Mediant™ 1000 MSBG

Multi-Service Business Gateway

SIP Protocol

User's Manual

Version 6.4

March 2012

Document # LTRT-27003

AudioCodes

Table of Contents

1 Introduction 23

Getting Started 27

2 Assigning the VoIP LAN IP Address 29

2.1 Using CLI 29
2.2 Using the Web Interface 31
2.3 Using the FXS Voice Menu Guidance 32

3 Assigning Data-Router LAN and WAN IP Addresses 35

3.1 Configuring Data-Routing LAN Interface 35
3.1.1 Using CLI 36
3.1.2 Using Web Interface 36
3.2 Configuring Device's DHCP Server 38
3.3 Assigning a WAN IP Address 38
3.4 Assign WAN Interface to VoIP Traffic 39
3.5 Configuring Quality of Service 41
3.5.1 Defining Total WAN Bandwidth 41
3.5.2 Defining VoIP Tx Traffic Shaping Classes 42
3.5.3 Defining VoIP Traffic Matching Rules 44
3.6 Configuring Virtual Routing and Forwarding 48
3.7 Enabling Remote HTTP/S Web Management 48
3.8 Reset the Device and Reconnect with WAN 48

Management Tools 49

4 Web-Based Management 51

4.1 Getting Acquainted with the Web Interface 51
4.1.1 Computer Requirements 51
4.1.2 Accessing the Web Interface 52
4.1.3 Areas of the GUI 53
4.1.4 Toolbar Description 54
4.1.5 Navigation Tree 55
4.1.5.1 Displaying Navigation Tree in Basic and Full View 56
4.1.5.2 Showing / Hiding the Navigation Pane 57
4.1.6 Working with Configuration Pages 58
4.1.6.1 Accessing Pages 58
4.1.6.2 Viewing Parameters 58
4.1.6.3 Modifying and Saving Parameters 60
4.1.6.4 Entering Phone Numbers 61
4.1.6.5 Working with Tables 62
4.1.7 Searching for Configuration Parameters 66
4.1.8 Creating a Login Welcome Message 67
4.1.9 Getting Help 68
4.1.10 Logging off the Web Interface 69
4.2 Using the Home Page 70
4.2.1 Assigning a Port Name 72
4.2.2 Resetting an Analog Channel 72
4.2.3 Viewing Analog Port Information 73

4.2.4 Viewing Trunk Channels 73
4.2.5 Replacing Modules 74
4.3 Configuring Web User Accounts 76
4.4 Configuring Web Security Settings 79
4.5 Limiting OAMP Access to a Specific WAN Interface 80
4.6 Web Login Authentication using Smart Cards 81
4.7 Configuring Web and Telnet Access List 82
4.8 Configuring RADIUS Settings 84
5 CLI-Based Management 85
5.1 Configuring Telnet and SSH Settings 86
5.2 TACACS+ for CLI Login 87
5.2.1 Configuring TACACS+ using CLI 88
6 SNMP-Based Management 89
6.1 Configuring SNMP Community Strings 89
6.2 Configuring SNMP Trap Destinations 90
6.3 Configuring SNMP Trusted Managers 91
6.4 Configuring SNMP V3 Users 92
7 EMS-Based Management 95
8 INI File-Based Management 97
8.1 INI File Format 98
8.1.1 Configuring Individual ini File Parameters 98
8.1.2 Configuring ini File Table Parameters 98
8.1.3 General ini File Formatting Rules 100
8.2 Modifying an ini File 101
8.3 Secured Encoded ini File 102
General System Settings 103
9 Configuring Certificates 105
9.1 Replacing Device Certificate 105
9.2 Loading a Private Key 108
9.3 Mutual TLS Authentication 109
9.4 Self-Signed Certificates 110
10 Date and Time 111
10.1 Manual Date and Time 111
10.1.1 Configuring Date and Time Manually using Web Interface 111
10.1.2 Configuring Date and Time Manually using CLI 112
10.2 Automatic Date and Time through SNTP Server 112
10.2.1 Configuring SNTP using Web Interface 113
10.2.2 Configuring SNTP using CLI 114
VoIP Configuration 115
11 Network 117
11.1 Configuring IP Interface Settings 117

11.1.1 Network Configuration Overview 120
11.1.1.1 Multiple Network Interfaces and VLANs 121
11.1.1.2 Setting Up VoIP Networking 127
11.2 Configuring the IP Routing Table 132
11.2.1 Routing Table Columns 133
11.2.1.1 Destination Column 134
11.2.1.2 Prefix Length Column 134
11.2.1.3 Gateway Column 134
11.2.1.4 Interface Column 134
11.2.1.5 Metric Column 135
11.2.1.6 State Column 135
11.2.2 Routing Table Configuration Summary and Guidelines 135
11.2.3 Troubleshooting the Routing Table 135
11.3 Configuring QoS Settings 135
11.4 DNS 137
11.4.1 Configuring the Internal DNS Table 137
11.4.2 Configuring the Internal SRV Table 138
11.5 NAT (Network Address Translation) Support 139
11.5.1 First Incoming Packet Mechanism 139
11.5.2 No-Op Packets 140
11.6 Configuring NFS Settings 141
11.7 Robust Receipt of Media Streams 143
11.8 Multiple Routers Support 144
12 Security 145
12.1 Configuring Firewall Settings 145
12.2 Configuring General Security Settings 149
13 Media 151
13.1 Configuring Voice Settings 151
13.1.1 Voice Gain (Volume) Control 152
13.1.2 Silence Suppression (Compression) 152
13.1.3 Echo Cancellation 153
13.2 Fax and Modem Capabilities 155
13.2.1 Fax/Modem Operating Modes 156
13.2.2 Fax/Modem Transport Modes 156
13.2.2.1 T.38 Fax Relay Mode 156
13.2.2.2 G.711 Fax / Modem Transport Mode 158
13.2.2.3 Fax Fallback 158
13.2.2.4 Fax/Modem Bypass Mode 158
13.2.2.5 Fax / Modem NSE Mode 159
13.2.2.6 Fax / Modem Transparent with Events Mode 160
13.2.2.7 Fax / Modem Transparent Mode 160
13.2.2.8 RFC 2833 ANS Report upon Fax/Modem Detection 161
13.2.3 V.34 Fax Support 161
13.2.3.1 Bypass Mechanism for V.34 Fax Transmission 162
13.2.3.2 Relay Mode for T.30 and V.34 Faxes 162
13.2.4 V.152 Support 162
13.2.5 Fax Transmission behind NAT 163
13.3 Configuring RTP/RTCP Settings 164
13.3.1 Dynamic Jitter Buffer Operation 165
13.3.2 Comfort Noise Generation 166
13.3.3 Dual-Tone Multi-Frequency Signaling 167
13.3.3.1 Configuring DTMF Transport Types 167

13.3.3.2 Configuring RFC 2833 Payload 169
13.3.4 Configuring RTP Base UDP Port 170
13.3.5 RTP Control Protocol Extended Reports (RTCP XR) 170
13.4 Configuring IP Media Settings 172
13.4.1 Answer Machine Detector (AMD) 172
13.4.2 Automatic Gain Control (AGC) 177
13.5 Configuring General Media Settings 178
13.6 Configuring Analog Settings 178
13.7 Configuring DSP Templates 179
13.8 DSP Channel Resources for Transcoding 180
13.9 Configuring Media Realms 181
13.10 Configuring Media Security 184
13.11 Configuring Quality of Experience Parameters per Media Realm 185
13.12 Configuring Server for Media Quality of Experience 188
14 Services 189
14.1 Routing Based on LDAP Active Directory Queries 189
14.1.1 Configuring the LDAP Server 189
14.1.2 AD-Based Tel-to-IP Routing in Microsoft Lync 190
14.1.2.1 Querying the AD and Routing Priority 191
14.1.2.2 Configuring AD-Based Routing Rules 193
14.2 Least Cost Routing 196
14.2.1 Overview 196
14.2.2 Configuring LCR 199
14.2.2.1 Enabling the LCR Feature 200
14.2.2.2 Configuring Cost Groups 201
14.2.2.3 Configuring Time Bands for Cost Groups 202
14.2.2.4 Assigning Cost Groups to Routing Rules 204
15 Enabling Applications 205
16 Control Network 207
16.1 Configuring SRD Table 207
16.2 Configuring SIP Interface Table 210
16.3 Configuring IP Groups 212
16.4 Configuring Proxy Sets Table 219
16.5 Configuring NAT Translation per IP Interface 224
16.6 Multiple SIP Signaling and Media Interfaces using SRDs 225
17 SIP Definitions 235
17.1 Configuring SIP General Parameters 235
17.2 Configuring Advanced Parameters 236
17.3 Configuring Account Table 237
17.4 Configuring Proxy and Registration Parameters 240
18 Coders and Profiles 243
18.1 Configuring Coders 243
18.2 Configuring Coder Groups 244
18.3 Configuring Tel Profile 246
18.4 Configuring IP Profiles 248

19 GW and IP to IP 251

19.1 Digital PSTN 251
19.1.1 Configuring TDM Bus Settings 251
19.1.2 Configuring CAS State Machines 251
19.1.3 Configuring Trunk Settings 254
19.1.4 Configuring Digital Gateway Parameters 257
19.1.5 Tunneling Applications 258
19.1.5.1 TDM Tunneling 258
19.1.5.2 QSIG Tunneling 261
19.1.6 Advanced PSTN Configuration 262
19.1.6.1 Release Reason Mapping 262
19.1.6.2 ISDN Overlap Dialing 266
19.1.6.3 ISDN Non-Facility Associated Signaling (NFAS) 268
19.1.6.4 Redirect Number and Calling Name (Display) 270
19.2 Trunk Group 271
19.2.1 Configuring Trunk Group Table 271
19.2.2 Configuring Trunk Group Settings 273
19.3 Manipulation 277
19.3.1 Configuring General Settings 277
19.3.2 Configuring Number Manipulation Tables 277
19.3.3 Configuring Redirect Number IP to Tel 281
19.3.4 Configuring Redirect Number Tel to IP 283
19.3.5 Mapping NPI/TON to SIP Phone-Context 285
19.3.6 Numbering Plans and Type of Number 287
19.3.7 Configuring Release Cause Mapping 288
19.3.8 SIP Calling Name Manipulations 289
19.3.9 SIP Message Manipulation 289
19.3.10 Manipulating Number Prefix 290
19.4 Routing 291
19.4.1 Configuring General Routing Parameters 291
19.4.2 Configuring Outbound IP Routing Table 291
19.4.3 Configuring Inbound IP Routing Table 299
19.4.4 Mapping PSTN Release Cause to SIP Response 302
19.4.5 IP Destinations Connectivity Feature 302
19.4.6 Alternative Routing for Tel-to-IP Calls 303
19.4.6.1 Alternative Routing Based on IP Connectivity 303
19.4.6.2 Alternative Routing Based on SIP Responses 304
19.4.6.3 PSTN Fallback 306
19.4.7 Alternative Routing for IP-to-Tel Calls 307
19.4.7.1 Alternative Routing to Trunk upon Q.931 Call Release Cause Code 307
19.4.7.2 Alternative Routing to IP Destination upon Busy Trunk 308
19.5 Configuring DTMF and Dialing 310
19.6 Configuring Supplementary Services 310
19.6.1 Call Hold and Retrieve 312
19.6.2 BRI Suspend and Resume 314
19.6.3 Consultation Feature 314
19.6.4 Call Transfer 315
19.6.4.1 Consultation Call Transfer 315
19.6.4.2 Consultation Transfer for QSIG Path Replacement 316
19.6.4.3 Blind Call Transfer 316
19.6.5 Call Forward 317
19.6.5.1 Call Forward Reminder Ring 318
19.6.5.2 Call Forward Reminder (Off-Hook) Special Dial Tone 319
19.6.5.3 BRI Call Forwarding 319
19.6.6 Call Waiting 320

19.6.7 Message Waiting Indication 321
19.6.8 Caller ID 322
19.6.8.1 Caller ID Detection / Generation on the Tel Side 322
19.6.8.2 Debugging a Caller ID Detection on FXO 323
19.6.8.3 Caller ID on the IP Side 324
19.6.9 Three-Way Conferencing 325
19.6.10 Emergency E911 Phone Number Services 326
19.6.10.1 FXS Device Emulating PSAP using DID Loop-Start Lines 326
19.6.10.2 FXO Device Interworking SIP E911 Calls from Service Provider's IP Network to PSAP DID Lines 329
19.6.10.3 Pre-empting Existing Calls for E911 IP-to-Tel Calls 332
19.6.11 Enhanced 9-1-1 Support for Microsoft Lync Server 2010 333
19.6.11.1 About E9-1-1 Services 333
19.6.11.2 Microsoft Lync Server 2010 and E9-1-1 334
19.6.11.3 AudioCodes ELIN Gateway for Lync Server 2010 E9-1-1 Calls to PSTN 338
19.6.11.4 Configuring AudioCodes ELIN Gateway 342
19.6.12 Multilevel Precedence and Preemption 344
19.6.13 Denial of Collect Calls 346
19.6.14 Configuring ISDN Supplementary Services 346
19.6.15 Configuring Voice Mail Parameters 348
19.6.16 Advice of Charge Services for Euro ISDN 349
19.7 Analog Gateway 350
19.7.1 Configuring Keypad Features 350
19.7.2 Configuring Metering Tones 352
19.7.3 Configuring Charge Codes 353
19.7.4 Configuring FXO Settings 354
19.7.5 Configuring Authentication 355
19.7.6 Configuring Automatic Dialing 356
19.7.7 Configuring Caller Display Information 357
19.7.8 Configuring Call Forward 358
19.7.9 Configuring Caller ID Permissions 360
19.7.10 Configuring Call Waiting 361
19.7.11 Configuring FXS Distinctive Ringing and Call Waiting Tones per Source/Destination Number 362
19.7.12 FXS/FXO Coefficient Types 363
19.7.13 FXO Operating Modes 363
19.7.13.1 FXO Operations for IP-to-Tel Calls 363
19.7.13.2 FXO Operations for Tel-to-IP Calls 366
19.7.13.3 Call Termination on FXO Devices 368
19.7.14 Remote PBX Extension between FXO and FXS Devices 369
19.7.14.1 Dialing from Remote Extension (Phone at FXS) 370
19.7.14.2 Dialing from PBX Line or PSTN 370
19.7.14.3 Message Waiting Indication for Remote Extensions 371
19.7.14.4 Call Waiting for Remote Extensions 371
19.7.14.5 FXS Gateway Configuration 372
19.7.14.6 FXO Gateway Configuration 373
19.8 Dialing Plan Features 374
19.8.1 Dialing Plan Notation for Routing and Manipulation 374
19.8.2 Digit Mapping 375
19.8.3 External Dial Plan File 377
19.8.3.1 Modifying ISDN-to-IP Calling Party Number 378
19.8.4 Dial Plan Prefix Tags for IP-to-Tel Routing 379
19.9 SIP Call Routing Examples 381
19.9.1 SIP Call Flow Example 381
19.9.2 SIP Message Authentication Example 383
19.9.3 Establishing a Call between Two Devices 386
19.9.4 Trunk-to-Trunk Routing Example 387

19.9.5 SIP Trunking between Enterprise and ITSPs 388
19.10 IP-to-IP Routing Application 391
19.10.1 Theory of Operation 392
19.10.1.1 Proxy Sets 393
19.10.1.2 IP Groups 393
19.10.1.3 Inbound and Outbound IP Routing Rules 395
19.10.1.4 Accounts 395
19.10.2 IP-to-IP Routing Configuration Example 396
19.10.2.1 Step 1: Enable the IP-to-IP Capabilities 398
19.10.2.2 Step 2: Configure the Number of Media Channels 399
19.10.2.3 Step 3: Define a Trunk Group for the Local PSTN 399
19.10.2.4 Step 4: Configure the Proxy Sets 400
19.10.2.5 Step 5: Configure the IP Groups 402
19.10.2.6 Step 6: Configure the Account Table 404
19.10.2.7 Step 7: Configure IP Profiles for Voice Coders 405
19.10.2.8 Step 8: Configure Inbound IP Routing 407
19.10.2.9 Step 9: Configure Outbound IP Routing 408
19.10.2.10 Step 10: Configure Destination Phone Number Manipulation 410

20 Session Border Controller 411

20.1 SBC Overview 411
20.1.1 Overview 411
20.1.1.1 NAT Traversal 412
20.1.1.2 VoIP Firewall 412
20.1.1.3 Topology Hiding 412
20.1.1.4 SIP Normalization 412
20.1.1.5 Survivability 413
20.1.2 SIP Network Definitions 413
20.1.3 SIP Dialog Initiation Process 413
20.1.3.1 Determining Source and Destination URL 414
20.1.3.2 Source IP Group Classification 415
20.1.3.3 SBC IP-to-IP Routing 416
20.1.3.4 IP-to-IP Inbound and Outbound Manipulation 417
20.1.3.5 SIP Header Manipulation 419
20.1.4 User Registration and Internal Database 421
20.1.4.1 Initial Registration Request Processing 421
20.1.4.2 Internal Database 422
20.1.4.3 Routing using Internal Database 422
20.1.4.4 Registration Refreshes 423
20.1.4.5 Registration Restriction Control 423
20.1.5 SBC Media Handling 424
20.1.5.1 Media Anchoring without Transcoding (Transparent) 425
20.1.5.2 Media Anchoring with Transcoding 426
20.1.5.3 No Media Anchoring 428
20.1.5.4 Interworking DTMF Methods 429
20.1.5.5 Transcoding Modes 430
20.1.5.6 Restricting Coders 430
20.1.5.7 SRTP-RTP Transcoding 432
20.1.5.8 Multiple RTP Media Streams per Call Session 433
20.1.6 Fax Negotiation and Transcoding 433
20.1.7 SIP Dialog Admission Control 434
20.1.8 Limiting SBC Call Duration 434
20.1.9 SIP Authentication Server for SBC Users 434
20.1.10 Handling SIP 3xx Redirect Responses 435
20.1.11 Interworking SIP Diversion and History-Info Headers 436
20.1.12 Call Survivability 438
20.1.12.1 Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server for Survivability 438

20.1.12.2 BroadSoft's Shared Phone Line Call Appearance for SBC Survivability 439
20.1.12.3 Call Survivability for Call Centers 440
20.1.12.4 Survivability Mode Display on Aastra IP Phones 442
20.1.13 Call Forking 443
20.1.13.1 Initiating SIP Call Forking 443
20.1.13.2 SIP Forking Initiated by SIP Proxy Server 443
20.1.14 Alternative Routing on Detection of Failed SIP Response 444
20.2 SBC Configuration 445
20.2.1 Configuring General Settings 445
20.2.2 Configuring Admission Control 446
20.2.3 Configuring Allowed Coder Groups 448
20.2.4 Configuring SIP Message Policy Rules 449
20.2.5 Routing SBC 452
20.2.5.1 Configuring Classification Rules 452
20.2.5.2 Configuring Condition Rules 456
20.2.5.3 Configuring SBC IP-to-IP Routing 457
20.2.5.4 Configuring Alternative Routing Reasons 462
20.2.6 Manipulations SBC 464
20.2.6.1 Configuring Message Manipulations 464
20.2.6.2 Configuring IP-to-IP Inbound Manipulations 467
20.2.6.3 Configuring IP-to-IP Outbound Manipulations 469

21 Stand-Alone Survivability (SAS) Application 473

21.1 Overview 473
21.1.1 SAS Operating Modes 473
21.1.1.1 SAS Outbound Mode 474
21.1.1.2 SAS Redundant Mode 475
21.1.2 SAS Routing 477
21.1.2.1 SAS Routing in Normal State 477
21.1.2.2 SAS Routing in Emergency State 479
21.2 SAS Configuration 480
21.2.1 General SAS Configuration 480
21.2.1.1 Enabling the SAS Application 480
21.2.1.2 Configuring Common SAS Parameters 481
21.2.2 Configuring SAS Outbound Mode 483
21.2.3 Configuring SAS Redundant Mode 484
21.2.4 Configuring Gateway Application with SAS 484
21.2.4.1 Gateway with SAS Outbound Mode 484
21.2.4.2 Gateway with SAS Redundant Mode 486
21.2.5 Advanced SAS Configuration 488
21.2.5.1 Manipulating URI user part of Incoming REGISTER 488
21.2.5.2 Manipulating Destination Number of Incoming INVITE 489
21.2.5.3 SAS Routing Based on IP2IP Routing Table 492
21.2.5.4 Blocking Calls from Unregistered SAS Users 496
21.2.5.5 Configuring SAS Emergency Calls 497
21.2.5.6 Adding SIP Record-Route Header to SIP INVITE 498
21.2.5.7 Replacing Contact Header for SIP Messages 499
21.3 Viewing Registered SAS Users 499
21.4 SAS Cascading 500

22 Configuring the IP Media Parameters 503

22.1 Transcoding using Third-Party Call Control 503
22.1.1 Using RFC 4117 504
22.1.2 Using RFC 4240 - NetAnn 2-Party Conferencing 504
22.2 Overview 506
22.2.1 Conference Server 506

22.2.1.1 Simple Conferencing (NetAnn) 507
22.2.1.2 Advanced Conferencing (MSCML) 509
22.2.1.3 Conference Call Flow Example 514
22.2.2 Announcement Server 521
22.2.2.1 NetAnn Interface 521
22.2.2.2 MSCML Interface 522
22.2.2.3 Voice Streaming 531
22.2.2.4 Announcement Call Flow Example 537
22.2.3 Voice XML Interpreter 540
22.2.3.1 Features 540
22.2.3.2 Feature Key 540
22.2.3.3 VXML Scripts 540
22.2.3.4 Proprietary Extensions 541
22.2.3.5 Combining <audio> Elements 546
22.2.3.6 Notes Regarding Non-compliant Functionality 547
22.2.3.7 Supported Elements and Attributes 547
22.2.3.8 Example of UDT 'beep' Tone Definition 562
22.2.3.9 Limitations and Restrictions 562

Data-Router Settings 563

23 Getting Acquainted with Data Configuration Pages 565

23.1 Working with Tables 565
23.2 Using the Configuration Buttons 566

24 WAN Access 567

24.1 Configuring Ethernet WAN Interface 567
24.2 Configuring T1 WAN Interface 569
24.2.1 PPP over T1 WAN 570
24.2.2 HDLC over T1 WAN 572
24.2.3 ML-PPP over T1 WAN 573
24.3 Configuring SHDSL WAN Interface 576

25 Firewall and ACL 581

25.1 Configuring General Security Settings 582
25.2 Configuring LAN Restrictions 583
25.3 Configuring Port Forwarding 584
25.4 Configuring DMZ Host 586
25.5 Configuring Port Triggering 586
25.6 Configuring Website Restrictions 588
25.7 Configuring NAT 589
25.8 Configuring the Access List 591
25.9 Configuring Advanced Filtering 594

26 QoS 595

26.1 Configuring General QoS Settings 595
26.2 Configuring Matching Rules 596
26.3 Configuring Traffic Shaping 599
26.4 Configuring DSCP Settings 602
26.5 Configuring 802.1p Settings 604

27 VPN 605
27.1 Configuring IPSec 605
27.2 Configuring PPTP Server 608
27.3 Configuring L2TP Server 609
27.4 Creating VPN Users 610
28 Data Services 615
28.1 Configuring DDNS 615
28.2 Configuring DNS Server 617
28.3 Configuring DHCP Server 618
29 Data Routing 623
29.1 Configuring General Routing 623
29.2 Configuring BGP and OSPF 627
30 Objects and Rules 629
30.1 Configuring Protocols 629
30.2 Configuring Network Objects 630
31 Configuring Network Connections 633
31.1 Network Connection Wizard 634
31.2 LAN Switch 639
31.3 WAN Ethernet 641
31.3.1 Dynamic Host Control Protocol 641
31.3.2 Manual WAN IP Address 642
31.3.3 Point-to-Point Protocol over Ethernet (PPPoE) 643
31.4 LAN Bridge 645
31.5 Virtual LAN Interface (VLAN) 646
31.6 Point-to-Point Tunneling Protocol (PPTP) 648
31.7 Point-to-Point Tunneling Protocol Server (PPTP Server) 653
31.8 Layer 2 Tunneling Protocol (L2TP) 655
31.9 Layer 2 Tunneling Protocol Server (L2TP Server) 659
31.10 Internet Protocol Security (IPSec) 660
31.11 Internet Protocol Security Server (IPSec Server) 666
31.12 General Routing Encapsulation (GRE) 668
31.12.1 GRE Example Scenario 669
31.13 Editing Existing Connections 671

Maintenance 675

32 Basic Maintenance 677
32.1 Resetting the Device 677
32.2 Locking and Unlocking the Device 679
32.3 Saving Configuration 680
33 Software Upgrade 681
33.1 Loading Auxiliary Files 681
33.1.1 Auxiliary Configuration Files 684
33.1.1.1 Call Progress Tones File 684

33.1.1.2 Distinctive Ringing 687
33.1.1.3 Prerecorded Tones File 688
33.1.1.4 Voice Prompts File 689
33.1.1.5 CAS Files 690
33.1.1.6 Dial Plan File 690
33.1.1.7 User Information File 691
33.1.1.8 User Information File for PBX Extensions and "Global" Numbers 692
33.1.1.9 AMD Sensitivity File 694
33.2 Loading Software Upgrade Key 696
33.3 Software Upgrade Wizard 699
33.4 Backing Up and Loading Configuration File 702
34 Restoring Factory Default Settings 705
34.1 Restoring Defaults using CLI 705
34.2 Restoring Defaults using Hardware Reset Button 706
34.3 Restoring Defaults using an ini File 706
Status, Performance Monitoring & Reporting 707
35 System Status 709
35.1 Viewing Device Information 709
35.2 Viewing Ethernet Port Information 710
36 Carrier-Grade Alarms 711
36.1 Viewing Active Alarms 711
36.2 Viewing Alarm History 712
37 Performance Monitoring 713
37.1 Viewing Trunk Utilization 713
37.2 Viewing MOS per Media Realm 715
37.3 Viewing Quality of Experience 716
37.4 Viewing Average Call Duration 718
38 VoIP Status 719
38.1 Viewing Active IP Interfaces 719
38.2 Viewing Performance Statistics 719
38.3 Viewing Call Counters 720
38.4 Viewing SAS/SBC Registered Users 722
38.5 Viewing Call Routing Status 723
38.6 Viewing Registration Status 724
38.7 Viewing IP Connectivity 725
39 Data Status 727
39.1 Viewing WAN Status 727
39.2 Viewing Network Connection Statistics 727
39.3 Viewing Logged Security Events 728
39.4 Viewing QoS Queues Statistics 730
39.5 Viewing Logged Data Events 731

40 Reporting Information to External Party 733
40.1 Generating Call Detail Records 733
40.1.1 CDR Fields for Gateway Application 733
40.1.2 Release Reasons in CDR 735
40.1.3 CDR Fields for SBC 737
40.1.3.1 CDR Fields for SBC Signaling 737
40.1.3.2 CDR Fields for SBC Media 738
40.1.4 Supported RADIUS Attributes 739
40.2 Event Notification using X-Detect Header 742
40.3 Querying Device Channel Resources using SIP OPTIONS 744

Diagnostics 745

41 Configuring Syslog Settings 747

42 Capturing VoIP and Data-Router Network Traffic 749

43 Running Data-Router Diagnostic Tests 751
43.1 Internet Connectivity Diagnostic Tests 751
43.2 Running Diagnostic Tests 752

Appendices 753

A Configuration Parameters Reference 755
A.1 Networking Parameters 755
A.1.1 VoIP Multiple Network Interfaces and VLAN Parameters 755
A.1.2 VoIP Static Routing Parameters 756
A.1.3 Quality of Service Parameters 757
A.1.4 NAT and STUN Parameters 759
A.1.5 NFS Parameters 760
A.1.6 DNS Parameters 761
A.1.7 DHCP Parameters 762
A.1.8 NTP and Daylight Saving Time Parameters 763
A.2 Management Parameters 764
A.2.1 General Parameters 764
A.2.2 Web Parameters 765
A.2.3 Telnet Parameters 767
A.2.4 SNMP Parameters 767
A.2.5 CLI Parameters 771
A.2.6 Serial Parameters 772
A.3 Debugging and Diagnostics Parameters 773
A.3.1 General Parameters 773
A.3.2 Syslog, CDR and Debug Parameters 775
A.3.3 Resource Allocation Indication Parameters 778
A.4 Security Parameters 779
A.4.1 General Parameters 779
A.4.2 HTTPS Parameters 780
A.4.3 SRTP Parameters 781
A.4.4 TLS Parameters 783
A.4.5 SSH Parameters 785
A.4.6 OCSP Parameters 786
A.5 RADIUS Parameters 786
A.6 SIP Media Realm Parameters 788

A.7 Quality of Experience Reporting 789
A.8 Control Network Parameters 790
A.8.1 IP Group, Proxy, Registration and Authentication Parameters 790
A.8.2 Network Application Parameters 803
A.9 General SIP Parameters 805
A.10 Coders and Profile Parameters 831
A.11 Channel Parameters 840
A.11.1 Voice Parameters 840
A.11.2 Coder Parameters 843
A.11.3 DTMF Parameters 844
A.11.4 RTP, RTCP and T.38 Parameters 845
A.12 Gateway and IP-to-IP Parameters 850
A.12.1 Fax and Modem Parameters 850
A.12.2 DTMF and Hook-Flash Parameters 856
A.12.3 Digit Collection and Dial Plan Parameters 861
A.12.4 Voice Mail Parameters 863
A.12.5 Supplementary Services Parameters 869
A.12.6 PSTN Parameters 896
A.12.7 ISDN and CAS Interworking Parameters 912
A.12.8 Answer and Disconnect Supervision Parameters 929
A.12.9 Tone Parameters 934
A.12.10 Telephone Keypad Sequence Parameters 944
A.12.11 General FXO Parameters 947
A.12.12 FXS Parameters 950
A.12.13 Trunk Groups and Routing Parameters 951
A.12.14 IP Connectivity Parameters 959
A.12.15 Alternative Routing Parameters 960
A.12.16 Number Manipulation Parameters 962
A.12.17 LDAP Parameters 976
A.12.18 Least Cost Routing Parameters 977
A.13 SBC Parameters 979
A.14 Standalone Survivability Parameters 995
A.15 IP Media Parameters 999
A.16 Auxiliary and Configuration Files Parameters 1011
A.16.1 Auxiliary/Configuration File Name Parameters 1011
A.16.2 Automatic Update Parameters 1012

B SIP Message Manipulation Syntax 1015

B.1 Actions 1015
B.2 Header Types 1015
B.2.1 Accept 1015
B.2.2 Accept-Language 1016
B.2.3 Allow 1016
B.2.4 Call-Id 1017
B.2.5 Contact 1017
B.2.6 Cseq 1018
B.2.7 Diversion 1018
B.2.8 Event 1019
B.2.9 From 1020
B.2.10 History-Info 1020
B.2.11 Min-Se and Min-Expires 1021
B.2.12 P-Asserted-Identity 1022
B.2.13 P-Associated-Uri 1022
B.2.14 P-Called-Party-Id 1023
B.2.15 P-Charging-Vector 1024

B.2.16 P-Preferred-Identity 1024
B.2.17 Privacy 1025
B.2.18 Proxy-Require 1025
B.2.19 Reason 1026
B.2.20 Referred-By 1027
B.2.21 Refer-To 1027
B.2.22 Remote-Party-Id 1028
B.2.23 Request-Uri 1029
B.2.24 Require 1030
B.2.25 Resource-Priority 1031
B.2.26 Retry-After 1031
B.2.27 Server or User-Agent 1032
B.2.28 Service-Route 1032
B.2.29 Session-Expires 1033
B.2.30 Subject 1034
B.2.31 Supported 1034
B.2.32 To 1035
B.2.33 Unsupported 1036
B.2.34 Via 1036
B.2.35 Warning 1037
B.2.36 Unknown Header 1038
B.3 Structure Definitions 1039
B.3.1 Event Structure 1039
B.3.2 Host 1039
B.3.3 MLPP 1039
B.3.4 Privacy Struct 1040
B.3.5 Reason Structure 1040
B.3.6 SIPCapabilities 1040
B.3.7 URL 1041
B.4 Random Type 1042
B.4.1 Random Strings 1042
B.4.2 Random Integers 1042
B.5 Wildcarding for Header Removal 1042
B.6 Message Manipulation using SDP Conditions 1043
B.7 Copying Information between Messages using Variables 1044
B.8 Enum Definitions 1045
B.8.1 AgentRole 1045
B.8.2 Event Package 1045
B.8.3 MLPP Reason Type 1046
B.8.4 Number Plan 1046
B.8.5 NumberType 1046
B.8.6 Privacy 1047
B.8.7 Reason (Diversion) 1047
B.8.8 Reason (Reason Structure) 1047
B.8.9 Reason (Remote-Party-Id) 1050
B.8.10 Refresher 1050
B.8.11 Screen 1050
B.8.12 ScreenInd 1050
B.8.13 TransportType 1051
B.8.14 Type 1051
B.9 Actions and Types 1052
B.10 Syntax 1057
C DSP Templates 1061
C.1 Analog Interfaces 1061
C.2 Digital Interfaces 1062

C.3 Media Processing Interfaces 1063

D Technical Specifications 1065

Reader's Notes

Notice

This document describes the AudioCodes Mediant 1000 Multi-Service Business Gateway (MSBG).

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Before consulting this document, check the corresponding Release Notes regarding feature preconditions and/or specific support in this release. In cases where there are discrepancies between this document and the Release Notes, the information in the Release Notes supersedes that in this document. Updates to this document and other documents as well as software files can be downloaded by registered customers at http://www.audiocodes.com/downloads.

© Copyright 2012 AudioCodes Ltd. All rights reserved.

This document is subject to change without notice.

Date Published: March-09-2012

Trademarks

AudioCodes, AC, AudioCoded, Ardito, CTI2, CTI², CTI Squared, HD VoIP, HD VoIP Sounds Better, InTouch, IPmedia, Mediant, MediaPack, NetCoder, Netrake, Nuera, Open Solutions Network, OSN, Stretto, TrunkPack, VMAS, VoicePacketizer, VoIPerfect, VoIPerfectHD, What's Inside Matters, Your Gateway To VoIP and 3GX are trademarks or registered trademarks of AudioCodes Limited. All other products or trademarks are property of their respective owners. Product specifications are subject to change without notice.

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

Customer Support

Customer technical support and service are generally provided by AudioCodes' Distributors, Partners, and Resellers from whom the product was purchased. For technical support for products purchased directly from AudioCodes, or for customers subscribed to AudioCodes Customer Technical Support (ACTS), contact support@audiocodes.com.

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Related Documentation

Manual Name
SIP CPE Release Notes
Product Reference Manual for SIP CPE Devices
Mediant 1000 MSBG Hardware Installation Manual
MSBG Data CLI Reference Guide
MSBG VoIP and System CLI Reference Guide

Note: The scope of this document does not fully cover security aspects for deploying the device in your environment. Security measures should be done in accordance with your organization's security policies. For basic security guidelines, you can refer to AudioCodes Recommended Security Guidelines document.

Note: Throughout this manual, unless otherwise specified, the term *device* refers to the Mediant 1000 MSBG.

Note: Before configuring the device, ensure that it is installed correctly as instructed in the *Hardware Installation Manual*.

Note: The terms IP-to-Tel and Tel-to-IP refer to the direction of the call relative to the device. IP-to-Tel refers to calls received from the IP network and destined to the PSTN/PBX (i.e., telephone connected directly or indirectly to the device); Tel-to-IP refers to calls received from the PSTN/PBX and destined for the IP network.

Notes:

- FXO (Foreign Exchange Office) is the interface replacing the analog telephone and connects to a Public Switched Telephone Network (PSTN) line from the Central Office (CO) or to a Private Branch Exchange (PBX). The FXO is designed to receive line voltage and ringing current, supplied from the CO or the PBX (just like an analog telephone). An FXO VoIP device interfaces between the CO/PBX line and the Internet.
- FXS (Foreign Exchange Station) is the interface replacing the Exchange (i.e., the CO or the PBX) and connects to analog telephones, dial-up modems, and fax machines. The FXS is designed to supply line voltage and ringing current to these telephone devices. An FXS VoIP device interfaces between the analog telephone devices and the Internet.

Note: OPEN SOURCE SOFTWARE. Portions of the software may be open source software and may be governed by and distributed under open source licenses, such as the terms of the GNU General Public License (GPL), the terms of the Lesser General Public License (LGPL), BSD and LDAP, which terms are located at: http://www.audiocodes.com/support and all are incorporated herein by reference. If any open source software is provided in object code, and its accompanying license requires that it be provided in source code as well, Buyer may receive such source code by contacting AudioCodes, by following the instructions available on AudioCodes website.

Customer Information

1. This equipment, RMX module of Mediant 1000 MSBG complies with Part 68 of the FCC Rules and the requirements adopted by the ACTA. On the front panel of the module is a label that contains among other information, a product identifier in the format US:6NPDDNANRMX. If requested, this number must be provided to the telephone company.
2. This equipment is designed to be connected to the telephone network using RJ-48C connector, which complies with Part 68 rules and requirements adopted by ACTA. The service order codes (SOC) are 6.0F for digital interface and the Facility interface codes (FIC) are: 04DU9.1SN, 04DU9.1KN, 04DU9.BN, 04DU9.DN.
3. Should the product cause harm to the telephone network, the telephone company will notify you in advance that temporary discontinuance of service may be required. If advance notice is not practical, you will be notified as soon as possible. Also, you will be advised of your right to file a complaint with the FCC if it is necessary.
4. The telephone company may make changes in its facilities, equipment, operations or procedures that could affect the operation of the equipment. If this happens, the telephone company will provide advance notice in order for you to make necessary modifications to maintain uninterrupted service.
5. If trouble is experienced with this equipment, for repair or warranty information please contact AudioCodes Inc., 2099 Gateway Place, Suite 500, San Jose, CA, 95110, phone number (408) 441 1175, URL:www.audiocodes.com. If the equipment is causing harm to the telephone network, the telephone company may request to disconnect the equipment until the problem is resolved.
6. Installation is described in the product's Installation Manual. Connection to telephone company provided coin service is prohibited. Connection to party lines service is subject to state tariffs.

Reader's Notes

1 Introduction

The Mediant 1000 Multi-Service Business Gateway (MSBG) is a networking device that combines multiple service functions such as a Media Gateway, Session Border Controller (SBC), Data Router and Firewall, LAN switch, WAN access, Stand Alone Survivability (SAS) and an integrated general-purpose server.

The device offers enhanced dialing plans and voice routing capabilities along with SIP-to-SIP mediation, allowing enterprises to implement SIP Trunking services (IP-to-IP call routing) and IP-based Unified Communications, as well as flexible PSTN and legacy PBX connectivity.

The device's data routing capabilities supports static and dynamic routing protocols such as RIP/OSPF and BGP, Virtual Routing and Forwarding (VRF-Lite) where interfaces can be clustered into a VRF to provide segregated routing domains. In addition, the device supports Gigabit Ethernet, T1 WAN, and Symmetric High-Speed Digital Subscriber Line (SHDSL) WAN interfaces, providing flexibility in connecting to Service Providers.

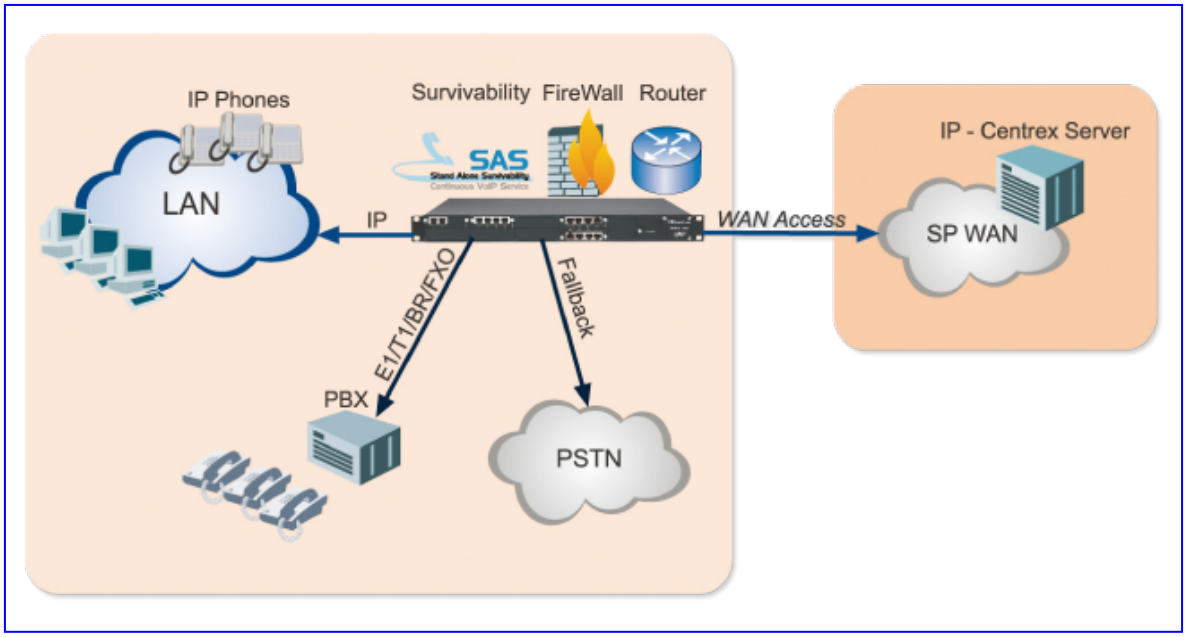
- The T1 WAN connection is through a Dual T1 line interface (according to ANSI T1.403-1999). The device uses its Dual T1 WAN Data Service Unit/Channel Service Unit (DSU/CSU) port interface to transmit and receive data using IP over Point-to-Point Protocol (PPP), IP over High-Level Data Link Control (HDLC), or IP over Multilink Point-to-Point Protocol (MLPPP) framing.
- The SHDSL WAN connection supports up to four copper wire pairs according to G.991.2, acting as a remote-terminal CPE device. Both ATM and EFM modes are supported. In the ATM mode, a variety of protocols are supported, including PPPoE, PPPoA, and RFC 2684 in both bridged (Ethernet-over-ATM) and routed (IP-over-ATM) variants. In the EFM mode, the SHDSL port functions as a logical Ethernet device.

The device is designed as a secured VoIP and data platform. Enhanced media gateway security features include, for example, SRTP for media, TLS for SIP control, IPSec for management. Data security functions include integrated Stateful Firewall, IDS/IPS, SSL for remote user access and site-to-site VPN. A fully featured enterprise class SBC provides a secured voice network deployment based on a Back-to-Back User Agent (B2BUA) implementation.

The device's Stand Alone Survivability (SAS) functionality offers service continuity to enterprises served by a centralized SIP-based IP-Centrex server or branch offices of distributed enterprises. SAS enables internal office communication between SIP clients, along with PSTN fallback in the case of disconnection from the centralized SIP IP-Centrex server or IP-PBX.

The device also provides an integrated Open Solution Network (OSN) Server module (based on an Intel CPU). The OSN can host a variety of third-party applications such as IP-PBX, Call Center, and Conferencing.

Figure 1-1: Mediant 1000 MSBG Typical Application



The device is a fully modular, 1U high, integrated access device. Six hot-swappable slots are available for any mixture of the available telephony modules:

- **Digital:**
  - The device supports up to four E1/T1/J1 spans, available in 1, 2, or 4 spans per module, for connecting the PSTN/PBX to the IP network. The digital modules provide RJ-48 ports. The digital module can also be configured with up to 1 or 2 paired spans for switching to the PSTN in case of power or network failure (PSTN Fallback).
  - The device also supports ISDN Basic Rate Interface (BRI) modules for connecting BRI-based PSTN or PBX lines to the IP network. Each BRI module supports four BRI ports (RJ-45). Up to five BRI modules can be housed in the device, supporting up to 20 BRI digital ports. The BRI module can be configured for PSTN Fallback, switching to the PSTN in case of power failure or network problems.
  - Depending on configuration, the device can provide IP Media channels at the expense of PSTN channels. These channels may be used for Media Server applications.
- **Analog:** The device's analog interface supports up to 24 analog ports (four ports per analog module) in various Foreign Exchange Office (FXO) or Foreign Exchange Station (FXS) configurations, supporting up to 24 simultaneous VoIP calls. The device supports up to six analog modules, each module providing four analog RJ-11 ports. The FXO module can be used to connect analog lines of an enterprise's PBX or the PSTN to the IP network. The FXS module can be used to connect legacy telephones, fax machines, and modems to the IP network. Optionally, the FXS module can be connected to the external trunk lines of a PBX. When deployed with a combination of FXO and FXS modules, the device can be used as a PBX for Small Office Home Office (SOHO) users, and businesses not equipped with a PBX. The FXS modules also support the Lifeline feature, enabling an FXS port to connect directly to the PSTN upon power and/or network failure.
- **Media Processing Module (MPM):** The device supports up to three MPM modules for providing IP media processing services such as conferencing, SBC, and IP-to-IP applications.

In total, the device has the capacity for 120 concurrent voice channels.

The device also provides the CRMX module for data routing, switching, security, and voice security (SBC) capabilities. This module provides three LAN ports and one WAN port that is available either as Gigabit Ethernet, fiber optic, SHDSL, or a Dual Data Service Unit/Channel Service Unit (DSU/CSU) T1 WAN port.

The device allows full management through its command line interface (CLI) as well as its HTTP/S-based embedded Web server. The user-friendly Web interface allows remote configuration using any standard Web browser (such as Microsoft™ Internet Explorer™).

Reader's Notes

Part I

Getting Started

Before you can begin configuring your device, you need to access it with the default VoIP LAN IP address and change this IP address and/or WAN IP address, if required, to suit your networking scheme. Once modified, you can then access the device using the new LAN or WAN IP address. This section describes how to perform this initialization process.

Note: Once you access the device through the default VoIP LAN interface, you can configure Web management access for one of the following interfaces:

- Any of the configured LAN data interfaces. The default LAN data interface is 192.168.0.1. This interface can be in a different subnet to the VoIP LAN IP address and with a different VLAN ID. This is useful, for example, if you want to separate management from the VoIP traffic.
- WAN port interface. In this setup, you need to enable remote HTTP/S access to the WAN port interface.

Reader's Notes

2 Assigning the VoIP LAN IP Address

The default VoIP LAN IP address is listed in the table below:

Table 2-1: Default VoIP LAN IP Address

IP Address	Value
IP Address	192.168.0.2
Subnet Mask	255.255.255.0
Default Gateway IP Address	192.168.0.1

You can use any of the following management tools to change the default VoIP LAN IP address:

- Embedded command line interface (CLI) - see 'Using CLI' on page 29
- Embedded HTTP-based Web server - see 'Using the Web Interface' on page 31
- Analog (FXS) telephone voice menu - see Using the FXS Voice Menu Guidance on page 32

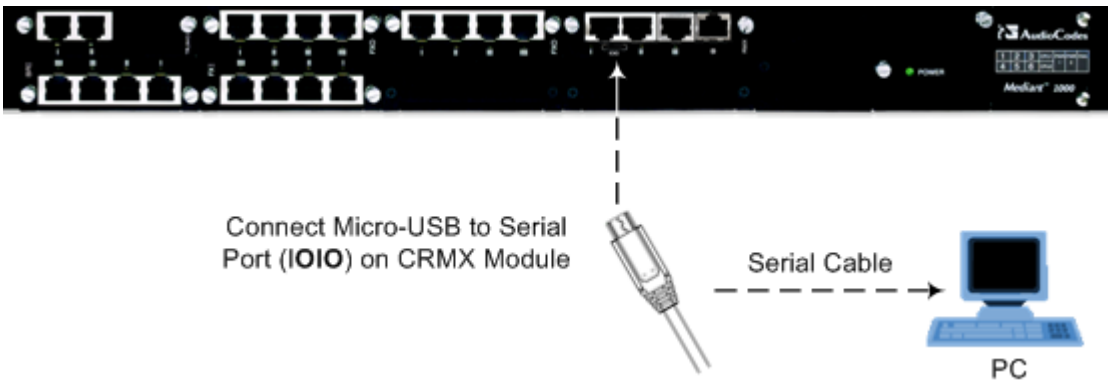
2.1 Using CLI

The procedure below describes how to assign a VoIP LAN IP address, using CLI.

➢ **To assign a LAN IP address using CLI:**

1. Connect the RS-232 port of the device to the serial communication port on your computer. For more information, refer to the *Hardware Installation Manual*.

Figure 2-1: Connecting Serial Port to PC for Serial Communication



2. Establish a serial communication link with the device using a terminal emulator program (such as HyperTerminal) with the following communication port settings:
   - Baud Rate: 115,200 bps
   - Data Bits: 8
   - Parity: None
   - Stop Bits: 1
   - Flow Control: None
3. At the CLI prompt, type the username (default is "Admin" - case sensitive), and then press Enter:

```
Username: Admin
```

4. At the prompt, type the password (default is "Admin" - case sensitive), and then press Enter:

   Password: **Admin**

5. At the prompt, type the following, and then press Enter:

   enable

6. At the prompt, type the password again, and then press Enter:

   Password: **Admin**

7. At the prompt, type the following command to access the VoIP interface, and then press Enter:

   # **configure-voip**

8. At the prompt, type the following command to access the Multiple Interface table, and then press Enter:

   (config-voip)# **interface network-if 0**

9. At the prompt, type the following command to set the IP address, and then press Enter:

   (network-if-0)# **set ip** <IP address>

10. At the prompt, type the following command to set the prefix length, and then press Enter:

    (network-if-0)# **set prefix-length** <prefix length, e.g., 16>

11. At the prompt, type the following command to set the Default Gateway, and then press Enter:

    (network-if-0)# **set gateway** <IP address>

12. At the prompt, type the following command to exit the Multiple Interface table, and then press Enter:

    (network-if-0)# **exit**

13. At the prompt, type the following command to exit the VoIP interface, and then press Enter:

    (config-voip)# **exit**

14. At the prompt, type the following command to save your changes and to reset the device, and then press Enter:

    # **reload**

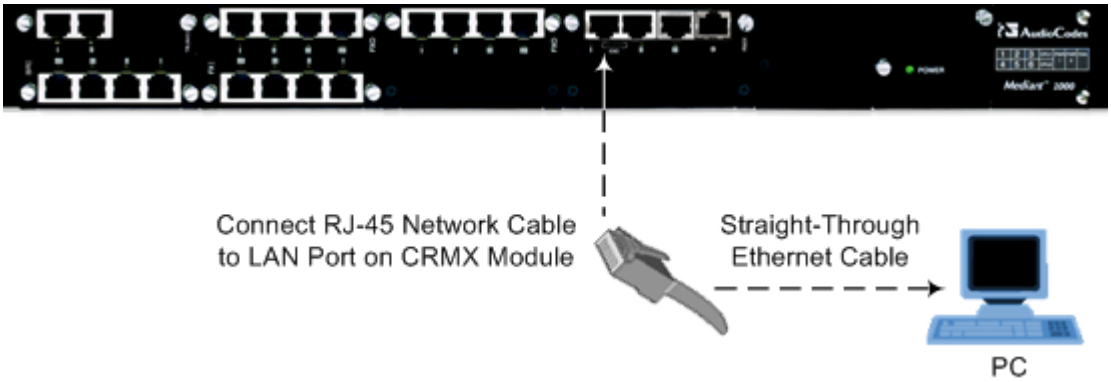
2.2 Using the Web Interface

The procedure below describes how to assign a LAN IP address, using the Web interface.

➢ **To assign an IP address using the Web interface:**

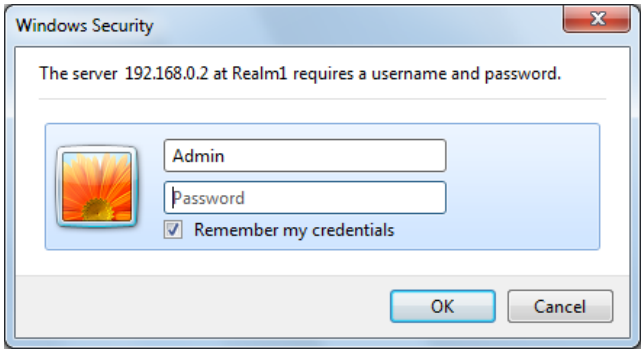
1. Disconnect any network cables from the device.
2. Connect one of the LAN ports of the device directly to the network interface of your computer, using a straight-through Ethernet cable.

Figure 2-2: Connecting LAN to PC for Initial Connectivity



3. Ensure that your computer is configured to automatically obtain an IP address. The device has an embedded DHCP server, which by default allocates an IP address to a connected computer.
4. On your computer, start a Web browser and in the URL address field, enter the default IP address of the device; the Web interface's Login screen appears:

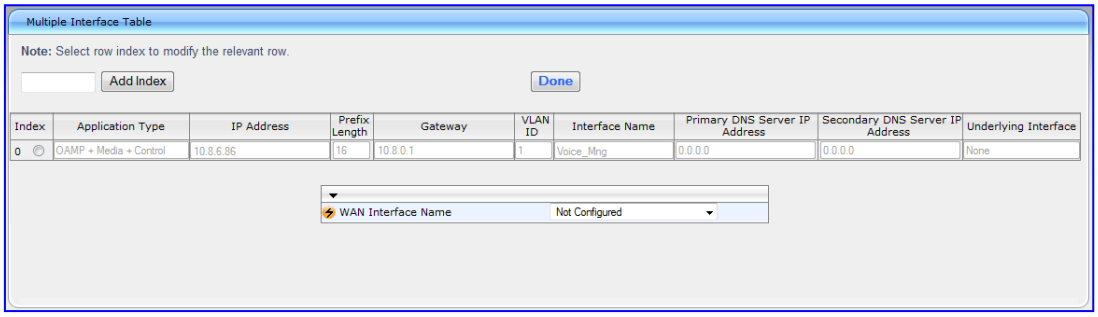
Figure 2-3: Login Screen



5. In the 'User Name' and 'Password' fields, enter the default login user name "Admin" (case-sensitive) and password "Admin" (case-sensitive), and then click **OK**; the device's Web interface is accessed.
6. Open the Multiple Interface Table page (**Configuration** tab > **VoIP** menu > **Network** submenu > **IP Settings**).

7. Select the 'Index' radio button corresponding to the "OAMP + Media + Control" application type, and then click **Edit**.

Figure 2-4: Multiple Interface Table



8. Change the IP address, subnet mask, and Default Gateway IP address to correspond with your network IP addressing scheme (e.g., 10.8.6.86).
9. Click **Apply**, and then click **Done** to validate your settings.
10. Assign the data-router LAN IP address and/or WAN IP address, as described in Assigning Data-Router LAN and WAN IP Addresses on page 34.

2.3 Using the FXS Voice Menu Guidance

You can assign an IP address that suits your networking scheme using a standard touch-tone telephone connected to one of the FXS ports. The voice menu can also be used to query and modify basic configuration parameters.

Notes:

- Assigning an IP address using the voice menu is applicable only when the device is installed with an FXS module.
- If you want to disable the voice menu, do one of the following:
  - Set the VoiceMenuPassword parameter to 'disable'.
  - Change the Web login password for the Admin user from its default value (i.e., "Admin") to any other value, and then reset the device.

➢ **To assign an IP address using the voice menu:**

1. Connect a telephone to one of the FXS ports.
2. Lift the handset and dial ***12345** (three stars followed by the digits 1, 2, 3, 4, and 5).
3. Wait for the 'configuration menu' voice prompt to be played.
4. To change the IP address:
   a. Press **1** followed by the pound key (#); the current IP address of the device is played.
   b. Press the # key.
   c. Dial the new IP address, using the star (*) key instead of periods (.), e.g., 192*168*0*4, and then press # to finish.
   d. Review the new IP address, and then press **1** to save.
5. To change the subnet mask:
   a. Press **2** followed by the # key; the current subnet mask of the device is played.
   b. Press the # key.
   c. Dial the new subnet mask (e.g., 255*255*0*0), and then press # to finish.
   d. Review the new subnet mask, and then press **1** to save.

6. To change the Default Gateway IP address:
   a. Press **3** followed by the # key; the current Default Gateway address is played.
   b. Press the # key.
   c. Dial the new Default Gateway address (e.g., 192*168*0*1), and then press # to finish.
   d. Review the new Default Gateway address, and then press **1** to save.
7. Hang up (on-hook) the handset.

Alternatively, initial configuration may be performed using an HTTP server, as discussed in the *Product Reference Manual* ('Automatic Update Facility'). The Voice Menu may be used to specify the configuration URL.

➢ **To set a configuration URL:**

1. Obtain the IP address of the configuration HTTP server (e.g., 36.44.0.6).
2. Connect a telephone to one of the FXS ports.
3. Lift the handset and dial ***12345** (three stars followed by the digits 1, 2, 3, 4, and 5).
4. Wait for the 'configuration menu' voice prompt to be played.
5. Dial **31** followed by the # key; the current IP address is played.
6. To change the IP address:
   a. Press the # key.
   b. Dial the configuration server's IP address. Use the star (*) key instead of dots ("."), e.g., 36*44*0*6, and then press # to finish.
   c. Review the configuration server's IP address, and then press **1** to save.
7. Dial **32** followed by the # key, and then do the following to change the configuration file name pattern:
   a. Press the # key.
   b. Select one of the patterns listed in the table below (*aa.bb.cc.dd* denotes the IP address of the configuration server):

#	Configuration File Name Pattern	Description
1	http://aa.bb.cc.dd/config.ini	Standard config.ini.
2	https://aa.bb.cc.dd/config.ini	Secure HTTP.
3	http://aa.bb.cc.dd/audiocodes/<MAC>.ini	The device's MAC address is appended to the file name (e.g., http://36.44.0.6/audiocodes/00908f012300.ini).
4	http://aa.bb.cc.dd:8080/config.ini	HTTP on port 8080.
5	http://aa.bb.cc.dd:1400/config.ini	HTTP on port 1400.
6	http://aa.bb.cc.dd/cgi-bin/acconfig.cgi?mac=<MAC>&ip=<IP>	Generating configuration per IP/MAC address dynamically, using a CGI script. See perl example below.

   a. Press the selected pattern code, and then press # to finish.
8. Press **1** to save, and then hang up the handset. The device retrieves the configuration from the HTTP server.

The following is an example perl CGI script, suitable for most Apache-based HTTP servers for generating configuration dynamically per pattern #6 above. Copy this script to /var/www/cgi-bin/acconfig.cgi on your Apache server and edit it as required:

```
#!/usr/bin/perl
use CGI;
$query = new CGI;
$mac = $query->param('mac');
$ip = $query->param('ip');
print "Content-type: text/plain\n\n";
print "; INI file generator CGI\n";
print "; Request for MAC=$mac IP=$ip\n\n";
print <<"EOF";
SyslogServerIP = 36.44.0.15
EnableSyslog = 1
SSHServerEnable = 1
EOF
```

The table below lists the configuration parameters that can be viewed and modified using the voice menu:

Table 2-2: Configuration Parameters Available via the Voice Menu

Item Number at Menu Prompt	Description
1	IP address.
2	Subnet mask.
3	Default Gateway IP address.
4	Primary DNS server IP address.
7	DHCP enable / disable.
31	Configuration server IP address.
32	Configuration file name pattern.
99	Voice menu password (initially 12345). **Note:** The voice menu password can also be changed using the Web interface or *ini* file parameter VoiceMenuPassword.

3 Assigning Data-Router LAN and WAN IP Addresses

Once you have changed the VoIP LAN IP address to suit your networking scheme, you need to configure the data-router LAN IP address and/or WAN IP address. You can setup your device for Web management access using one of the following interfaces:

- Any of the configured data-router LAN interfaces. The default LAN data interface is 192.168.0.1.
- WAN interface. In this setup, you need to enable remote HTTP/S access to the WAN port interface.

The procedure for basic setup and configuring the data-router LAN IP address and/or WAN IP address is as follows:

1. Configure the Data-Router LAN interface - see 'Configuring Data-Routing LAN Interface' on page 35
2. Configure the DHCP server - see 'Configuring Device's DHCP Server' on page 38
3. Configure the WAN IP address - see 'Assigning a WAN IP Address' on page 38
4. Assign the WAN interface to the VoIP traffic - see 'Assigning WAN Interface to VoIP Traffic' on page 39
5. Configure Quality of Service - see 'Configuring Quality of Service' on page 41
6. Configure VRF - see 'Configuring Virtual Routing and Forwarding' on page 48
7. Enable remote Web management - see 'Enabling Remote HTTP/S Web Management' on page 48

3.1 Configuring Data-Routing LAN Interface

This section describes how to assign an IP address to the data-router LAN interface. The default LAN IP address of the data-router is listed in the table below.

Table 3-1: Default IP Address of LAN Data-Router

Parameter	Default Value
IP Address	192.168.0.1
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0

Note: The LAN IP address of the data-router must be in the same subnet as the LAN IP address of the VoIP interface.

3.1.1 Using CLI

The procedure below describes how to configure the data-router LAN IP address, using the CLI.

➢ **To assign an IP address to the LAN data-router using CLI:**

1. Establish serial communication with the device.
2. At the prompt, type the following command to access the data-router interface, and then press Enter:

```
# configure data
```

3. At the prompt, type the following command to access the VLAN 1 LAN switch interface, and then press Enter:

```
(config-data)# interface vlan 1
```

4. At the prompt, type the following command to set the IP address and subnet, and then press Enter:

```
(conf-if-VLAN 1)# ip address <IP address> <subnet>
```

5. At the prompt, type the following command to exit the interface mode, and then press Enter:

```
(conf-if-VLAN 1)# exit
```

6. At the prompt, type the following command to exit the data-router interface, and then press Enter:

```
(config-data)# exit
```

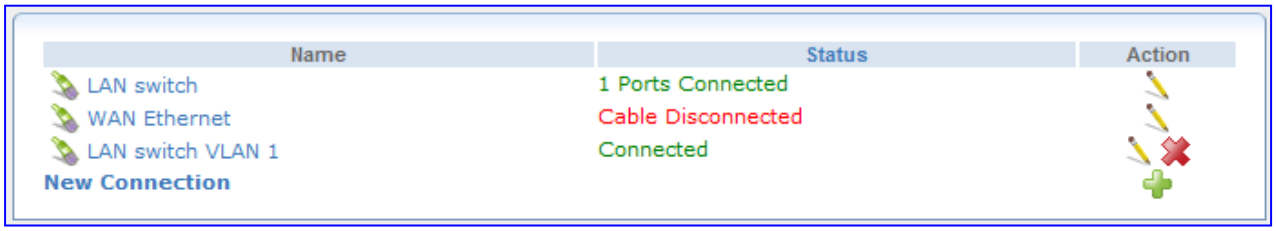
3.1.2 Using Web Interface

The procedure below describes how to configure the data-router LAN IP address, using the Web interface.

➢ **To assign an IP address to the LAN data-router using the Web interface:**

1. Access the Web interface using the new VoIP LAN IP address.
2. Open the Connections page (**Configuration** tab > **Data** menu > **Data System** > **Connections**).

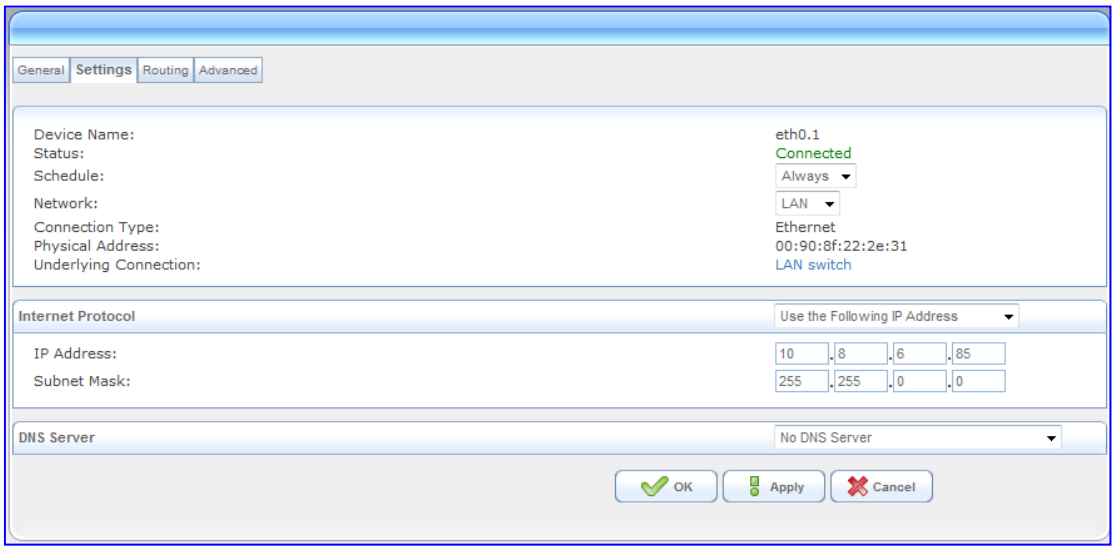
Figure 3-1: Connections Page



3. Click the **Edit** button corresponding to the "LAN Switch VLAN 1" connection, and then click the **Settings** tab.

4. In the 'IP Address' and 'Subnet Mask' fields, enter the required IP address and subnet respectively.

Figure 3-2: Defining LAN Data Routing IP Address



5. Click **OK** to save your settings.

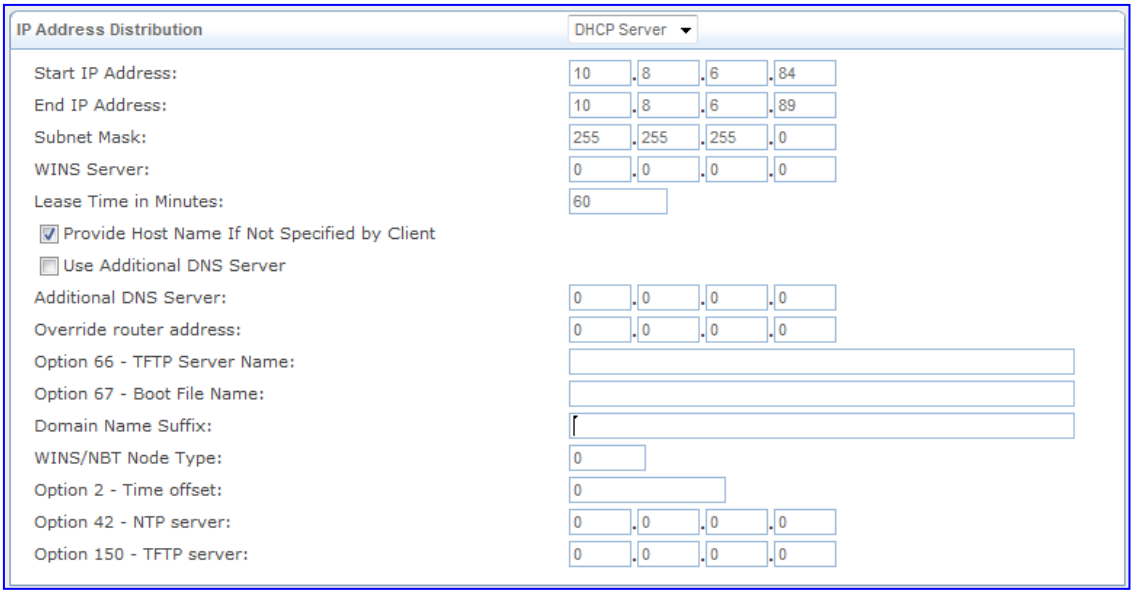
3.2 Configuring Device's DHCP Server

The device is supplied with an enabled embedded DHCP server for the LAN, and with default IP pool addresses relating to the default subnet LAN. You can either disable the DHCP server or modify the IP address pool accordingly. The device's DHCP server uses this setting to allocate IP addresses to the computers connected to its LAN interface.

➢ **To configure DHCP on the device:**

1. Open the DHCP Server page (**Configuration** tab > **Data** menu > **Data Services** > **DHCP Server**).
2. Click the **Edit** button corresponding to the "LAN Switch VLAN 1" connection.
3. From the 'IP Address Distribution' drop-down list, select **DHCP Server**.
4. Define the IP address pool.

Figure 3-3: Configuring the DHCP Server



5. Click **OK**.
6. If required, refresh the address by disconnecting the cable and then reconnecting it again, or by performing the following in Windows' command line interface:

```
ipconfig /release
ipconfig /renew
```

3.3 Assigning a WAN IP Address

If you require remote management through the WAN interface, then follow the procedure described below.

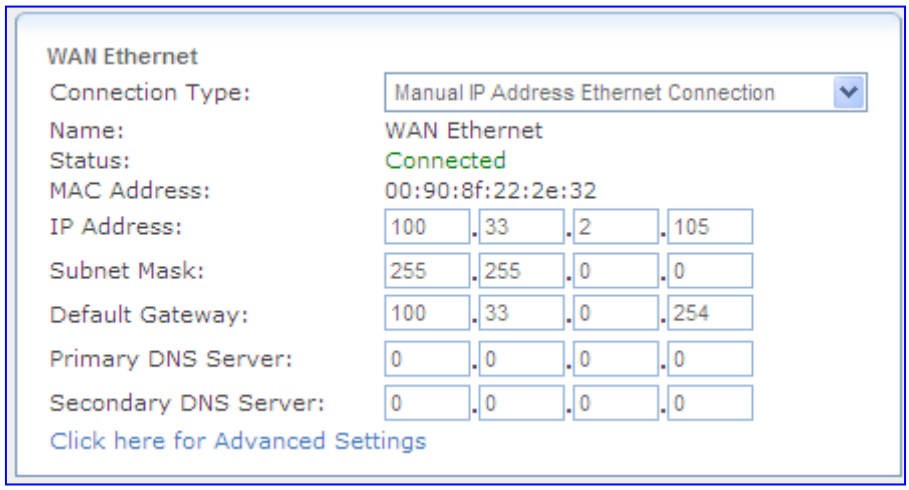
Note: Before you configure the WAN interface connection, ensure that you have all the required information from your Internet Service Provider (ISP).

➢ **To assign a WAN IP address:**

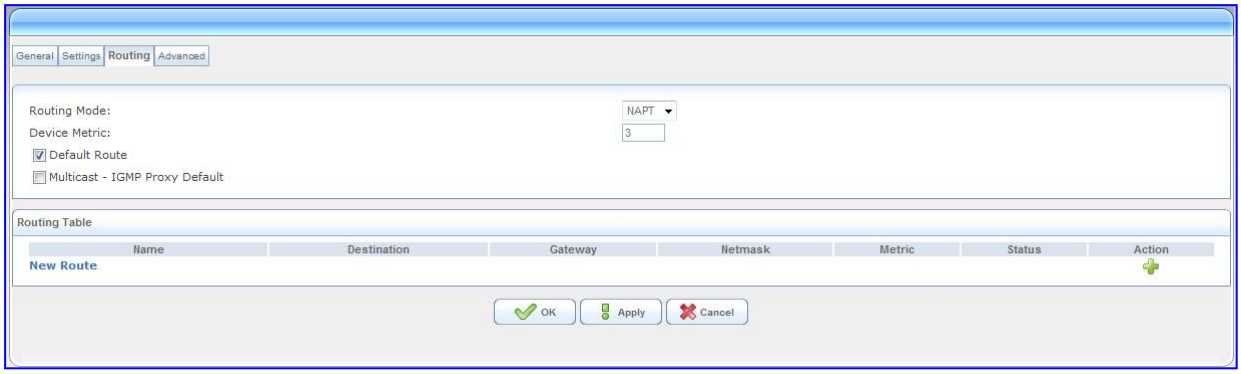
1. Connect the WAN port to the WAN network (i.e., ADSL or Cable modem). For information on cabling the WAN port, refer to the *Hardware Installation Manual*.

2. Open the Settings page (**Configuration** tab > **Data** menu > **WAN Access** > **Settings**).

Figure 3-4: Selecting WAN Connection



3. From the 'Connection Type' drop-down lists, select the required WAN connection type, and then configure the subsequent parameters (e.g., IP address of 100.33.2.105).
4. If you configured the WAN interface to obtain an IP address from a DHCP server, then wait until the acquired IP address is displayed and make a note of it. You need this IP address for accessing the device in subsequent sessions.
5. Configure the WAN interface operating mode for Network Address Port Translation (NAPT):
   a. Click the **Click here for Advanced Settings** link, and then select the **Routing** tab; the Routing page appears:



   b. From the 'Routing Mode' drop-down list, select **NAPT**.
   c. Select the 'Default Route' check box.
   d. Click **OK**.

3.4 Assign WAN Interface to VoIP Traffic

If you require remote management through the WAN interface, then follow the procedure described in this section.

Once you have configured the WAN IP address, you need to associate it with VoIP traffic (i.e., SIP signaling and media / RTP interfaces). The available WAN interfaces depend on the hardware configuration (e.g., Ethernet or SHDSL) and/or whether VLANs are configured for the WAN interface. If VLANs are configured, then you can select the WAN VLAN on which you want to run the SIP signaling and media interfaces.

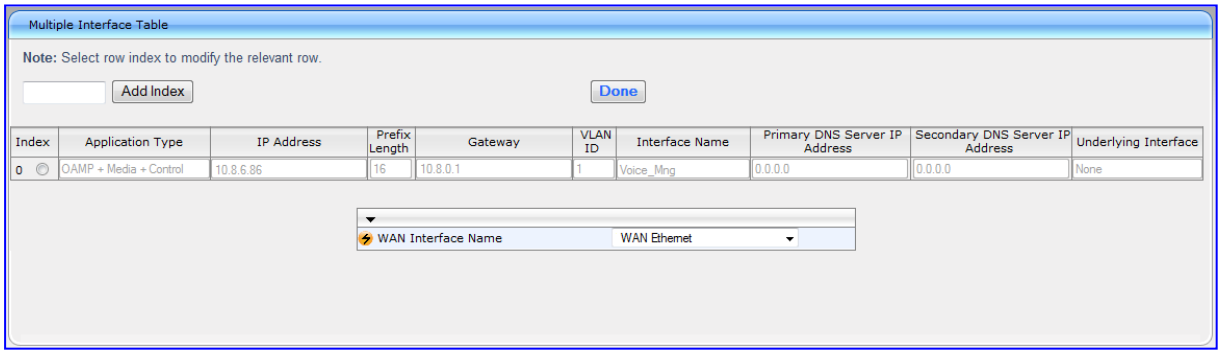
Once this association is set, VoIP traffic is sent through the WAN and incoming traffic is identified as coming from the WAN. The device automatically configures the required port forwarding and static NAT rules.

Note: If you do not assign the WAN interface to SIP and media interfaces, then the WAN interface may not be used for VoIP traffic. In such cases, the VoIP traffic can be sent and received within the LAN, or sent to the WAN through a third-party LAN router. If a third-party router is used as the interface to the WAN, then you need to configure NAT rules (in the NAT Translation table) to translate the VoIP LAN IP addresses (configured in the Multiple Interface table and associated with SIP and media interfaces) into global, public IP addresses.

➢ To assign a WAN interface to VoIP traffic:

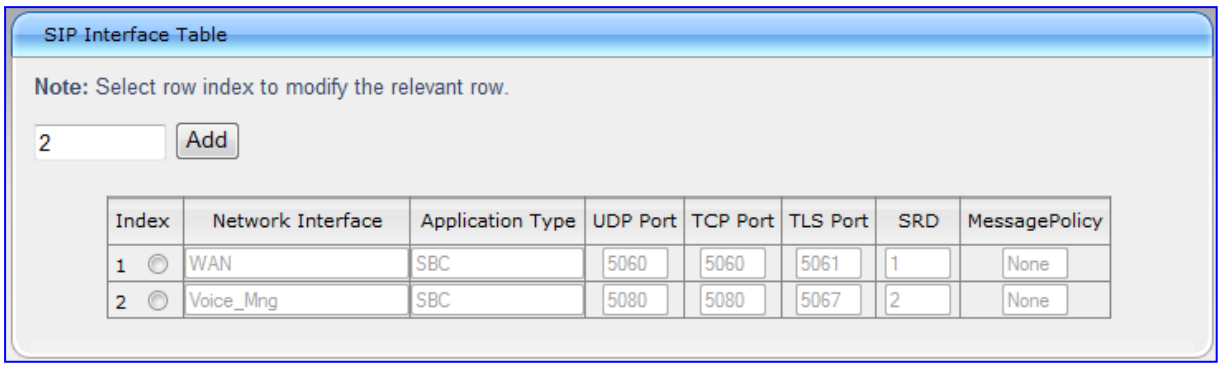
1. Select the WAN interface:
   a. Open the Multiple Interface Table page (**Configuration** tab > **VoIP** menu > **Network** submenu > **IP Settings**).

Figure 3-5: Selecting WAN Interface for VoIP Traffic



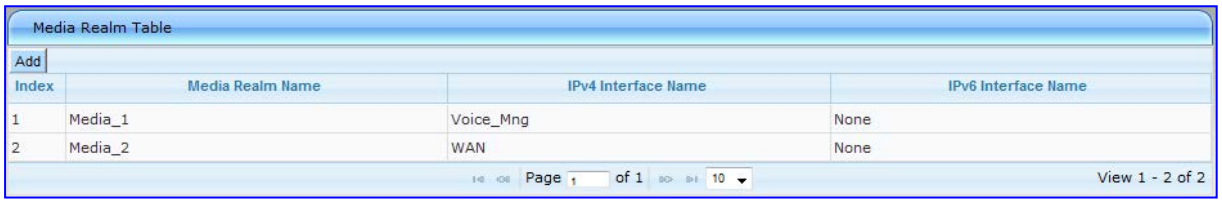
   b. From the 'WAN Interface Name' drop-down list, select the WAN interface for VoIP traffic.
   c. Click **Done**, and then reset the device for your setting to take effect.
2. Assign the selected WAN interface to SIP signaling and RTP (media) interfaces. This is done in the SIP Interface and Media Realm tables respectively (whereby the WAN interface is denoted as "WAN"):
   a. Open the SIP Interface Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **SIP Interface Table**) and configure SIP interfaces on the WAN interface.

Figure 3-6: Assigning SIP Interface to WAN



b. Open the Media Realm Table page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Media Realm Configuration**) and configure media interfaces on the WAN interface.

Figure 3-7: Assigning WAN Interface to Media Realm



c. Configure SRDs and associate them with these SIP signaling and media interfaces. Configure other SIP settings as required.

3.5 Configuring Quality of Service

This section describes how to configure the device to guarantee appropriate handling of VoIP services. As VoIP traffic is delay-sensitive, you need to prioritize VoIP traffic over other classes of traffic. This is achieved by configuring Quality of Service (QoS).

The QoS configuration includes the following stages:

- Configuring the total WAN bandwidth (typically assigned by your ITSP) - see 'Defining Total WAN Bandwidth' on page 41
- Configuring traffic shaping classes with the minimum bandwidth guaranteed for VoIP traffic - see 'Defining VoIP Tx Shaping Class' on page 42
- Assigning the above VoIP traffic class to VoIP RTP media packets and VoIP signaling (UDP, TCP, and TLS) traffic matching rules - 'Defining VoIP Traffic Matching Rules' on page 44

3.5.1 Defining Total WAN Bandwidth

To define traffic shaping on the device so that packets will not be “dropped” by your ITSP, you should configure your device with the total WAN bandwidth allocated by your ITSP. In other words, the ITSP can be considered the bottleneck of the network and thus, the device needs to accommodate its outgoing traffic to this bandwidth.

Note: When the device uses T1 WAN interface, the maximum T1 WAN Tx bandwidth per physical T1 line is 1.544 Mbps.

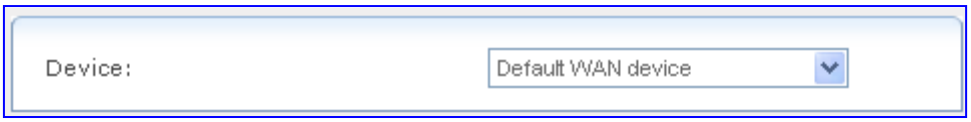
➢ **To define the WAN bandwidth traffic shaping class:**

1. Open the Traffic Shaping page (**Configuration** tab > **Data** menu > **QoS** > **Traffic Shaping**).

Figure 3-8: Traffic Shaping Page

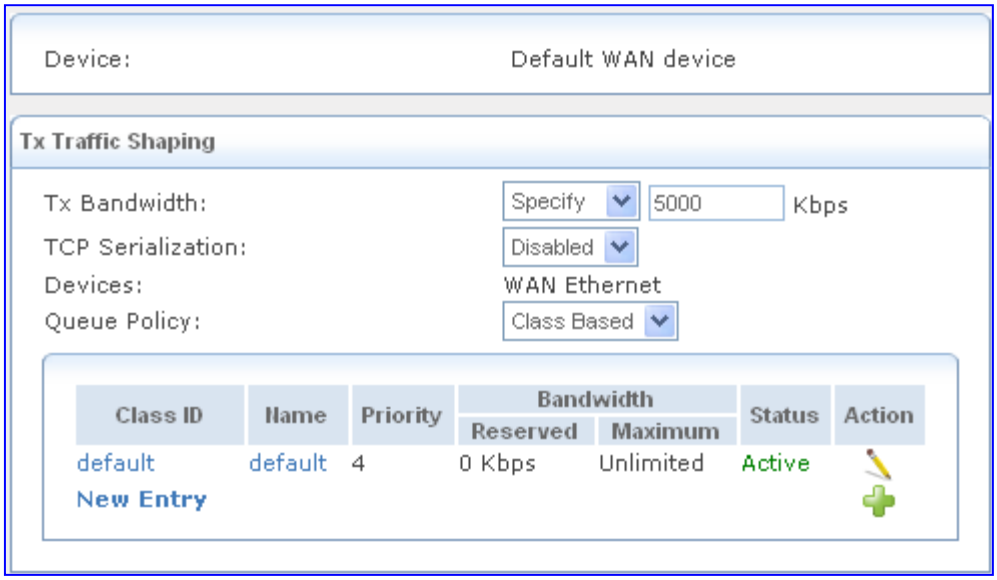
2. Click the **New** button; the following page appears.

Figure 3-9: Selecting Device for Traffic Shaping



3. From the 'Device' drop-down list, select **Default WAN device**, and then click **OK**; the following page appears:

Figure 3-10: Defining Traffic Shaping



4. In the 'Tx Bandwidth' field, specify the total WAN bandwidth (in Kbps) allocated by your ITSP. For example, 5 Mbps (5000 Kbps), as shown in the figure above.
5. Click **OK**.

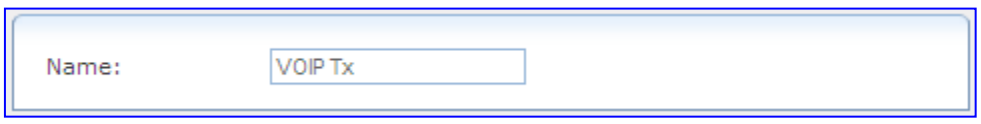
3.5.2 Defining VoIP Tx Traffic Shaping Classes

Once you have defined the total bandwidth allocated to the device's WAN interface, you need to define a traffic shaping class to reserve a minimum bandwidth (e..g., of 1 Mbps) for Tx VoIP traffic (SIP signaling and RTP packets) out of the total bandwidth (e.g., 5 Mbps).

➢ **To define traffic shaping class for VoIP traffic:**

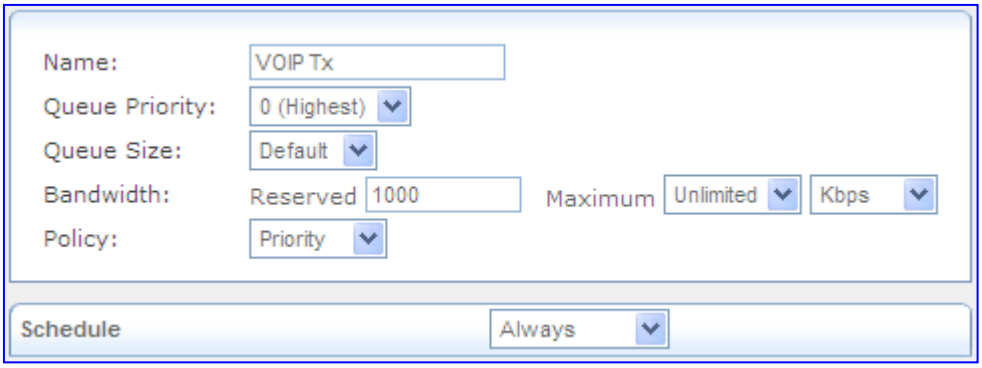
1. Open the Traffic Shaping page (see 'Defining Total WAN Bandwidth' on page 41) and then click the **Edit** button corresponding to the 'Default WAN Device' entry.
2. In the 'Tx Traffic Shaping' group, click the **New** button.
3. Assign a name to the new class (e.g., "VOIP Tx"), and then click **OK**; the page closes and you are returned to the previous page.

Figure 3-11: Adding Class Rule



4. Click the newly added class name; the following page appears:

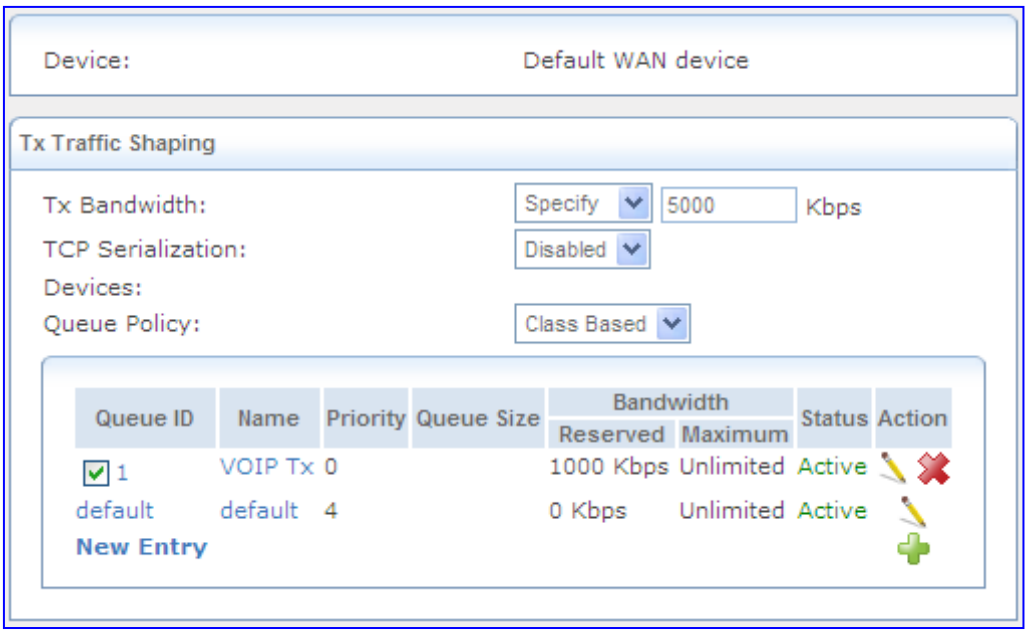
Figure 3-12: Defining Shaping Class (for VoIP Tx Traffic)



5. Configure the following:
   a. From the 'Queue Priority' drop-down list, select **0 (Highest)**, i.e., the highest priority.
   b. In the 'Bandwidth - Reserved' field, enter "1000" (i.e., 1 Mbps).
   c. From the 'Maximum' drop-down list, select **Unlimited**.
   d. From the 'Policy' drop-down list, select **Priority**.
   e. Click **OK**.

The figure below shows an example of the configured traffic shaping classes for Tx VoIP traffic.

Figure 3-13: Configured Traffic Shaping for Total WAN and VoIP Bandwidth



3.5.3 Defining VoIP Traffic Matching Rules

Once you have configured the VoIP Tx traffic shaping class (e.g., "VOIP Tx"), you need to configure traffic matching rules (QoS outbound rules) for VoIP RTP media traffic and SIP signaling traffic, and then assign the shaping class to these traffic rules.

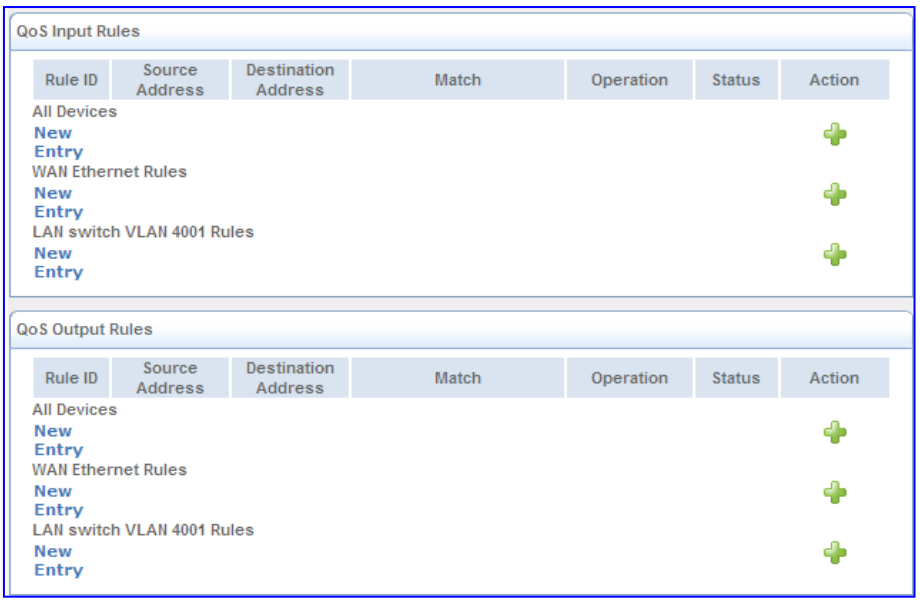
The following matching rules need to be configured (assuming standard ports are used):

- **SIP TCP connections:**
  - Incoming TCP connection to WAN port 5060-5061 must be matched to Traffic Shaping class "VOIP Tx". Outbound packets sent on this connection will originate from port 5060.
  - Outbound TCP connections from WAN port 5060-5061 sent to any destination must be matched to Traffic Shaping class "VOIP Tx". Outbound packets on this connection will originate from an arbitrary port, but will be destined to port 5060-5061.
- **SIP UDP connections:** Outbound UDP packets sent from WAN port 5060 to any destination must be matched to Traffic Shaping class "VOIP Tx".
- **RTP UDP:** Outbound UDP packets sent from WAN port 7000-8000 to any destination must be matched to Traffic Shaping class "VOIP Tx".

➢ **To define VoIP traffic matching rules:**

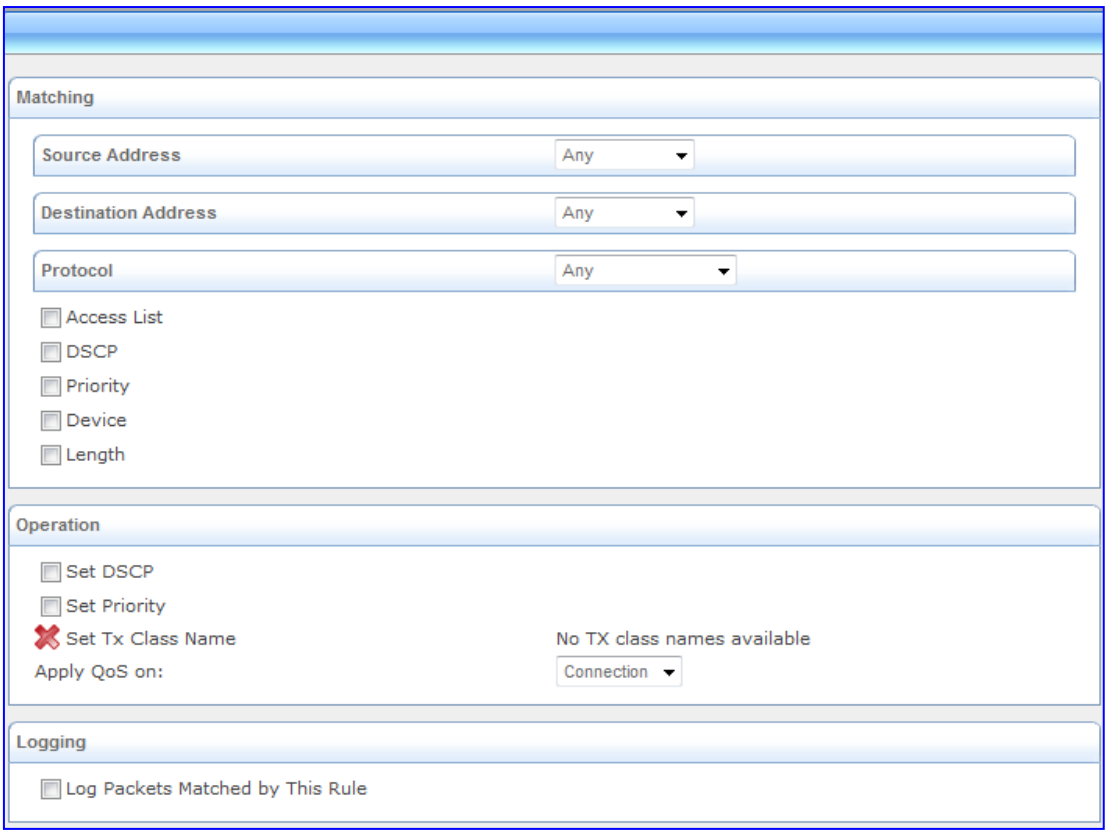
1. Open the Match Rules page (**Configuration** tab > **Data** menu > **QoS** > **Match Rules**).

Figure 3-14: Match Rules Page



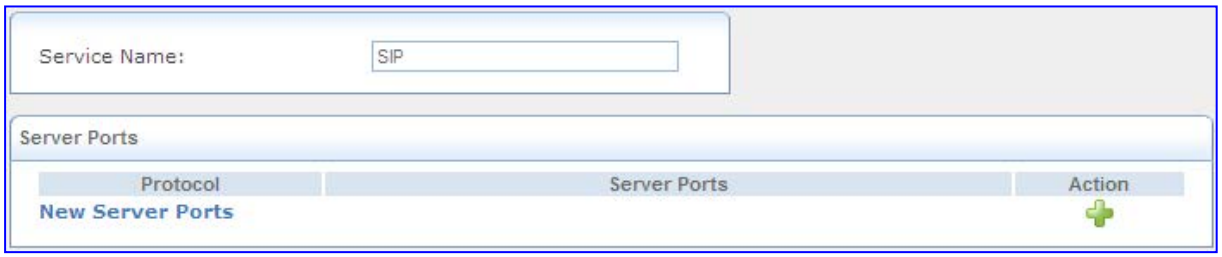
2. In the QoS Output Rules table, click the **New** button corresponding to the 'WAN Ethernet Rules' rule ID; the following page appears:

Figure 3-15: Adding a Traffic Priority Rule



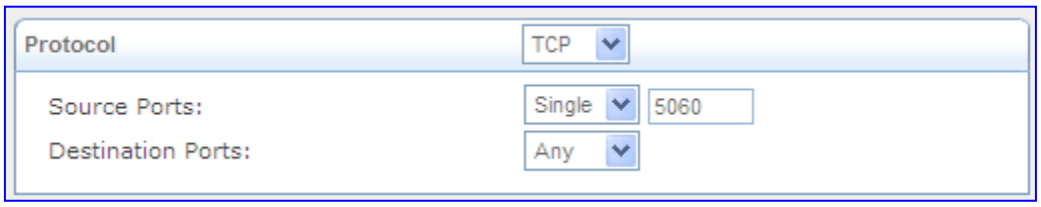
3. Add a new output traffic rule for VoIP SIP signaling to the WAN:
   a. From the 'Protocol' drop-down list, select **Show All Services** to view all protocols and then select **SIP**; "SIP" is displayed under the **Protocol** group.
   b. From the drop-down list appearing below "SIP", select **User Defined**; the following appears.

Figure 3-16: Defining Incoming SIP Ports



   c. In the 'Service Name' field, enter the service, and then click **New Server Ports**; the following page appears.

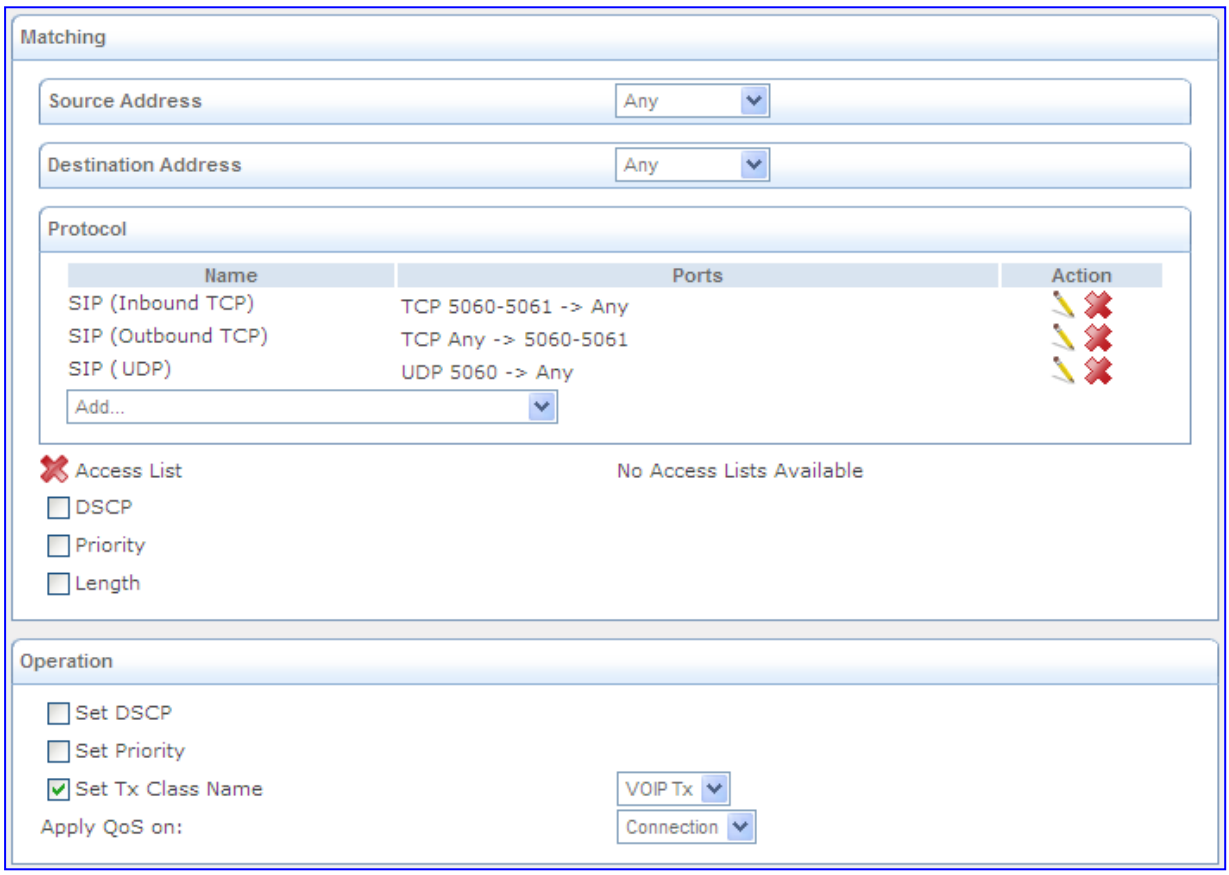
Figure 3-17: Defining SIP Ports (e.g. TCP)



d. Define the TCP and UDP protocol and their ports, and then click **OK**.
e. Click **OK** again.
f. Perform steps 2b through 2e to configure ports for outgoing SIP packets.
g. Under the **Operation** group, select the 'Set Tx Class Name' check box, and then from the corresponding drop-down list, select the traffic shaping class **VOIP Tx**, which you defined for Rx packets (in 'Defining VoIP Tx Shaping Class' on page 42).

   The configured incoming and outgoing SIP ports are shown below:

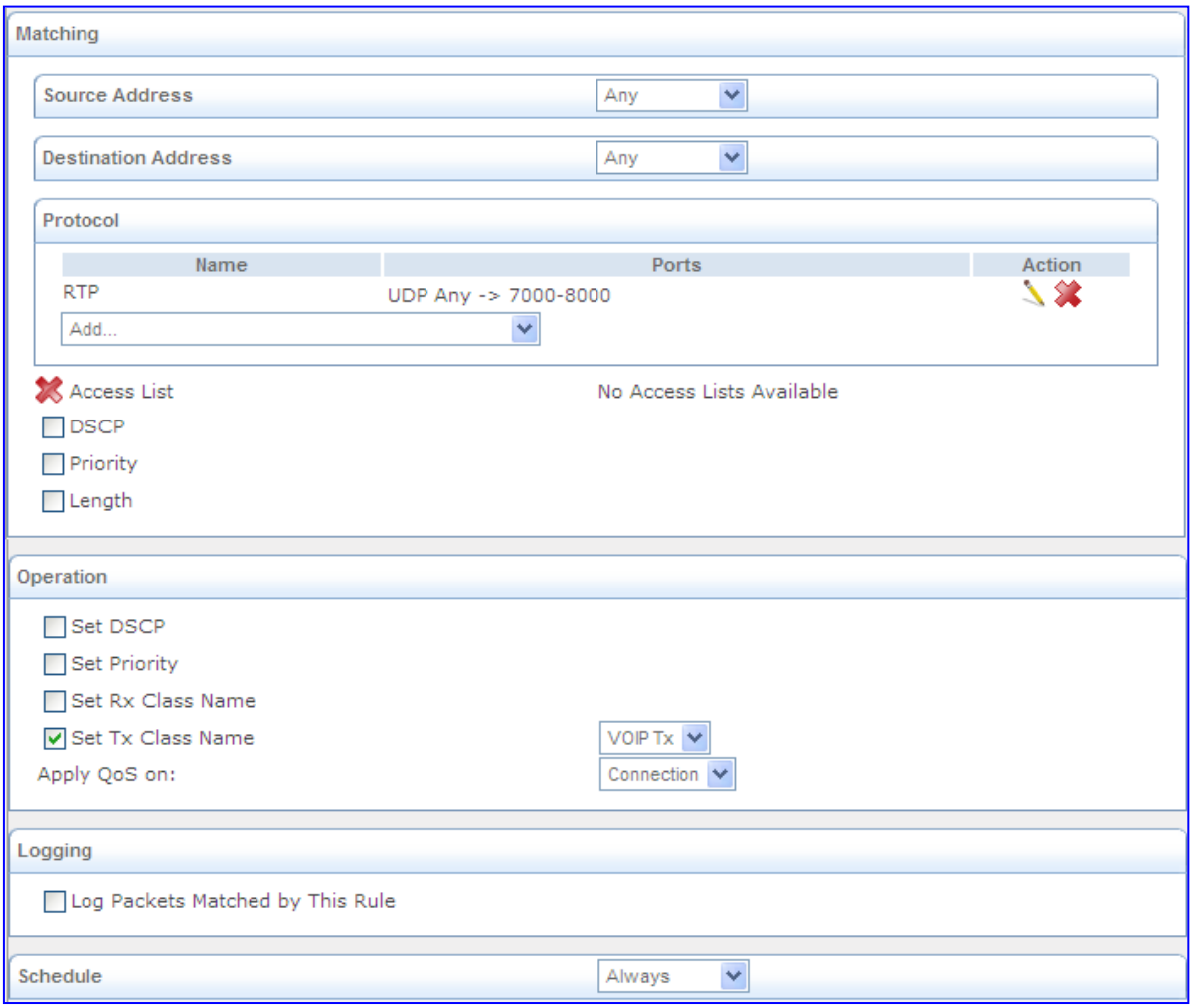
Figure 3-18: Traffic Matching Rule for Received SIP Signaling Traffic



h. Click **OK**.

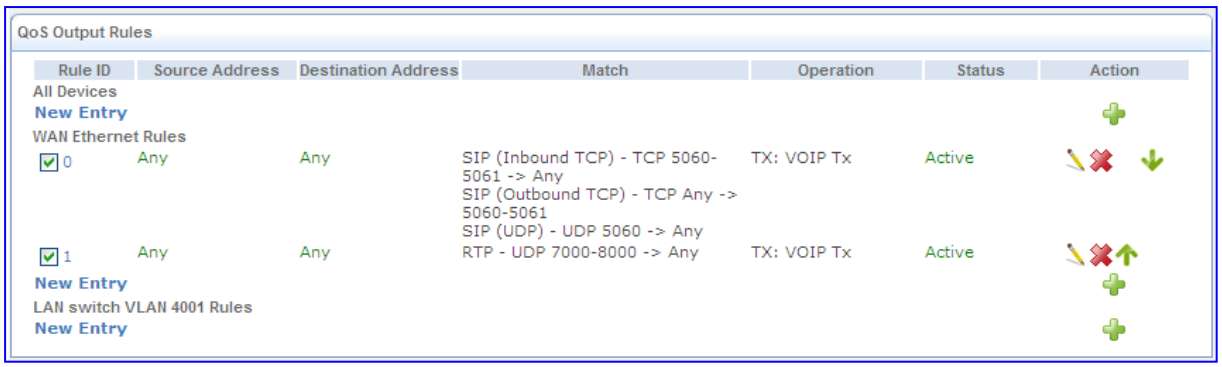
4. Add a new traffic matching rule for transmitted (Tx) VoIP RTP packets to the WAN. Do steps 2 through 3, but for the **Protocol** group, select 'RTP' and only port 'UDP', as shown below.

Figure 3-19: Matching Rule for Received RTP Traffic



The final traffic matching rule configuration for WAN Tx RTP and SIP signaling is shown below.

Figure 3-20: Traffic Matching Rule for WAN Tx/Rx RTP and SIP Signaling



3.6 Configuring Virtual Routing and Forwarding

The Virtual Routing and Forwarding (VRF) Lite feature enables the ability to use a single physical router as several logical routers. Each VRF is associated with its own routing table. To create fully separated logical routers on the same physical router, every interface can be mapped to a specified VRF and static routes can be added to it.

For VRF configuration, refer to the *MSBG Series CLI Reference Guide*.

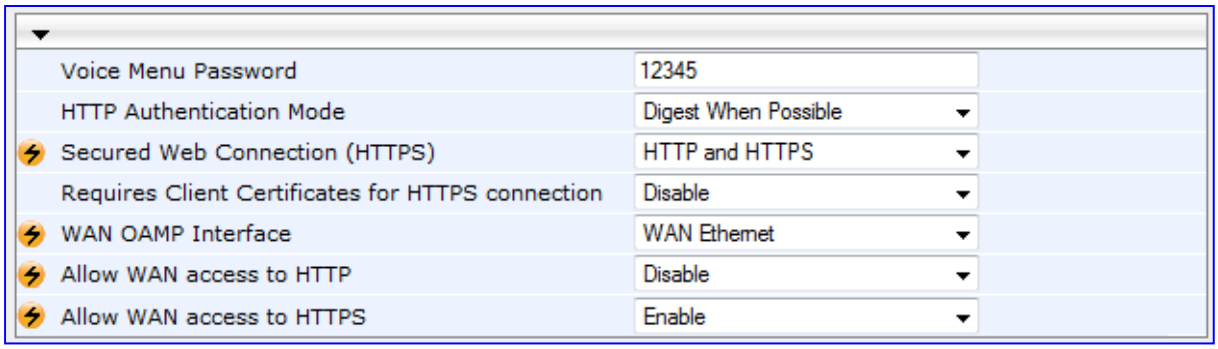
3.7 Enabling Remote HTTP/S Web Management

To access the Web interface remotely through the WAN port using HTTP or HTTPS, you need to enable remote management as described in the procedure below.

➢ **To define WAN HTTP/S port for remote Web management:**

1. Open the WEB Security Settings page (**Configuration** tab > **System** menu > **Management** submenu > **WEB Security Settings**).

Figure 3-21: Defining WAN HTTP Port



2. From the 'WAN OAMP Interface' drop-down list, select the WAN interface.
3. From the 'Allow WAN access to HTTPS' drop-down list, select **Enable**.
4. Click **Submit**.

3.8 Reset the Device and Reconnect with WAN

When you have completed all the procedures described in the previous sections, you need to reset the device to apply your changes. Once the settings are applied, you can access the device's management interface through the WAN IP address.

Part II

Management Tools

This part provides an overview of the various management tools that can be used to configure the device and describes how to configure the management settings. The following management tools can be used to configure the device:

- Embedded HTTP/S-based Web server - see 'Web-based Management' on page 51
- Command Line Interface (CLI) - see 'CLI-Based Management' on page 85
- Configuration *INI* file - see 'INI File-Based Management' on page 97
- AudioCodes Element Management System - see 'EMS-Based Management' on page 95
- Simple Network Management Protocol (SNMP) browser software - see 'SNMP-Based Management' on page 89

Note: Some configuration settings can only be done using specific management tools. For example, the *ini* file method provides many parameters that are not supported in the Web interface.

Reader's Notes

4 Web-Based Management

The device's embedded Web server (hereafter referred to as the *Web interface*) provides FCAPS (fault management, configuration, accounting, performance, and security) functionality. The Web interface allows you to remotely configure the device for quick-and-easy deployment, including the loading of software (.cmp), configuration (.ini), and auxiliary files. The Web interface provides real-time, online monitoring of the device, including display of alarms and their severity. In addition, the Web interface displays performance statistics of voice calls, data routing, and various traffic parameters.

The Web interface provides a user-friendly, graphical user interface (GUI), which can be accessed using any standard Web browser (e.g., Microsoft™ Internet Explorer). Access to the Web interface is controlled by various security mechanisms such as login user name and password, read-write privileges, and limiting access to specific IP addresses.

Notes:

- For a detailed description of all the parameters in the Web interface, see 'Configuration Parameters Reference' on page 752.
- The parameters in the Web interface can alternatively be configured using their corresponding *ini* file parameters, which are enclosed in square brackets "[...]" in 'Configuration Parameters Reference' on page 752.
- The Web interface allows you to configure most of the device's settings. However, additional configuration parameters may exist that are not provided in the Web interface and which can only be configured using *ini* file parameters. These parameters are listed without a corresponding Web parameter name in 'Configuration Parameters Reference' on page 752.
- Some Web interface pages are Software Upgrade Key dependant. These pages appear only if the installed Software Upgrade Key supports the features related to the pages. For viewing your Software Upgrade Key, see 'Loading Software Upgrade Key' on page 696.

4.1 Getting Acquainted with the Web Interface

This section provides a description of the Web interface, including the areas of the GUI, navigation, and configuration methods.

4.1.1 Computer Requirements

The client computer requires the following to work with the Web interface of the device:

- A network connection to the device.
- One of the following Web browsers:
  - Microsoft™ Internet Explorer™ (version 6.0 or later)
  - Mozilla Firefox® (versions 2 or 3)
- The following recommended screen resolutions: 1024 x 768 pixels, or 1280 x 1024 pixels.

Note: Your Web browser must be JavaScript-enabled to access the Web interface.

4.1.2 Accessing the Web Interface

The procedure below describes how to access the Web interface.

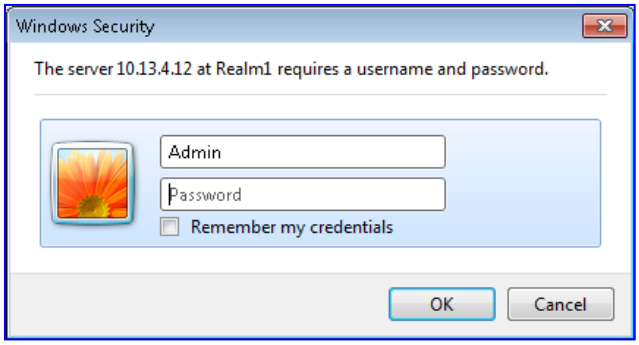
When initially accessing the Web interface, use

Note: For assigning an IP address to the device, refer to the *Installation Manual*.

➢ **To access the Web interface:**

1. Open a standard Web browser (see 'Computer Requirements' on page 51).
2. In the Web browser, specify the IP address of the device (e.g., http://10.1.10.10); the Web interface's Login window appears, as shown below**:**

Figure 4-1: Login Screen



3. In the 'User Name' and 'Password' fields, enter the case-sensitive, user name and password respectively.

Notes:

- The default user name and password is "Admin". To change the login user name and password, see 'Configuring the Web User Accounts' on page 76.
- If you want the Web browser to remember your password, select the 'Remember my credentials' check box. The next time you log in to the Web interface, instead of entering your credentials as described in Step 3 above, all you need to do is to click **OK** twice in succession.

4. Click **OK**; the Web interface is accessed, displaying the Home page (for a detailed description of the Home page, see 'Using the Home Page' on page 70).

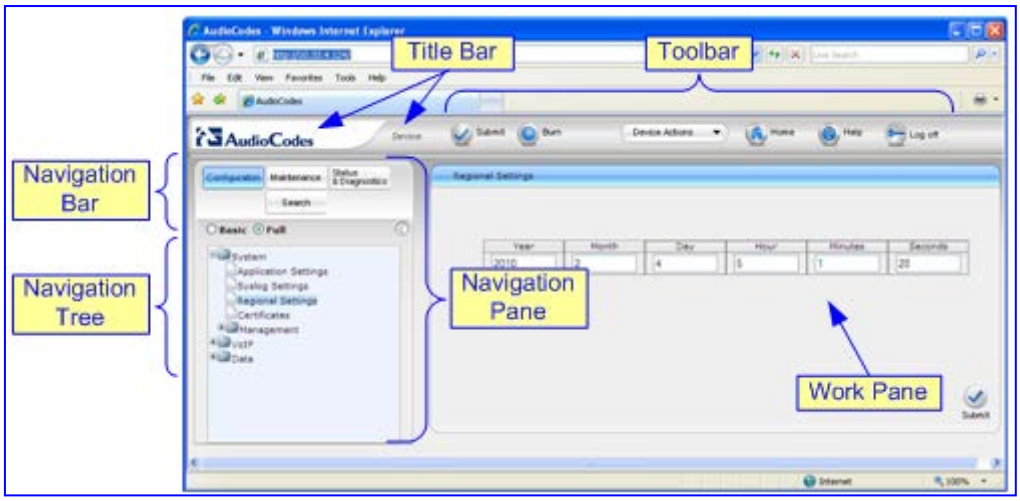
Note: If access to the Web interface is denied ("Unauthorized") due to Microsoft Internet Explorer security settings, do the following:

1. Delete all cookies in the Temporary Internet Files folder. If this does not resolve the problem, the security settings may need to be altered (continue with Step 2).
2. In Internet Explorer, navigate to **Tools** menu > **Internet Options** > **Security** tab > **Custom Level**, and then scroll down to the Logon options and select **Prompt for username and password**. Select the **Advanced** tab, and then scroll down until the HTTP 1.1 Settings are displayed and verify that **Use HTTP 1.1** is selected.
3. Quit the Web browser and start it again.

4.1.3 Areas of the GUI

The figure below displays the areas of the Web interface GUI:

Figure 4-2: Areas of the Web GUI



The Web GUI consists of the following main areas:

- **Title bar:** Displays the corporate logo image and product name.
- **Toolbar:** Provides frequently required command buttons (see 'Toolbar Description' on page 54).
- **Navigation Pane:** Includes the following areas:
  - **Navigation bar:** Provides tabs for accessing the configuration menus (see 'Navigation Tree' on page 55) and searching Web interface parameters (see 'Searching for Configuration Parameters' on page 66).
  - **Navigation tree:** Displays the elements pertaining to the selected tab on the Navigation bar (tree-like structure of the configuration menus or Search engine).
- **Work pane:** Displays configuration pages in which configuration is done (see 'Working with Configuration Pages' on page 58).

4.1.4 Toolbar Description

The toolbar provides frequently required command buttons, as described in the table below:

Table 4-1: Description of Toolbar Buttons

Icon	Button Name	Description
	Submit	Applies parameter settings to the device (see 'Saving Configuration' on page 680). **Note:** This icon is grayed out when not applicable to the currently opened page.
	Burn	Saves parameter settings to flash memory (see 'Saving Configuration' on page 680).
	Device Actions	Opens a drop-down menu list with frequently needed commands: ▪ **Load Configuration File:** opens the Configuration File page for loading an *ini* file (see 'Backing Up and Loading Configuration File' on page 702). ▪ **Save Configuration File:** opens the Configuration File page for saving the *ini* file to a folder on a computer (see 'Backing Up and Loading Configuration File' on page 702). ▪ **Reset:** opens the Maintenance Actions page for resetting the device (see 'Resetting the Device' on page 677). ▪ **Software Upgrade Wizard:** starts the Software Upgrade wizard for upgrading the device's software (see 'Software Upgrade Wizard' on page 699).
	Home	Opens the Home page (see 'Using the Home Page' on page 70).
	Help	Opens the Online Help topic of the currently opened configuration page (see 'Getting Help' on page 68).
	Log off	Logs off a session with the Web interface (see 'Logging Off the Web Interface' on page 69).

Note: If you modify parameters that take effect only after a device reset, after you click the **Submit** button, the toolbar displays "Reset" (in red color), as shown in the figure below. This is a reminder that you need to later save your settings to flash memory and reset the device.

Figure 4-3: "Reset" Displayed on Toolbar

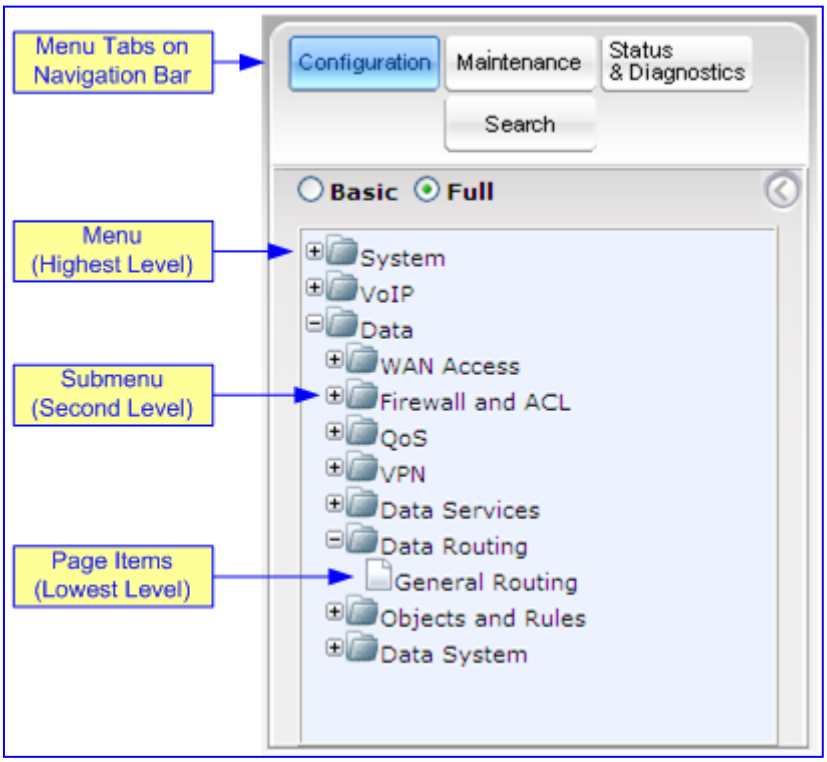
4.1.5 Navigation Tree

The Navigation tree is located in the Navigation pane. It displays the menus pertaining to the selected menu tab on the Navigation bar and is used for accessing the configuration pages. The Navigation tree displays a tree-like structure of menus. You can drill-down to the required page item level to open its corresponding page in the Work pane.

The terminology used throughout this manual for referring to the hierarchical structure of the tree is as follows:

- *menu*: first level (highest level)
- *submenu*: second level - contained within a menu
- *page item*: last level (lowest level in a menu) - contained within a menu or submenu

Figure 4-4: Navigation Tree



➢ **To view menus in the Navigation tree:**

- On the Navigation bar, select the required tab - **Configuration**, **Maintenance**, or **Status & Diagnostics**.

➢ **To navigate to a page:**

1. Navigate to the required page item, by performing the following:
   - Drilling-down using the **plus** ⊞ sign to expand the menu and submenus.
   - Drilling-up using the **minus** ⊟ sign to collapse the menu and submenus.
2. Select the required page item; the page opens in the Work pane.

4.1.5.1 Displaying Navigation Tree in Basic and Full View

You can view an expanded or reduced Navigation tree display regarding the number of listed menus and submenus. This is relevant when using the configuration tabs (**Configuration**, **Maintenance**, and **Status & Diagnostics**) on the Navigation bar.

The Navigation tree menu can be displayed in one of two views:

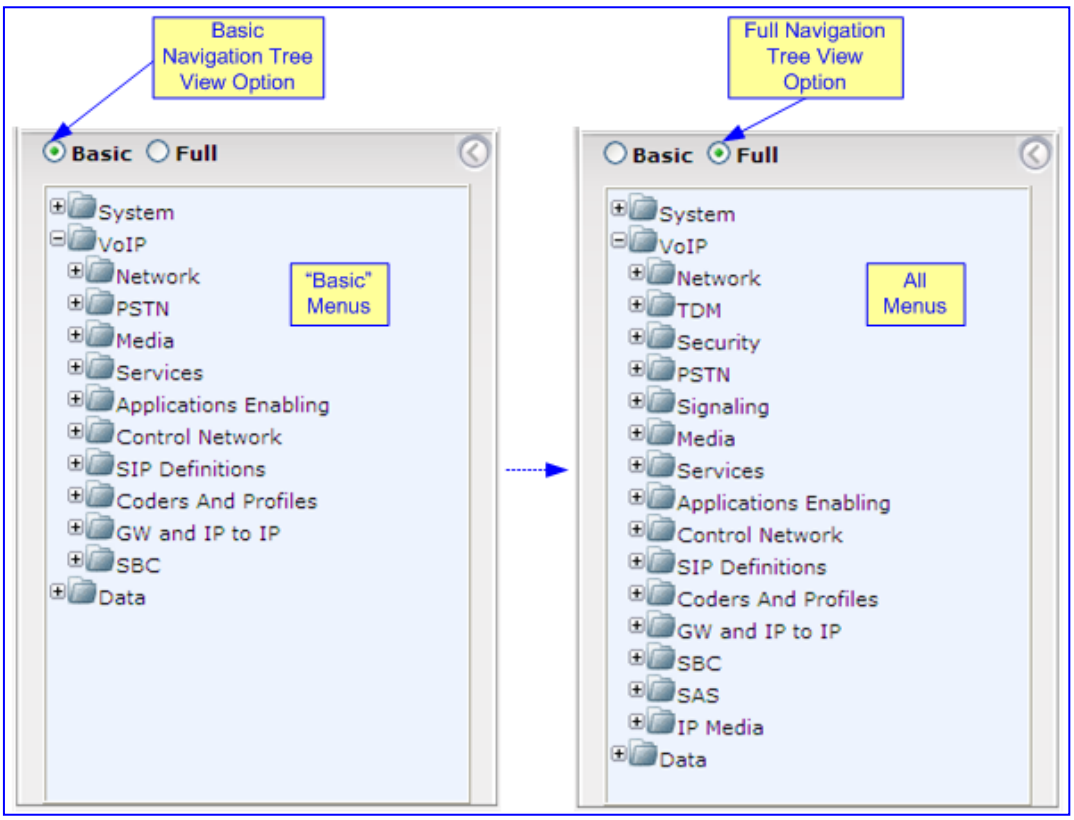
- **Basic:** displays only commonly used menus
- **Full:** displays all the menus pertaining to a configuration tab

The advantage of the Basic view is that it prevents "cluttering" of the Navigation tree with menus that may not be required. Therefore, a Basic view allows you to easily locate required menus.

➢ **To toggle between Full and Basic view:**

- Select the **Basic** option, located below the Navigation bar, to display a reduced menu tree; select the **Full** option to display all the menus. By default, the **Basic** option is selected.

Figure 4-5: Basic and Full View



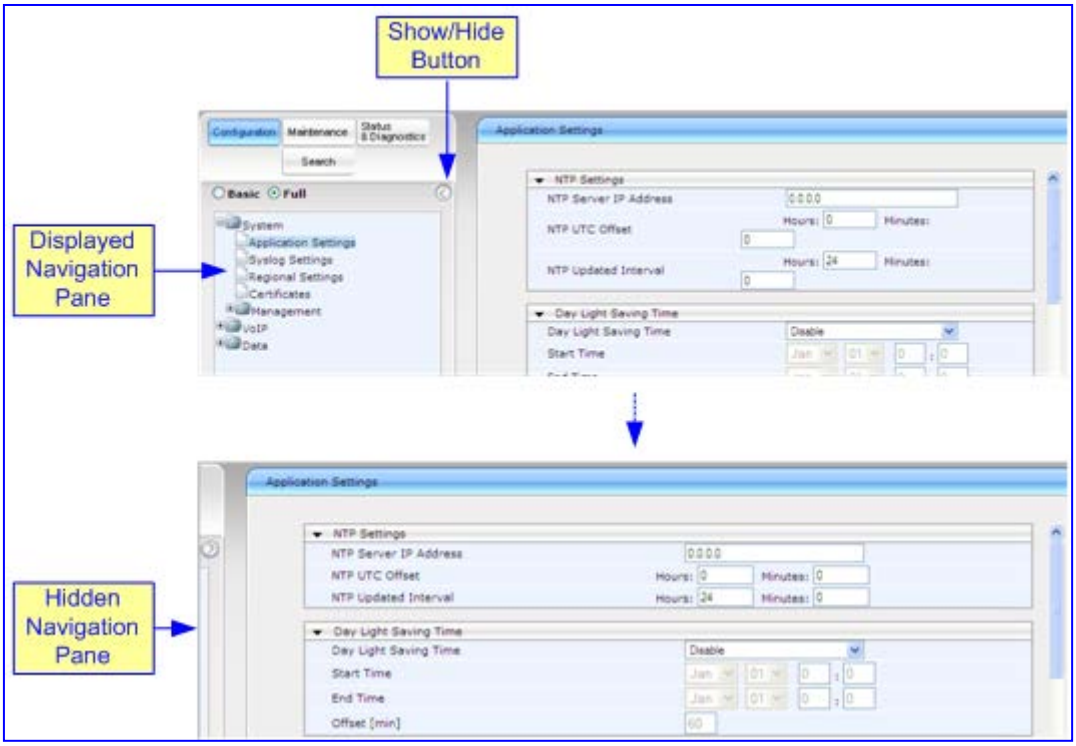
Note: After you reset the device, the Web GUI is displayed in Basic view.

4.1.5.2 Showing / Hiding the Navigation Pane

The Navigation pane can be hidden to provide more space for elements displayed in the Work pane. This is especially useful when the Work pane displays a table that's wider than the Work pane and to view all the columns, you need to use scroll bars. The arrow button located just below the Navigation bar is used to hide and show the Navigation pane.

- **To hide the Navigation pane:** click the left-pointing arrow ; the pane is hidden and the button is replaced by the right-pointing arrow button.
- **To show the Navigation pane:** click the right-pointing arrow ; the pane is displayed and the button is replaced by the left-pointing arrow button.

Figure 4-6: Showing and Hiding Navigation Pane



4.1.6 Working with Configuration Pages

The configuration pages contain the parameters for configuring the device and are displayed in the Work pane, located to the right of the Navigation pane.

4.1.6.1 Accessing Pages

The configuration pages are accessed by clicking the required page item in the Navigation tree.

➢ **To open a configuration page:**

1. On the Navigation bar, click the required tab:
   - **Configuration**
   - **Maintenance**
   - **Status & Diagnostics**

   The menus pertaining to the selected tab appear in the Navigation tree.
2. In the Navigation tree, drill-down to the required submenu and then click the required page item; the page opens in the Work pane.

You can also access previously opened pages by clicking the Web browser's **Back** button until you have reached the required page. This is useful if you want to view pages in which you have performed configurations in the current Web session.

Notes:

- You can also access certain pages from the **Device Actions** button located on the toolbar (see 'Toolbar Description' on page 54).
- To view all the menus in the Navigation tree, ensure that the Navigation tree is in Full view (see 'Displaying Navigation Tree in Basic and Full View' on page 56).
- To get Online Help for the currently displayed page, see 'Getting Help' on page 68.
- Certain pages may not be accessible or may be read-only if your Web user account's access level is low (see 'Configuring the Web User Accounts' on page 76). If a page is read-only, 'Read-Only Mode' is displayed at the bottom of the page.

4.1.6.2 Viewing Parameters

For convenience, some pages allow you to view a reduced or expanded display of parameters. The Web interface provides two methods for displaying page parameters:

- Displaying "basic" and "advanced" parameters - see 'Displaying Basic and Advanced Parameters' on page 59
- Displaying parameter groups - see 'Showing / Hiding Parameter Groups' on page 60

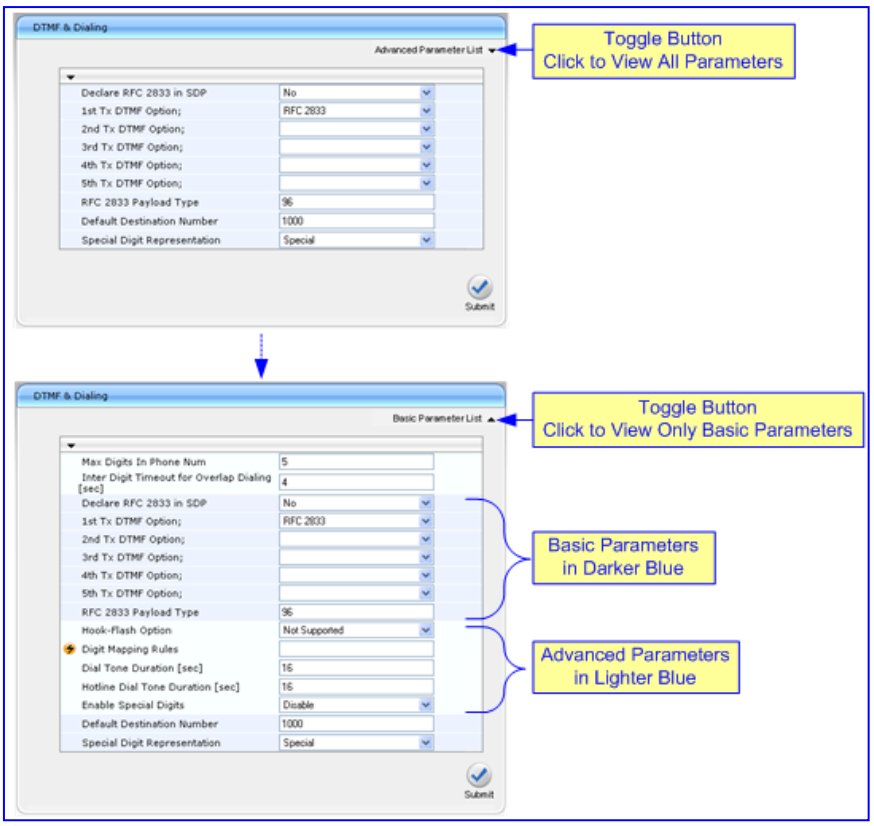
4.1.6.2.1 Displaying Basic and Advanced Parameters

Some pages provide you with an **Advanced Parameter List** / **Basic Parameter List** toggle button that allows you to show or hide advanced parameters (in addition to displaying the basic parameters). This button is located on the top-right corner of the page and has two states:

- **Advanced Parameter List** button with down-pointing arrow: click this button to display all parameters.
- **Basic Parameter List** button with up-pointing arrow: click this button to show only common (*basic*) parameters.

The figure below shows an example of a page displaying basic parameters only, and then showing advanced parameters as well, using the **Advanced Parameter List** button.

Figure 4-7: Toggling between Basic and Advanced View



For ease of identification, the basic parameters are displayed with a darker blue color background than the advanced parameters.

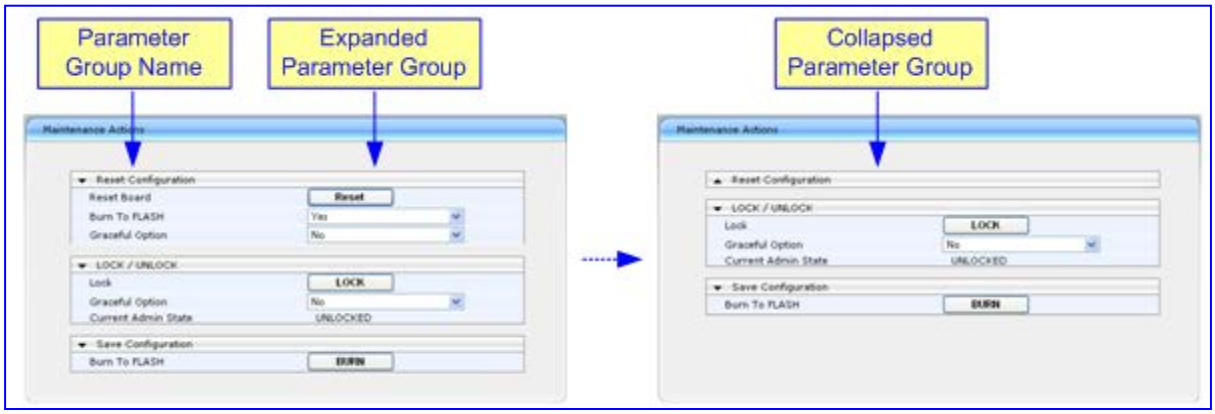
Notes:

- When the Navigation tree is in Full mode (see 'Navigation Tree' on page 55), configuration pages display all their parameters (i.e., the Advanced Parameter List view is displayed).
- If a page contains only basic parameters, the **Basic Parameter List** button is not displayed.
- After you reset the device, the Web pages display only the basic parameters.

4.1.6.2.2 Showing / Hiding Parameter Groups

Some pages provide groups of parameters, which can be hidden or shown. To toggle between hiding and showing a group, simply click the group title button that appears above each group. The button appears with a down-pointing or up-pointing arrow, indicating that it can be collapsed or expanded when clicked, respectively.

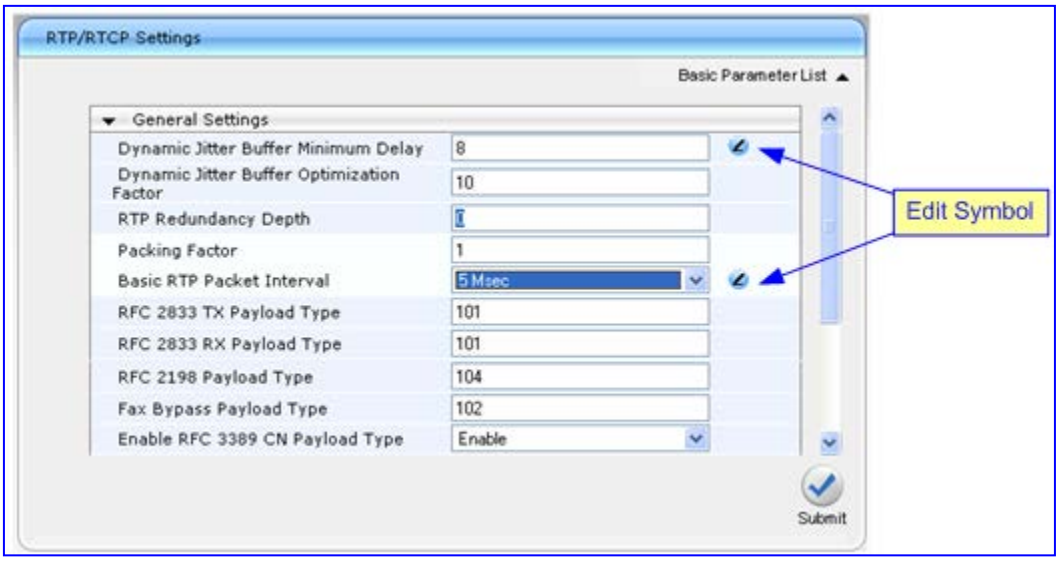
Figure 4-8: Expanding and Collapsing Parameter Groups



4.1.6.3 Modifying and Saving Parameters

When you modify a parameter value on a page, the **Edit** symbol appears to the right of the parameter. This is useful for indicating the parameters that you have currently modified (before applying the changes). After you apply your modifications, the symbols disappear.

Figure 4-9: Edit Symbol after Modifying Parameter Value



➢ **To save configuration changes on a page to the device's volatile memory (RAM), do one of the following:**

- On the toolbar, click the **Submit** button.
- At the bottom of the page, click the **Submit** button.

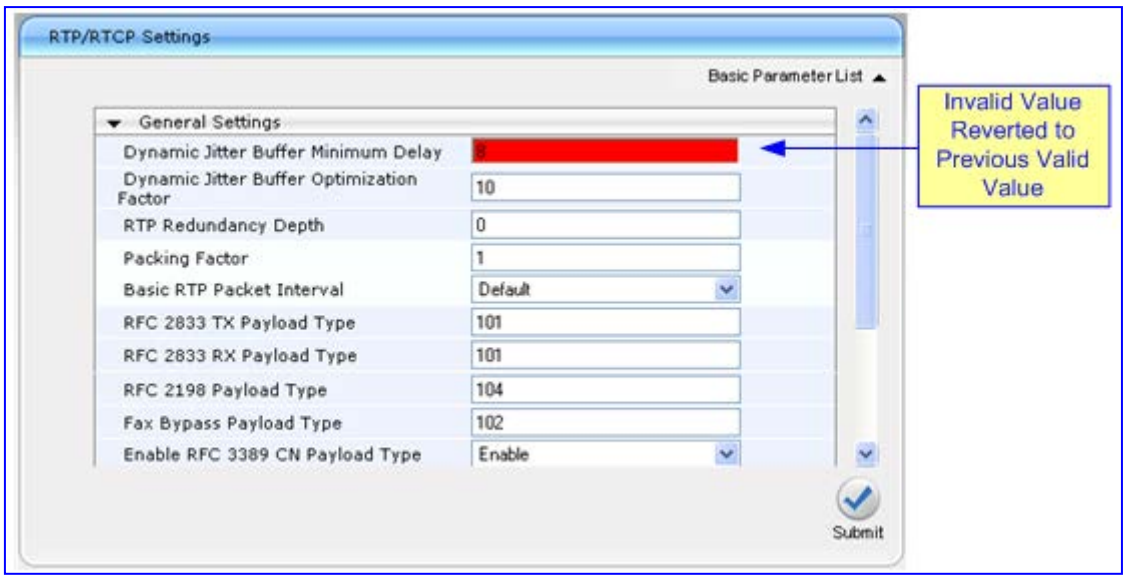
When you click **Submit**, modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect; other parameters displayed on the page with the lightning symbol are not changeable on-the-fly and require a device reset (see 'Resetting the Device' on page 677) before taking effect.

Notes:

- Parameters saved to the volatile memory (by clicking **Submit**), revert to their previous settings after a hardware or software reset (or if the device is powered down). Therefore, to ensure parameter changes (whether on-the-fly or not) are retained, save ('burn') them to the device's non-volatile memory, i.e., flash (see 'Saving Configuration' on page 680).
- If you modify a parameter value and then attempt to navigate away from the page without clicking **Submit**, a message box appears notifying you of this. Click **Yes** to save your modifications or **No** to ignore them.

If you enter an invalid parameter value (e.g., not in the range of permitted values) and then click **Submit**, a message box appears notifying you of the invalid value. In addition, the parameter value reverts to its previous value and is highlighted in red, as shown in the figure below:

Figure 4-10: Value Reverts to Previous Valid Value



4.1.6.4 Entering Phone Numbers

Phone numbers or prefixes that you need to configure throughout the Web interface must be entered only as digits without any other characters. For example, if you wish to enter the phone number 555-1212, it must be entered as 5551212 without the hyphen (-). If the hyphen is entered, the entry is invalid.

4.1.6.5 Working with Tables

This section describes how to work with configuration tables, which are provided in basic or enhanced design (depending on the configuration page).

4.1.6.5.1 Basic Design Tables

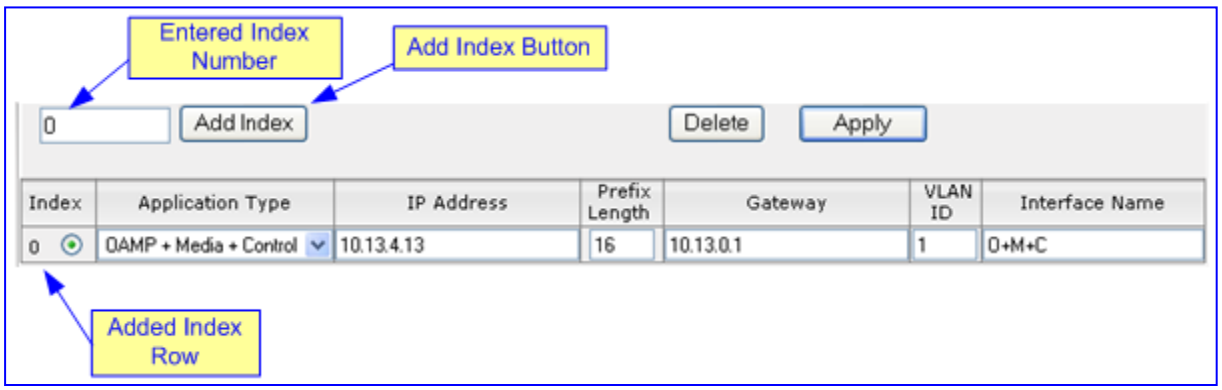
The basic design tables provide the following command buttons:

- **Add Index:** adds an index entry to the table.
- **Duplicate:** duplicates a selected, existing index entry.
- **Compact:** organizes the index entries in ascending, consecutive order.
- **Delete:** deletes a selected index entry.
- **Apply:** saves the configuration.

➢ **To add an entry to a table:**

1. In the 'Add Index' field, enter the desired index entry number, and then click **Add Index**; an index entry row appears in the table:

Figure 4-11: Adding an Index Entry to a Table



2. Click **Apply** to save the index entry.

Notes:

- Before you can add another index entry, ensure that you have applied the previously added index entry (by clicking **Apply**).
- If you leave the 'Add' field blank and then click **Add Index**, the existing index entries are all incremented by one and the newly added index entry is assigned the index 0.

➢ **To copy an existing index table entry:**

1. In the 'Index' column, select the index that you want to duplicate; the **Edit** button appears.
2. Click **Edit**; the fields in the corresponding index row become available.
3. Click **Duplicate**; a new index entry is added with identical settings as the selected index in Step 1. In addition, all existing index entries are incremented by one and the newly added index entry is assigned the index 0.

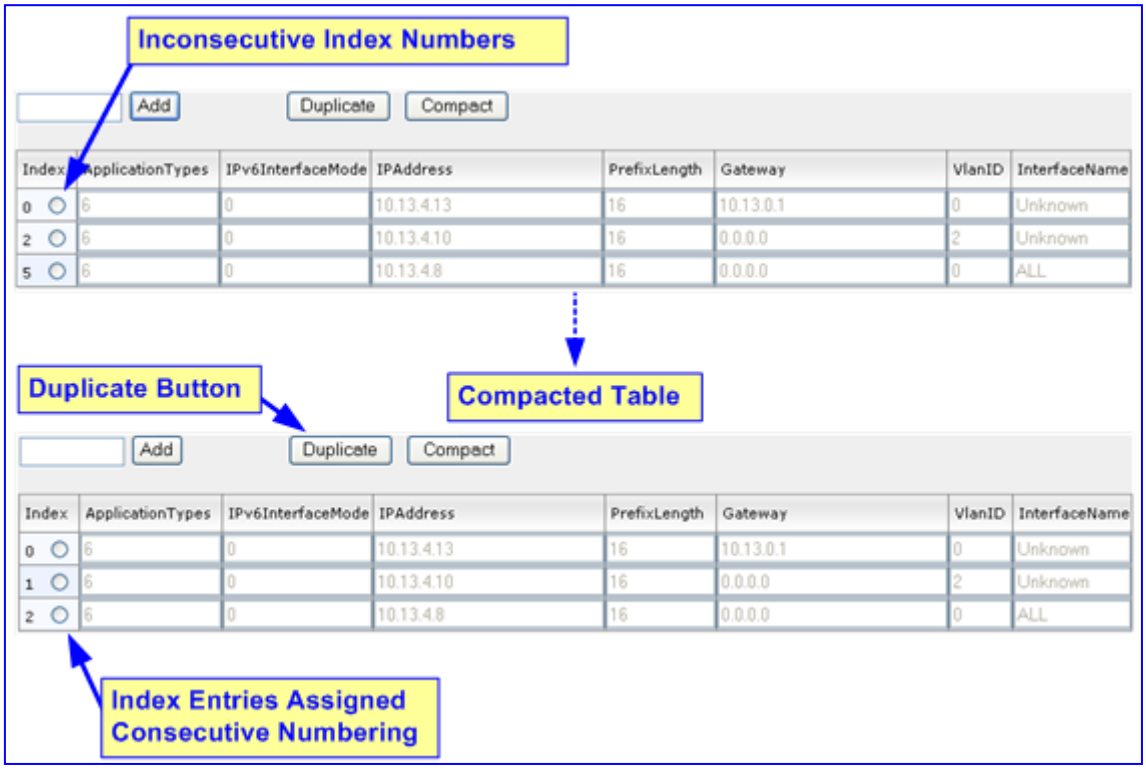
➢ **To edit an index table entry:**

1. In the 'Index' column, select the index corresponding to the table row that you want to edit.
2. Click **Edit**; the fields in the corresponding index row become available.
3. Modify the values as required, and then click **Apply**; the new settings are applied.

➢ **To organize the index entries in ascending, consecutive order:**

- Click **Compact**; the index entries are organized in ascending, consecutive order, starting from index 0. For example, if you added three index entries 0, 4, and 6, then the index entry 4 is re-assigned index number 1 and the index entry 6 is re-assigned index number 2.

Figure 4-12: Compacting a Web Interface Table



➢ **To delete an index table entry:**

1. In the 'Index' column, select the index corresponding to the table row that you want to delete.
2. Click **Delete**; the table row is removed from the table.

4.1.6.5.2 Enhanced Design Tables

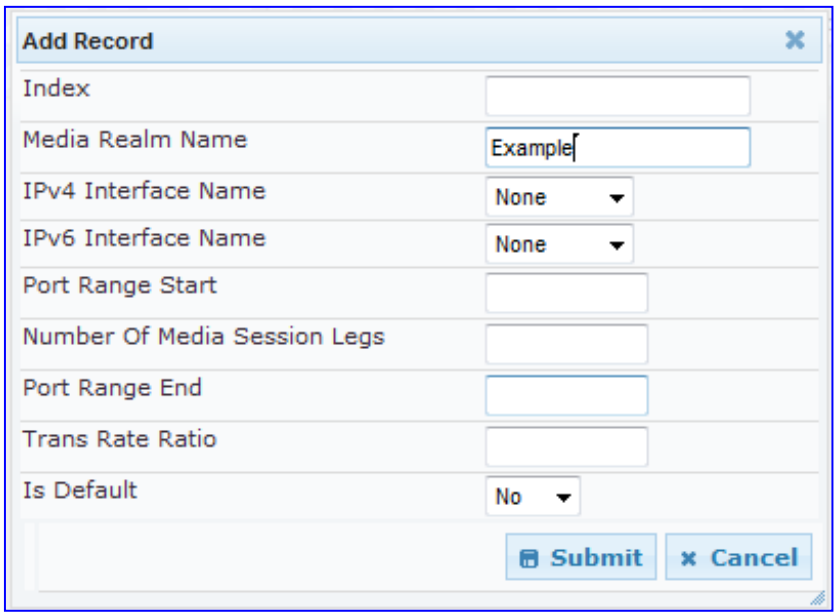
The enhanced table structure includes the following buttons:

- **Add:** adds a row entry to the table
- **Edit:** edits the selected table row
- **Delete:** deletes a selected table row
- **View/Unview:** shows or hides all configuration settings of selected table rows

➢ **To add an entry:**

1. Click the **Add** button; the Add Record dialog box appears:

Figure 4-13: Add Record Dialog Box



2. Configure the required parameters, and then click **Submit** to apply your changes (or **Cancel** to ignore your changes); the new row entry is added to the table. If the configuration is invalid, the index of the table row is highlighted in red, as shown below:

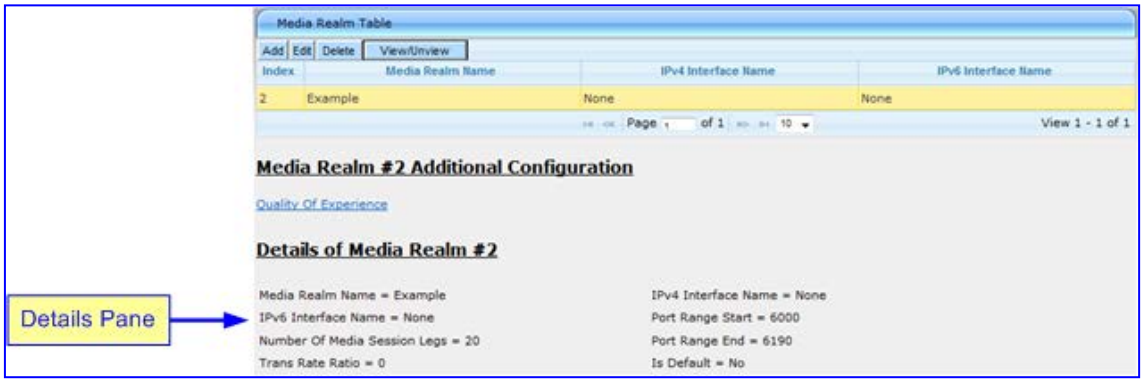
Figure 4-14: Index Highlighted in Red

By default, the table displays 10 entries per page. However, you can change this to 5 by selecting **5** from the drop-down list located immediately below the table. If your table spans over multiple pages, you can navigate between the pages by clicking the left and right arrow buttons located immediately below the table.

➢ **To view the configuration settings of an entry:**

1. Select the table row that you want to view, and then click the **View/Unview** button; a Details pane appears below the table, displaying the configuration settings of the selected row, as shown below:

Figure 4-15: Displayed Details Pane



2. To hide the Details pane, click the **View/Unview** button again.

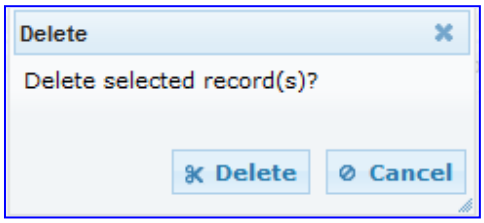
➢ **To edit an entry:**

1. Select the table row that you want to modify, and then click the **Edit** button; the Edit Record dialog box appears.
2. Make the required changes, and then click **Submit**.

➢ **To delete an entry:**

1. Select the table row that you want to delete, and then click the **Delete** button; the Delete message box appears:

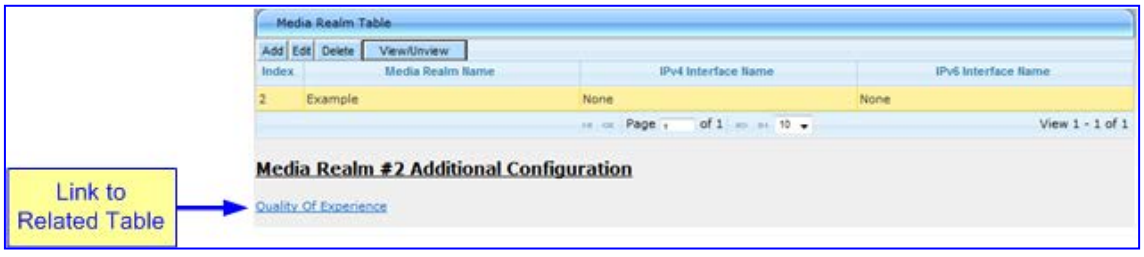
Figure 4-16: Delete Message Box



2. Click **Delete** to confirm deletion (or **Cancel** to abort the process).

Some tables provide a link to a related table for advanced configuration of a selected row entry, as shown below:

Figure 4-17: Link to Related Table



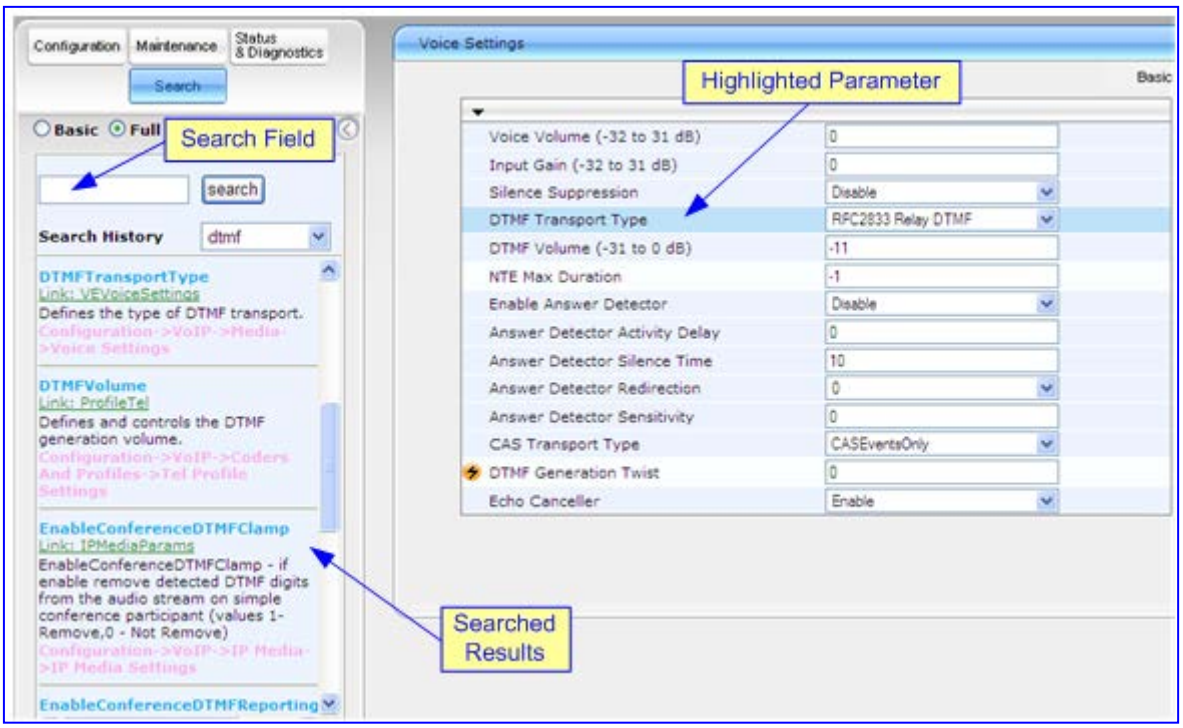
4.1.7 Searching for Configuration Parameters

The Web interface provides a search engine that allows you to search any *ini* file parameter that is configurable in the Web interface (i.e., has a corresponding Web parameter). You can search for a specific parameter (e.g., "EnableIPSec") or a substring of that parameter (e.g., "sec"). If you search for a substring, all parameters containing the searched substring in their names are listed.

➢ **To search for *ini* file parameters configurable in the Web interface:**

1. On the Navigation bar, click the **Search** tab; the Search engine appears in the Navigation pane.
2. In the 'Search' field, enter the parameter name or substring of the parameter name that you want to search. If you have done a previous search for such a parameter, instead of entering the required string, you can use the 'Search History' drop-down list to select the string saved from a previous search.
3. Click **Search**; a list of located parameters based on your search appears in the Navigation pane. Each searched result displays the following:
   - *ini* file parameter name
   - Link (in green) to its location (page) in the Web interface
   - Brief description of the parameter
4. In the searched list, click the required parameter (link in green) to open the page in which the parameter appears; the relevant page opens in the Work pane and the searched parameter is highlighted in the page for easy identification, as shown in the figure below:

Figure 4-18: Searched Result Screen



4.1.8 Creating a Login Welcome Message

You can create a Welcome message box (alert message) that appears after each successful login to the Web interface. The *WelcomeMessage* ini file parameter table allows you to create the Welcome message. Up to 20 lines of character strings can be defined for the message. If this parameter is not configured, no Welcome message box is displayed after login.

An example of a Welcome message is shown in the figure below:

Figure 4-19: User-Defined Web Welcome Message after Login

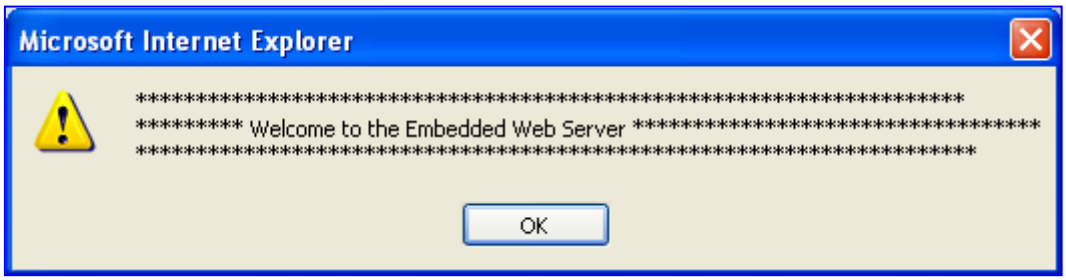


Table 4-2: ini File Parameter for Welcome Login Message

Parameter	Description
WelcomeMessage	Defines the Welcome message that appears after a successful login to the Web interface. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; [\WelcomeMessage] For Example: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; WelcomeMessage 1 = "**********"; WelcomeMessage 2 = "********* This is a Welcome message **"; WelcomeMessage 3 = "**********"; [\WelcomeMessage] **Note:** Each index represents a line of text in the Welcome message box. Up to 20 indices can be defined.

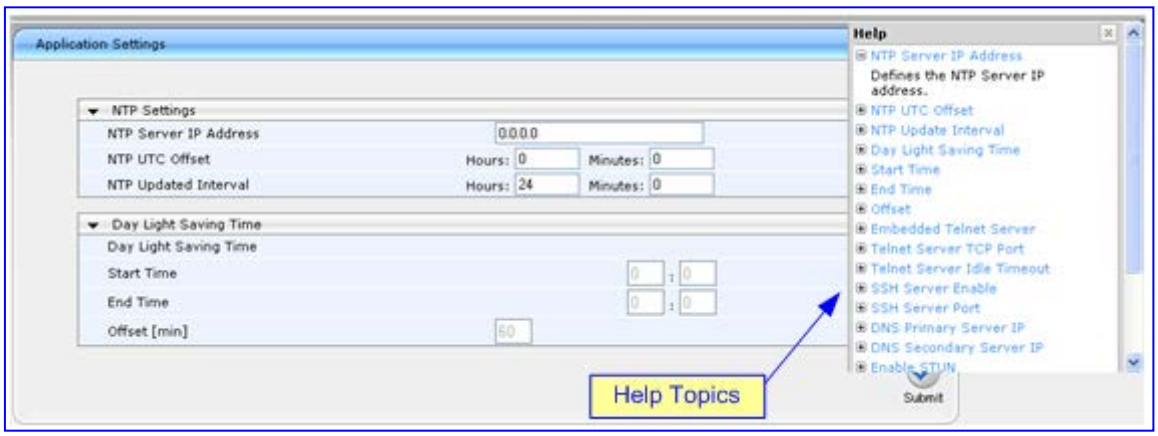
4.1.9 Getting Help

The Web interface provides you with context-sensitive Online Help. The Online Help provides brief descriptions of parameters pertaining to the currently opened page.

➢ **To view the Help topic of a currently opened page:**

1. On the toolbar, click the **Help** button; the Help topic pertaining to the opened page appears, as shown below:

Figure 4-20: Help Topic for Current Page



2. To view a description of a parameter, click the **plus** ⊞ sign to expand the parameter. To collapse the description, click the **minus** ⊟ sign.
3. To close the Help topic, click the **close** button located on the top-right corner of the Help topic window or simply click the **Help** button.

Note: Instead of clicking the **Help** button for each page you open, you can open it once for a page and then simply leave it open. Each time you open a different page, the Help topic pertaining to that page is automatically displayed.

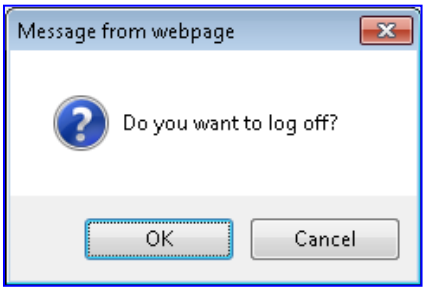
4.1.10 Logging off the Web Interface

You can log off the Web interface and re-access it with a different user account. For more information on Web User Accounts, see 'Configuring Web User Accounts' on page 76.

➢ **To log off the Web interface:**

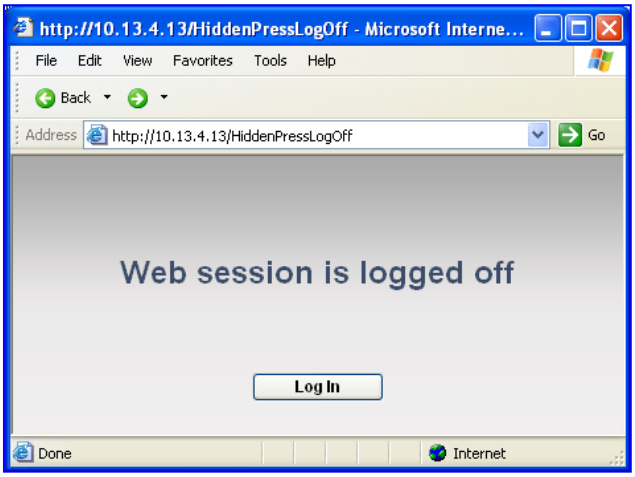
1. On the toolbar, click the **Log Off** button; the Log Off confirmation message box appears:

Figure 4-21: Log Off Confirmation Box



2. Click **OK**; the Web session is logged off and the **Log In** button appears.

Figure 4-22: Web Session Logged Off



To log in again, simply click the **Log In** button, and then in the Login window, enter your user name and password (see 'Accessing the Web Interface' on page 52).

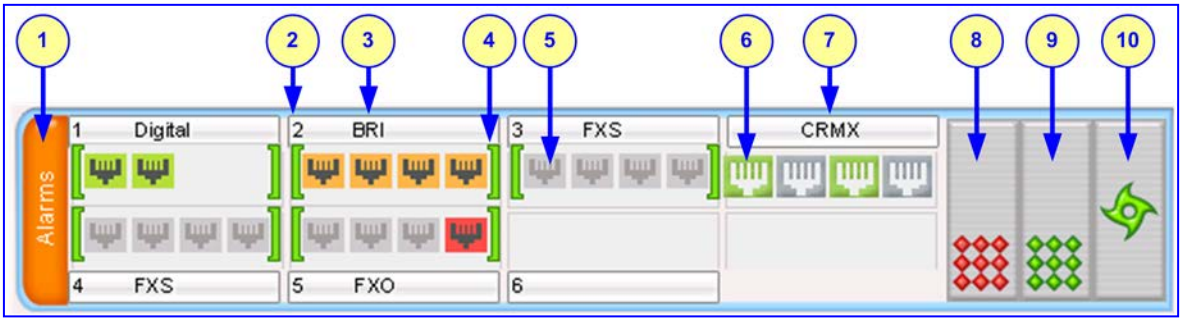
4.2 Using the Home Page

By default, the Home page is displayed when you access the device's Web interface. The Home page provides you with a graphical display of the device's front panel, displaying color-coded status icons for monitoring the functioning of the device. The Home page also displays general device information (in the 'General Information' pane) such as the device's IP address and firmware version.

➢ **To access the Home page:**

- On the toolbar, click the **Home** icon.

Figure 4-23: Home Page



Note: The displayed number and type of telephony interfaces depends on the device's hardware configuration. The displayed WAN port type depends on the ordered hardware configuration (e.g., Gigabit Ethernet or SHDSL).

In addition to the color-coded status information depicted on the graphical display of the device (as described in the subsequent table), the Home page displays various read-only information in the General Information pane:

- **IP Address:** IP address of the device
- **Subnet Mask:** subnet mask address of the device
- **Default Gateway Address:** default gateway used by the device
- **Digital Port Number:** number of digital PRI ports (appears only if the device houses a DIGITAL module)
- **BRI Port Number:** number of BRI ports (appears only if the device houses a BRI module)
- **Analog Port Number:** number of analog (FXS / FXO) ports (appears only if the device houses any of these analog modules)
- **Firmware Version:** software version currently running on the device
- **Protocol Type:** signaling protocol currently used by the device (i.e. SIP)
- **Gateway Operational State:** operational state of the device:
  - "LOCKED" - device is locked (i.e. no new calls are accepted)
  - "UNLOCKED" - device is not locked
  - "SHUTTING DOWN" - device is currently shutting down

Table 4-3: Description of the Areas of the Home Page

<table>
<tr><th>Item #</th><th>Description</th></tr>
<tr><td>1</td><td>Displays the highest severity of an active alarm raised (if any) by the device:
Green = No alarms
Red = Critical alarm
Orange = Major alarm
Yellow = Minor alarm
To view a list of active alarms in the Active Alarms page (see Viewing Active Alarms on page 711), click the Alarms area.</td></tr>
<tr><td>2</td><td>Module slot number (1 to 6).</td></tr>
<tr><td>3</td><td>Module type: FXS, FXO, DIGITAL (i.e., E1/T1), BRI, IPMEDIA, and CRMX.</td></tr>
<tr><td>4</td><td>Module status icon: (green): Module has been inserted or is correctly configured (gray): Module was removed. 'Reserved' is displayed alongside the module's name (red): Module failure. 'Failure' is displayed instead of the module's name</td></tr>
<tr><td>5</td><td>Port (trunk or channel) status icon (see Viewing Trunks' Channels on page 73).
<table>
<tr><th>Icon</th><th>Trunk Description (Digital Module)</th><th>Channel Description (Analog Module)</th></tr>
<tr><td> (grey)</td><td>Disable: Trunk not configured (not in use)</td><td>Inactive: Channel is currently on-hook</td></tr>
<tr><td> (green)</td><td>Active - OK: Trunk synchronized</td><td>Call Connected: Active RTP stream</td></tr>
<tr><td> (yellow)</td><td>RAI Alarm: Remote Alarm Indication (RAI), also known as the Yellow Alarm</td><td>-</td></tr>
<tr><td> (red)</td><td>LOS / LOF Alarm: Loss due to LOS (Loss of Signal) or LOF (Loss of Frame)</td><td>Not Connected: No analog line is connected to this port or port out of service due to Serial Peripheral Interface (SPI) failure (applicable only to FXO interfaces)</td></tr>
<tr><td> (blue)</td><td>AIS Alarm: Alarm Indication Signal (AIS), also known as the Blue Alarm</td><td>Handset Offhook: Channel is off-hook, but there is no active RTP session</td></tr>
<tr><td> (orange)</td><td>D-Channel Alarm: D-channel alarm</td><td>-</td></tr>
</table></td></tr>
</table>

Item #	Description
6	Ethernet LAN port status icons: ▪ (green): Ethernet link is working ▪ (gray): Ethernet link is not connected The first three ports from the left depict the LAN ports; the fourth port depicts the WAN. You can also view detailed Ethernet port information in the Ethernet Port Information page (see Viewing Ethernet Port Information on page 710), by clicking the icon.
7	CRMX module for data.
8	Power Supply Unit 1 status icon: ▪ (green): Power supply is operating ▪ (red): Power supply failure or no power supply unit installed
9	Power Supply Unit 2 status indicator. See Item #8 for a description.
10	Fan tray unit status icon: ▪ (green): Fan tray operating ▪ (red): Fan tray failure

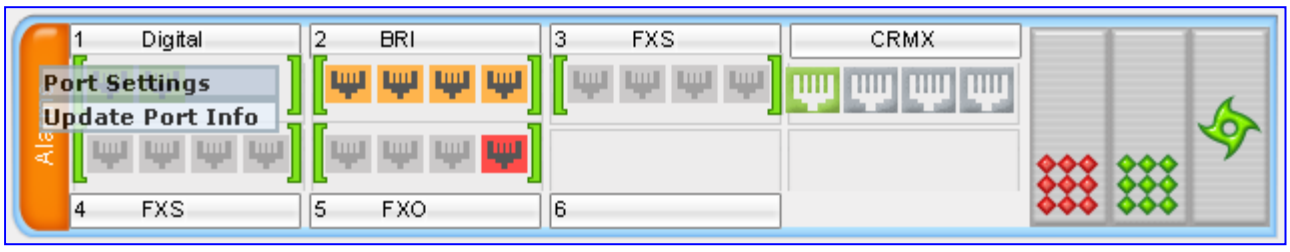
4.2.1 Assigning a Port Name

The Home page allows you to assign an arbitrary name or a brief description to each port. This description appears as a tooltip when you move your mouse over the port.

➢ **To add a port description:**

1. Click the required port icon; a shortcut menu appears, as shown below:

Figure 4-24: Assigning Port Name



2. From the shortcut menu, choose **Update Port Info**; a text box appears.
3. Type a brief description for the port, and then click **Apply Port Info**.

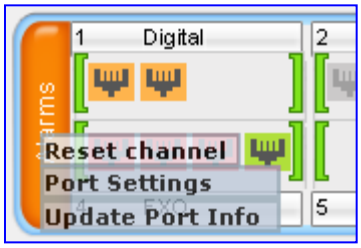
4.2.2 Resetting an Analog Channel

The Home page allows you to inactivate (*reset*) an FXO or FXS analog channel. This is sometimes useful, for example, when the device (FXO) is connected to a PBX and the communication between the two can't be disconnected (e.g., when using reverse polarity).

➢ **To reset a channel:**

- Click the required **FXS** or **FXO** port icon, and then from the shortcut menu, choose **Reset Channel**; the channel is changed to inactive (i.e., the port icon is displayed in grey).

Figure 4-25: Resetting Analog Port



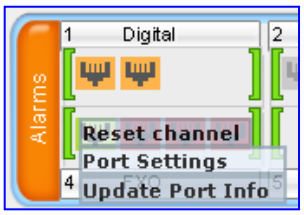
4.2.3 Viewing Analog Port Information

The Home page allows you to view detailed information on a specific FXS or FXO analog port such as RTP/RTCP and voice settings.

➢ **To view detailed port information:**

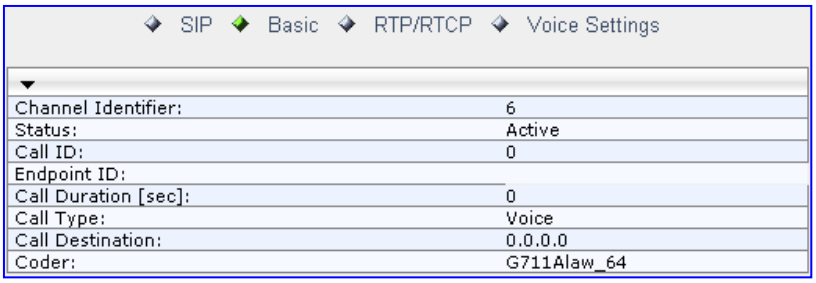
1. Click the port for which you want to view port settings; the shortcut menu appears.

Figure 4-26: Viewing Analog Port Information



2. From the shortcut menu, click **Port Settings**; the Basic Channel Information page appears.

Figure 4-27: Basic Information Screen



3. To view RTP/RTCP or voice settings, click the relevant button.

4.2.4 Viewing Trunk Channels

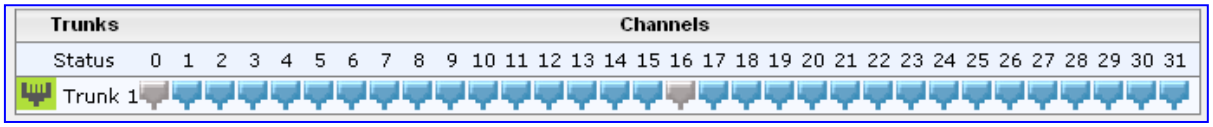
The Home page allows you to drill-down to view a detailed status of the channels pertaining to a trunk In addition, you can view the trunk's configuration.

➢ **To view a detailed status of a trunk's channels:**

1. In the Home page, click the trunk port icon of whose status you want to view; a shortcut menu appears.

2. From the shortcut menu, choose **Port Settings**; the Trunks & Channels Status page pertaining to the specific trunk appears:

Figure 4-28: Trunks and Channels Status Screen



The color-coding for the status of the trunk's channels status is described in the table below:

Table 4-4: Color-Coding Status for Trunk Channels

Icon	Color	Label	Description
	Light blue	Inactive	Configured, but currently no call
	Green	Active	Call in progress (RTP traffic)
	Purple	SS7	Configured for SS7 **Note:** Currently, SS7 is not supported.
	Grey	Non Voice	Not configured
	Blue	ISDN Signaling	Configured as a D-channel
	Yellow	CAS Blocked	-

3. To view the configuration settings of the trunk and/or to modify the trunk's settings, click the Trunk icon, and then from the shortcut menu, choose Port Settings; the Trunk Settings page appears. For more information on configuring the trunk, see Configuring the Trunk Settings on page 254.

4.2.5 Replacing Modules

To replace the device's modules, you must use the Web interface in combination with physical removal and insertion of the modules. In other words, when you replace a module, you first need to 'software-remove' it, then extract it physically from the chassis and insert a new module, and then 'software-insert' it using the Web interface. The software removal and insertion is performed in the Home page.

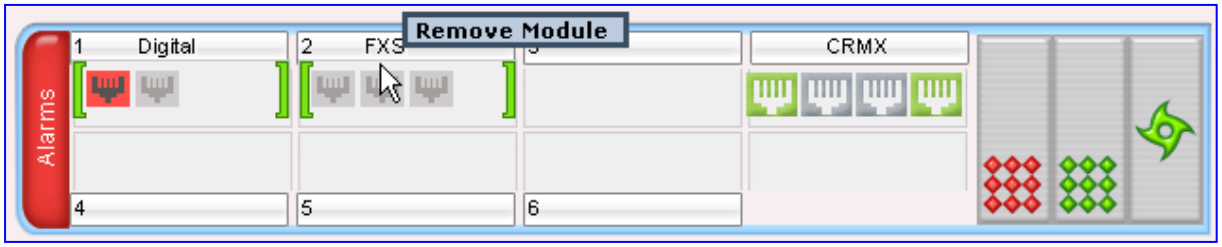
Warnings:

- A module must be replaced with the same type of module and in the same module slot number. For example, a module with two digital spans in Slot 1 must be replaced with a module with two digital spans in Slot 1.
- When only one module is available, removal of the module causes the device to reset.
- Before inserting a module into a previously empty slot, you must power down the device.

➢ **To replace a module:**

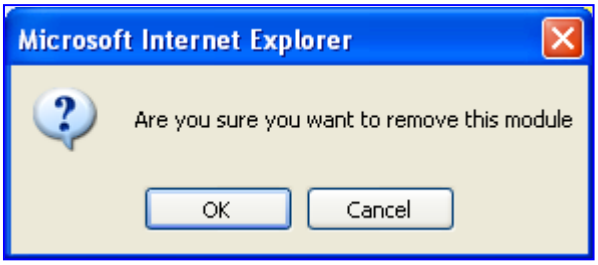
1. Remove the module by performing the following:
   a. In the Home page, click the title of the module that you want to replace; the **Remove Module** button appears:

Figure 4-29: Removing Module



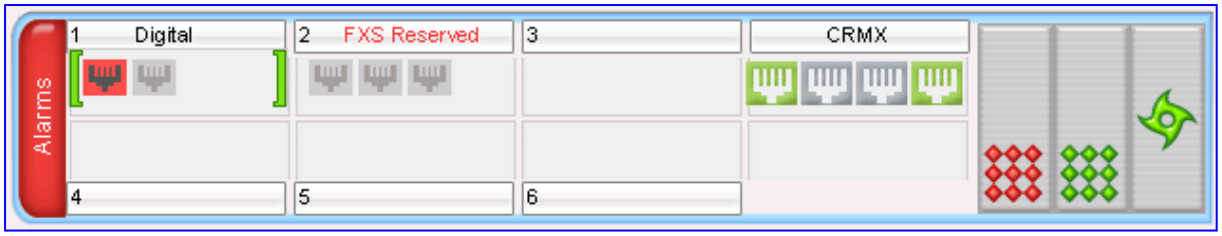
b. Click the **Remove Module** button; a message box appears requesting you to confirm module removal:

Figure 4-30: Module Removal Confirmation Message Box



c. Click **OK** to confirm removal; after a few seconds, the module is software-removed, the module status icon turns to grey, and the name of the module is suffixed with the word 'Reserved':

Figure 4-31: Module Removed



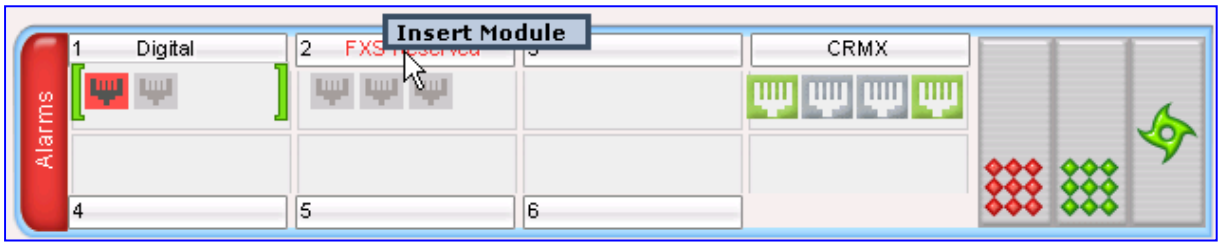
d. Physically remove the module (refer to the *Installation Manual*).

2. Insert the replaced module, by performing the following:

a. Physically insert the replaced module (refer to the *Installation Manual*) into the same slot in which the previous module resided.

b. In the Home page, click the title of the module ("<module type> Reserved") that you want to replace; the **Insert Module** button appears:

Figure 4-32: Inserting Module



c. Click the **Insert Module** button; a message appears informing you that this may take a few seconds. When the message disappears, the module is inserted, which is indicated by the disappearance of the word 'Reserved' from the module's name.

4.3 Configuring Web User Accounts

To prevent unauthorized access to the Web interface, two Web user accounts are available (primary and secondary) with assigned user name, password, and access level. When you login to the Web interface, you are requested to provide the user name and password of one of these Web user accounts. If the Web session is idle (i.e., no actions are performed) for more than five minutes, the Web session expires and you are once again requested to login with your user name and password. Up to five Web users can simultaneously open (log in to) a session on the device's Web interface. Users can be banned for a period of time upon a user-defined number of unsuccessful login attempts. Login information (such as how many login attempts were made and the last successful login time) can be presented to the user.

Each Web user account is composed of three attributes:

- **User name and password:** enables access (login) to the Web interface.
- **Access level:** determines the extent of the access (i.e., availability of pages and read / write privileges). The available access levels and their corresponding privileges are listed in the table below:

Table 4-5: Web User Accounts Access Levels and Privileges

Access Level	Numeric Representation*	Privileges
Security Administrator	200	Read / write privileges for all pages.
Administrator	100	read / write privileges for all pages except security-related pages, which are read-only.
User Monitor	50	No access to security-related and file-loading pages; read-only access to the other pages. This read-only access level is typically applied to the secondary Web user account.
No Access	0	No access to any page.
* The numeric representation of the access level is used only to define accounts in a RADIUS server (the access level ranges from 1 to 255).		

The default attributes for the two Web user accounts are shown in the following table:

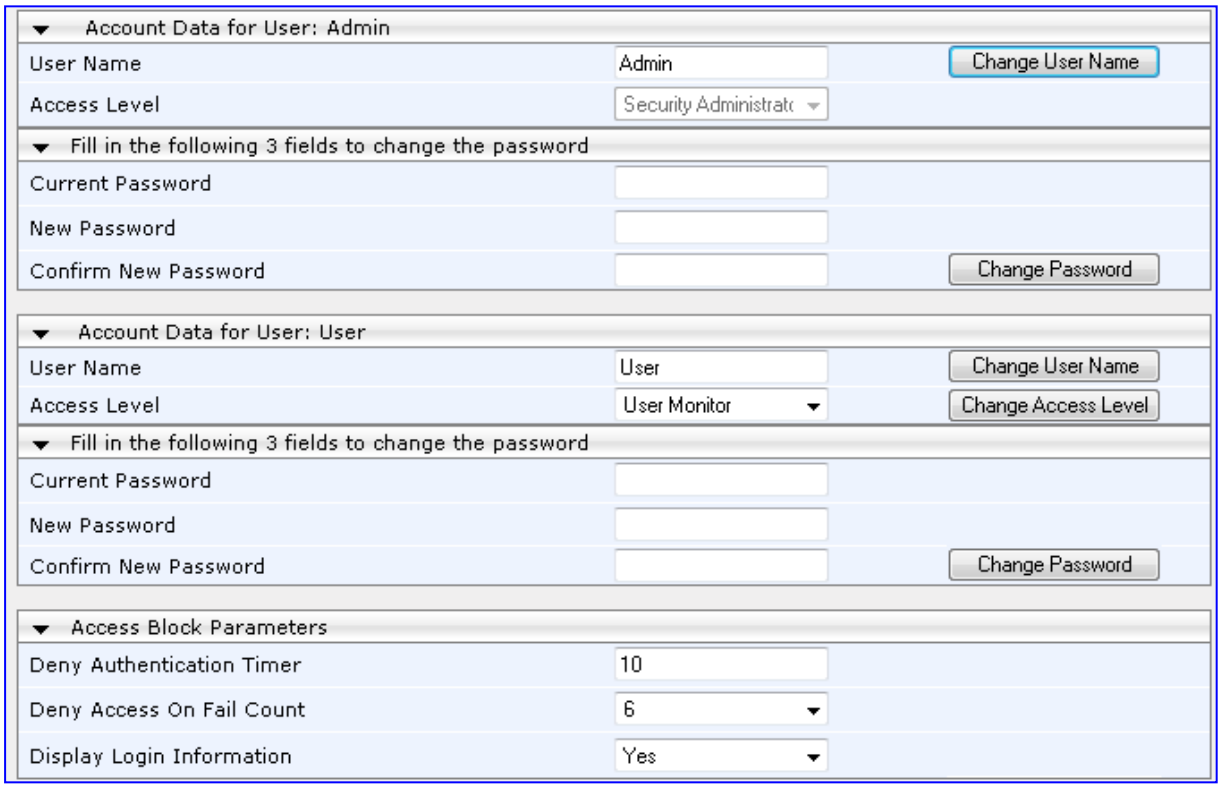
Table 4-6: Default Attributes for the Web User Accounts

Account / Attribute	User Name (Case-Sensitive)	Password (Case-Sensitive)	Access Level
Primary Account	Admin	Admin	Security Administrator **Note:** The Access Level cannot be changed for this account type.
Secondary Account	User	User	User Monitor

➢ **To change the Web user accounts attributes:**

1. Open the Web User Accounts page (**Configuration** tab > **System** menu > **Web User Accounts**).

Figure 4-33: WEB User Accounts Page (for Users with 'Security Administrator' Privileges)



Note: If you are logged into the Web interface as the Security Administrator, both Web user accounts are displayed on the Web User Accounts page (as shown above). If you are logged in with the secondary user account, only the details of the secondary account are displayed on the page.

2. To change the access level of the secondary account:
   a. From the 'Access Level' drop-down list, select the new access level.
   b. Click **Change Access Level**; the new access level is applied immediately.

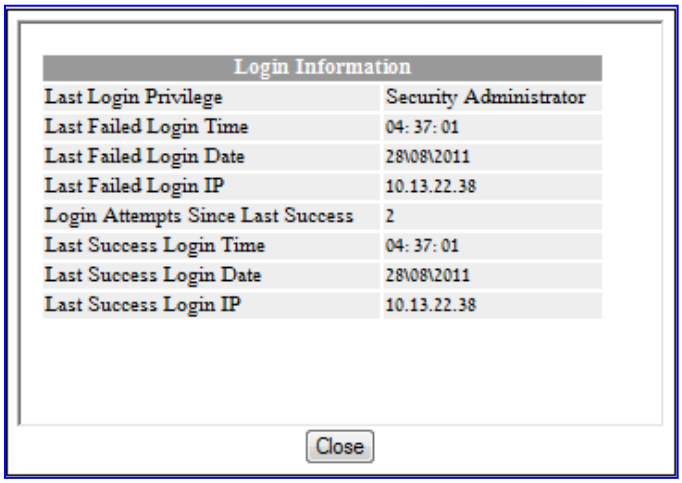
Notes:

- The access level of the primary Web user account is 'Security Administrator', which cannot be modified.
- The access level of the secondary account can only be modified by the primary account user or a secondary account user with 'Security Administrator' access level.

3. To change the user name of an account, perform the following:
   a. In the field 'User Name', enter the new user name (maximum of 19 case-sensitive characters).
   b. Click **Change User Name**; if you are currently logged into the Web interface with this account, the 'Enter Network Password' dialog box appears, requesting you to enter the new user name.

4. To change the password of an account, perform the following:
   a. In the field 'Current Password', enter the current password.
   b. In the fields 'New Password' and 'Confirm New Password', enter the new password (maximum of 19 case-sensitive characters).
   c. Click **Change Password**; if you are currently logged into the Web interface with this account, the 'Enter Network Password' dialog box appears, requesting you to enter the new password.
5. To prevent user access after a specific number of failed logins, do the following:
   a. From the 'Deny Access On Fail Count' drop-down list, select the number of failed logins after which the user is prevented access to the device for a user-defined time (see next step).
   b. In the 'Deny Authentication Timer' field, enter the interval (in seconds) that the user needs to wait before a new login attempt from the same IP address can be done after reaching the number of failed login attempts (defined in the previous step).
6. To display user login information upon a successful login, from the 'Display Login Information' drop-down list, select **Yes**. After you login, the following window is displayed:

Figure 4-34: Login Information Window



7. Click **Submit** to apply your changes.

Notes:

- For security, it's recommended that you change the default user name and password.
- A Web user with access level 'Security Administrator' can change all attributes of all the Web user accounts. Web users with an access level other than 'Security Administrator' can only change their own password and user name.
- To reset the two Web user accounts' user names and passwords to default, set the *ini* file parameter ResetWebPassword to 1.
- To access the Web interface with a different account, click the **Log off** button located on the toolbar, click any button or page item, and then re-access the Web interface with a different user name and password.
- You can set the entire Web interface to read-only (regardless of Web user account's access level), by using the *ini* file parameter DisableWebConfig (see 'Web and Telnet Parameters' on page 764).
- Access to the Web interface can be disabled, by setting the ini file parameter DisableWebTask to 1. By default, access is enabled.
- You can define additional Web user accounts using a RADIUS server (refer to the *Product Reference Manual*).
- For secured HTTP connection (HTTPS), refer to the *Product Reference Manual*.

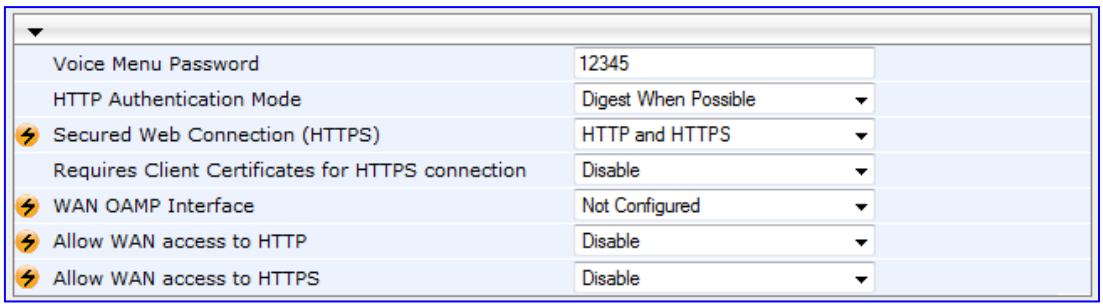
4.4 Configuring Web Security Settings

The WEB Security Settings page is used to define a secure Web access communication method. For a description of these parameters, see 'Web and Telnet Parameters' on page 764.

➢ **To define Web access security:**

1. Open the WEB Security Settings page (**Configuration** tab > **System** menu > **Management** submenu > **WEB Security Settings**).

Figure 4-35: Web Security Page



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

4.5 Limiting OAMP Access to a Specific WAN Interface

You can limit the access of OAMP applications (such as HTTP, HTTPS, Telnet, and SSH) to a specific WAN interface. This OAMP-interface binding can then be associated with a Virtual Routing and Forwarding (VRF).

➢ **To limit OAMP access on a specific WAN interface, using CLI.**

1. Enable WAN management access for specific OAMP applications (see 'Enabling Remote HTTP/S Web Management' on page 48), using any of the following commands:

```
set wan-ssh-allow
set wan-telnet-allow
set wan-snmp-allow
set wan-http-allow
set wan-https-allow
```

2. Define the WAN interface for the OAMP applications, using the OAMPWanInterfaceName ini file parameter or the following CLI command:

```
bind GigabitEthernet <slot/port.vlanId> oamp
bind vlan <vlanId> oamp
```

The following example enables WAN access for Telnet on interface GigabitEthernet 0/0.4 (GigabitEthernet 0/0.4 may be associated with a VRF):

```
(config-system)# cli-terminal
(cli-terminal)# set wan-telnet-allow on
(cli-terminal)# exit
(config-system)# bind GigabitEthernet 0/0.5 oamp
```

➢ **To define the WAN OAMP interface, using Web interface:**

1. Open the WEB Security Settings page (see 'Configuring Web Security Settings' on page 79).
2. From the 'WAN OAMP Interface' drop-down list, select the required WAN interface.
3. Click **Submit** to apply your changes.

4.6 Web Login Authentication using Smart Cards

You can enable Web login authentication using certificates from a third-party, common access card (CAC) with user identification. When a user attempts to access the device through the Web browser (HTTPS), the device retrieves the Web user's login username (and other information, if required) from the CAC. The user attempting to access the device is only required to provide the login password. Typically, a TLS connection is established between the CAC and the device's Web interface, and a RADIUS server is implemented to authenticate the password with the username. Therefore, this feature implements a two-factor authentication - what the user has (i.e., the physical card) and what the user knows (i.e., the login password).

This feature is enabled using the EnableMgmtTwoFactorAuthentication parameter.

Note: For specific integration requirements for implementing a third-party smart card for Web login authentication, contact your AudioCodes representative.

➢ **To login to the Web interface using CAC:**

1. Insert the Common Access Card into the card reader.
2. Access the device using the following URL: https://<host name or IP address>; the device prompts for a username and password.
3. Enter the password only. As some browsers require that the username be provided, it's recommended to enter the username with an arbitrary value.

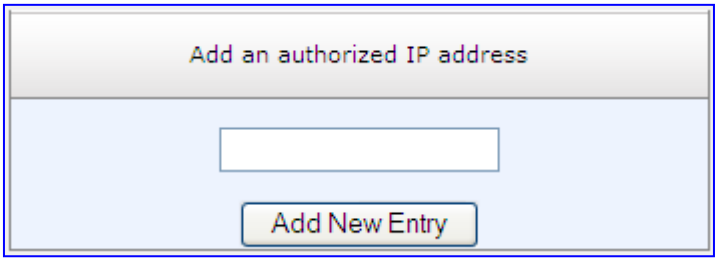
4.7 Configuring Web and Telnet Access List

The Web & Telnet Access List page is used to define IP addresses (up to ten) that are permitted to access the device's Web, Telnet, and SSH interfaces. Access from an undefined IP address is denied. If no IP addresses are defined, this security feature is inactive and the device can be accessed from any IP address. The Web and Telnet Access List can also be defined using the *ini* file parameter WebAccessList_x (see 'Web and Telnet Parameters' on page 764).

➢ **To add authorized IP addresses for Web, Telnet, and SSH interfaces access:**

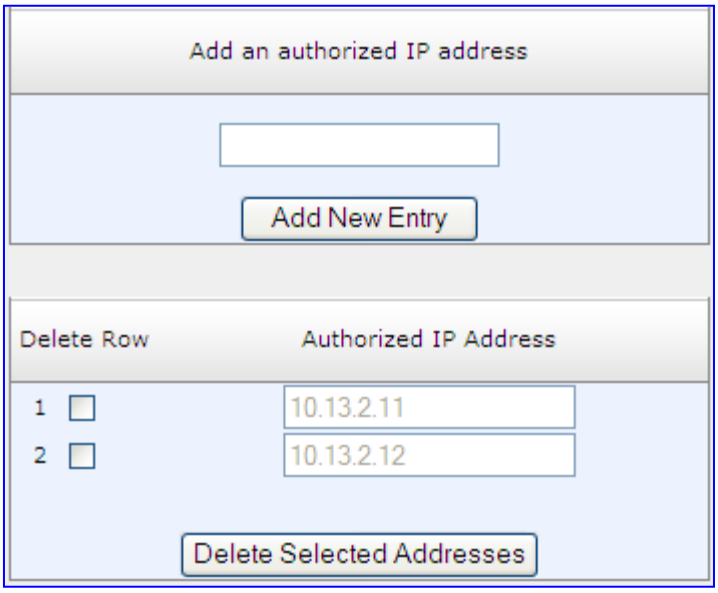
1. Open the Web & Telnet Access List page (**Configuration** tab > **System** menu > **Management** submenu > **Web & Telnet Access List**).

Figure 4-36: Web & Telnet Access List Page - Add New Entry



2. To add an authorized IP address, in the 'Add an authorized IP address' field, enter the required IP address, and then click **Add New Entry**; the IP address you entered is added as a new entry to the Web & Telnet Access List table.

Figure 4-37: Web & Telnet Access List Table



3. To delete authorized IP addresses, select the Delete Row check boxes corresponding to the IP addresses that you want to delete, and then click **Delete Selected Addresses**; the IP addresses are removed from the table and these IP addresses can no longer access the Web and Telnet interfaces.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Notes:

- The first authorized IP address in the list must be your PC's (terminal) IP address; otherwise, access from your PC is denied.
- Delete your PC's IP address last from the 'Web & Telnet Access List page. If it is deleted before the last, subsequent access to the device from your PC is denied.

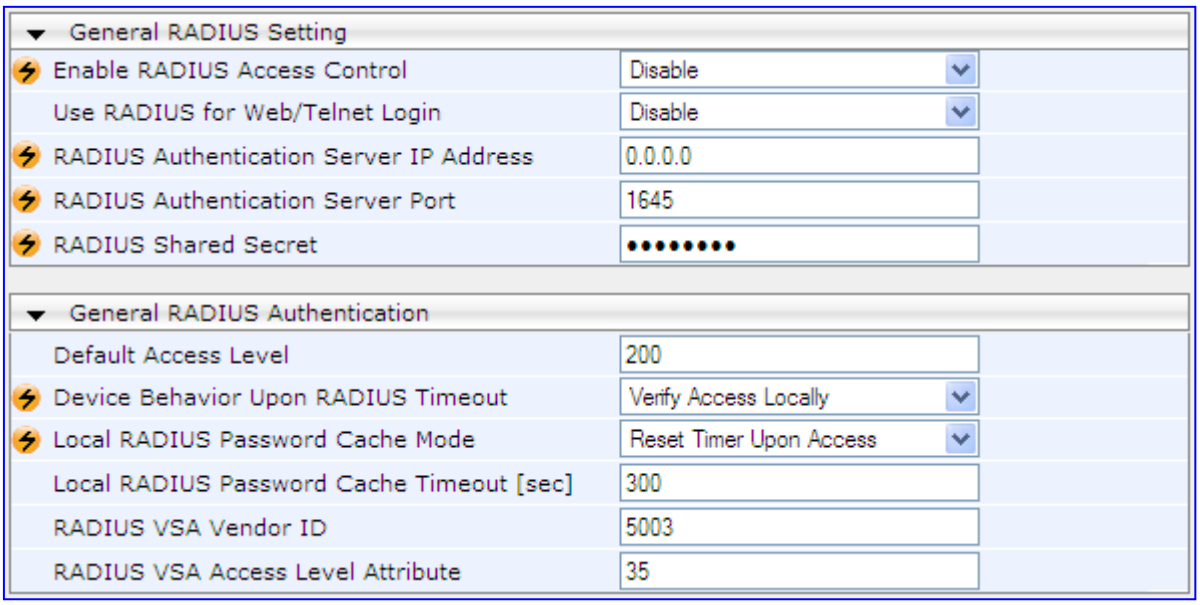
4.8 Configuring RADIUS Settings

The RADIUS Settings page is used for configuring the Remote Authentication Dial In User Service (RADIUS) accounting parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 752.

➢ **To configure RADIUS:**

1. Open the RADIUS Settings page (**Configuration** tab > **System** menu > **Management** submenu > **RADIUS Settings**).

Figure 4-38: RADIUS Parameters Page



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

5 CLI-Based Management

This section provides an overview of the CLI-based management and configuration relating to CLI management.

The CLI can be accessed by using the RS-232 serial port or by using SSH or Telnet through the Ethernet interface.

Once logged into the CLI with your username and password, you can configure the device by accessing one of the following modes:

- Basic command mode: Provides general CLI commands, for example, to display system information and activate debugging. This mode is accessed immediately after you login to the CLI.
- Enable command mode: Provides the configuration commands.

  To access this mode, type the following:

  ```
  # enable
  # Password: <password>
  ```

  This mode groups the commands under the following command sets:

  - configure-system: This contains the general and system related configuration commands, for example, Syslog configuration. This set is accessed by typing the following:

    ```
    # configure system
    ```

  - configure-data: This contains the data-router configuration commands. This set is accessed by typing the following:

    ```
    # configure data
    ```

  - configure-voip: This contains VoIP-related configuration commands, for example, SIP, VoIP network interfaces, and VoIP media configurations. This set is accessed by typing the following:

    ```
    # configure voip
    ```

Notes:

- For information on accessing the CLI interface, see 'Using CLI' on page 29.
- For more information on using CLI and for a description of the CLI commands, refer to the following documents: *MSBG Data CLI Reference Guide* and *MSBG VoIP and System CLI Reference Guide*.

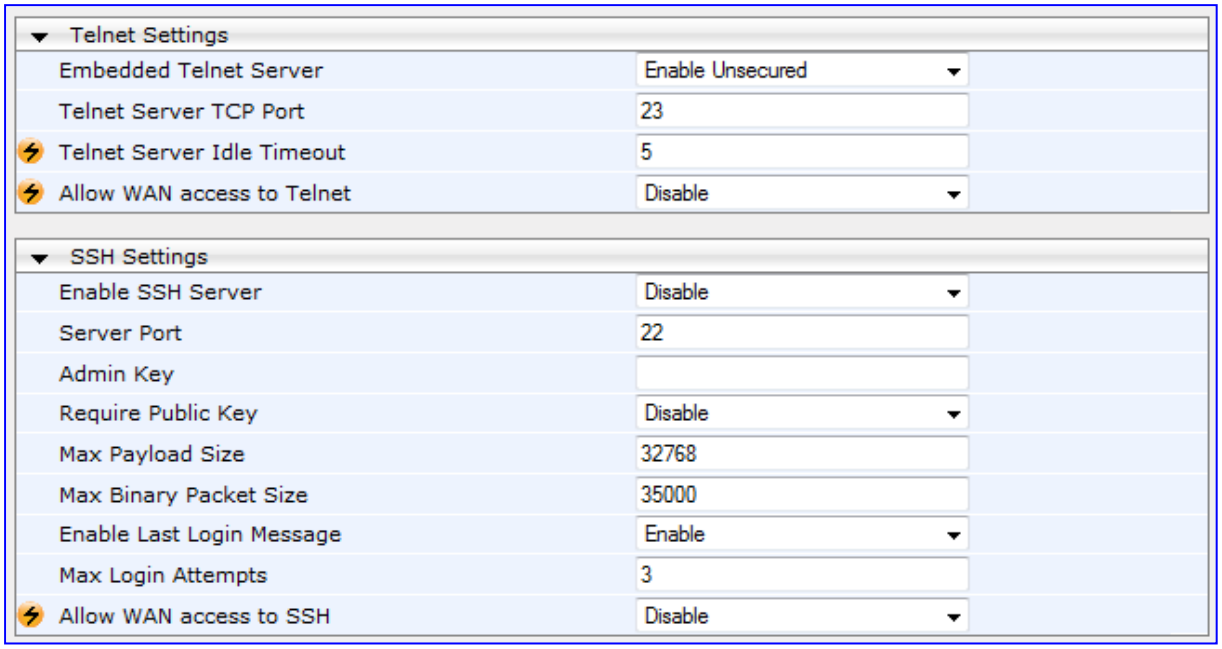
5.1 Configuring Telnet and SSH Settings

The Telnet/SSH Settings page is used to define Telnet and Secure Shell (SSH). For a description of these parameters, see 'Web and Telnet Parameters' on page 764.

➢ **To define Telnet and SSH:**

1. Open the Telnet/SSH Settings page (**Configuration** tab > **System** menu > **Management** submenu > **Telnet/SSH Settings**).

Figure 5-1: Telnet/SSH Settings Page



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

5.2 TACACS+ for CLI Login

This section describes how to enable and configure Terminal Access Controller Access-Control System (TACACS+). TACACS+ is a security protocol for centralized username and password verification. TACACS+ can be used for validating users attempting to gain access to the device through CLI. TACACS+ services are maintained on a database on a TACACS+ daemon. You must have access to and must configure a TACACS+ server before configuring TACACS+ on your device.

TACACS+ can provide the following services:

- Authentication: provides authentication through login and password dialog
- Authorization: manages user capabilities for the duration of the user's session by placing restrictions on what commands a user may execute
- Accounting: collects and sends information for auditing and reporting to the TACACS+ daemon

The TACACS+ protocol provides authentication between the device and the TACACS+ daemon, and it ensures confidentiality as all protocol exchanges between a network access server and a TACACS+ daemon are encrypted. You need a system running TACACS+ daemon software to use the TACACS+ functionality on your network access server.

When a user attempts a simple ASCII login by authenticating to a network access server using TACACS+, the following typically occurs:

1. When the connection is established, the network access server contacts the TACACS+ daemon to obtain a username prompt, which is then displayed to the user. The user enters a username and the network access server then contacts the TACACS+ daemon to obtain a password prompt. The network access server displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.
2. The network access server eventually receives one of the following responses from the TACACS+ daemon:
   - ACCEPT: The user is authenticated and service may begin. If the network access server is configured to require authorization, authorization will begin at this time.
   - REJECT: The user has failed to authenticate. The user may be denied further access.
   - ERROR: An error occurred at some time during authentication. This can be at the daemon or in the network connection between the daemon and the network access server. If an ERROR response is received, the device typically attempts to use an alternative method for authenticating the user.
3. If TACACS+ authorization is needed, the TACACS+ daemon is again contacted for each CLI command entered by the user, and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the CLI command is allowed; otherwise, it is rejected.

5.2.1 Configuring TACACS+ using CLI

The following CLI commands are used for enabling and configuring TACACS+:

- **aaa authentication login tacacs+:** enables TACACS+
- **tacacs-server host <host-ip>:** defines the IP address of the TACACS+ server (up to two servers can be configured)
- **tacacs-server port <port-num>:** defines the TCP port number for the TACACS+ service
- **tacacs-server key <password>:** defines the shared secret between the TACACS+ server and the device
- **tacacs-server timeout <seconds>:** defines how much time to wait for a TACACS+ response before failing the authentication

The procedure below describes how to configure TACACS+ through CLI.

➢ **To configure TACACS+ through the CLI:**

1. Establish serial communication with the device.
2. At the prompt, type the following command to access the data interface and then press Enter:

```
# configure data
```

3. At the prompt, type the TACACS+ commands, as required, to enable and configure TACACS+. Below shows an example configuration:

```
(config-data)# aaa authentication login tacacs+
(config-data)# tacacs-server host 192.168.1.55
(config-data)# tacacs-server key Tumble
```

6 SNMP-Based Management

The device provides an embedded SNMP Agent to operate with a third-party SNMP Manager (e.g., element management system or EMS) for operation, administration, maintenance, and provisioning (OAMP) of the device. The SNMP Agent supports standard Management Information Base (MIBs) and proprietary MIBs, enabling a deeper probe into the interworking of the device. The SNMP Agent can also send unsolicited events (SNMP traps) towards the SNMP Manager. All supported MIB files are supplied to customers as part of the release.

This section provides configuration relating to SNMP management.

Note: For more information on SNMP support, refer to the *Product Reference Manual*.

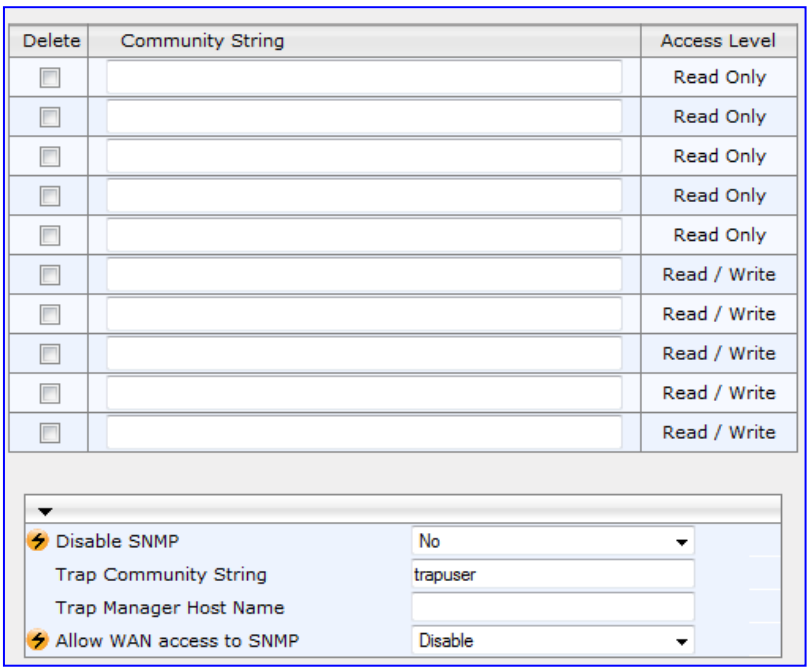
6.1 Configuring SNMP Community Strings

The SNMP Community String page allows you to configure up to five read-only and up to five read-write SNMP community strings, and to configure the community string that is used for sending traps. For more information on SNMP community strings, refer to the *Product Reference Manual*. For detailed descriptions of the SNMP parameters, see 'SNMP Parameters' on page 767.

➢ **To configure the SNMP community strings:**

1. Open the SNMP Community String page (**Maintenance** tab > **System** menu > **Management** submenu > **SNMP** submenu > **SNMP Community String**).

Figure 6-1: SNMP Community String Page



2. Configure the SNMP community strings parameters according to the table below.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

To delete a community string, select the **Delete** check box corresponding to the community string that you want to delete, and then click **Submit**.

Table 6-1: SNMP Community String Parameters Description

Parameter	Description
Community String	▪ Read Only **[SNMPReadOnlyCommunityString_x]:** Up to five read-only community strings (up to 19 characters each). The default string is 'public'. ▪ Read / Write **[SNMPReadWriteCommunityString_x]:** Up to five read / write community strings (up to 19 characters each). The default string is 'private'.
Trap Community String **[SNMPTrapCommunityString]**	Community string used in traps (up to 19 characters). The default string is 'trapuser'.

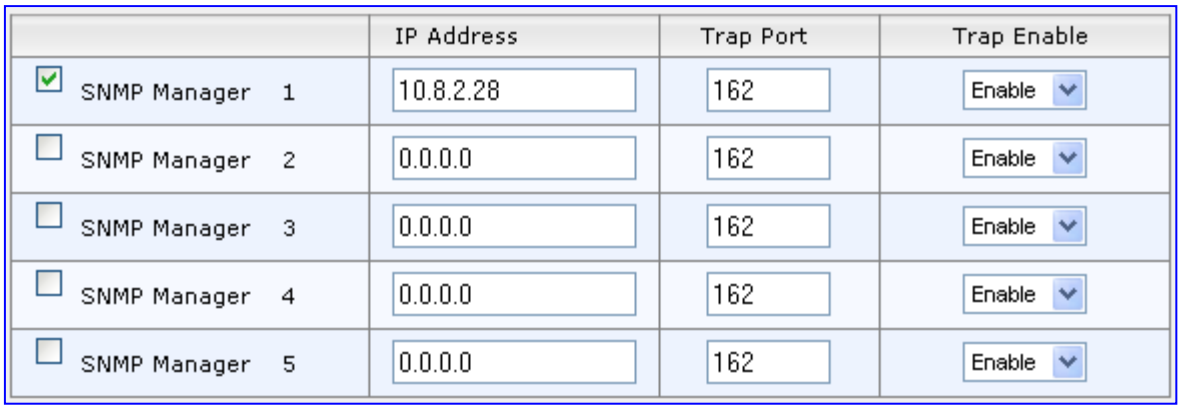
6.2 Configuring SNMP Trap Destinations

The SNMP Trap Destinations page allows you to configure up to five SNMP trap managers.

➢ **To configure SNMP trap destinations:**

1. Open the SNMP Trap Destinations page (**Maintenance** tab > **System** menu > **Management** submenu > **SNMP** submenu > **SNMP Trap Destinations**).

Figure 6-2: SNMP Trap Destinations Page



2. Configure the SNMP trap manager parameters according to the table below.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Note: Only table row entries whose corresponding check boxes are selected are applied when clicking **Submit**; otherwise, settings revert to their defaults.

Table 6-2: SNMP Trap Destinations Parameters Description

Parameter	Description
SNMP Manager **[SNMPManagerIsUsed_x]**	Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used

Parameter	Description
	to receive SNMP traps. ▪ **[0]** (Check box cleared) = Disabled (default) ▪ **[1]** (Check box selected) = Enabled
IP Address **[SNMPManagerTableIP_x]**	IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to these IP addresses. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255.
Trap Port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to these ports. The valid SNMP trap port range is 100 to 4000. The default port is 162.
Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Activates or de-activates the sending of traps to the corresponding SNMP Manager. ▪ **[0]** Disable = Sending is disabled. ▪ **[1]** Enable = Sending is enabled (default).

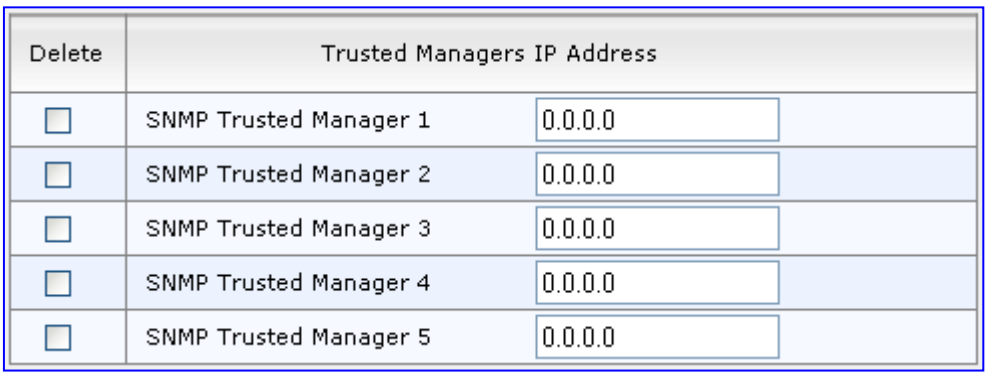
6.3 Configuring SNMP Trusted Managers

The SNMP Trusted Managers page allows you to configure up to five SNMP Trusted Managers, based on IP addresses. By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests.

➢ **To configure SNMP Trusted Managers:**

1. Open the SNMP Trusted Managers page (**Maintenance** tab > **System** menu > **Management** submenu > **SNMP** submenu > **SNMP Trusted Managers**).

Figure 6-3: SNMP Trusted Managers



2. Select the check box corresponding to the SNMP Trusted Manager that you want to enable and for whom you want to define an IP address.
3. Define an IP address in dotted-decimal notation.
4. Click Submit to apply your changes.
5. To save the changes, see 'Saving Configuration' on page 680.

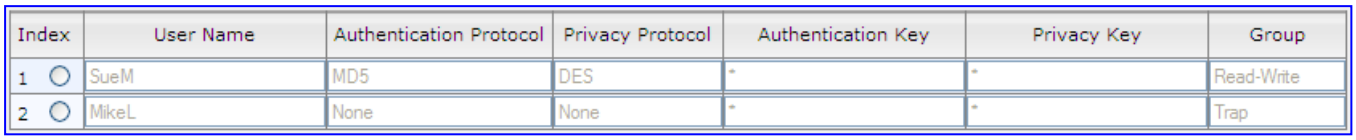
6.4 Configuring SNMP V3 Users

The SNMP v3 Users page allows you to configure authentication and privacy for up to 10 SNMP v3 users.

➢ **To configure the SNMP v3 users:**

1. Open the SNMP v3 Users page (**Maintenance** tab > **System** menu > **Management** submenu > **SNMP** submenu > **SNMP V3 Users**).

Figure 6-4: SNMP V3 Setting Page



Index	User Name	Authentication Protocol	Privacy Protocol	Authentication Key	Privacy Key	Group
1	SueM	MD5	DES	*	*	Read-Write
2	MikeL	None	None	*	*	Trap

2. To add an SNMP v3 user, in the 'Add Index' field, enter the desired row index, and then click **Add Index**. A new row appears.
3. Configure the SNMP V3 Setting parameters according to the table below.
4. Click the **Apply** button to save your changes.
5. To save the changes, see 'Saving Configuration' on page 680.

Notes:

- For a description of the web interface's table command buttons (e.g., **Duplicate** and **Delete**), see 'Working with Tables' on page 62.
- You can also configure SNMP v3 users using the *ini* file table parameter SNMPUsers (see 'SNMP Parameters' on page 767).

Table 6-3: SNMP V3 Users Parameters

Parameter	Description
Index **[SNMPUsers_Index]**	The table index. The valid range is 0 to 9.
User Name **[SNMPUsers_Username]**	Name of the SNMP v3 user. This name must be unique.
Authentication Protocol **[SNMPUsers_AuthProtocol]**	Authentication protocol of the SNMP v3 user. ▪ **[0]** None (default) ▪ **[1]** MD5 ▪ **[2]** SHA-1
Privacy Protocol **[SNMPUsers_PrivProtocol]**	Privacy protocol of the SNMP v3 user. ▪ **[0]** None (default) ▪ **[1]** DES ▪ **[2]** 3DES ▪ **[3]** AES-128 ▪ **[4]** AES-192 ▪ **[5]** AES-256
Authentication Key **[SNMPUsers_AuthKey]**	Authentication key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
Privacy Key	Privacy key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and

Parameter	Description
[SNMPUsers_PrivKey]	keys are localized.
Group **[SNMPUsers_Group]**	The group with which the SNMP v3 user is associated. ▪ **[0]** Read-Only (default) ▪ **[1]** Read-Write ▪ **[2]** Trap **Note:** All groups can be used to send traps.

Reader's Notes

7 EMS-Based Management

AudioCodes Element Management System (EMS)is an advanced solution for standards-based management of MSBGs within VoP networks, covering all areas vital for the efficient operation, administration, management and provisioning (OAM&P) of AudioCodes' families of MSBGs. The EMS enables Network Equipment Providers (NEPs) and System Integrators (SIs) the ability to offer customers rapid time-to-market and inclusive, cost-effective management of next-generation networks. The standards-compliant EMS uses distributed SNMP-based management software, optimized to support day-to-day Network Operation Center (NOC) activities, offering a feature-rich management framework. It supports fault management, configuration and security.

Note: For more information on using the EMS tool, refer to the *EMS User's Manual* and *EMS Server IOM Manual*.

Reader's Notes

8 INI File-Based Management

The *ini* file is a text-based file (created using, for example, Notepad) that can contain any number of parameters settings. The *ini* file can be loaded to the device using the following methods:

- Web interface (see 'Backing Up and Loading Configuration File' on page 702)
- AudioCodes' BootP/TFTP utility (refer to the Product Reference Manual)
- Any standard TFTP server

When loaded to the device, the configuration settings of the *ini* file are saved to the device's non-volatile memory. If a parameter is excluded from the loaded *ini* file, the following occurs, depending on how you load the file:

- Using the Load Auxiliary Files page (see 'Loading Auxiliary Files' on page 681): current settings are retained for excluded parameters
- All other methods: default value is assigned to excluded parameters (according to the .cmp file running on the device), thereby, overriding values previously defined for these parameters

Notes:

- For a list and description of the *ini* file parameters, see 'Configuration Parameters Reference' on page 752.
- Some parameters are configurable only through the *ini* file (and not the Web interface).
- To restore the device to default settings using the *ini* file, see 'Restoring Factory Default Settings' on page 705.

8.1 INI File Format

The *ini* file can be configured with any number of parameters. These *ini* file parameters can be one of the following types:

- Individual parameters (see 'Configuring Individual ini File Parameters' on page 98)
- Table parameters (see 'Configuring ini File Table Parameters' on page 98)

8.1.1 Configuring Individual ini File Parameters

The format of individual *ini* file parameters includes an optional, subsection name (group name) to conveniently group similar parameters by their functionality. Following this line are the actual parameter settings. These format lines are shown below:

```
[subsection name]
; the subsection name is optional.
Parameter_Name = Parameter_Value
Parameter_Name = Parameter_Value
; Remark
; For example:
[System Parameters]
SyslogServerIP = 10.13.2.69
EnableSyslog = 1
; these are a few of the system-related parameters.
```

For general *ini* file formatting rules, see 'General ini File Formatting Rules' on page 100.

8.1.2 Configuring ini File Table Parameters

The *ini* file table parameters allow you to configure tables which can include multiple parameters (*columns*) and row entries (*indices*). When loading an *ini* file to the device, it's recommended to include only tables that belong to applications that are to be configured (dynamic tables of other applications are empty, but static tables are not).

The *ini* file table parameter is composed of the following elements:

- **Title of the table:** The name of the table in square brackets (e.g., **[MY**_TABLE_NAME]).
- **Format line:** Specifies the columns of the table (by their string names) that are to be configured.
  - The first word of the Format line must be 'FORMAT', followed by the Index field name and then an equal (=) sign. After the equal sign, the names of the columns are listed.
  - Columns must be separated by a comma (,).
  - The Format line must only include columns that can be modified (i.e., parameters that are not specified as read-only). An exception is Index fields, which are mandatory.
  - The Format line must end with a semicolon (;).

- **Data line(s):** Contain the actual values of the columns (parameters). The values are interpreted according to the Format line.
  - The first word of the Data line must be the table's string name followed by the Index field.
  - Columns must be separated by a comma (,).
  - A Data line must end with a semicolon (;).
- **End-of-Table Mark:** Indicates the end of the table. The same string used for the table's title, preceded by a backslash (\), e.g., **[\MY**_TABLE_NAME].

The following displays an example of the structure of an *ini* file table parameter.

```
[Table_Title]
; This is the title of the table.
FORMAT Index = Column_Name1, Column_Name2, Column_Name3;
; This is the Format line.
Index 0 = value1, value2, value3;
Index 1 = value1, $$, value3;
; These are the Data lines.
[\Table_Title]
; This is the end-of-the-table-mark.
```

The *ini* file table parameter formatting rules are listed below:

- Indices (in both the Format and the Data lines) must appear in the same order. The Index field must never be omitted.
- The Format line can include a subset of the configurable fields in a table. In this case, all other fields are assigned with the pre-defined default values for each configured line.
- The order of the fields in the Format line isn't significant (as opposed to the Index fields). The fields in the Data lines are interpreted according to the order specified in the Format line.
- The double dollar sign ($$) in a Data line indicates the default value for the parameter.
- The order of the Data lines is insignificant.
- Data lines must match the Format line, i.e., it must contain exactly the same number of Indices and Data fields and must be in exactly the same order.
- A row in a table is identified by its table name and Index field. Each such row may appear only once in the *ini* file.
- Table dependencies: Certain tables may depend on other tables. For example, one table may include a field that specifies an entry in another table. This method is used to specify additional attributes of an entity, or to specify that a given entity is part of a larger entity. The tables must appear in the order of their dependency (i.e., if Table X is referred to by Table Y, Table X must appear in the *ini* file before Table Y).

For general *ini* file formatting rules, see 'General ini File Formatting Rules' on page 100.

The table below displays an example of an *ini* file table parameter:

```
[ CodersGroup0 ]
FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime,
CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce;
CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0;
CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0;
[ \CodersGroup0 ]
```

Note: Do not include read-only parameters in the *ini* file table parameter as this can cause an error when attempting to load the file to the device.

8.1.3 General ini File Formatting Rules

The *ini* file must adhere to the following formatting rules:

- The *ini* file name must not include hyphens (-) or spaces; if necessary, use an underscore (_) instead.
- Lines beginning with a semi-colon (;) are ignored. These can be used for adding remarks in the *ini* file.
- A carriage return (i.e., Enter) must be done at the end of each line.
- The number of spaces before and after the equals sign (=) is irrelevant.
- Subsection names for grouping parameters are optional.
- If there is a syntax error in the parameter name, the value is ignored.
- Syntax errors in the parameter's value can cause unexpected errors (parameters may be set to the incorrect values).
- Parameter string values that denote file names (e.g., CallProgressTonesFileName) must be enclosed with inverted commas ('…'), e.g., CallProgressTonesFileName = 'cpt_usa.dat'
- The parameter name is not case-sensitive.
- The parameter value is not case-sensitive, except for coder names.
- The *ini* file must end with at least one carriage return.

8.2 Modifying an ini File

You can modify an *ini* file currently used by the device. Modifying an *ini* file instead of loading an entirely new *ini* file preserves the device's current configuration.

➢ **To modify an *ini* file:**

1. Save the current *ini* file from the device to your PC, using the Web interface (see 'Backing Up and Loading Configuration File' on page 702).
2. Open the *ini* file (using a text file editor such as Notepad), and then modify the *ini* file parameters according to your requirements.
3. Save the modified *ini* file, and then close the file.
4. Load the modified *ini* file to the device, using the Web interface (see 'Backing Up and Loading Configuration File' on page 702).

Tip: Before loading the *ini* file to the device, verify that the file extension of the *ini* file is correct, i.e., *.ini*.

8.3 Secured Encoded ini File

The *ini* file contains sensitive information that is required for the functioning of the device. The file may be loaded to the device using HTTP. These protocols are not secure and are vulnerable to potential hackers. To overcome this security threat, the AudioCodes' TrunkPack Downloadable Conversion Utility (DConvert) utility allows you to binary-encode (encrypt) the *ini* file before loading it to the device (refer to the *Product Reference Manual)*.

Notes:

- The procedure for loading an encoded *ini* file is identical to the procedure for loading an unencoded *ini* file (see 'Backing Up and Loading Configuration File' on page 702).
- If you download from the device (to a folder on your PC) an *ini* file that was loaded encoded to the device, the file is saved as a regular *ini* file (i.e., unencoded).

Part III

General System Settings

This part provides general system configurations.

Reader's Notes

9 Configuring Certificates

The Certificates page is used for configuring secure communication using HTTPS and SIP TLS. This page allows you to do the following:

- Replace the device's certificate - see 'Replacing Device Certificate' on page 105
- Load a new private key from an external source - see 'Loading a Private Key' on page 108
- Configure trusted root certificates - see 'Mutual TLS Authentication' on page 109
- Regenerate keys and self-signed certificates - see 'Self-Signed Certificates' on page 110

Note: The device is shipped with a working TLS configuration. Therefore, configure certificates only if required.

9.1 Replacing Device Certificate

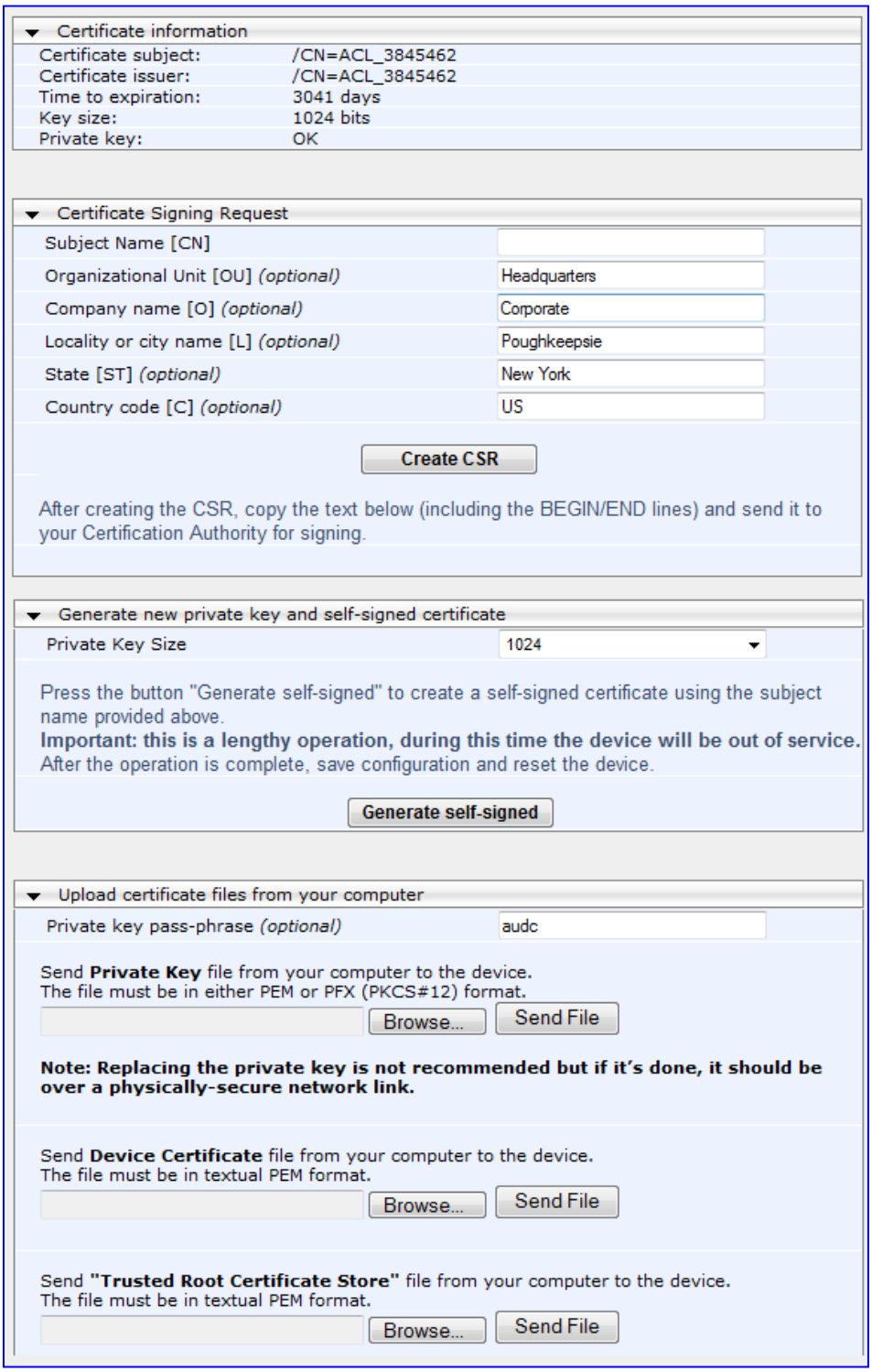
The device is supplied with a working Transport Layer Security (TLS) configuration consisting of a unique self-signed server certificate. If an organizational Public Key Infrastructure (PKI) is used, you may wish to replace this certificate with one provided by your security administrator.

➢ **To replace the device's certificate:**

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This DNS name is used to access the device and therefore, must be listed in the server certificate.
2. If the device is operating in HTTPS mode, then set the 'Secured Web Connection (HTTPS)' field (HTTPSOnly) to **HTTP and HTTPS** (see 'Configuring Web Security Settings' on page 79). This ensures that you have a method for accessing the device in case the new certificate does not work. Restore the previous setting after testing the configuration.

3. Open the Certificates page (**Configuration** tab > **System** menu > **Certificates**).

Figure 9-1: Certificates Page



4. Under the **Certificate Signing Request** group, do the following:
   a. In the 'Subject Name **[CN]**' field, enter the DNS name.
   b. Fill in the rest of the request fields according to your security provider's instructions.
   c. Click **Create CSR**; a textual certificate signing request is displayed.

5. Copy the text and send it to your security provider. The security provider (also known as Certification Authority or CA) signs this request and then sends you a server certificate for the device.
6. Save the certificate to a file (e.g., cert.txt). Ensure that the file is a plain-text file containing the 'BEGIN CERTIFICATE' header, as shown in the example of a Base64-Encoded X.509 Certificate below:

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIEAgAAADANBgkqhkiG9w0BAQQFADA/MQswCQYDVQQGEwJGUj
ETMBEGA1UEChMKQ2VydGlwb3N0ZTEbMBkGA1UEAxMSQ2VydGlwb3N0ZSBTZXJ2ZXVy
MB4XDTk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1UEBhMCRlIxEz
ARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9zdGUgU2VydmV1cjCC
ASEwDQYJKoZIhvcNAQEBBQADggEOADCCAQkCggEAPqd4MziR4spWldGRx8bQrhZkon
WnNm`+Yhb7+4Q67ecf1janH7GcN/SXsfx7jJpreWULf7v7Cvpr4R7qIJcmdHIntmf7
JPM5n6cDBv17uSW63er7NkVnMFHwK1QaGFLMybFkzaeGrvFm4k3lRefiXDmuOe+FhJ
gHYezYHf44LvPRPwhSrzi9+Aq3o8pWDguJuZDIUP1F1jMa+LPwvREXfFcUW+w==
-----END CERTIFICATE-----
```

7. Scroll down to the **Upload certificates files from your computer** group, click the **Browse** button corresponding to the 'Send Device Certificate...' field, navigate to the cert.txt file, and then click **Send File**.
8. After the certificate successfully loads to the device, save the configuration with a device reset (see 'Saving Configuration' on page 680); the Web interface uses the provided certificate.
9. Open the Certificates page again and verify that under the **Certificate information** group (at the top of the page), the 'Private key' read-only field displays "OK"; otherwise, consult your security administrator.
10. If the device was originally operating in HTTPS mode and you disabled it in Step 2, then return it to HTTPS by setting the 'Secured Web Connection (HTTPS)' field to **HTTPS Only**.

Notes:

- The certificate replacement process can be repeated when necessary (e.g., the new certificate expires).
- It is possible to use the IP address of the device (e.g., 10.3.3.1) instead of a qualified DNS name in the Subject Name. This is not recommended since the IP address is subject to changes and may not uniquely identify the device.
- The device certificate can also be loaded via the Automatic Update Facility, using the HTTPSCertFileName *ini* file parameter.

9.2 Loading a Private Key

The device is shipped with a self-generated random private key, which cannot be extracted from the device. However, some security administrators require that the private key be generated externally at a secure facility and then loaded to the device through configuration. Since private keys are sensitive security parameters, take precautions to load them over a physically-secure connection such as a back-to-back Ethernet cable connected directly to the managing computer.

➢ **To replace the device's private key:**

1. Your security administrator should provide you with a private key in either textual PEM (PKCS #7) or PFX (PKCS #12) format. The file may be encrypted with a short pass-phrase, which should be provided by your security administrator.
2. If the device is operating in HTTPS mode, then set the 'Secured Web Connection (HTTPS)' field (HTTPSOnly) to **HTTP and HTTPS** (see 'Configuring Web Security Settings' on page 79). This ensures that you have a method for accessing the device in case the new configuration does not work. Restore the previous setting after testing the configuration.
3. Open the Certificates page (**Configuration** tab > **System** menu > **Certificates**) and scroll down to the **Upload certificate files from your computer** group.
4. Fill in the 'Private key pass-phrase' field, if required.
5. Click the **Browse** button corresponding to the 'Send Private Key' field, navigate to the key file, and then click **Send File**.
6. If the security administrator has provided you with a device certificate file, load it using the 'Send Device Certificate' field.
7. After the files successfully load to the device, save the configuration with a device reset (see 'Saving Configuration' on page 680); the Web interface uses the new configuration.
8. Open the Certificates page again, and verify that under the **Certificate information** group (at the top of the page) the 'Private key' read-only field displays "OK"; otherwise, consult your security administrator.
9. If the device was originally operating in HTTPS mode and you disabled it in Step 2, then enable it by setting the 'Secured Web Connection (HTTPS)' field to **HTTPS Only**.

9.3 Mutual TLS Authentication

By default, servers using TLS provide one-way authentication. The client is certain that the identity of the server is authentic. When an organizational PKI is used, two-way authentication may be desired - both client and server should be authenticated using X.509 certificates. This is achieved by installing a client certificate on the managing PC and loading the root CA's certificate to the device's Trusted Root Certificate Store. The Trusted Root Certificate file may contain more than one CA certificate combined, using a text editor.

Since X.509 certificates have an expiration date and time, the device must be configured to use NTP (see 'Simple Network Time Protocol Support' on page 112) to obtain the current date and time. Without the correct date and time, client certificates cannot work.

➢ **To enable mutual TLS authentication for HTTPS:**

1. Set the 'Secured Web Connection (HTTPS)' field to **HTTPS Only** (see 'Configuring Web Security Settings' on page 79) to ensure you have a method for accessing the device in case the client certificate does not work. Restore the previous setting after testing the configuration.
2. Open the Certificates page (see 'Replacing Device Certificate' on page 105).
3. In the **Upload certificate files from your computer** group, click the **Browse** button corresponding to the 'Send Trusted Root Certificate Store ...' field, navigate to the file, and then click **Send File**.
4. When the operation is complete, set the 'Requires Client Certificates for HTTPS connection' field to **Enable** (see 'Configuring Web Security Settings' on page 79).
5. Save the configuration with a device reset (see 'Saving Configuration' on page 680).

When a user connects to the secured Web interface of the device:

- If the user has a client certificate from a CA that is listed in the Trusted Root Certificate file, the connection is accepted and the user is prompted for the system password.
- If both the CA certificate and the client certificate appear in the Trusted Root Certificate file, the user is not prompted for a password (thus, providing a single-sign-on experience - the authentication is performed using the X.509 digital signature).
- If the user does not have a client certificate from a listed CA or does not have a client certificate, the connection is rejected.

Notes:

- The process of installing a client certificate on your PC is beyond the scope of this document. For more information, refer to your operating system documentation, and/or consult your security administrator.
- The root certificate can also be loaded via the Automatic Update facility, using the HTTPSRootFileName *ini* file parameter.
- You can enable Online Certificate Status Protocol (OCSP) on the device to check whether a peer's certificate has been revoked by an OCSP server. For more information, refer to the *Product Reference Manual*.

9.4 Self-Signed Certificates

The device is shipped with an operational, self-signed server certificate. The subject name for this default certificate is 'ACL_nnnnnnn', where *nnnnnnn* denotes the serial number of the device. However, this subject name may not be appropriate for production and can be changed while still using self-signed certificates.

➢ **To change the subject name and regenerate the self-signed certificate:**

1. Before you begin, ensure the following:
   - You have a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device and should therefore, be listed in the server certificate.
   - No traffic is running on the device. The certificate generation process is disruptive to traffic and should be executed during maintenance time.
2. Open the Certificates page (see 'Replacing Device Certificate' on page 105).
3. In the 'Subject Name **[CN]**' field, enter the fully-qualified DNS name (FQDN) as the certificate subject, select the desired private key size (in bits), and then click **Generate self-signed**; after a few seconds, a message appears displaying the new subject name.
4. Save the configuration with a device reset (see 'Saving Configuration' on page 680) for the new certificate to take effect.

10 Date and Time

The date and time of the device can be configured manually or it can be obtained automatically from a Simple Network Time Protocol (SNTP) server.

10.1 Manual Date and Time

The date and time of the device can be configured manually.

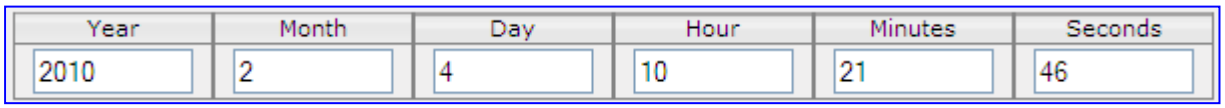
10.1.1 Configuring Date and Time Manually using Web Interface

The Regional Settings page allows you to define and view the device's internal date and time.

➢ **To configure the device's date and time:**

1. Open the Regional Settings page (**Configuration** tab > **System** menu > **Regional Settings**).

Figure 10-1: Regional Settings Page



2. Enter the current date and time in the geographical location in which the device is installed.
3. Click the **Submit** button; the date and time are automatically updated.

Notes:

- If the device is configured to obtain the date and time from an Simple Network Time Protocol Support (SNTP) server, the fields on this page display the received date and time and are read-only.
- After performing a hardware reset, the date and time are returned to their defaults and therefore, should be updated.

10.1.2 Configuring Date and Time Manually using CLI

The procedure below describes how to configure the date and time manually using the Web interface.

➢ **To configure date and time manually using the CLI:**

1. Access the system mode:
```
# configure system
```
2. At the prompt, enter the following command, and then press Enter:
```
(configure-system) # clock
```
3. Set (or display) the date using the following command:
```
(clock) # set date
```
4. Set (or display) the time using the following command:
```
(clock) # set time
```

10.2 Automatic Date and Time through SNTP Server

The Simple Network Time Protocol (SNTP) client functionality generates requests and reacts to the resulting responses using the NTP version 3 protocol definitions (according to RFC 1305). Through these requests and responses, the NTP client synchronizes the system time to a time source within the network, thereby eliminating any potential issues should the local system clock 'drift' during operation. By synchronizing time to a network time source, traffic handling, maintenance, and debugging become simplified for the network administrator.

The NTP client follows a simple process in managing system time: the NTP client requests an NTP update, receives an NTP response, and then updates the local system clock based on a configured NTP server within the network.

The client requests a time update from a specified NTP server at a specified update interval. In most situations, this update interval is every 24 hours based on when the system was restarted. The NTP server identity (as an IP address) and the update interval are user-defined (using the *ini* file parameters NTPServerIP and NTPUpdateInterval respectively), or an SNMP MIB object (refer to the *Product Reference Manual*).

When the client receives a response to its request from the identified NTP server, it must be interpreted based on time zone or location offset that the system is to a standard point of reference called the Universal Time Coordinate (UTC). The time offset that the NTP client uses is configurable using the *ini* file parameter NTPServerUTCOffset, or via an SNMP MIB object (refer to the *Product Reference Manual*).

If required, the clock update is performed by the client as the final step of the update process. The update is performed in such a way as to be transparent to the end users. For instance, the response of the server may indicate that the clock is running too fast on the client. The client slowly robs bits from the clock counter to update the clock to the correct time. If the clock is running too slow, then in an effort to catch the clock up, bits are added to the counter, causing the clock to update quicker and catch up to the correct time. The advantage of this method is that it does not introduce any disparity in the system time that is noticeable to an end user or that could corrupt call timeouts and timestamps.

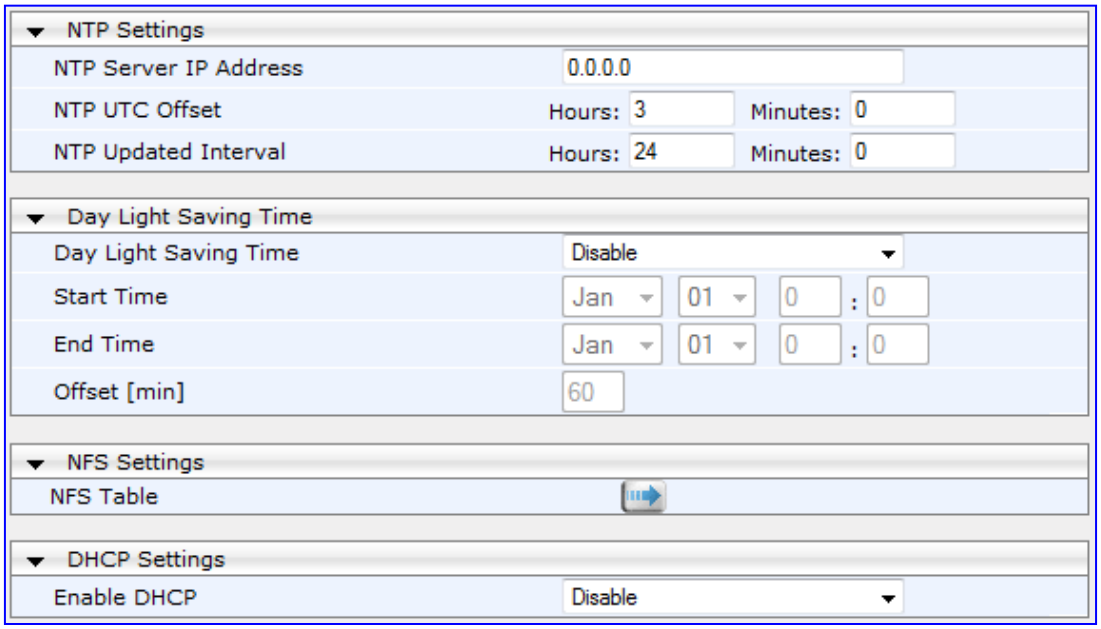
10.2.1 Configuring SNTP using Web Interface

The procedure below describes how to configure SNTP using the Web interface.

➢ **To configure SNTP using the Web interface:**

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).

Figure 10-2: Applications Settings Page



2. Configure the NTP parameters:
   - 'NTP Server IP Address' (NTPServerIP) - defines the IP address of the NTP server
   - 'NTP UTC Offset' (NTPServerUTCOffset) - defines the time offset in relation to the UTC. For example, if your region is 2 hours ahead of the UTC, enter "2".
   - 'NTP Updated Interval' (NTPUpdateInterval) - defines the period after which the date and time of the device is updated
3. Configure daylight saving, if required:
   - 'Day Light Saving Time' (DayLightSavingTimeEnable) - enables daylight saving time
   - 'Start Time' (DayLightSavingTimeStart) and 'End Time' (DayLightSavingTimeEnd) - defines the period for which daylight saving time is relevant.
   - 'Offset' (DayLightSavingTimeOffset) - defines the offset in minutes to add to the time for daylight saving. For example, if your region has daylight saving of one hour, the time received from the NTP server is 11:00, and the UTC offset for your region is +2 (i.e., 13:00), you need to enter "60" to change the local time to 14:00.
4. Verify that the device is set to the correct date and time. You can do this by viewing the date and time in the Regional Settings page, as described in 'Configuring Date and Time' on page 111.

10.2.2 Configuring SNTP using CLI

The procedure below describes how to configure SNTP using the Web interface.

➢ **To configure SNTP using the CLI:**

1. Configure the NTP parameters:
   - **`primary-server`** - defines the IP address of the NTP server
   - **`utc-offset`** - defines the time offset in relation to the UTC. For example, if your region is 2 hours ahead of the UTC, enter "2".
   - **`update-interval`** - defines the period after which the date and time of the device is updated
2. Configure daylight saving, if required:
   - **`summer-time`** - enables daylight saving time
   - **`start and end`** - defines the period for which daylight saving time is relevant.
   - **`offset`** - defines the offset in minutes to add to the time for daylight saving. For example, if your region has daylight saving of one hour, the time received from the NTP server is 11:00, and the UTC offset for your region is +2 (i.e., 13:00), you need to enter "60" to change the local time to 14:00.

Note: For more information on these parameters, refer to the *VoIP CLI Reference Guide*.

Part IV

VoIP Configuration

This part describes VoIP configuration.

Reader's Notes

11 Network

This section describes the network-related configuration.

11.1 Configuring IP Interface Settings

The Multiple Interface Table page allows you to configure logical VoIP network interfaces. Each interface can be defined with the following:

- Application type allowed on the interface:
  - Control - call control signaling traffic (i.e., SIP)
  - Media - RTP traffic
  - Operations, Administration, Maintenance and Provisioning (OAMP) - management (such as Web- and SNMP-based management)
- IP address and subnet
- VLAN ID
- Default Gateway
- Primary and secondary DNS IP address

You can configure up to 12 interfaces - up to 11 Control and/or Media interfaces, and 1 OAMP interface.

The default VoIP interface is as follows:

- Application type: OAMP + Media + Control
- IP address: 192.168.0.2 with prefix length 24 (i.e., subnet mask 255.255.255.0)
- Default gateway: 192.168.0.1
- Name: "Voice"
- VLAN ID: 1

When using data-routing functionality, the network interfaces for the data-router are configured using the Data Settings menu (see the Part on Data Router Configuration).

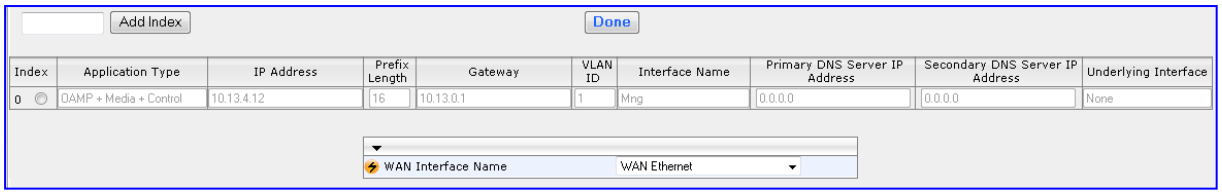
Notes:

- When operating with both voice and data-routing functionalities, it is recommended to define the default gateway IP address for the VoIP network interfaces in the same subnet and with the same VLAN ID as the IP address defined in the data-routing configuration section.
- For more information and examples of VoIP network interfaces configuration, see 'Network Configuration' on page 120.
- You can define firewall rules (access list) to deny (block) or permit (allow) packets received from a specific IP interface configured in this table. These rules are configured using the AccessList parameter (see 'Configuring the Access List' on page 591).
- You can view currently active configured IP interfaces in the 'IP Active Interfaces page (see 'Viewing Active IP Interfaces' on page 719).
- You can also configure this table using the *ini* file table parameter InterfaceTable (see 'Networking Parameters' on page 755).
- For configuring Web interface tables, see 'Working with Tables' on page 62.

➢ **To configure VoIP network interfaces:**

1. Open the Multiple Interface Table page (**Configuration** tab > **VoIP** menu > **Network** submenu > **IP Settings**).

Figure 11-1: Multiple Interface Table Page



2. In the 'Add Index' field, enter the desired index number for the new interface, and then click **Add Index**; the index row is added to the table.
3. Configure the interface according to the table below.
4. Click the **Apply** button; the interface is added to the table and the **Done** button appears.
5. Click **Done** to validate the interface. If the interface is not valid (e.g., if it overlaps with another interface in the table or if it does not adhere to the other rules as summarized in 'Multiple Interface Table Configuration Summary and Guidelines' on page 126), a warning message is displayed.
6. Save the changes to flash memory and reset the device (see 'Saving Configuration' on page 680).

To view network interfaces that are currently active, click the **IP Interface Status Table** button. For a description of this display, see 'Viewing Active IP Interfaces' on page 719.

Table 11-1: Multiple Interface Table Parameters Description

Parameter	Description
Table parameters	
Index	Table index row of the interface. The range is 0 to 11.
Web: Application Type EMS: Application Types **[InterfaceTable_ApplicationTypes]**	Types of applications that are allowed on the specific interface. ▪ **[0]** OAMP = Only Operations, Administration, Maintenance and Provisioning (OAMP) applications (e.g., Web, Telnet, SSH, and SNMP) are allowed on the interface. ▪ **[1]** Media = Only Media (i.e., RTP streams of voice) is allowed on the interface. ▪ **[2]** Control = Only Call Control applications (e.g., SIP) are allowed on the interface. ▪ **[3]** OAMP + Media = Only OAMP and Media applications are allowed on the interface. ▪ **[4]** OAMP + Control = Only OAMP and Call Control applications are allowed on the interface. ▪ **[5]** Media + Control = Only Media and Call Control applications are allowed on the interface. ▪ **[6]** OAMP + Media + Control = All application types are allowed on the interface.

Parameter	Description
	Notes: ▪ A single OAMP interface (and only one) must be configured. This OAMP interface can be combined with Media and Control interfaces. ▪ At least one interface for Media traffic and at least one interface for Control traffic must be configured. These interfaces can be combined (i.e., Media + Control, or OAMP + Media + Control). ▪ At least one IPv4 interface must be configured.
Web/EMS: IP Address **[InterfaceTable_IPAddres]**	The IPv4 IP address in dotted-decimal notation.
Web/EMS: Prefix Length **[InterfaceTable_PrefixLength]**	Defines the Classless Inter-Domain Routing (CIDR)-style representation of a dotted decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted decimal format (e.g. 192.168.0.0/16 is synonymous with 192.168.0.0 and a subnet of 255.255.0.0. Defines the number of '1' bits in the subnet mask (i.e., replaces the standard dotted-decimal representation of the subnet mask for IPv4 interfaces). For example: A subnet mask of 255.0.0.0 is represented by a prefix length of 8 (i.e., 11111111 00000000 00000000 00000000), and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (i.e., 11111111 11111111 11111111 11111100). The prefix length is a Classless Inter-Domain Routing (CIDR) style presentation of a dotted-decimal subnet notation. The CIDR-style presentation is the latest method for interpretation of IP addresses. Specifically, instead of using eight-bit address blocks, it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes (refer to http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing for more information). For IPv4 Interfaces, the prefix length values range from 0 to 31. **Note:** Subnets of different interfaces must not overlap in any way (e.g., defining two interfaces with 10.0.0.1/8 and 10.50.10.1/24 is invalid). Each interface must have its own address space.
Web/EMS: Gateway **[InterfaceTable_Gateway]**	Defines the IP address of the default gateway for this interface. **Notes:** ▪ A default gateway can be defined for each interface. ▪ The default gateway's IP address must be in the same subnet as the interface address.
Web/EMS: VLAN ID **[InterfaceTable_VlanID]**	Defines the VLAN ID for each interface. **Note:** The VLAN ID must be unique for each interface.
Web/EMS: Interface Name **[InterfaceTable_InterfaceName]**	Defines a string (up to 16 characters) to name this interface. This name is displayed in management interfaces (Web, CLI and SNMP) for clarity (and has no functional use), as well as in the Media Realm table and SIP Interface table.

Parameter	Description
	Notes: ▪ This parameter is mandatory. ▪ The name must be unique for each interface.
WAN Interface Name **[WanInterfaceName]**	Assigns a WAN interface to the VoIP traffic (i.e., SIP signaling and media / RTP interfaces). The available WAN interface options depends on the hardware configuration (e.g., Ethernet, T1, or SHDSL) and/or whether VLANs are defined for the WAN interface (see Virtual LAN Interface (VLAN) on page 646. If VLANs are configured, for example, for the Ethernet WAN interface (see Data Settings on page 565), then you can select the WAN VLAN on which you want to run these SIP signaling and/or media interfaces. The WAN interface can be assigned to SIP signaling and media interfaces in the SIP Interface table (see Configuring SIP Interface Table on page 210) and Media Realm table (see Configuring Media Realms on page 181), where the WAN interface is denoted as "WAN". Once this association is set, VoIP traffic is sent via the WAN and incoming traffic is identified as coming from the WAN. The device also automatically configures the required port forwarding and static NAT rules. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If you do not assign the WAN interface to SIP and media interfaces, then the WAN interface may not be used for VoIP traffic. In such scenarios, the VoIP traffic can be sent and received within the LAN, or sent to the WAN via a third-party LAN router. If a third-party router is used as the interface to the WAN, then you need to define NAT rules (using the NATTranslation parameter) to translate the VoIP LAN IP addresses (defined in the Multiple Interface table and associated with SIP and media interfaces) into global, public IP addresses. ▪ This parameter is applicable only if the data-routing functionality is supported (i.e., relevant Software Upgrade Feature Key is installed on the device).
Web/EMS: Primary DNS Server IP address **[InterfaceTable_PrimaryDNSServerIPAddress]**	Defines the IP address (in dotted-decimal notation) of the primary DNS server that is used for translating domain names into IP addresses for each interface. **Note:** This parameter is optional.
Web/EMS: Secondary DNS Server IP address **[InterfaceTable_SecondaryDNSServerIPAddress]**	Defines the IP address (in dotted-decimal notation) of the secondary DNS server that is used for translating domain names into IP addresses for each interface. **Note:** This parameter is optional.

11.1.1 Network Configuration Overview

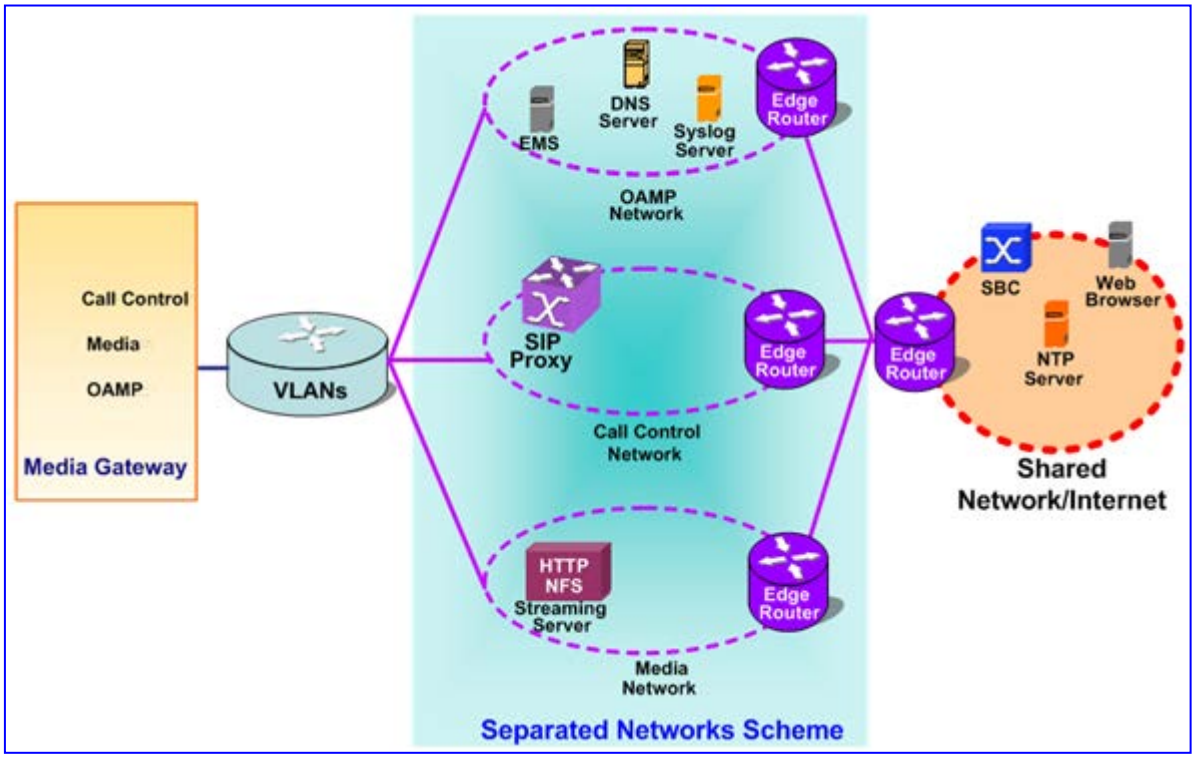
The device allows you to configure multiple IP addresses with associated VLANs for the VoIP network, using the Multiple Interface table. Complementing this table is the Routing table, which allows you to define VoIP network static routing rules for non-local hosts/subnets. This section describes the various network configuration options offered by the device.

This section covers the VoIP network configuration (interfaces, static routing rules, and QoS definitions). For configuring the data-routing network (interfaces, routing, firewall, NAT, etc.), see Data Settings on page 565.

11.1.1.1 Multiple Network Interfaces and VLANs

A need often arises to have logically separated network segments for various applications (for administrative and security reasons). This can be achieved by employing Layer-2 VLANs and Layer-3 subnets.

Figure 11-2: Multiple Network Interfaces



The figure depicts a typical configuration featuring in which the device is configured with three network interfaces for:

- Operations, Administration, Maintenance, and Provisioning (OAMP) applications
- Call Control applications
- Media

The Multiple Interfaces scheme allows the configuration of different IP addresses, each associated with a unique VLAN ID. The configuration is performed using the Multiple Interface table, which is configurable using the *ini* file, Web, and SNMP interfaces.

11.1.1.1.1 Overview of Multiple Interface Table

The Multiple Interfaces scheme allows you to define different IP addresses and VLANs in a table format, as shown below:

Table 11-2: Multiple Interface Table

Index Mode	Application	Interface	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name
0	OAMP	IPv4	10.31.174.50	16	0.0.0.0	4	ManagementIF

Index Mode	Application	Interface	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name
1	Control	IPv4	10.32.174.50	16	0.0.0.0	5	ControlIF
2	Media	IPv4	10.33.174.50	16	10.33.0.1	6	Media1IF
3	Media	IPv4	10.34.174.50	16	0.0.0.0	7	Media2IF
4	Media	IPv4	10.35.174.50	16	10.35.0.1	8	Media3IF
5	Media	IPv4	10.36.174.50	16	0.0.0.0	9	Media4IF
6	Media	IPv4	10.37.174.50	16	0.0.0.0	10	Media5IF
7	Media	IPv4	10.38.174.50	16	0.0.0.0	11	Media6IF
8	Media	IPv4	10.39.174.50	16	10.39.0.1	12	Media7IF
9	Media	IPv4	10.40.174.50	16	10.40.0.1	13	Media8IF
10	Media & Control	IPv4	10.41.174.50	16	0.0.0.0	14	MediaCtrl9IF
11	Media	IPv4	10.42.174.50	16	0.0.0.0	15	Media10IF
12	Media	IPv4	10.43.174.50	16	10.43.0.1	16	Media11IF
13	Media	IPv4	10.44.174.50	16	0.0.0.0	17	Media12IF
14	Media & Control	IPv4	10.45.174.50	16	10.45.0.1	18	Media13IF

Complementing this network configuration is the On-Board Ethernet Switch configuration. This allows configuring the VLAN IDs accessible through each physical port, as well as the Native VLAN ID of each physical port. The On-Board Ethernet Switch configuration is described in LAN Switch on page 639. In addition, Layer3 (DiffServ) and Layer 2 (VLAN priority) Quality of Service parameters are also configurable.

Each row of the table defines a logical IP interface with its own IP address, subnet mask (represented by Prefix Length), VLAN ID, name, and application types that are allowed on this interface. Multiple interfaces can be defined with a default gateway. Traffic from this interface destined to a subnet which does not meet any of the routing rules (either local or static routes) are forwarded to this gateway.

Index Column:

This column holds the index of each interface. Possible values are 0 to 11. Each interface index must be unique.

Application Types Column:

This column defines the types of applications that are allowed on this interface:

- OAMP – Operations, Administration, Maintenance and Provisioning applications such as Web, Telnet, SSH, SNMP
- CONTROL – Call Control protocols (i.e., SIP)
- MEDIA – RTP streams of voice
- Various combinations of the above mentioned types

The following table shows the possible values of this column and their descriptions:

Table 11-3: Application Types

Value	Description
0	OAMP: only OAMP applications are allowed on this interface.

Value	Description
1	MEDIA: only Media (RTP) are allowed on this interface.
2	CONTROL: only Call Control applications are allowed on this interface.
3	OAMP & MEDIA: only OAMP and Media (RTP) applications are allowed on this interface.
4	OAMP & CONTROL: only OAMP and Call Control applications are allowed on this interface.
5	MEDIA & CONTROL: only Media (RTP) and Call Control applications are allowed on this interface.
6	OAMP, MEDIA & CONTROL: all of the application types are allowed on this interface.

For valid configuration guidelines, see 'Multiple Interface Table Configuration Summary and Guidelines' on page 126.

Interface Mode Column:

The Interface Mode column determines the method that this interface uses to acquire its IP address. For IPv4 Manual IP Address assignment, use "IPv4 Manual" (10).

IP Address and Prefix Length Columns:

These columns allow the user to configure an IPv4 IP address and its related subnet mask.

The Prefix Length column holds the Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted-decimal format, in other words, 192.168.0.0/16 is synonymous with 192.168.0.0 and a subnet 255.255.0.0 (Refer to http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing for more information).

This CIDR notation lists the number of '1' bits in the subnet mask. So, a subnet mask of 255.0.0.0 (when broken down to its binary format) is represented by a prefix length of 8 (11111111 00000000 00000000 00000000), and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (11111111 11111111 11111111 11111100).

Each interface must have its own address space. Two interfaces may not share the same address space, or even part of it. The IP address should be configured as a dotted-decimal notation.

For IPv4 interfaces, the prefix length values range from 0 to 30.

Gateway Column:

This column defines a default gateway for each interface. A default gateway can be defined for each interface. When traffic is sent from this interface to an unknown destination (i.e., not in the same subnet and not defined for any static routing rule), it is forwarded to this default gateway. The default gateway's address must be on the same subnet as the interface address. A separate routing table allows configuring additional static routing rules. See 'Configuring the IP Routing Table' on page 132 for more details.

Note: In the example below, the default gateway for the OAMP application is 192.168.0.1, whereas for Media & Control applications it is 200.200.85.1.

Table 11-4: Configured Default Gateway Example

Index	Application Type	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	IPv4 Manual	192.168.0.2	16	192.168.0.1	100	Mgmt
1	Media & Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	200	CntrlMedia

A separate routing table allows configuring static routing rules. Configuring the following routing enable OAMP applications to access peers on subnet 17.17.0.0 through the gateway 192.168.10.1 (which is not the default gateway of the interface), and Media & Conrol applications to access peers on subnet 171.79.39.0 through the gateway 200.200.85.10 (which is not the default gateway of the interface).

Table 11-5: Separate Routing Table Example

Destination	Prefix Length	Gateway	Interface	Metric	Status
17.17.0.0	16	192.168.10.1	0	1	Active
171.79.39.0	24	200.200.85.10	1	1	Active

VLAN ID Column:

This column defines the VLAN ID for each interface. This column must hold a unique value for each interface of the same address family.

Interface Name Column:

This column allows the configuration of a short string (up to 16 characters) to name this interface. This name is displayed in management interfaces (Web, CLI, and SNMP) and is used in the Media Realm table. This column must have a unique value for each interface (no two interfaces can have the same name) and must not be left blank.

Primary / Secondary DNS Server IP Address Columns:

Defines the primary and secondary DNS server IP addresses for translating domain names into IP addresses.

11.1.1.1.2 Booting using DHCP

The *DHCPEnable* parameter enables the device to boot while acquiring an IP address from a DHCP server. Note that when using this method, Multiple Interface table/VLANs and other advanced configuration options are disabled.

11.1.1.1.3 Quality of Service Parameters

The device allows you to specify values for Layer-3 priorities, by assigning values to the following service classes:

- Premium Media service class – used for RTP Media traffic
- Premium Control Service class – used for Call Control traffic
- Gold Service class – used for streaming applications
- Bronze Service class – used for OAMP applications

The Layer-3 QoS parameters defines the values of the DiffServ field in the IP Header of the frames related to a specific service class. The Layer-2 QoS parameters defines the values

for the 3 priority bits in the VLAN tag (according to the IEEE 802.1p standard) according to the value of the DiffServ field found in the packet IP header.

The DiffServ Table (DiffServToVlanPriority) allows you to configure DiffServ-to-VLAN Priority mapping (Layer 2 class of service). For each packet sent to the LAN, the VLAN Priority of the packet is set according to the DiffServ value in the IP header of the packet. For Layer-3 CoS, you can use the PremiumServiceClassMediaDiffServ, PremiumServiceClassControlDiffServ, GoldServiceClassDiffServ, and BronzeServiceClassDiffServ parameters.

The mapping of an application to its CoS and traffic type is shown in the table below:

Table 11-6: Traffic/Network Types and Priority

Application	Traffic / Network Types	Class-of-Service (Priority)
Debugging interface	Management	Bronze
Telnet	Management	Bronze
DHCP	Management	Network
Web server (HTTP)	Management	Bronze
SNMP GET/SET	Management	Bronze
Web server (HTTPS)	Management	Bronze
RTP traffic	Media	Premium media
RTCP traffic	Media	Premium media
T.38 traffic	Media	Premium media
SIP	Control	Premium control
SIP over TLS (SIPS)	Control	Premium control
Syslog	Management	Bronze
SNMP Traps	Management	Bronze
DNS client	Varies according to DNS settings: ▪ OAMP ▪ Control	Depends on traffic type: ▪ Control: Premium Control ▪ Management: Bronze
NTP	Varies according to NTP settings (EnableNTPasOAM): ▪ OAMP ▪ Control	Depends on traffic type: ▪ Control: Premium control ▪ Management: Bronze

11.1.1.1.4 Assigning NTP Services to Application Types

NTP applications can be associated with different application types (OAMP or Control) in different setups. The table below describes the parameter for configuring this:

Table 11-7: Application Type Parameters

Parameter	Description
EnableNTPasOAM	Determines the application type for NTP services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control.

Parameter	Description
	Note: For this parameter to take effect, a device reset is required.

11.1.1.1.5 Multiple Interface Table Configuration Summary and Guidelines

Multiple Interface table configuration must adhere to the following rules:

- Up to 12 different interfaces may be defined.
- The indices used must be in the range between 0 and 11.
- Each interface must have its own subnet. Defining two interfaces with addresses in the same subnet (i.e. two interfaces with 192.168.0.1/16 and 192.168.100.1/16) is illegal.
- Subnets in different interfaces must not be overlapping in any way (i.e. defining two interfaces with 10.0.0.1/8 and 10.50.10.1/24 is invalid). Each interface must have its own address space.
- The Prefix Length replaces the dotted decimal Subnet Mask presentation. This column must have a value of 0-30 for IPv4 interfaces.
- Only one IPv4 interface with OAMP "Application Types" **must** be configured. At least one IPv4 interface with CONTROL "Application Types" **must** be configured. At least one IPv4 interface with MEDIA "Application Types" **must** be configured. These application types **may** be mixed (i.e. OAMP and CONTROL). Here are some examples for interface configuration:
  - One IPv4 interface with "Application Types" OAMP, MEDIA & CONTROL (without VLANs).
  - One IPv4 interface with "Application Types" OAMP, one other or more IPv4 interfaces with "Application Types" CONTROL, and one or more IPv4 interfaces with "Application Types" MEDIA (with VLANs).
  - One IPv4 interface with "Application Types" OAMP & MEDIA, one other or more IPv4 interfaces with "Application Types" MEDIA & CONTROL.
  - Other configurations are also possible while keeping to the above-mentioned rule.
- Each network interface may be defined with a default gateway. This default gateway address must be in the same subnet as the associated interface. Additional routing rules may be specified in the Routing table ('Configuring the IP Routing Table' on page 132).
- The Interface Name column may have up to 16 characters. This column allows the user to name each interface with an easier name to associate the interface with. This column must have a unique value to each interface and must not be left blank.
- Primary and Secondary DNS server address may be configured for each interface. **Note:** Currently, the device supports DNS configuration for only one interface.
- For IPv4 interfaces, the "Interface Mode" column must be set to "IPv4 Manual" (numeric value 10).
- Quality of Service parameters specify the DiffServ field in the IP header according to service classes. DiffServ-to-VLAN Priority mapping allows associating each DiffServ value with a VLAN priority (according to IEEE 802.1p standard).
- Network Configuration changes are offline. The new configuration should be saved and becomes available at the next startup.

Upon system start up, the Multiple Interface table is parsed and passes comprehensive validation tests. If any errors occur during this validation phase, the device sends an error message to the Syslog server and falls back to a "safe mode", using a single interface and no VLANs. Ensure that you view the Syslog messages that the device sends in system startup to see if any errors occurred.

Note: When configuring the device using the Web interface, it is possible to perform a quick validation of the configured Multiple Interface table and VLAN definitions, by clicking the **Done** button in the Multiple Interface Table Web page. It is highly recommended to perform this when configuring Multiple Interfaces and VLANs, using the Web Interface to ensure the configuration is complete and valid.

11.1.1.1.6 Troubleshooting the Multiple Interface Table

If any of the Multiple Interface table guidelines are violated, the device falls back to a "safe mode" configuration, working temporarily with IP address 192.168.0.2. For more information on validation failures, consult the Syslog messages.

Validation failures may be caused by one of the following:

- One of the Application Types (OAMP, CONTROL, MEDIA) is missing in the IPv4 interfaces.
- There are too many interfaces with "Application Types" of OAMP. Only one interface defined but the "Application Types" column is not set to "OAM + Media + Control" (numeric value 6).
- An IPv4 interface was defined with "Interface Type" different than "IPv4 Manual" (10).
- Two interfaces have the exact VLAN ID value.
- Two interfaces have the same name.
- Two interfaces share the same address space or subnet.

Apart from these validation errors, connectivity problems may be caused by one of the following:

- Trying to access the device with untagged traffic when VLANs are on and Native VLAN is not configured properly.
- Routing Table is not configured properly.

11.1.1.2 Setting Up VoIP Networking

11.1.1.2.1 Using the ini File

When configuring the network configuration using the *ini* File, use a textual presentation of the Interface and Routing Tables, as well as some other parameters. The following shows an example of a full network configuration, consisting of **all** the parameters described in this section:

```
;  *** TABLE InterfaceTable ***
[ InterfaceTable ]
FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress,
InterfaceTable_UnderlyingInterface;
InterfaceTable 0 = 0, 10, 192.168.0.2, 16, 192.168.0.1, 501, Mgmt, , , ;
InterfaceTable 1 = 5, 10, 10.32.174.50, 24, 10.32.174.1, 2012,
MediaCntrl, , , ;
InterfaceTable 2 = 1, 10, 10.33.174.50, 24, 10.33.174.1, 2013, Media1, ,
```

```
, ;
InterfaceTable 3 = 2, 10, 10.34.174.50, 24, 10.34.174.1, 2014, Cntrl1, ,
, ;
[ \InterfaceTable ]

; Routing Table Configuration:
[ StaticRouteTable ]
FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName,
StaticRouteTable_Destination, StaticRouteTable_PrefixLength,
StaticRouteTable_Gateway, StaticRouteTable_Description;
StaticRouteTable 0 = 0, 10.31.174.0, 24, 192.168.11.1, ;
StaticRouteTable 1 = 1, 174.96.151.15, 24, 10.32.174.12, ;
StaticRouteTable 2 = 3, 10.35.174.0, 24, 10.34.174.240, ;
[ \StaticRouteTable ]

; Layer 3 QoS parameters (DiffServ):
PremiumServiceClassMediaDiffServ = 46
PremiumServiceClassControlDiffServ = 40
GoldServiceClassDiffServ = 26
BronzeServiceClassDiffServ = 10

;  *** TABLE DiffServToVlanPriority ***
[ DiffServToVlanPriority ]
FORMAT DiffServToVlanPriority_Index = DiffServToVlanPriority_DiffServ,
DiffServToVlanPriority_VlanPriority;
DiffServToVlanPriority 0 = 0, 7;
DiffServToVlanPriority 1 = 46, 6;
DiffServToVlanPriority 2 = 40, 6;
DiffServToVlanPriority 3 = 26, 4;
DiffServToVlanPriority 4 = 10, 2;
[ \DiffServToVlanPriority ]

; Application Type for NTP applications:
EnableNTPasOAM = 1
```

This *ini* file shows the following:

- A Multiple Interface table with a an interface for OAMP applications (192.168.0.2/16), an interface for Media & Control applications (10.32.174.50/24), an interface for Media applications (10.33.174.50/24), and an interface for Control applications (10.34.174.50/24).
- Each interface is defined with its own VLAN ID, Default Gateway, and name.
- A Routing table is configured with three static routing rules:
  - Directing all traffic destined to subnet 10.31.174.0/24 to 192.168.11.1 over Interface Index 0
  - Directing all traffic destined to subnet 174.96.151.15/24 to 10.32.174.12 over Interface Index 1
  - Directing all traffic destined to subnet 10.35.174.0/24 to 10.34.174.240 over Interface Index 3
- Layer-3 QoS values are assigned:
  - For Media Service class, the default DiffServ value is set to 46
  - For Control Service class, the default DiffServ value is set to 40
  - For Gold Service class, the default DiffServ value is set to 26
  - For Bronze Service class, the default DiffServ value is set to 10

- Layer-2 QoS values are assigned:
  - For packets sent with DiffServ value of 46, set VLAN priority to 6
  - For packets sent with DiffServ value of 40, set VLAN priority to 6
  - For packets sent with DiffServ value of 26, set VLAN priority to 4
  - For packets sent with DiffServ value of 10, set VLAN priority to 2
- The NTP applications are configured to serve as OAMP applications.

Notes:

- Lines that begin with a semicolon are considered a remark and are ignored.
- When using the *ini* file, the Multiple Interface table and the DiffServ To VLAN Priority table must have the prefix and suffix to allow the INI File parser to correctly recognize and parse the table.

11.1.1.2.2 Networking Configuration Examples

This section provides examples of network configurations (and their corresponding *ini* file configuration).

Example 1 - One VoIP Interface for All Applications: Multiple Interface table with a single interface for OAMP, Media and Control applications:

Table 11-8: Multiple Interface Table - Example 1

Index	Allowed Applications	Interface Mode	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name	Primary/Secondary DNS Server IP Address	Underlying Interface
0	OAMP, Media & Control	IPv4	192.168.0.2	16	192.168.0.1	1	myInterface	-	-

Additional static routing rules:

Table 11-9: Routing Table - Example 1

Destination	Prefix Length	Gateway	Interface	Metric
201.201.0.0	16	192.168.11.10	0	1
202.202.0.0	16	192.168.11.1	0	1

The NTP applications remain with their default application types.

The corresponding *ini* file configuration is shown below:

```
; Interface Table Configuration:
[InterfaceTable]
FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingInterface;
```

```
InterfaceTable 0  = 6, 10, 192.168.0.2, 16, 192.168.0.1, 1, myInterface,
, , ;
[\InterfaceTable]

; Routing Table Configuration:
[ StaticRouteTable ]
FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName,
StaticRouteTable_Destination, StaticRouteTable_PrefixLength,
StaticRouteTable_Gateway, StaticRouteTable_Description;
StaticRouteTable 0 = 0, 201.201.0.0, 16, 192.168.11.10, ;
StaticRouteTable 1 = 0, 202.202.0.0, 16, 192.168.11.1, ;
[ \StaticRouteTable ]
```

Example 2 - Three VoIP Interfaces, One for each Application Exclusively: the Multiple Interface table is configured with three interfaces, one exclusively for each application type: one interface for OAMP applications, one for Call Control applications, and one for RTP Media applications:

Table 11-10: Multiple Interface Table - Example 2

Index	Allowed Applications	Interface Mode	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name	Primary/Secondary DNS Server IP Address	Underlying Interface
0	OAMP	IPv4 Manual	192.168.0.2	16	192.168.0.1	1	ManagementIF	-	-
1	Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	200	myControlIF	-	-
2	Media	IPv4 Manual	211.211.85.14	24	211.211.85.1	211	myMediaIF	-	-

Additional static routing rules:

Table 11-11: Routing Table - Example2

Destination	Prefix Length	Gateway	Interface	Metric
176.85.49.0	24	192.168.11.1	0	1

All other parameters are set to their respective default values. The NTP application remains with its default application types.

The corresponding *ini* file configuration is shown below:

```
; Interface Table Configuration:
[InterfaceTable]
FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingInterface;
InterfaceTable 0  = 0, 10, 192.168.0.2,  16, 192.168.0.1, 1, ManagementIF, ,
```

```
,;
InterfaceTable 1  = 2, 10, 200.200.85.14,  24, 200.200.85.1, 200,
myControlIF, , ,;
InterfaceTable 2  = 1, 10, 211.211.85.14,  24, 211.211.85.1, 211,
myMediaIF, , ,;
[\InterfaceTable]
; Routing Table Configuration:
[ StaticRouteTable ]
FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName,
StaticRouteTable_Destination, StaticRouteTable_PrefixLength,
StaticRouteTable_Gateway, StaticRouteTable_Description;
StaticRouteTable 0 = 0, 176.85.49.0, 24, 192.168.11.1, ;
[ \StaticRouteTable ]
```

Example 3 - Three Interfaces: one exclusively for management (OAMP applications) and two others for Call Control and RTP (Control and Media applications) :

Table 11-12: Multiple Interface Table - Example 3

Index	Allowed Applications	Interface Mode	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name	Primary/Secondary DNS Server IP Address	Underlying Interface
0	OAMP	IPv4 Manual	192.168.0.2	16	192.168.0.1	1	Mgmt	-	-
1	Media & Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	201	MediaCntrl1	-	-
2	Media & Control	IPv4 Manual	200.200.86.14	24	200.200.86.1	202	MediaCntrl2	-	-

Additional static routing rules

Table 11-13: Routing Table - Example 3

Destination	Destination Subnet Mask/Prefix Length	Gateway	Interface	Metric
176.85.49.0	24	192.168.0.10	0	1

All other parameters are set to their respective default values. The NTP application remains with its default application types.

The corresponding *ini* file configuration is shown below:

```
; Interface Table Configuration:
[InterfaceTable]
FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingInterface;
```

```
InterfaceTable 0  = 0, 10, 192.168.0.2,  16, 192.168.0.1, 1, Mgmt,,,;
InterfaceTable 1  = 5, 10, 200.200.85.14,  24, 200.200.85.1, 201,
MediaCntrl1,,,;
InterfaceTable 2  = 5, 10, 200.200.86.14,  24, 200.200.86.1, 202,
MediaCntrl2,,,;

[\InterfaceTable]
; Routing Table Configuration:
[ StaticRouteTable ]
FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName,
StaticRouteTable_Destination, StaticRouteTable_PrefixLength,
StaticRouteTable_Gateway, StaticRouteTable_Description;
StaticRouteTable 0 = 0, 176.85.49.0, 24, 192.168.11.1, ;
[ \StaticRouteTable ]
```

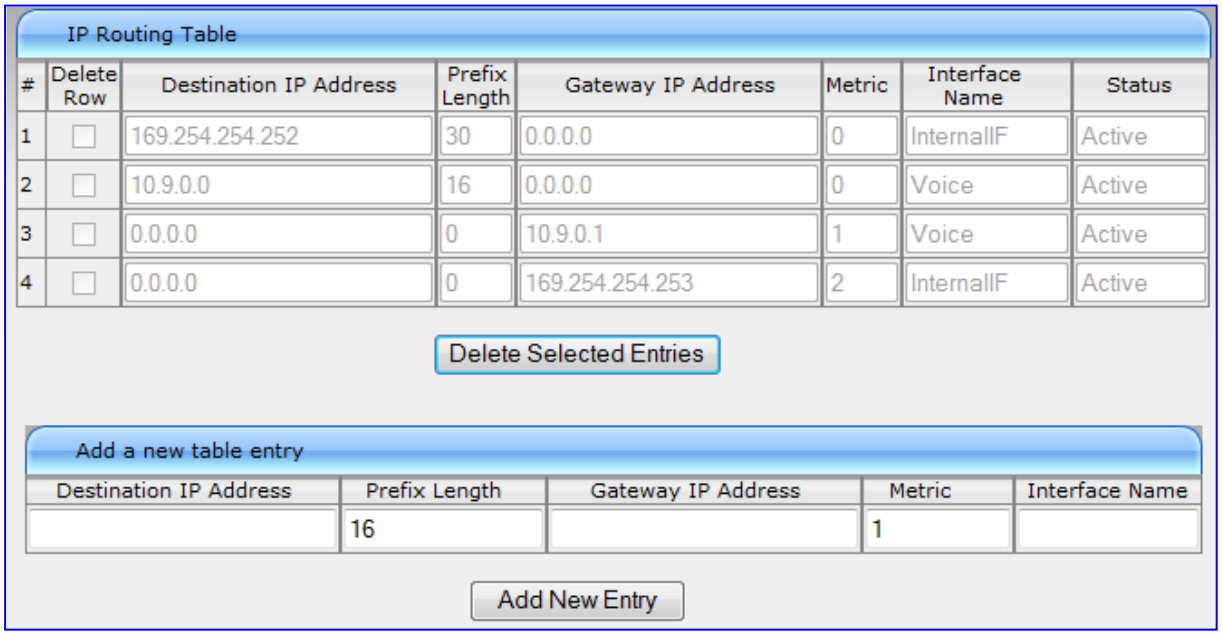
11.2 Configuring the IP Routing Table

The IP Routing Table page allows you to define up to 30 static IP routing rules for the device. These rules can be associated with a network interface (defined in the Multiple Interface table) and therefore, the routing decision is based on the source subnet/VLAN. If not associated with an IP interface, the static IP rule is based on destination IP address.

➢ **To configure static IP routing:**

1. Open the IP Routing Table page (**Configuration** tab > **VoIP** menu > **Network** submenu > **IP Routing Table**).

Figure 11-3: IP Routing Table Page



2. In the Add a new table entry table, add a new static routing rule according to the parameters described in the table below.
3. Click **Add New Entry**; the new routing rule is added to the IP routing table.

To delete a routing rule from the table, select the 'Delete Row' check box corresponding to the required routing rule, and then click **Delete Selected Entries**.

Notes:

- You can delete only inactive routing rules.
- You can also configure the IP Routing table using the *ini* file table parameter StaticRouteTable.

Table 11-14: IP Routing Table Description

Parameter	Description
Destination IP Address **[StaticRouteTable_Destination]**	Specifies the IP address of the destination host/network.
Prefix Length **[StaticRouteTable_PrefixLength]**	Specifies the subnet mask of the destination host/network.
The address of the host/network you want to reach is determined by an AND operation that is applied to the fields 'Destination IP Address' and 'Destination Mask'. For example, to reach the network 10.8.x.x, enter 10.8.0.0 in the field 'Destination IP Address' and 255.255.0.0 in the field 'Destination Mask'. As a result of the AND operation, the value of the last two octets in the field 'Destination IP Address' is ignored. To reach a specific host, enter its IP address in the field 'Destination IP Address' and 255.255.255.255 in the field 'Destination Mask'.	
Gateway IP Address **[StaticRouteTable_Gateway]**	The IP address of the router (next hop) to which the packets are sent if their destination matches the rules in the adjacent columns. **Note:** The Gateway address must be in the same subnet as the IP address of the interface over which you configure this static routing rule.
Metric	The number of hops needed to get to the specified destination. **Note:** The recommended value for this parameter is 1.
Interface Name **[StaticRouteTable_InterfaceName]**	Assigns a network interface to the routing rule. This value is the index of the network interface as defined in the Multiple Interface table (see 'Configuring IP Interface Settings' on page 117). **Note:** The IP address of the 'Gateway IP Address' field must be in the same subnet as this interface's IP address.
Status	Read-only field displaying the status of the static IP route: ▪ "Active" - routing rule is used ny the device ▪ "Inactive" - routing rule is not applied

11.2.1 Routing Table Columns

Each row of the Routing table defines a static routing rule. Traffic destined to the subnet specified in the routing rule is re-directed to the defined gateway, reachable through the specified interface.

The IP Routing table consists of the following:

Table 11-15: IP Routing Table Layout

Destination	Prefix Length	Gateway	Interface	Metric	Status
201.201.0.0	16	192.168.0.1	0	1	Active
202.202.0.0	16	192.168.0.2	0	1	Active
203.203.0.0	16	192.168.0.3	0	1	Active
225.225.0.0	16	192.168.0.25	0	1	Inactive

11.2.1.1 Destination Column

This column defines the destination of the route rule. The destination can be a single host or a whole subnet, depending on the Prefix Length/Subnet Mask specified for this routing rule.

11.2.1.2 Prefix Length Column

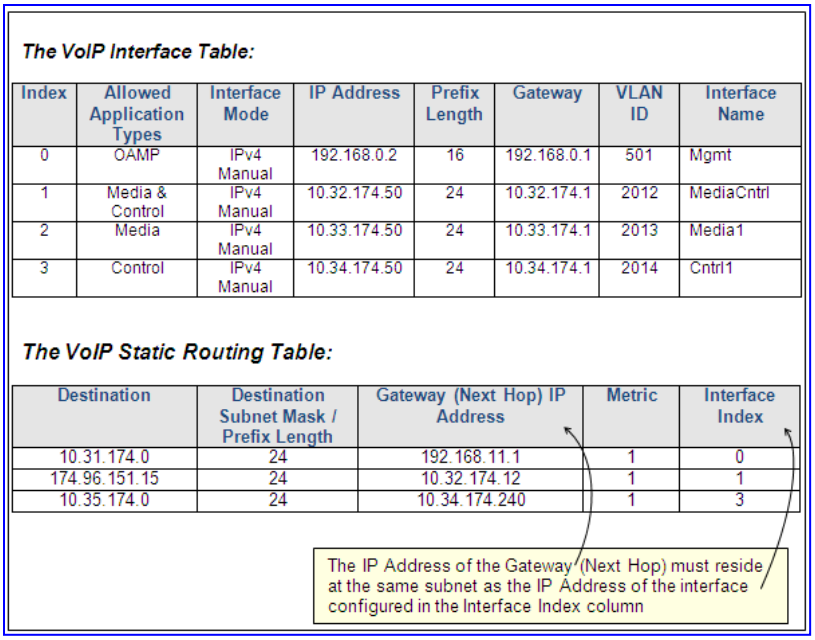
The Prefix Length column holds the Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits that are set in the dotted-decimal format. For example, 16 is synonymous with subnet 255.255.0.0.

11.2.1.3 Gateway Column

The Gateway column defines the IP address of the next hop used for traffic destined to the subnet/host as defined in the destination/mask columns. This gateway address must be on the same subnet as the IP address of the interface configured in the Interface column.

11.2.1.4 Interface Column

This column defines the interface index (in the Multiple Interface table) from which the gateway address is reached.



11.2.1.5 Metric Column

The Metric column must be set to 1 for each static routing rule.

11.2.1.6 State Column

The State column displays the state of each static route. Possible values are "Active" and "Inactive". When the destination IP address is not on the same segment with the next hop or the interface does not exist, the route state changes to "Inactive".

11.2.2 Routing Table Configuration Summary and Guidelines

The Routing table configurations must adhere to the following rules:

- Up to 30 different static routing rules may be defined.
- The Prefix Length replaces the dotted-decimal subnet mask presentation. This column must have a value of 0-31 for IPv4 interfaces.
- The "Gateway" IP Address must be on the same subnet as the IP address of the interfaces configured in the Interface Index column.
- The “Metric” column must be set to 1.
- Network Configuration changes are offline. The new configuration should be saved and will be available at the next startup.

11.2.3 Troubleshooting the Routing Table

When adding a new static routing rule, the added rule passes a validation test. If errors are found, the routing rule is rejected and is not added to the IP Routing table. Failed routing validations may result in limited connectivity (or no connectivity) to the destinations specified in the incorrect routing rule. For any error found in the Routing table or failure to configure a routing rule, the device sends a notification message to the Syslog server reporting the problem.

Common routing rule configuration errors may include the following:

- The IP address specified in the "Gateway" column is unreachable from the interface specified in the "Interface" column.
- The same destination is defined in two different routing rules.
- More than 30 routing rules were defined.

Note: If a routing rule is required to access OAMP applications (for remote management, for instance) and this route is not configured correctly, the route is not added and the device is not accessible remotely. To restore connectivity, the device must be accessed locally from the OAMP subnet and the required routes be configured.

11.3 Configuring QoS Settings

The Diff Serv Table page is used for configuring the Layer-2 and Layer-3 Quality of Service (QoS) parameters for VoIP. DiffServ is an architecture providing different types or levels of service for IP traffic. DiffServ (according to RFC 2474), prioritizes certain traffic types based on their priority, thereby, accomplishing a higher-level QoS at the expense of other

traffic types. By prioritizing packets, DiffServ routers can minimize transmission delays for time-sensitive packets such as VoIP packets.

This page allows you to assign Differentiated Services (DiffServ) to four classes of traffic (Media Premium, Control Premium, Gold, and Bronze) and to assign VLAN priorities (IEEE 802.1p) to various values of DiffServ. For a detailed description of the parameters appearing on this page, see 'Networking Parameters' on page 755. For a description on QoS and the mapping of each application to a class of service, see 'Quality of Service Parameters' on page 124.

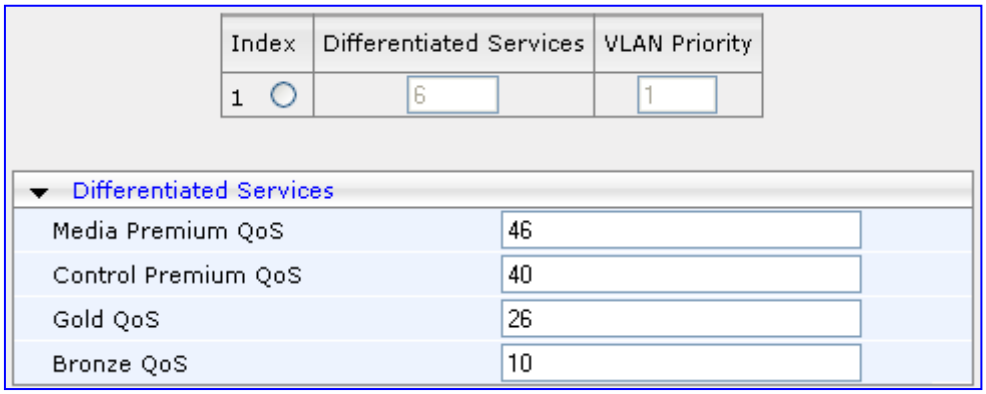
Notes:

- For the settings of this table to take effect, a device reset is required.
- You can also configure the DiffServ table using the ini file table parameter DiffServToVlanPriority.

➢ **To configure QoS:**

1. Open the Diff Serv Table page (**Configuration** tab > **VoIP** menu > **Network** submenu > **QoS Settings**).

Figure 11-4: DiffServ Table Page



2. Configure DiffServ to VLAN priority mapping (Layer-2 QoS):
   a. Enter an index entry, and then click Add.
   b. In the 'Differentiated Services' field, enter the DiffServ value (0-63) and its corresponding VLAN priority level (0-7).
   c. Click Apply.
3. Configure the desired DiffServ (Layer-3 QoS) values for the following traffic classes:
   - Media Premium QoS: this affects Media RTP packets sent by the VoIP towards the LAN.
   - Control Premium QoS: this affects Control Protocol (SIP) packets sent by the VoIP towards the LAN.
   - Gold QoS: this affects HTTP Streaming packets sent by the VoIP towards the LAN.
   - Bronze QoS: this affects OAMP packets sent by the VoIP towards the LAN.
4. Click Submit to apply your changes.
5. Save the changes to flash memory and reset the device (see 'Saving Configuration' on page 680).

11.4 DNS

You can use the device's embedded domain name server (DNS) or an external, third-party DNS to translate domain names into IP addresses. This is useful if domain names are used as the destination in call routing.

The device supports the configuration of the following DNS types:

- Internal DNS table - see 'Configuring the Internal DNS Table' on page 137
- Internal SRV table - see 'Configuring the Internal SRV Table' on page 138

11.4.1 Configuring the Internal DNS Table

The Internal DNS Table page, similar to a DNS resolution translates up to 20 host (domain) names into IP addresses (e.g., when using the Outbound IP Routing Table for Tel-to-IP call routing). Up to four different IP addresses can be assigned to the same host name (typically used for alternative Tel-to-IP call routing).

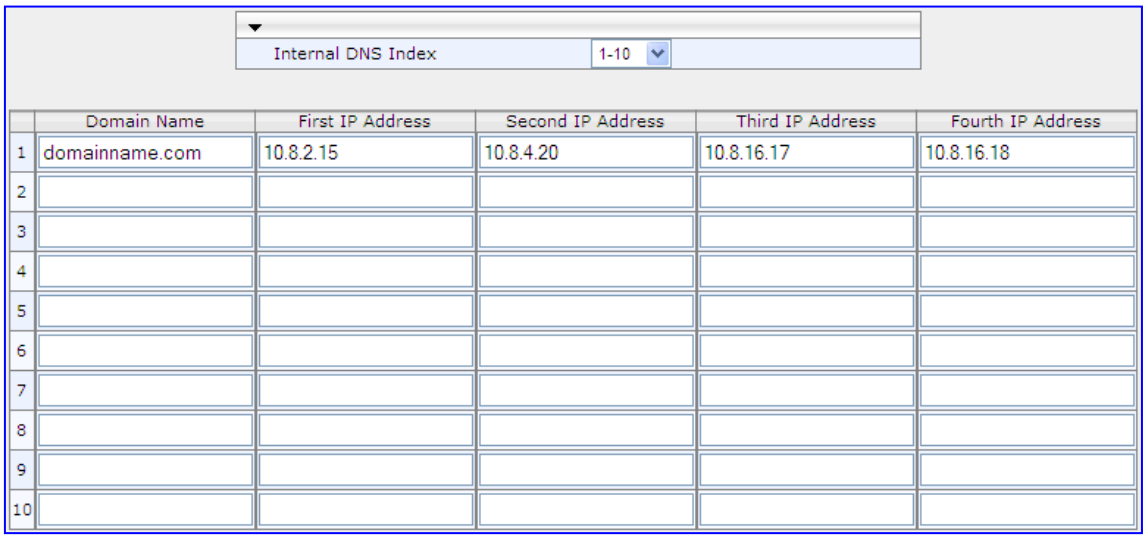
Notes:

- The device initially attempts to resolve a domain name using the Internal DNS table. If the domain name isn't listed in the table, the device performs a DNS resolution using an external DNS server (defined in the Multiple Interface table - see 'Configuring IP Interface Settings' on page 117).
- You can also configure the DNS table using the *ini* file table parameter DNS2IP (see 'DNS Parameters' on page 761).

➢ **To configure the internal DNS table:**

1. Open the Internal DNS Table page (**Configuration** tab > **VoIP** menu > **Network** submenu > **DNS** submenu > **Internal DNS Table**).

Figure 11-5: Internal DNS Table Page



2. In the 'Domain Name' field, enter the host name to be translated. You can enter a string of up to 31 characters.
3. In the 'First IP Address' field, enter the first IP address (in dotted-decimal format notation) to which the host name is translated.

4. Optionally, in the 'Second IP Address', 'Third IP Address', and 'Second IP Address' fields, enter the next IP addresses to which the host name is translated.
5. Click Submit to apply your changes.
6. To save the changes to flash memory, see 'Saving Configuration' on page 680.

11.4.2 Configuring the Internal SRV Table

The Internal SRV Table page resolves host names to DNS A-Records. Three different A-Records can be assigned to each host name. Each A-Record contains the host name, priority, weight, and port.

Notes:

- If the Internal SRV table is configured, the device initially attempts to resolve a domain name using this table. If the domain name isn't found, the device performs an Service Record (SRV) resolution using an external DNS server (defined in the Multiple Interface table - see 'Configuring IP Interface Settings' on page 117).
- You can also configure the Internal SRV table using the *ini* file table parameter SRV2IP (see 'DNS Parameters' on page 761).

➢ **To configure the Internal SRV table:**

1. Open the Internal SRV Table page (**Configuration** tab > **VoIP** menu > **Network** submenu > **DNS** submenu > **Internal SRV Table**).

Figure 11-6: Internal SRV Table Page

	Domain Name	Transport Type	DNS Name 1	Priority	Weight	Port	DNS Name 2	Priority	Weight	Port	DNS Name 3	Priority	Weight	Port
1		UDP												
2		UDP												
3		UDP												
4		UDP												
5		UDP												
6		UDP												
7		UDP												
8		UDP												
9		UDP												
10		UDP												

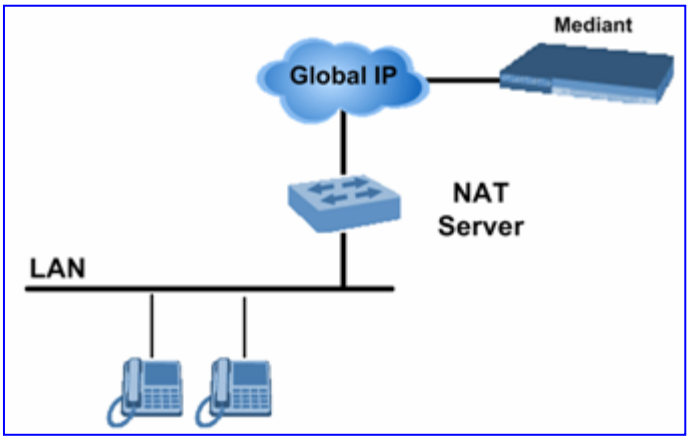
2. In the 'Domain Name' field, enter the host name to be translated. You can enter a string of up to 31 characters.
3. From the 'Transport Type' drop-down list, select a transport type.
4. In the 'DNS Name 1' field, enter the first DNS A-Record to which the host name is translated.
5. In the 'Priority', 'Weight' and 'Port' fields, enter the relevant values
6. Repeat steps 4 through 5, for the second and third DNS names, if required.
7. Repeat steps 2 through 6, for each entry.
8. Click Submit to apply your changes.
9. To save the changes so they are available after a hardware reset or power fail, see 'Saving Configuration' on page 680.

11.5 NAT (Network Address Translation) Support

Network Address Translation (NAT) is a mechanism that maps a set of internal IP addresses used within a private network to global IP addresses, providing transparent routing to end hosts. The primary advantages of NAT include (1) Reduction in the number of global IP addresses required in a private network (global IP addresses are only used to connect to the Internet); (2) Better network security by hiding its internal architecture.

The following figure illustrates the device's supported NAT architecture.

Figure 11-7: NAT Support



The design of SIP creates a problem for VoIP traffic to pass through NAT. SIP uses IP addresses and port numbers in its message body and the NAT server can't modify SIP messages and therefore, can't change local to global addresses. Two different streams traverse through NAT: signaling and media. A device (located behind a NAT) that initiates a signaling path has problems in receiving incoming signaling responses (they are blocked by the NAT server). Furthermore, the initiating device must notify the receiving device where to send the media.

To resolve these issues, the following mechanisms are available:

- First Incoming Packet Mechanism (see 'First Incoming Packet Mechanism' on page 139)
- RTP No-Op packets according to the avt-rtp-noop draft (see 'No-Op Packets' on page 140)

For information on SNMP NAT traversal, refer to the *Product Reference Manual*.

11.5.1 First Incoming Packet Mechanism

If the remote device resides behind a NAT device, it's possible that the device can activate the RTP/RTCP/T.38 streams to an invalid IP address / UDP port. To avoid such cases, the device automatically compares the source address of the incoming RTP/RTCP/T.38 stream with the IP address and UDP port of the remote device. If the two are not identical, the transmitter modifies the sending address to correspond with the address of the incoming stream. The RTP, RTCP and T.38 can thus have independent destination IP addresses and UDP ports.

You can disable the NAT mechanism by setting the *ini* file parameter DisableNAT to 1. The two parameters EnableIpAddrTranslation and EnableUdpPortTranslation allow you to specify the type of compare operation that occurs on the first incoming packet. To compare only the IP address, set EnableIpAddrTranslation to 1, and EnableUdpPortTranslation to 0. In this case, if the first incoming packet arrives with only a difference in the UDP port, the sending addresses won't change. If both the IP address and UDP port need to be compared, then both parameters need to be set to 1.

11.5.2 No-Op Packets

The device's No-Op packet support can be used to verify Real-Time Transport Protocol (RTP) and T.38 connectivity, and to keep NAT bindings and Firewall pinholes open. The No-Op packets are available for sending in RTP and T.38 formats.

You can control the activation of No-Op packets by using the *ini* file parameter NoOpEnable. If No-Op packet transmission is activated, you can control the time interval in which No-Op packets are sent in the case of silence (i.e., no RTP or T.38 traffic). This is performed using the *ini* file parameter NoOpInterval. For a description of the RTP No-Op *ini* file parameters, see 'Networking Parameters' on page 755.

- **RTP No-Op:** The RTP No-Op support complies with IETF Internet-Draft draft-wing-avt-rtp-noop-03 ("A No-Op Payload Format for RTP"). This IETF document defines a No-Op payload format for RTP. The draft defines the RTP payload type as dynamic. You can control the payload type with which the No-Op packets are sent. This is performed using the RTPNoOpPayloadType *ini* parameter (see 'Networking Parameters' on page 755). AudioCodes' default payload type is 120.
- **T.38 No-Op:** T.38 No-Op packets are sent only while a T.38 session is activated. Sent packets are a duplication of the previously sent frame (including duplication of the sequence number).

Note: Receipt of No-Op packets is always supported.

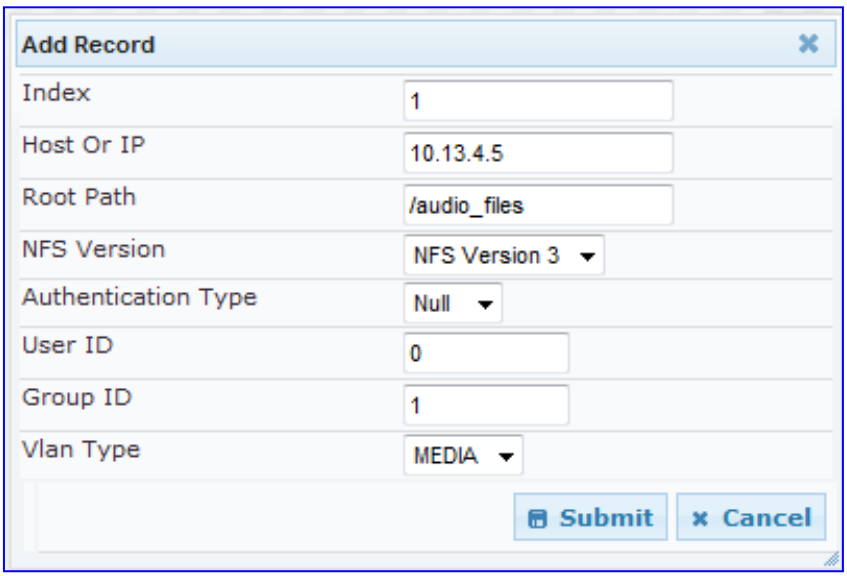
11.6 Configuring NFS Settings

Network File System (NFS) enables the device to access a remote server's shared files and directories, and to handle them as if they're located locally. You can configure up to 16 different NFS file systems. As a file system, the NFS is independent of machine types, operating systems, and network architectures. NFS is used by the device to load the *cmp*, *ini*, and auxiliary files, using the Automatic Update mechanism (refer to the *Product Reference Manual*). Note that an NFS file server can share multiple file systems. There must be a separate row for each remote file system shared by the NFS file server that needs to be accessed by the device.

➢ **To add remote NFS file systems:**

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).
2. Under the NFS Settings group, click the **NFS Table** button; the NFS Settings page appears.
3. Click the **Add** button; the Add Record dialog box appears:

Figure 11-8: Add Record Dialog Box for NFS



4. Configure the NFS parameters according to the table below.
5. Click the **Submit** button; the remote NFS file system is immediately applied, which can be verified by the appearance of the 'NFS mount was successful' message in the Syslog server.
6. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Notes:

- To avoid terminating current calls, a row must not be deleted or modified while the device is currently accessing files on that remote NFS file system.
- The combination of 'Host Or IP' and 'Root Path' must be unique for each row in the table. For example, the table must include only one row with a Host/IP of 192.168.1.1 and Root Path of /audio.
- For configuring Web interface tables, see 'Working with Tables' on page 62.
- You can also configure the NFS table using the *ini* file table parameter NFSServers (see 'NFS Parameters' on page 760).

Table 11-16: NFS Settings Parameters

Parameter	Description
Index	The row index of the remote file system. The valid range is 1 to 16.
Host Or IP	The domain name or IP address of the NFS server. If a domain name is provided, a DNS server must be configured.
Root Path	Path to the root of the remote file system in the format: /**[path].** For example, '/audio'.
NFS Version	NFS version used to access the remote file system. ▪ **[2]** NFS Version 2 ▪ **[3]** NFS Version 3 (default)
Authentication Type	Authentication method used for accessing the remote file system. ▪ **[0]** Null ▪ **[1]** Unix (default)
User ID	User ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 0.
Group ID	Group ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 1.
VLAN Type	The VLAN type for accessing the remote file system. ▪ **[0]** OAM ▪ **[1]** MEDIA (default) **Note:** This parameter applies only if VLANs are enabled or if Multiple IPs is configured (see 'Network Configuration' on page 120).

11.7 Robust Receipt of Media Streams

This mechanism filters out unwanted RTP streams that are sent to the same port number on the device. These multiple RTP streams can result from traces of previous calls, call control errors, and deliberate attacks. When more than one RTP stream reaches the device on the same port number, the device accepts only one of the RTP streams and rejects the rest of the streams.

The RTP stream is selected according to the following: The first packet arriving on a newly opened channel sets the source IP address and UDP port from which further packets are received. Thus, the source IP address and UDP port identify the currently accepted stream. If a new packet arrives whose source IP address or UDP port are different to the currently accepted RTP stream, one of the following occurs:

- The device reverts to the new RTP stream when the new packet has a source IP address and UDP port that are the same as the remote IP address and UDP port that were stated during the opening of the channel.
- The packet is dropped when the new packet has any other source IP address and UDP port.

11.8 Multiple Routers Support

Multiple routers support is designed to assist the device when it operates in a multiple routers network. The device learns the network topology by responding to Internet Control Message Protocol (ICMP) redirections and caches them as routing rules (with expiration time).

When a set of routers operating within the same subnet serve as devices to that network and intercommunicate using a dynamic routing protocol, the routers can determine the shortest path to a certain destination and signal the remote host the existence of the better route. Using multiple router support, the device can utilize these router messages to change its next hop and establish the best path.

Note: Multiple Routers support is an integral feature that doesn't require configuration.

12 Security

This section describes the VoIP security-related configuration.

12.1 Configuring Firewall Settings

The device provides an internal firewall, allowing you (the security administrator) to define network traffic filtering rules. You can add up to 50 ordered firewall rules.

The access list provides the following firewall rules:

- Block traffic from known malicious sources
- Only allow traffic from known friendly sources, and block all others
- Mix allowed and blocked network sources
- Limit traffic to a pre-defined rate (blocking the excess)
- Limit traffic to specific protocols, and specific port ranges on the device

For each packet received on the network interface, the table is scanned from the top down until a matching rule is found. This rule can either deny (*block*) or permit (*allow*) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted. For more information on the internal firewall, refer to the *Product Reference Manual*.

Notes:

- It is recommended to add a rule at the end of your table that blocks all traffic and add firewall rules above it (in the table) that allow traffic (with bandwidth limitations). To block all traffic, the following must be set:
  - IP address to 0.0.0.0
  - Prefix length of 0 (implies the rule can match any IP address)
  - Local port range 0-65535
  - Protocol "Any"
  - Action Upon Match "block"
- You can also configure the firewall settings using the *ini* file table parameter AccessList (see 'Security Parameters' on page 779).

➢ To add firewall rules:

1. Open the Firewall Settings page (**Configuration** tab > **VoIP** menu > **Security** submenu > **Firewall Settings**).

Figure 12-1: Firewall Settings Page

Edit Rule	Rule Status	Source IP	Source Port	Prefix Length	Local Port Range	Protocol	Use Specific Interface	Interface Name	Packet Size	Byte rate	Burst Bytes	Action Upon Match	Match Count
1	Not Active	mgmt.customer.com	0	32	0 - 80	tcp	Enable	Mng	0	0	0	Allow	0
2	Not Active	192.0.0.0	0	9	0-65535	Any	Disable	None	0	40000	50000	ALLOW	0
3	Not Active	10.31.4.0	0	24	4000-9000	Any	Disable	None	0	0	0	BLOCK	0
4	Not Active	10.4.0.0	0	16	4000-9000	Any	Disable	None	0	0	0	BLOCK	0

2. In the 'Add' field, enter the index of the access rule that you want to add, and then click **Add**; a new firewall rule index appears in the table.
3. Configure the firewall rule's parameters according to the table below.
4. Click one of the following buttons:
   - **Apply:** saves the new rule (without activating it).
   - **Duplicate Rule:** adds a new rule by copying a selected rule.
   - **Activate:** saves the new rule and activates it.
   - **Delete:** deletes the selected rule.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

The previous figure shows the following access list settings:

- **Rule #1:** traffic from the host 'mgmt.customer.com' destined to TCP ports 0 to 80, is always allowed.
- **Rule #2:** traffic from the 192.xxx.yyy.zzz subnet, is limited to a rate of 40 Kbytes per second (with an allowed burst of 50 Kbytes). Note that the rate is specified in bytes, not bits, per second; a rate of 40000 bytes per second, nominally corresponds to 320 kbps.
- **Rule #3:** traffic from the subnet 10.31.4.xxx destined to ports 4000-9000 is always blocked, regardless of protocol.
- **Rule #4:** traffic from the subnet 10.4.xxx.yyy destined to ports 4000-9000 is always blocked, regardless of protocol.
- All other traffic is allowed

➢ To edit a rule:

1. In the 'Edit Rule' column, select the rule that you want to edit.
2. Modify the fields as desired.
3. Click the **Apply** button to save the changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

➢ To activate a de-activated rule:

1. In the 'Edit Rule' column, select the de-activated rule that you want to activate.
2. Click the **Activate** button; the rule is activated.

➢ To de-activate an activated rule:

1. In the 'Edit Rule' column, select the activated rule that you want to de-activate.
2. Click the **DeActivate** button; the rule is de-activated.

➢ **To delete a rule:**

1. Select the radio button of the entry you want to activate.
2. Click the **Delete Rule** button; the rule is deleted.
3. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 12-1: Internal Firewall Parameters

Parameter	Description
Rule Status	A read-only field indicating whether the rule is active or not. **Note:** After device reset, all rules are active.
Source IP **[AccessList_Source_IP]**	IP address (or DNS name) or a specific host name of the source network (i.e., from where the incoming packet is received).
Source Port **[AccessList_Source_Port]**	Defines the source UDP/TCP ports (on the remote host) from where packets are sent to the device. The valid range is 0 to 65535. **Note:** When set to 0, this field is ignored and any source port matches the rule.
Prefix Length **[AccessList_PrefixLen]**	IP network mask. 32 for a single host, or the appropriate value for the source IP addresses. ▪ A value of 8 corresponds to IPv4 subnet class A (network mask of 255.0.0.0). ▪ A value of 16 corresponds to IPv4 subnet class B (network mask of 255.255.0.0). ▪ A value of 24 corresponds to IPv4 subnet class C (network mask of 255.255.255.0). The IP address of the sender of the incoming packet is trimmed in accordance with the prefix length (in bits) and then compared to the parameter 'Source IP'.
Source Port **[AccessList_Source_Port]**	Defines the source UDP or TCP ports (on the remote host) from where packets are sent to the device. The valid range is 0 to 65535. **Note:** When set to 0, this field is ignored and any port matches the rule.
Local Port Range **[AccessList_Start_Port]** **[AccessList_End_Port]**	The destination UDP/TCP ports (on this device) to which packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.
Protocol **[AccessList_Protocol]**	The protocol type (e.g., UDP, TCP, ICMP, ESP or 'Any'), or the IANA protocol number (in the range of 0 (Any) to 255). **Note:** This field also accepts the abbreviated strings 'SIP' and 'HTTP'. Specifying these strings implies selection of the TCP or UDP protocols, and the appropriate port numbers as defined on the device.
Use Specific Interface **[AccessList_Use_Specific_Interface]**	Determines whether you want to apply the rule to a specific network interface defined in the Multiple Interface table (i.e., packets received from that defined in the Source IP field and received on this network interface):

Parameter	Description
	▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ If enabled, then in the 'Interface Name' field (described below), select the interface to which the rule is applied. ▪ If disabled, then the rule applies to all interfaces.
Interface Name **[AccessList_Interface_ID]**	The network interface to which you want to apply the rule. This is applicable if you enabled the 'Use Specific Interface' field. The list displays interface names as defined in the Multiple Interface table (see 'Configuring IP Interface Settings' on page 117).
Packet Size **[AccessList_Packet_Size]**	Maximum allowed packet size. The valid range is 0 to 65535. **Note:** When filtering fragmented IP packets, this field relates to the overall (re-assembled) packet size, and not to the size of each fragment.
Byte Rate **[AccessList_Byte_Rate]**	Expected traffic rate (bytes per second). This field defines the allowed bandwidth for the specified protocol. In addition to this field, the 'Burst Bytes' field provides additional allowance such that momentary bursts of data may utilize more than the defined byte rate, without being interrupted. For example, if 'Byte Rate' is set to 40000 and 'Burst Bytes' to 50000, then this implies the following: the allowed bandwidth is 40000 bytes/sec with extra allowance of 50000 bytes; if, for example, the actual traffic rate is 45000 bytes/sec, then this allowance would be consumed within 10 seconds, after which all traffic exceeding the allocated 40000 bytes/sec is dropped. If the actual traffic rate then slowed to 30000 bytes/sec, then the allowance would be replenished within 5 seconds.
Burst Bytes **[AccessList_Byte_Burst]**	Tolerance of traffic rate limit (number of bytes).
Action Upon Match **[AccessList_Allow_Type]**	Action upon match (i.e., 'Allow' or 'Block').
Match Count **[AccessList_MatchCount]**	A read-only field displaying the number of packets accepted/rejected by the specific rule.

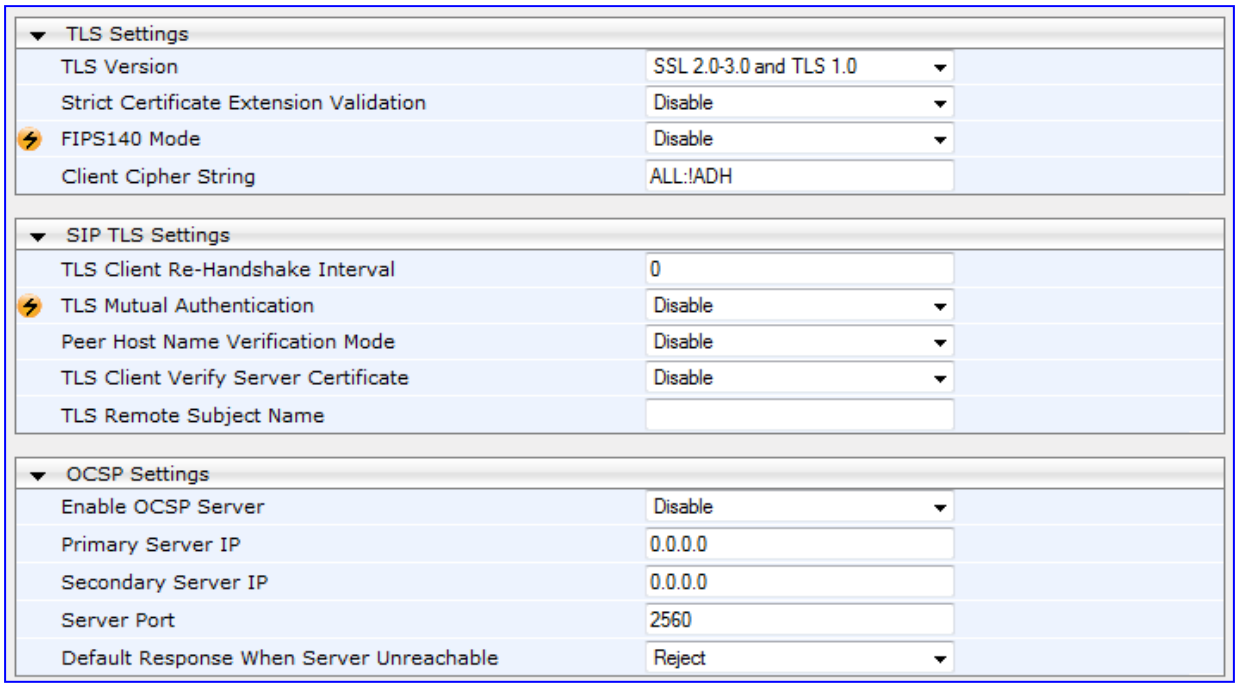
12.2 Configuring General Security Settings

The General Security Settings page is used to configure various security features. For a description of the parameters appearing on this page, refer 'Configuration Parameters Reference' on page 752.

➢ **To configure the general security parameters:**

1. Open the General Security Settings page (**Configuration** tab > **VoIP** menu > **Security** submenu > **General Security Settings**).

Figure 12-2: General Security Settings Page



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, refer to 'Saving Configuration' on page 680.

Reader's Notes

13 Media

This section describes the media-related configuration.

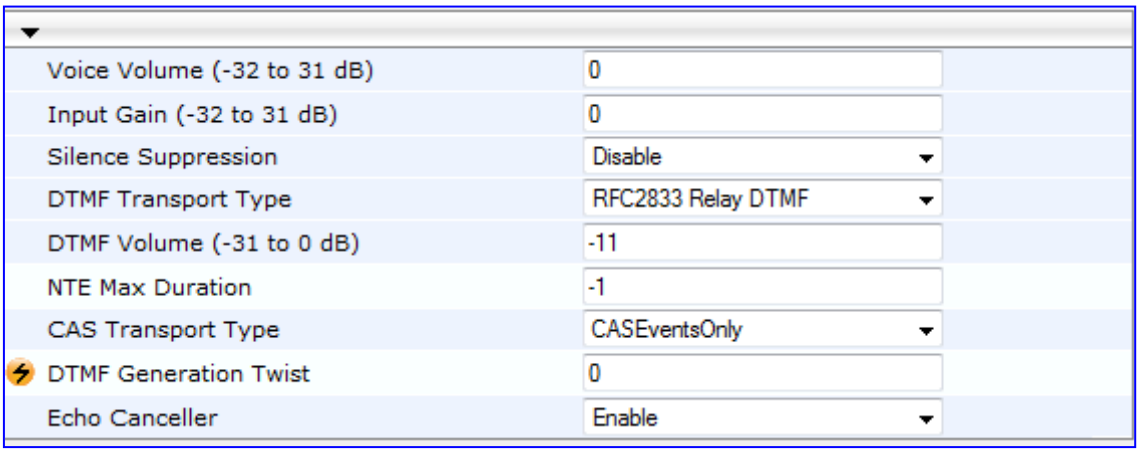
13.1 Configuring Voice Settings

The Voice Settings page configures various voice parameters such as voice volume, silence suppression, and DTMF transport type. For a detailed description of these parameters, see 'Configuration Parameters Reference' on page 752.

➢ **To configure the voice parameters:**

1. Open the Voice Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Voice Settings**).

Figure 13-1: Voice Settings Page



2. Configure the Voice parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

13.1.1 Voice Gain (Volume) Control

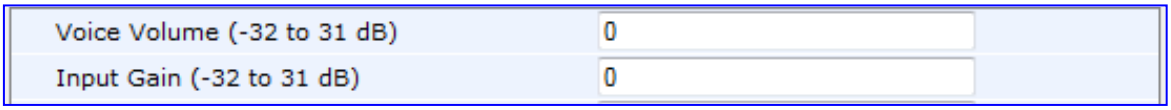
The device allows you to configure the level of the received (input gain) Tel-to-IP signal and the level of the transmitted (output gain) IP-to-Tel signal. The gain can be set between -32 and 31 decibels (dB).

The procedure below describes how to configure gain control using the Web interface:

➢ **To configure gain control using the Web interface:**

1. Open the Voice Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Voice Settings**).

Figure 13-2: Voice Volume Parameters in Voice Settings Page



2. Configure the following parameters:
   - 'Voice Volume' (*VoiceVolume*) - Defines the voice gain control (in decibels) for IP-to-Tel
   - 'Input Gain' (*InputGain*) - Defines the PCM input gain control (in decibels) for Tel-to-IP
3. Click **Submit** to apply your settings.

The procedure below describes how to configure gain control using CLI.

➢ **To configure gain control using CLI:**

1. Access the VoIP configuration section of the CLI:
```
# (configure-voip) media voice-processing
```
2. Configure the following parameters:
   - **input-gain:** Defines the PCM input gain control (in decibels) for Tel-to-IP.
   - **voice-volume:** Defines the voice gain control (in decibels) for IP-to-Tel.

13.1.2 Silence Suppression (Compression)

Silence suppression (compression) is a method for conserving bandwidth on VoIP calls by not sending packets when silence is detected. The device uses its VAD feature to detect periods of silence in the voice channel during an established call. When silence is detected, it stops sending packets in the channel.

The procedure below describes how to enable silence suppression using the Web interface.

➢ **To enable silence suppression using the Web interface:**

1. Open the Voice Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Voice Settings**).
2. Set the 'Silence Suppression' (*EnableSilenceCompression*) field to **Enable**.
3. Click **Submit** to apply your changes.

The procedure below describes how to configure silence suppression using CLI.

➢ **To configure silence suppression using CLI:**

1. Access the VoIP configuration section of the CLI:
```
# (configure-voip) media voice-processing
```
2. Configure the following parameter:
   - `silence-compression-mode` Enables silence suppression.

13.1.3 Echo Cancellation

The device supports adaptive linear (line) echo cancellation according to G.168-2002. Echo cancellation is a mechanism that removes echo from the voice channel. Echoes are reflections of the transmitted signal.

In this line echo, echoes are generated when two-wire telephone circuits (carrying both transmitted and received signals on the same wire pair) are converted to a four-wire circuit. Echoes are reflections of the transmitted signal, which result from impedance mismatch in the hybrid (bi-directional 2-wire to 4-wire converting device).

An estimated echo signal is built by feeding the decoder output signal to an RLS-like adaptive filter, which adapts itself to the characteristics of the echo path. The 'estimated echo signal' (the output of this filter) is then subtracted from the input signal (which is the sum of the desired input signal and the undesired echo) to provide a clean signal. To suppress the remaining residual echo, a Non Linear Processor (NLP) is used, as well as a double-talk (two people speak at the same time) detector that prevents false adaptation during near-end speech.

The procedure below describes how to configure echo cancellation using the Web interface:

➢ **To configure echo cancellation using the Web interface:**

1. Open the Voice Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Voice Settings**).
2. Set the 'Echo Canceller' field (*EnableEchoCanceller*) to **Enable**.
3. Open the General Media Settings page (Configuration tab > VoIP menu > Media submenu > General Media Settings).
4. From the 'Max Echo Canceller Length' drop-down list (MaxEchoCancellerLength), select the maximum echo path delay (tail length) for the echo canceller.

Note: The following additional echo cancellation parameters are configurable only through the *ini* file:

- *ECHybridLoss* **-** defines the four-wire to two-wire worst-case Hybrid loss
- *ECNLPMode* **-** defines the echo cancellation Non-Linear Processing (NLP) mode
- *EchoCancellerAggressiveNLP* **-** enables Aggressive NLP at the first 0.5 second of the call

The procedure below describes how to configure echo cancellation using CLI.

➢ **To configure echo cancellation using CLI:**

1. Access the VoIP configuration section of the CLI:
```
# (configure-voip) media voice-processing
```
2. Configure the following parameters:
   - `echo-canceller-NLP-mode`**:** defines the echo cancellation non-linear processing mode.
   - `echo-canceller-aggressive-NLP`**:** enables Aggressive Echo Canceller NLP at first 0.5 second of the call
   - `echo-canceller-enable`**:** enables the Echo Canceller.
   - `echo-canceller-freeze`**:** freezes echo canceller internal state and avoids additional adaptations.
   - `echo-canceller-hybrid-loss`**:** sets worst case ratio between signal level transmitted to hybrid and echo level returning from hybrid.
   - `max-echo-canceller-length`**:** defines the maximum EC (Echo Canceller) length.

13.2 Fax and Modem Capabilities

This section describes the device's fax and modem capabilities, and includes the following main subsections:

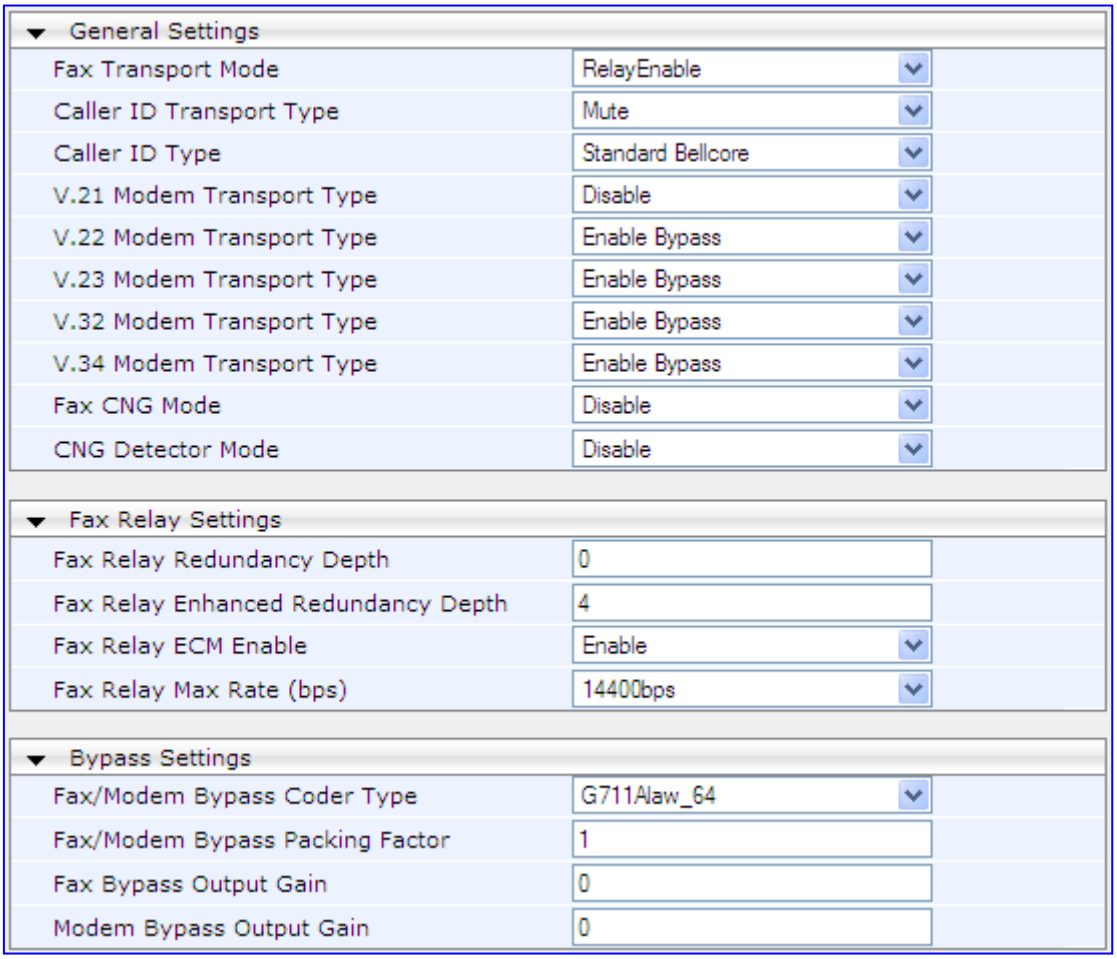
- Fax and modem operating modes (see 'Fax/Modem Operating Modes' on page 156)
- Fax and modem transport modes (see 'Fax/Modem Transport Modes' on page 156)
- V.34 fax support (see 'V.34 Fax Support' on page 161)
- V.152 support (see 'V.152 Support' on page 162)

The fax and modem parameters can e configured in the 'Fax/Modem/CID Settings page. For a detailed description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

➢ **To configure the fax and modem parameters:**

1. Open the Fax/Modem/CID Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Fax/Modem/CID Settings**).

Figure 13-3: Fax/Modem/CID Settings Page



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Note: Some SIP parameters override these fax and modem parameters (see the parameter IsFaxUsed, and V.152 parameters in Section 'V.152 Support' on page 162).

13.2.1 Fax/Modem Operating Modes

The device supports two modes of operation:

- Fax/modem negotiation that is not performed during the establishment of the call.
- Voice-band data (VBD) mode for V.152 implementation (see 'V.152 Support' on page 162): fax/modem capabilities are negotiated between the device and the remote endpoint at the establishment of the call. During a call, when a fax/modem signal is detected, transition from voice to VBD (or T.38) is automatically performed and no additional SIP signaling is required. If negotiation fails (i.e., no match is achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

13.2.2 Fax/Modem Transport Modes

The device supports the following transport modes for fax per modem type (V.22/V.23/Bell/V.32/V.34):

- T.38 fax relay (see 'T.38 Fax Relay Mode' on page 156)
- G.711 Transport: switching to G.711 when fax/modem is detected (see 'G.711 Fax / Modem Transport Mode' on page 158)
- Fax fallback to G.711 if T.38 is not supported (see 'Fax Fallback' on page 158)
- Fax and modem bypass: a proprietary method that uses a high bit rate coder (see 'Fax/Modem Bypass Mode' on page 158)
- NSE Cisco's Pass-through bypass mode for fax and modem (see 'Fax / Modem NSE Mode' on page 159)
- Transparent with events: passing the fax / modem signal in the current voice coder with adaptations (see 'Fax / Modem Transparent with Events Mode' on page 160)
- Transparent: passing the fax / modem signal in the current voice coder (see 'Fax / Modem Transparent Mode' on page 160)
- RFC 2833 ANS Report upon Fax/Modem Detection (see 'RFC 2833 ANS Report upon Fax/Modem Detection' on page 161)

'Adaptations' refer to automatic reconfiguration of certain DSP features for handling fax/modem streams differently than voice.

13.2.2.1 T.38 Fax Relay Mode

In Fax Relay mode, fax signals are transferred using the T.38 protocol. T.38 is an ITU standard for sending fax across IP networks in real-time mode. The device currently supports only the T.38 UDP syntax.

T.38 can be configured in the following ways:

- Switching to T.38 mode using SIP Re-INVITE messages (see 'Switching to T.38 Mode using SIP Re-INVITE' on page 157)
- Automatically switching to T.38 mode without using SIP Re-INVITE messages (see 'Automatically Switching to T.38 Mode without SIP Re-INVITE' on page 157)

When fax transmission ends, the reverse switching from fax relay to voice is automatically performed at both the local and remote endpoints.

You can change the fax rate declared in the SDP, using the parameter FaxRelayMaxRate (this parameter doesn't affect the actual transmission rate). In addition, you can enable or disable Error Correction Mode (ECM) fax mode using the FaxRelayECMEnable parameter.

When using T.38 mode, you can define a redundancy feature to improve fax transmission over congested IP networks. This feature is activated using the FaxRelayRedundancyDepth and FaxRelayEnhancedRedundancyDepth parameters. Although this is a proprietary redundancy scheme, it should not create problems when working with other T.38 decoders.

13.2.2.1.1 Switching to T.38 Mode using SIP Re-INVITE

In the Switching to T.38 Mode using SIP Re-INVITE mode, upon detection of a fax signal the terminating device negotiates T.38 capabilities using a Re-INVITE message. If the far-end device doesn't support T.38, the fax fails. In this mode, the parameter FaxTransportMode is ignored.

To configure T.38 mode using SIP Re-INVITE messages, set IsFaxUsed to 1. Additional configuration parameters include the following:

- FaxRelayEnhancedRedundancyDepth
- FaxRelayRedundancyDepth
- FaxRelayECMEnable
- FaxRelayMaxRate

Note: The terminating gateway sends T.38 packets immediately after the T.38 capabilities are negotiated in SIP. However, the originating device by default, sends T.38 (assuming the T.38 capabilities are negotiated in SIP) only after it receives T.38 packets from the remote device. This default behavior cannot be used when the originating device is located behind a firewall that blocks incoming T.38 packets on ports that have not yet received T.38 packets from the internal network. To resolve this problem, the device should be configured to send CNG packets in T.38 upon CNG signal detection (CNGDetectorMode = 1).

13.2.2.1.2 Automatically Switching to T.38 Mode without SIP Re-INVITE

In the Automatically Switching to T.38 Mode without SIP Re-INVITE mode, when a fax signal is detected, the channel automatically switches from the current voice coder to answer tone mode, and then to T.38-compliant fax relay mode.

To configure automatic T.38 mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 1
- Additional configuration parameters:
  - FaxRelayEnhancedRedundancyDepth
  - FaxRelayRedundancyDepth
  - FaxRelayECMEnable
  - FaxRelayMaxRate

13.2.2.2 G.711 Fax / Modem Transport Mode

In this mode, when the terminating device detects fax or modem signals (CED or AnsAM), it sends a Re-INVITE message to the originating device requesting it to re-open the channel in G.711 VBD with the following adaptations:

- Echo Canceller = off
- Silence Compression = off
- Echo Canceller Non-Linear Processor Mode = off
- Dynamic Jitter Buffer Minimum Delay = 40
- Dynamic Jitter Buffer Optimization Factor = 13

After a few seconds upon detection of fax V.21 preamble or super G3 fax signals, the device sends a second Re-INVITE enabling the echo canceller (the echo canceller is disabled only on modem transmission).

A 'gpmd' attribute is added to the SDP according to the following format:

- **For G.711A-law:** a=gpmd:0 vbd=yes;ecan=on (or off, for modems)
- **For G.711 µ-law:** a=gpmd:8 vbd=yes;ecan=on (or off for modems)

The parameters FaxTransportMode and VxxModemTransportType are ignored and automatically set to the mode called 'transparent with events'.

To configure fax / modem transparent mode, set IsFaxUsed to 2.

13.2.2.3 Fax Fallback

In this mode, when the terminating device detects a fax signal, it sends a Re-INVITE message to the originating device with T.38. If the remote device doesn't support T.38 (replies with SIP response 415 'Media Not Supported'), the device sends a new Re-INVITE with G.711 VBD with the following adaptations:

- Echo Canceller = on
- Silence Compression = off
- Echo Canceller Non-Linear Processor Mode = off
- Dynamic Jitter Buffer Minimum Delay = 40
- Dynamic Jitter Buffer Optimization Factor = 13

When the device initiates a fax session using G.711, a 'gpmd' attribute is added to the SDP according to the following format:

- **For G.711A-law:** a=gpmd:0 vbd=yes;ecan=on
- **For G.711 µ-law:** a=gpmd:8 vbd=yes;ecan=on

In this mode, the parameter FaxTransportMode is ignored and automatically set to 'transparent'.

To configure fax fallback mode, set IsFaxUsed to 3.

13.2.2.4 Fax/Modem Bypass Mode

In this proprietary mode, when fax or modem signals are detected, the channel automatically switches from the current voice coder to a high bit-rate coder (according to the parameter FaxModemBypassCoderType). In addition, the channel is automatically reconfigured with the following fax / modem adaptations:

- Disables silence suppression
- Enables echo cancellation for fax
- Disables echo cancellation for modem
- Performs certain jitter buffering optimizations

The network packets generated and received during the bypass period are regular voice RTP packets (per the selected bypass coder), but with a different RTP payload type (according to the parameters FaxBypassPayloadType and ModemBypassPayloadType). During the bypass period, the coder uses the packing factor, which is defined by the parameter FaxModemBypassM. The packing factor determines the number of coder payloads (each the size of FaxModemBypassBasicRTPPacketInterval) that are used to generate a single fax/modem bypass packet. When fax/modem transmission ends, the reverse switching, from bypass coder to regular voice coder is performed.

To configure fax / modem bypass mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 2
- V21ModemTransportType = 2
- V22ModemTransportType = 2
- V23ModemTransportType = 2
- V32ModemTransportType = 2
- V34ModemTransportType = 2
- BellModemTransportType = 2
- Additional configuration parameters:
  - FaxModemBypassCoderType
  - FaxBypassPayloadType
  - ModemBypassPayloadType
  - FaxModemBypassBasicRTPPacketInterval
  - FaxModemBypasDJBufMinDelay

Note: When the device is configured for modem bypass and T.38 fax, V.21 low-speed modems are not supported and fail as a result.

Tip: When the remote (non-AudioCodes') gateway uses G711 coder for voice and doesn't change the coder payload type for fax or modem transmission, it is recommended to use the Bypass mode with the following configuration:

- EnableFaxModemInbandNetworkDetection = 1
- FaxModemBypassCoderType = same coder used for voice
- FaxModemBypassM = same interval as voice
- ModemBypassPayloadType = 8 if voice coder is A-Law; 0 if voice coder is Mu-Law

13.2.2.5 Fax / Modem NSE Mode

In this mode, fax and modem signals are transferred using Cisco-compatible Pass-through bypass mode. Upon detection of fax or modem answering tone signal, the terminating device sends three to six special NSE RTP packets (using NSEpayloadType, usually 100). These packets signal the remote device to switch to G.711 coder (according to the parameter FaxModemBypassCoderType). After a few NSE packets are exchanged between the devices, both devices start using G.711 packets with standard payload type (8 for G.711 A-Law and 0 for G.711 Mu-Law). In this mode, no Re-INVITE messages are sent. The voice channel is optimized for fax/modem transmission (same as for usual bypass mode).

The parameters defining payload type for the proprietary AudioCodes' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass.

When configured for NSE mode, the device includes in its SDP the following line:

```
a=rtpmap:100 X-NSE/8000
```

(where 100 is the NSE payload type)

The Cisco gateway must include the following definition: "modem passthrough nse payload-type 100 codec g711alaw".

To configure NSE mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 2
- NSEMode = 1
- NSEPayloadType = 100
- V21ModemTransportType = 2
- V22ModemTransportType = 2
- V23ModemTransportType = 2
- V32ModemTransportType = 2
- V34ModemTransportType = 2
- BellModemTransportType = 2

13.2.2.6 Fax / Modem Transparent with Events Mode

In this mode, fax and modem signals are transferred using the current voice coder with the following automatic adaptations:

- Echo Canceller = on (or off, for modems)
- Echo Canceller Non-Linear Processor Mode = off
- Jitter buffering optimizations

To configure fax / modem transparent with events mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 3
- V21ModemTransportType = 3
- V22ModemTransportType = 3
- V23ModemTransportType = 3
- V32ModemTransportType = 3
- V34ModemTransportType = 3
- BellModemTransportType = 3

13.2.2.7 Fax / Modem Transparent Mode

In this mode, fax and modem signals are transferred using the current voice coder without notifications to the user and without automatic adaptations. It's possible to use the Profiles mechanism (see 'Coders and Profile Definitions' on page 243) to apply certain adaptations to the channel used for fax / modem (e.g., to use the coder G.711, to set the jitter buffer optimization factor to 13, and to enable echo cancellation for fax and disable it for modem).

To configure fax / modem transparent mode, use the following parameters:

- IsFaxUsed = 0
- FaxTransportMode = 0
- V21ModemTransportType = 0
- V22ModemTransportType = 0
- V23ModemTransportType = 0
- V32ModemTransportType = 0
- V34ModemTransportType = 0
- BellModemTransportType = 0
- Additional configuration parameters:
  - CodersGroup
  - DJBufOptFactor
  - EnableSilenceCompression
  - EnableEchoCanceller

Note: This mode can be used for fax, but is not recommended for modem transmission. Instead, use the modes Bypass (see 'Fax/Modem Bypass Mode' on page 158) or Transparent with Events (see 'Fax / Modem Transparent with Events Mode' on page 160) for modem.

13.2.2.8 RFC 2833 ANS Report upon Fax/Modem Detection

The device (terminator gateway) sends RFC 2833 ANS/ANSam events upon detection of fax and/or modem answer tones (i.e., CED tone). This causes the originator to switch to fax/modem. This parameter is applicable only when the fax or modem transport type is set to bypass, Transparent-with-Events, V.152 VBD, or G.711 transport. When the device is located on the originator side, it ignores these RFC 2833 events

Relevant parameters:

- IsFaxUsed = 0 or 3
- FaxTransportMode = 2
- FaxModemNTEMode = 1
- VxxModemTransportType = 2

13.2.3 V.34 Fax Support

V.34 fax machines can transmit data over IP to the remote side using various methods. The device supports the following modes for transporting V.34 fax data over IP:

- Bypass mechanism for V.34 fax transmission (see 'Bypass Mechanism for V.34 Fax Transmission' on page 162)
- T38 Version 0 relay mode, i.e., fallback to T.38 (see 'Relay Mode for T.30 and V.34 Faxes' on page 162)

Note: The CNG detector is disabled (CNGDetectorMode = 0) in all the subsequent examples.

13.2.3.1 Bypass Mechanism for V.34 Fax Transmission

In this proprietary scenario, the device uses bypass (or NSE) mode to transmit V.34 faxes, enabling the full utilization of its speed.

Configure the following parameters to use bypass mode for both T.30 and V.34 faxes:

- FaxTransportMode = 2 (Bypass)
- V34ModemTransportType = 2 (Modem bypass)
- V32ModemTransportType = 2
- V23ModemTransportType = 2
- V22ModemTransportType = 2

Configure the following parameters to use bypass mode for V.34 faxes and T.38 for T.30 faxes:

- FaxTransportMode = 1 (Relay)
- V34ModemTransportType = 2 (Modem bypass)
- V32ModemTransportType = 2
- V23ModemTransportType = 2
- V22ModemTransportType = 2

13.2.3.2 Relay Mode for T.30 and V.34 Faxes

In this scenario, V.34 fax machines are forced to use their backward compatibility with T.30 faxes and operate in the slower T.30 mode.

Use the following parameters to use T.38 mode for both V.34 and T.30 faxes:

- FaxTransportMode = 1 (Relay)
- V34ModemTransportType = 0 (Transparent)
- V32ModemTransportType = 0
- V23ModemTransportType = 0
- V22ModemTransportType = 0

13.2.4 V.152 Support

The device supports the ITU-T recommendation V.152 (Procedures for Supporting Voice-Band Data over IP Networks). Voice-band data (VBD) is the transport of modem, facsimile, and text telephony signals over a voice channel of a packet network with a codec appropriate for such signals.

For V.152 capability, the device supports T.38 as well as VBD codecs (i.e., G.711 A-law and G.711 µ-law). The selection of capabilities is performed using the coders table (see 'Configuring Coders' on page 243).

When in VBD mode for V.152 implementation, support is negotiated between the device and the remote endpoint at the establishment of the call. During this time, initial exchange of call capabilities is exchanged in the outgoing SDP. These capabilities include whether VBD is supported and associated RTP payload types ('gpmd' SDP attribute), supported codecs, and packetization periods for all codec payload types ('ptime' SDP attribute). After this initial negotiation, no Re-INVITE messages are necessary as both endpoints are synchronized in terms of the other side's capabilities. If negotiation fails (i.e., no match was achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

Below is an example of media descriptions of an SDP indicating support for V.152. In the example, V.152 implementation is supported (using the dynamic payload type 96 and G.711 u-law as the VBD codec) as well as the voice codecs G.711 µ-law and G.729.

```
v=0
o=-  0 0 IN IPV4 <IPAdressA>
s=-
t=0 0
p=+1
c=IN IP4  <IPAddressA
m=audio <udpPort A> RTP/AVP 18 0
a=ptime:10
a=rtpmap:96 PCMU/8000
a=gpmd: 96 vbd=yes
```

Instead of using VBD transport mode, the V.152 implementation can use alternative relay fax transport methods (e.g., fax relay over IP using T.38). The preferred V.152 transport method is indicated by the SDP 'pmft' attribute. Omission of this attribute in the SDP content means that VBD mode is the preferred transport mechanism for voice-band data. To configure T.38 mode, use the CodersGroup parameter.

Note: You can also configure the device to handle G.711 coders received in INVITE SDP offers as VBD coders, using the HandleG711asVBD parameter. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing G.729 and G.711 VBD coders, allowing subsequent bypass (passthrough) sessions if fax / modem signals are detected during the call.

13.2.5 Fax Transmission behind NAT

The device supports transmission from fax machines (connected to the device) located inside (behind) a Network Address Translation (NAT). Generally, the firewall blocks T.38 (and other) packets received from the WAN, unless the device behind the NAT sends at least one IP packet from the LAN to the WAN through the firewall. If the firewall blocks T.38 packets sent from the termination IP fax, the fax fails.

To overcome this, the device sends No-Op ("no-signal") packets to open a pinhole in the NAT for the answering fax machine. The originating fax does not wait for an answer, but immediately starts sending T.38 packets to the terminating fax machine.

This feature is enabled using the T38FaxSessionImmediateStart parameter. The No-Op packets are enabled using the NoOpEnable and NoOpInterval parameters.

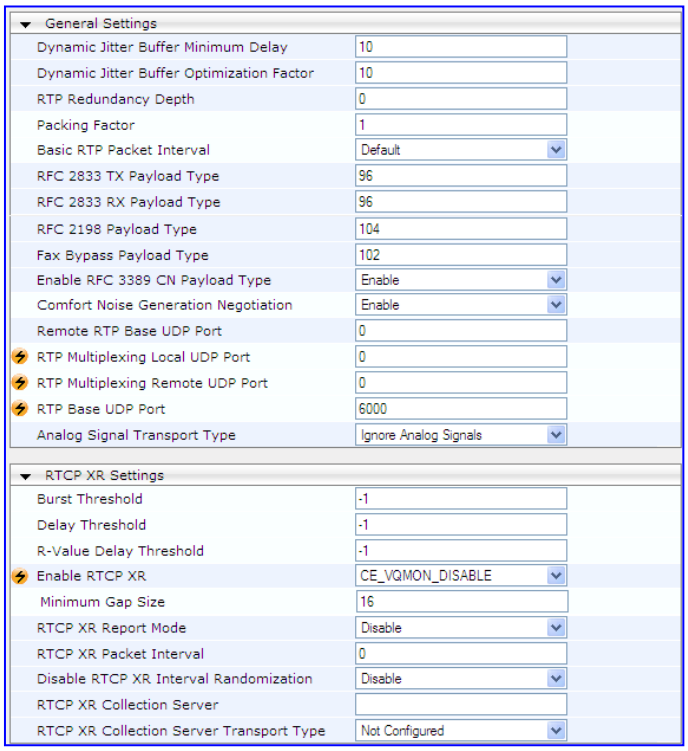
13.3 Configuring RTP/RTCP Settings

The RTP/RTCP Settings page configures the Real-Time Transport Protocol (RTP) and Real-Time Transport (RTP) Control Protocol (RTCP) parameters. For a detailed description of the parameters appearing on this page, refer to 'Configuration Parameters Reference' on page 752.

➢ **To configure the RTP/RTCP parameters:**

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **RTP/RTCP Settings**).

Figure 13-4: RTP/RTCP Settings Page



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, refer to 'Saving Configuration' on page 680.

13.3.1 Dynamic Jitter Buffer Operation

Voice frames are transmitted at a fixed rate. If the frames arrive at the other end at the same rate, voice quality is perceived as good. In many cases, however, some frames can arrive slightly faster or slower than the other frames. This is called jitter (delay variation), and degrades the perceived voice quality. To minimize this problem, the device uses a jitter buffer. The jitter buffer collects voice packets, stores them and sends them to the voice processor in evenly spaced intervals.

The device uses a dynamic jitter buffer that can be configured using the following parameters:

- **Minimum delay:** DJBufMinDelay (0 msec to 150 msec)
  Defines the starting jitter capacity of the buffer. For example, at 0 msec, there is no buffering at the start. At the default level of 10 msec, the device always buffers incoming packets by at least 10 msec worth of voice frames.
- **Optimization Factor:** DJBufOptFactor (0 to 12, 13)
  Defines how the jitter buffer tracks to changing network conditions. When set at its maximum value of 12, the dynamic buffer aggressively tracks changes in delay (based on packet loss statistics) to increase the size of the buffer and doesn't decay back down. This results in the best packet error performance, but at the cost of extra delay. At the minimum value of 0, the buffer tracks delays only to compensate for clock drift and quickly decays back to the minimum level. This optimizes the delay performance but at the expense of a higher error rate.

The default settings of 10 msec Minimum delay and 10 Optimization Factor should provide a good compromise between delay and error rate. The jitter buffer 'holds' incoming packets for 10 msec before making them available for decoding into voice. The coder polls frames from the buffer at regular intervals in order to produce continuous speech. As long as delays in the network do not change (jitter) by more than 10 msec from one packet to the next, there is always a sample in the buffer for the coder to use. If there is more than 10 msec of delay at any time during the call, the packet arrives too late. The coder tries to access a frame and is not able to find one. The coder must produce a voice sample even if a frame is not available. It therefore compensates for the missing packet by adding a Bad-Frame-Interpolation (BFI) packet. This loss is then flagged as the buffer being too small. The dynamic algorithm then causes the size of the buffer to increase for the next voice session. The size of the buffer may decrease again if the device notices that the buffer is not filling up as much as expected. At no time does the buffer decrease to less than the minimum size configured by the Minimum delay parameter.

For certain scenarios, the **Optimization Factor is set to 13**: One of the purposes of the Jitter Buffer mechanism is to compensate for clock drift. If the two sides of the VoIP call are not synchronized to the same clock source, one RTP source generates packets at a lower rate, causing under-runs at the remote Jitter Buffer. In normal operation (optimization factor 0 to 12), the Jitter Buffer mechanism detects and compensates for the clock drift by occasionally dropping a voice packet or by adding a BFI packet.

Fax and modem devices are sensitive to small packet losses or to added BFI packets. Therefore, to achieve better performance during modem and fax calls, the Optimization Factor should be set to 13. In this special mode the clock drift correction is performed less frequently - only when the Jitter Buffer is completely empty or completely full. When such condition occurs, the correction is performed by dropping several voice packets simultaneously or by adding several BFI packets simultaneously, so that the Jitter Buffer returns to its normal condition.

The procedure below describes how to configure the jitter buffer using the Web interface.

➢ **To configure jitter buffer using the Web interface:**

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **RTP/RTCP Settings**).
2. Configure the following parameters:
   - 'Dynamic Jitter Buffer Minimum Delay' (*DJBufMinDelay*) - Defines the minimum delay (in msec) for the Dynamic Jitter Buffer.
   - 'Dynamic Jitter Buffer Optimization Factor' (*DJBufOptFactor*) - Defines the Dynamic Jitter Buffer frame error/delay optimization factor.
3. Click **Submit** to apply your settings.

The procedure below describes how to configure the jitter buffer using CLI.

➢ **To configure jitter buffer using CLI:**

1. Access the VoIP configuration section of the CLI:
   ```
   # (configure-voip) media voice-processing
   ```
2. Configure the following parameters:
   - **`jitter-buffer-minimum-delay`** Defines the minimum delay (in msec) for the Dynamic Jitter Buffer.
   - **`jitter-buffer-optimization-factor`** Defines the Dynamic Jitter Buffer frame error/delay optimization factor.

13.3.2 Comfort Noise Generation

The device can generate artificial background noise ("comfort" noise) in the voice channel during periods of silence (i.e. no party is speaking). This is useful in that it reassures the calling parties that the call is still connected. The device detects silence using its Voice Activity Detection feature. When the CNG is enabled and silence is detected, the device transmits Silence Identifier Descriptors (SIDs) parameters to reproduce the local background noise at the remote (receiving) side.

The Comfort Noise Generation (CNG) support also depends on the silence suppression (SCE) setting for the coder used in the voice channel. For more information, see the description of the CNG-related parameters.

The procedure below describes how to configure CNG using the Web interface.

➢ **To configure CNG using the Web interface:**

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **RTP/RTCP Settings**).
2. Set the 'Comfort Noise Generation' (*ComfortNoiseNegotiation*) parameter to **Enable**.
3. Click **Submit** to apply your changes.

The procedure below describes how to configure the CNG using CLI.

➢ **To configure CNG using CLI:**

1. Access the VoIP configuration section of the CLI:
   ```
   # (configure-voip) media voice-processing
   ```
2. Configure the following parameter:
   - **`com-noise-gen-nego`** Enables negotiation of CNG.

13.3.3 Dual-Tone Multi-Frequency Signaling

13.3.3.1 Configuring DTMF Transport Types

The device supports various methods to transport DTMF digits over the IP network to the remote endpoint. These methods and their configuration are described below:

- **Using INFO message according to Nortel IETF draft:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0
  - TxDTMFOption = 1

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0).
- **Using INFO message according to Cisco's mode:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0
  - TxDTMFOption = 3

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0).
- **Using NOTIFY messages according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01**: DTMF digits are carried to the remote side using NOTIFY messages. To enable this mode, define the following:
  - RxDTMFOption = 0
  - TxDTMFOption = 2

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0).
- **Using RFC 2833 relay with Payload type negotiation:** DTMF digits are carried to the remote side as part of the RTP stream in accordance with RFC 2833 standard. To enable this mode, define the following:
  - RxDTMFOption = 3
  - TxDTMFOption = 4

  Note that to set the RFC 2833 payload type with a different value (other than its default), configure the RFC2833PayloadType parameter. The device negotiates the RFC 2833 payload type using local and remote SDP and sends packets using the payload type from the received SDP. The device expects to receive RFC 2833 packets with the same payload type as configured by the RFC2833PayloadType parameter. If the remote side doesn't include 'telephony-event' in its SDP, the device sends DTMF digits in transparent mode (as part of the voice stream).
- **Sending DTMF digits (in RTP packets) as part of the audio stream (DTMF Relay is disabled):** This method is typically used with G.711 coders; with other low-bit rate (LBR) coders, the quality of the DTMF digits is reduced. To enable this mode, define the following:
  - RxDTMFOption = 0 (i.e., disabled)
  - TxDTMFOption = 0 (i.e., disabled)
  - DTMFTransportType = 2 (i.e., transparent)
- **Using INFO message according to Korea mode:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0 (i.e., disabled)
  - TxDTMFOption = 3

Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0).

Notes:

- The device is always ready to receive DTMF packets over IP in all possible transport modes: INFO messages, NOTIFY, and RFC 2833 (in proper payload type) or as part of the audio stream.
- To exclude RFC 2833 Telephony event parameter from the device's SDP, set RxDTMFOption to 0 in the *ini* file.

The following parameters affect the way the device handles the DTMF digits:

- TxDTMFOption, RxDTMFOption, RFC2833TxPayloadType, and RFC2833RxPayloadTypeand
- MGCPDTMFDetectionPoint, DTMFVolume, DTMFTransportType, DTMFDigitLength, and DTMFInterDigitInterval

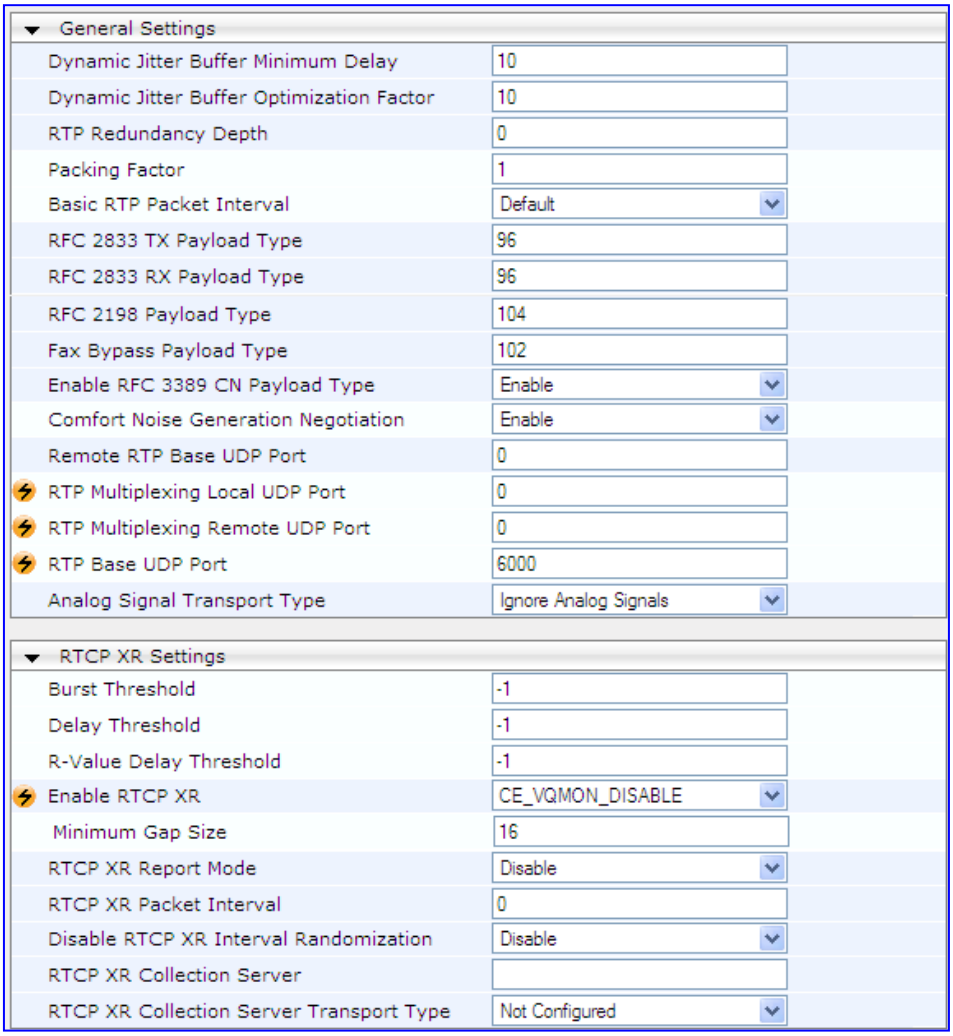
13.3.3.2 Configuring RFC 2833 Payload

The procedure below describes how to configure the RFC 2833 payload using the Web interface:

➢ **To configure RFC 2833 payload using the Web interface:**

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **RTP/RTCP Settings**).

Figure 13-5: RTP/RTCP Settings Page



2. Configure the following parameters:
   - 'RTP Redundancy Depth' (RTPRedundancyDepth) - enables the device to generate RFC 2198 redundant packets.
   - 'Enable RTP Redundancy Negotiation' (EnableRTPRedundancyNegotiation) - enables the device to included the RTP redundancy dynamic payload type in the SDP, according to RFC 2198.
   - 'RFC 2198 Payload Type' (RFC2198PayloadType) - defines the RTP redundancy packet payload type according to RFC 2198.
   - 'RFC 2833 TX Payload Type' (RFC2833TxPayloadType) - defines the Tx RFC 2833 DTMF relay dynamic payload type.

- 'RFC 2833 RX Payload Type' (RFC2833RxPayloadType) - defines the Rx RFC 2833 DTMF relay dynamic payload type.

3. Click **Submit** to apply your settings.

13.3.4 Configuring RTP Base UDP Port

You can configure the range of UDP ports for RTP, RTCP, and T.38. The UDP port range can be configured using media realms in the Media Realm table, allowing you to assign different port ranges (media realms) to different interfaces. However, if you do not use media realms, you can configure the lower boundary of the UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2), using the 'RTP Base UDP Port' (BaseUDPport) parameter. For example, if the Base UDP Port is set to 6000, then one channel may use the ports RTP 6000, RTCP 6001, and T.38 6002, while another channel may use RTP 6010, RTCP 6011, and T.38 6012. The range of possible UDP ports is 6,000 to 64,000 (default base UDP port is 6000).

The port range is calculated using the 'RTP Base UDP Port' (BaseUDPport) parameter as follows: **BaseUDPPort to (BaseUDPPort + <channels -1> * 10)**

The maximum (when all channels are required) UDP port range is calculated as follows:

- BaseUDPport to (BaseUDPport + 329*10) - for example, if the BaseUDPPort is set to 6,000, then the UDP port range is 6,000 to 9,290

Notes:

- The device allocates the UDP ports randomly to the channels.
- If you are using Media Realms (see 'Configuring Media Realms' on page 181), the port range configured for the Media Realm must be within this range defined by the BaseUDPPort parameter.

The procedure below describes how to configure the RTP base UDP port using the Web interface.

➢ **To configure the RTP base UDP port:**

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **RTP/RTCP Settings**).
2. Set the 'RTP Base UDP Port' parameter to the required value.
3. Click **Submit**.
4. Reset the device for the settings to take effect.

13.3.5 RTP Control Protocol Extended Reports (RTCP XR)

RTP Control Protocol Extended Reports (RTCP XR) is a VoIP management control that defines a set of metrics containing information for assessing VoIP call quality and for diagnosing problems. RTCP XR (RFC 3611) extends the RTCP reports defined in RFC 3550 by providing additional VoIP metrics. RTCP XR information publishing is implemented in the device according to <draft-johnston-sipping-rtcp-summary-07>. This draft defines how a SIP User Agent (UA) publishes the detailed information to a defined collector. RTCP XR measures VoIP call quality such as packet loss, delay, signal / noise / echo levels, estimated R-factor, and mean opinion score (MOS). RTCP XR measures these parameters using metrics (refer to the Product Reference Manual).

RTCP XR messages containing key call-quality-related metrics are exchanged periodically (user-defined) between the device and the SIP UA. This allows an analyzer to monitor these metrics midstream, or a device to retrieve them using SNMP. The device can send RTCP XR reports to an Event State Compositor (ESC) server using PUBLISH messages.

These reports can be sent at the end of each call (configured using RTCPXRReportMode) and according to a user-defined interval (RTCPInterval or DisableRTCPRandomize) between consecutive reports.

To enable RTCP XR reporting, the VQMonEnable ini file parameter must be set to 1. In addition, the device must be installed with the appropriate Software Upgrade Key.

The procedure below describes how to configure RTCP XR using the Web interface.

➢ **To configure RTCP XR:**

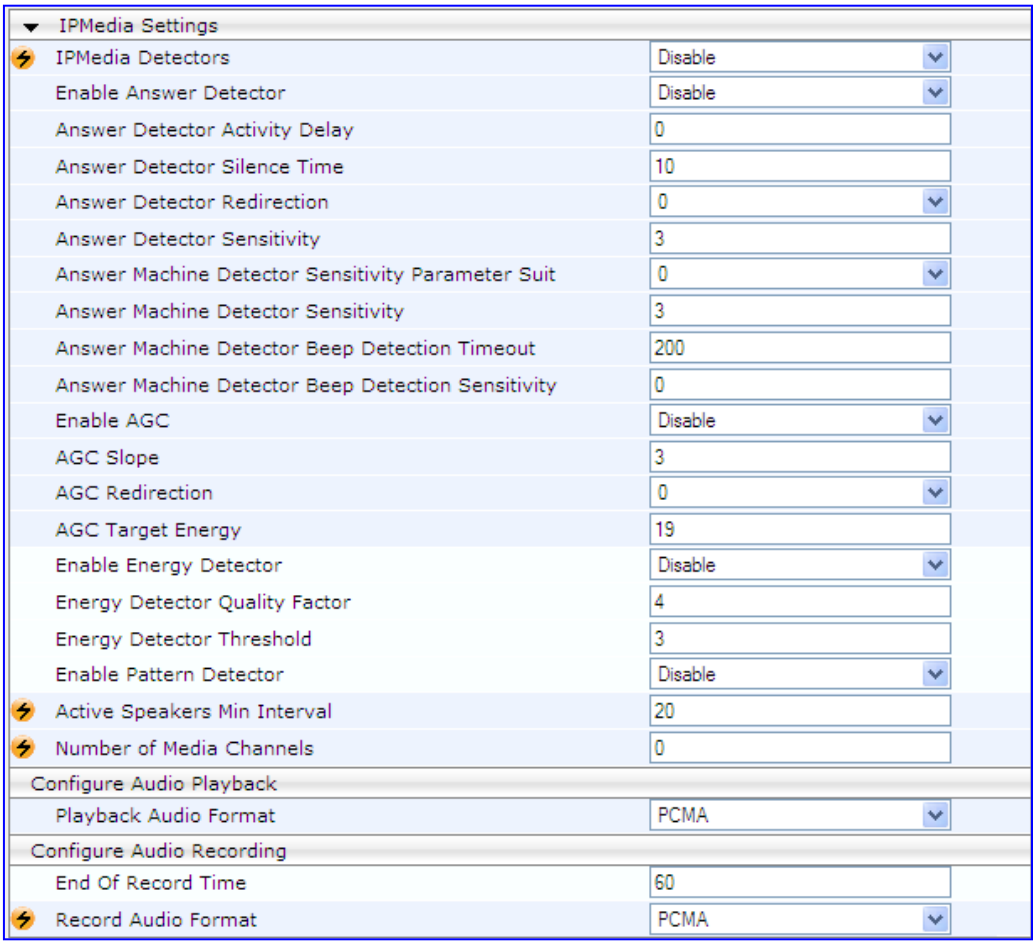
1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **RTP/RTCP Settings**).
2. Configure the following parameters:
   - 'Enable RTCP XR' (*VQMonEnable*) - enables voice quality monitoring and RTCP XR.
   - 'Minimum Gap Size' (*VQMonGMin*) - defines the voice quality monitoring - minimum gap size (number of frames).
   - 'Burst Threshold' (*VQMonBurstHR*) - defines the voice quality monitoring - excessive burst alert threshold.
   - 'Delay Threshold' (*VQMonDelayTHR*) - defines the voice quality monitoring - excessive delay alert threshold.
   - 'R-Value Delay Threshold' (*VQMonEOCRValTHR*) - defines the voice quality monitoring - end of call low quality alert threshold.
   - 'RTCP XR Packet Interval' (*RTCPInterval*) - defines the time interval between adjacent RTCP reports.
   - 'Disable RTCP XR Interval Randomization' (*DisableRTCPRandomize*) - determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval.
   - 'RTCP XR Collection Server Transport Type' (*RTCPXRESCTransportType*) - determines the transport layer for outgoing SIP dialogs initiated by the device to the RTCP-XR Collection Server.
   - 'RTCP XR Collection Server' (*RTCPXREscIP*) - defines the IP address of the Event State Compositor (ESC).
   - 'RTCP XR Report Mode' (*RTCPXRReportMode*) - determines whether RTCP XR reports are sent to the ESC and defines the interval in which they are sent.
3. Click **Submit**.
4. Reset the device for the settings to take effect.

13.4 Configuring IP Media Settings

The IPMedia Settings page allows you to configure the IP media parameters. For a detailed description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

➢ **To configure the IP media parameters:**

1. Open the IPMedia Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **IPMedia Settings**).



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

13.4.1 Answer Machine Detector (AMD)

The device provides answering machine detection (AMD) capabilities that can detect for example, if a human voice or an answering machine is answering the call. AMD is useful for automatic dialing applications.

The device supports up to four AMD parameter suites, where each parameter suite defines the AMD sensitivity levels of detection. The detection sensitivity levels can range from 0 to 15, depending on parameter suite. The level is selected using the AMDSensitivityLevel parameter. The Parameter Suite(s) can be loaded to the device in the Web interface as an auxiliary file (see 'Loading Auxiliary Files' on page 681) or loaded remotely through the ini file (using the AMDSensitivityFileName and AMDSensitivityFileUrl parameters). In addition,

you can configure AMD per call, based on the called number or Trunk Group. This is achieved by defining the AMD parameters for a specific IP Profile (IPProfile parameter) and then assigning the IP Profile to a Trunk Group in the Inbound IP Routing table (PSTNPrefix parameter).

The device also supports the detection of beeps at the end of an answering machine message. This allows users of third-party, Application servers to leave voice messages after an answering machine plays a "beep" sound.

The device supports two methods for detecting and reporting beeps (configured using the AMDBeepDetectionMode parameter):

- Using the AMD detector. This detector is integrated in the existing AMD feature. The beep detection timeout and beep detection sensitivity are configurable using the AMDBeepDetectionTimeout and AMDBeepDetectionSensitivity parameters respectively.
- Using the Call Progress Tone detector - several beep tones (Tone Type #46) can be configured in the CPT file.

The detection of beeps is done using the X-Detect header extension. The device sends a SIP INFO message containing one of the following field values:

- Type=AMD and SubType=Beep
- Type=CPT and SubType=Beep

Upon every AMD activation, the device can send a SIP INFO message to an Application server notifying it of one of the following:

- Human voice has been detected
- Answering machine has been detected
- Silence (i.e., no voice detected) has been detected

The table below shows the success rates of the AMD feature for correctly detecting live and fax calls:

Table 13-1: Approximate AMD Detection Normal Sensitivity (Based on North American English)

AMD Detection Sensitivity	Performance	
	Success Rate for Live Calls	Success Rate for Answering Machine
0 (Best for Answering Machine)	-	-
1	82.56%	97.10%
2	85.87%	96.43%
3 (Default)	88.57%	94.76%
4	88.94%	94.31%
5	90.42%	91.64%
6	90.66%	91.30%
7 (Best for Live Calls)	94.72%	76.14%

Table 13-2: Approximate AMD Detection High Sensitivity (Based on North American English)

AMD Detection Sensitivity	Performance	
	Success Rate for Live Calls	Success Rate for Answering Machine
0 (Best for Answering Machine)	72%	97%
1	77%	96%
2	79%	95%
3	80%	95%
4	84%	94%
5	86%	93%
6	87%	92%
7	88%	91%
8 (default)	90%	89%
9	90%	88%
10	91%	87%
11	94%	78%
12	94%	73%
13	95%	65%
14	96%	62%
15 (Best for Live Calls)	97%	46%

A pre-requisite for enabling the AMD feature is to set the *ini* file parameter EnableDSPIPMDetectors to 1. In addition, to enable voice detection, required once the AMD detects the answering machine, the *ini* file parameter EnableVoiceDetection must be set to 1.

Note: The device's AMD feature is based on voice detection for North American English. If you want to implement AMD in a different language or region, you must provide AudioCodes with a database of recorded voices in the language on which the device's AMD mechanism can base its voice detector algorithms for detecting these voices. The data needed for an accurate calibration should be recorded under the following guidelines:

- Statistical accuracy: The number of recordings should be large (i.e., about 100) and varied. The calls must be made to different people, at different times. The calls must be made in the specific location in which the device's AMD mechanism is to operate.
- Real-life recording: The recordings should simulate real-life answering of a person picking up the phone without the caller speaking (until the AMD decision).
- Normal environment interferences: The environment should almost simulate real-life scenarios, i.e., not sterile but not too noisy either. Interferences, for example, could include background noises of other people talking, spikes, and car noises.

The SIP call flows below show an example of implementing the device's AMD feature. This scenario example allows a third-party Application server to play a recorded voice message to an answering machine.

1. Upon detection by the device of the answering machine, the device sends a SIP INFO message to the Application server:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac1566945480
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c1505895240
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29758@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.40A.040.004
Content-Type: application/x-detect
Content-Length: 30
Type= AMD
SubType= AUTOMATA
```

2. The device then detects the start of voice (i.e., the greeting message of the answering machine), and then sends the following to the Application server:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c419779142
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29753@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.40A.040.004
Content-Type: application/x-detect
Content-Length: 34
Type= PTT
SubType= SPEECH-START
```

3. Upon detection of the end of voice (i.e., end of the greeting message of the answering machine), the device sends the Application server the following:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c419779142
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29753@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.40A.040.004
Content-Type: application/x-detect
Content-Length: 34
Type= PTT
SubType= SPEECH-END
```

4. The Application server now sends its message to the answering message.

If the device detects voice and not an answering machine, the SIP INFO message includes:

```
Type= AMD
SubType= VOICE
```

If the device detects silence, the SIP INFO message includes the SubType **SILENT**.

13.4.2 Automatic Gain Control (AGC)

Automatic Gain Control (AGC) adjusts the energy of the output signal to a required level (volume). This feature compensates for near-far gain differences. AGC estimates the energy of the incoming signal (from the IP or PSTN, determined by the parameter AGCRedirection), calculates the essential gain, and then performs amplification. Feedback ensures that the output signal is not clipped. You can define the required Gain Slope in decibels per second (using the parameter AGCGainSlop) and the required signal energy threshold (using the parameter AGCTargetEnergy).

When the AGC first detects an incoming signal, it begins operating in Fast Mode, which allows the AGC to adapt quickly when a conversation starts. This means that the Gain Slope is 8 dB/sec for the first 1.5 seconds. After this period, the Gain Slope is changed to the user-defined value. You can disable or enable the AGC's Fast Mode feature, using the *ini* file parameter AGCDisableFastAdaptation. After Fast Mode is used, the signal should be off for two minutes in order to have the feature turned on again.

Note: The AGC feature requires that the device be installed with the **IP Media Detectors** Feature Key

The procedure below describes how to configure AGC using the Web interface:

➢ **To configure AGC using the Web interface:**

1. Open the IPMedia Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **IPMedia Settings**).
2. Configure the following parameters:
   - 'Enable AGC' (*EnableAGC*) - Enables the AGC mechanism.
   - 'AGC Slope' (*AGCGainSlope*) - Determines the AGC convergence rate.
   - 'AGC Redirection' (*AGCRedirection*) - Determines the AGC direction.
   - 'AGC Target Energy' - Defines the signal energy value (dBm) that the AGC attempts to attain.
3. Click **Submit** to apply your settings.

Note: The following additional parameters can be configured using either the EMS or ini file:

- AGCMinGain - Defines the minimum gain (in dB) by the AGC when activated
- AGCMaxGain - Defines the maximum gain (in dB) by the AGC when activated.
- AGCDisableFastAdaptation - Enables the AGC Fast Adaptation mode

The procedure below describes how to configure AGC using CLI.

➢ **To configure AGC using CLI:**

1. Access the VoIP configuration section of the CLI:
```
# (configure-voip) media voice-processing
```
2. Configure the following parameters:
   - **AGC-disable-fast-adaptation:** Disables the AGC FastAdaptation mode.
   - **AGC-enable:** Activates the AGC.
   - **AGC-gain-slope**: Determines the AGC convergence rate.
   - **AGC-max-gain:** Defines the maximum signal gain of the AGC **[dB].**
   - **AGC-min-gain:** Defines the minimum signal gain of the AGC **[dB].**
   - **AGC-redirection:** Redirects the AGC output towards the Tel instead of the network.
   - **AGC-target-energy:** Determines the target signal energy level of the AGC **[-dBm**]

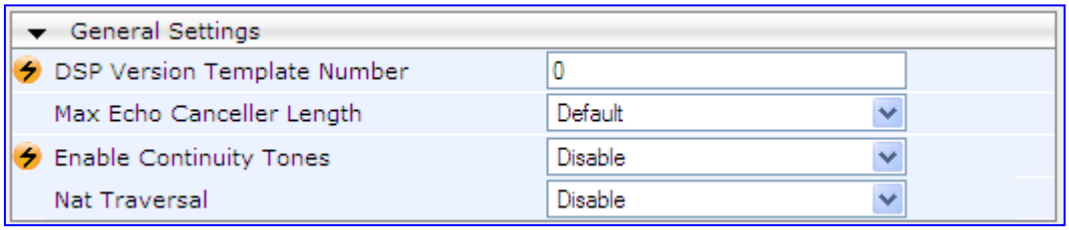
13.5 Configuring General Media Settings

The General Media Settings page allows you to configure various media parameters. For a detailed description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

➢ **To configure general media parameters:**

1. Open the General Media Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **General Media Settings**).

Figure 13-6: General Media Settings Page



2. Configure the parameters as required.
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

13.6 Configuring Analog Settings

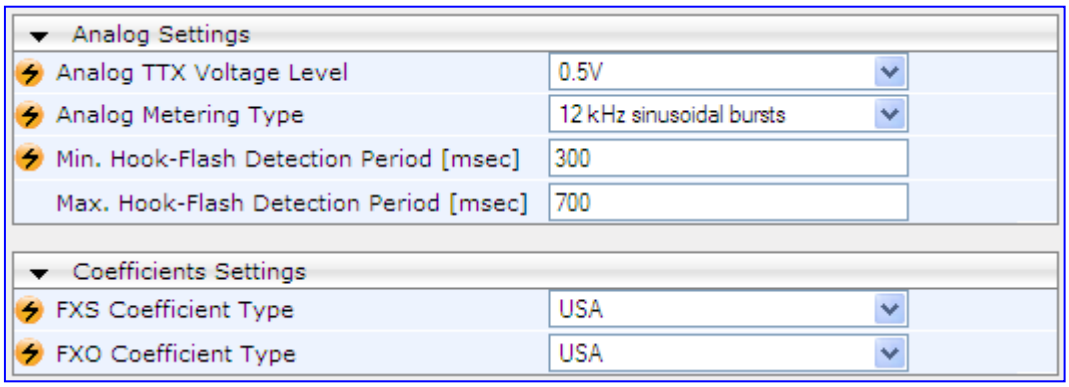
The Analog Settings page allows you to configure various analog parameters. For a detailed description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

This page also selects the type (USA or Europe) of FXS and/or FXO coefficient information. The FXS coefficient contains the analog telephony interface characteristics such as DC and AC impedance, feeding current, and ringing voltage.

➢ **To configure the analog parameters:**

1. Open the Analog Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Analog Settings**).

Figure 13-7: Analog Settings Page



2. Configure the parameters as required.
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

13.7 Configuring DSP Templates

The DSP Templates page allows you to load up to two DSP templates to the device. In addition, you can define the percentage of DSP resources allocated per DSP template.

➢ **To select DSP templates:**

1. Open the DSP Templates page (**Configuration** tab > **VoIP** menu > **Media** submenu > **DSP Templates**).
2. In the 'Add Index' field, enter the index number to add a new row in the table.
3. In the 'DSP Template Number' field, enter the desired DSP template number.
4. In the 'DSP Resources Percentage' field, enter the desired resource percentage for the specified template.
5. Click **Apply** to save your settings.
6. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Notes:

- You must either use the 'DSP Templates page or the DSPVersionTemplateNumber parameter to select the DSP template, not both. The DSP Templates page must be used only when two concurrent DSP templates are required; the DSPVersionTemplateNumber parameter must be used only when a single template is used.
- If no entries are defined, the device uses the default DSP template (i.e., Template 0).
- For supported DSP templates, see 'DSP Templates' on page 1061.
- For configuring the Web interface's tables, see 'Working with Tables' on page 62.

13.8 DSP Channel Resources for Transcoding

In addition to the DSP resources (channels) required for PRI trunk interfaces, the device provides flexibility in making DSP resources readily available for SBC and IP-to-IP functionality. This is achieved by obtaining DSP resources from the Media Processing Module (MPM) and by "borrowing" DSP resources from the PRI interfaces (i.e., TRUNKS module). DSP channels are required for the following:

- **IP-to-IP application:** Each IP-to-IP call session includes two legs, requiring two DSP resources.
- **SBC application:** Typically, the SBC application does not require DSP resources. However, it does need DSP resources for transcoding, where it uses two DSP resources.

Note: In some scenarios, the IP-to-IP application does not require DSP resources.

To support these capabilities and to allow optimal management of the required DSP resources, you need to ensure suitable hardware and software configuration, as described below:

- **Hardware Configuration:** The device can obtain DSP resources using one of the following hardware configurations:
  - **Media Processing Module (MPM) Modules:** Provide DSP resources for the SBC application, the IP-to-IP application, and/or IP media for IP media functionality (such as IP-based conferencing). The device can house up to three MPM modules. The DSP resources allocation is as follows:
    - Without Conferencing: When the MPM modules are installed in chassis slots #3, #4, and #5, the device supports up to 120 DSP resources, where each module provides up to 40 DSP resources.
    - With Conferencing: When the MPM modules are installed in chassis slots #4, #5, and #6, the device supports up to 100 DSP resources (up to 60 conference participants). These channels are allocated as follows:
      - ✓ MPM module in Slot #6 provides 20 channels (and enables conferencing)
      - ✓ MPM modules in slots #4 and #5 each provide 40 channels (i.e., total of 80 channels)

Notes:

- If the device is installed with all three MPM modules, no other telephony interface module can be installed in the device.
- It is recommended not to install an MPM module in Sot #6 as this reduces the available channels for transcoding to 20 (instead of 40).

  - **TRUNKS (PRI) Modules:** DSP resources can be obtained from installed TRUNKS modules. In other words, some trunk span interfaces are "disabled".

Note: DSP resources cannot be "borrowed" from span interfaces that use CAS.

- **Combination of MPM and PRI Modules:** DSP resources can be obtained from the MPM and TRUNKS modules (i.e., some span interfaces are "disabled"). For example, to achieve 120 channels (with conferencing), you need to use two MPM modules (installed in slots #5 and #6) and one TRUNKS module providing two PRI spans (installed in Slot #1).

■ **Software Configuration:** The device must be configured as follows:

- Set the EnableSBCApplication parameter to 1. This setting enables the SBC application, if required.
- Set the Enable IP2IPApplication parameter to 1. This setting enables the IP-to-IP application, if required.
- Set the MediaChannels parameter to the maximum number of required IP media channels, regardless of the module from where the channels are obtained.

Note: Setting the MediaChannels parameter to a value that is greater than the available DSP resources from the MPM module(s) can result in "stealing" of DSP resources from the B-channels of the PRI spans.

- Set the IPmediaChannels parameter to the number of DSP channels that you want to "borrow" (use) from each TRUNKS module for the SBC application and IP-to-IP application. The setting below shows an example of borrowing DSP channels from TRUNKS modules #1 and #2:

```
[IPMediaChannels]
FORMAT IPMediaChannels_Index = IPMediaChannels_ModuleID,
IPMediaChannels_DSPChannelsReserved;
IPMediaChannels 1 = 1, 15;
IPMediaChannels 2 = 2, 10;
[\IPMediaChannels]
```

Notes:

- To borrow DSP channels from the TRUNKS modules, you must also set the EnableIPMediaChannels parameter to 1.
- The value of IPMediaChannels_DSPChannelsReserved must be in multiples of 5.
- By default, the MPM module is set to the maximum number of IP media channels. Therefore, there is no need to define it with the IPmediaChannels parameter.
- By default, a TRUNKS module is set to 0 (i.e., no IP media channels).

13.9 Configuring Media Realms

The Media Realm Table page allows you to define a pool of up to 64 SIP media interfaces, termed *Media Realms*. Media Realms allow you to divide a Media-type interface (defined in the Multiple Interface table - see 'Configuring IP Interface Settings' on page 117) into several realms, where each realm is specified by a UDP port range. In addition, you can define the maximum number of sessions per Media Realm. Once created, Media Realms can be assigned to IP Groups (in the IP Group table - see 'Configuring IP Groups' on page 212) or SRDs (in the SRD table - see 'Configuring SRD Table' on page 207).

For each Media Realm you can configure Quality of Experience parameters and their thresholds for reporting to the AudioCodes SEM server used for monitoring the quality of

calls. For configuring this, see 'Configuring Quality of Experience Parameters per Media Realm' on page 185.

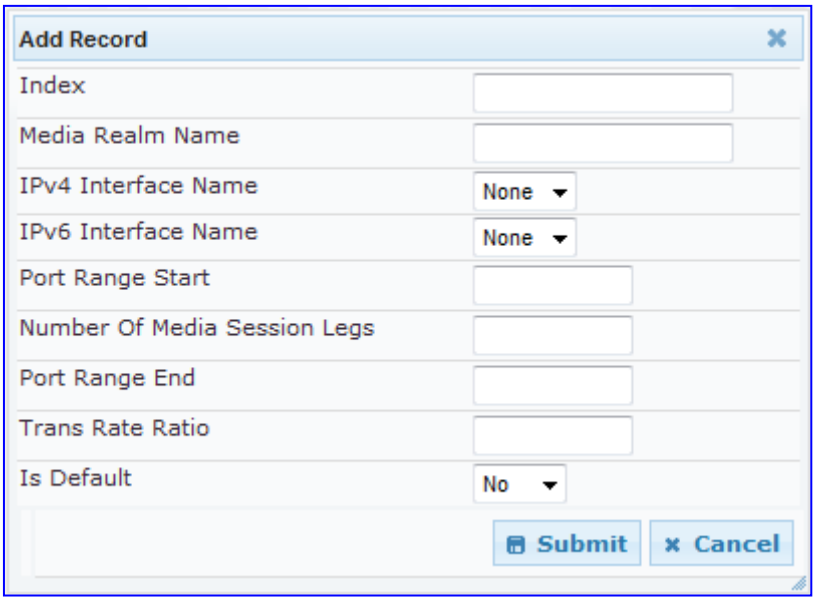
Notes:

- If different Media Realms are assigned to an IP Group and to an SRD, the IP Group's Media Realm takes precedence.
- For this setting to take effect, a device reset is required.
- You can also configure the Media Realm table using the *ini* file table parameter CpMediaRealm.

➢ **To define a Media Realm:**

1. Open the Media Realm Table page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Media Realm Configuration**).
2. Click the **Add** button; the following appears:

Figure 13-8: Add Record Dialog Box



3. Configure the parameters as required. See the table below for a description of each parameter
4. Click **Submit** to apply your settings.
5. Reset the device to save the changes to flash memory (see 'Saving Configuration' on page 680).

Table 13-3: Media Realm Table Parameter Descriptions

Parameter	Description
Index **[CpMediaRealm_Index]**	Defines the required table index number.
Media Realm Name **[CpMediaRealm_MediaRealmName]**	Defines an arbitrary, identifiable name for the Media Realm. The valid value is a string of up to 40 characters. **Notes:** ▪ This parameter is mandatory. ▪ The name assigned to the Media Realm must be unique. ▪ This Media Realm name is used in the SRD and IP

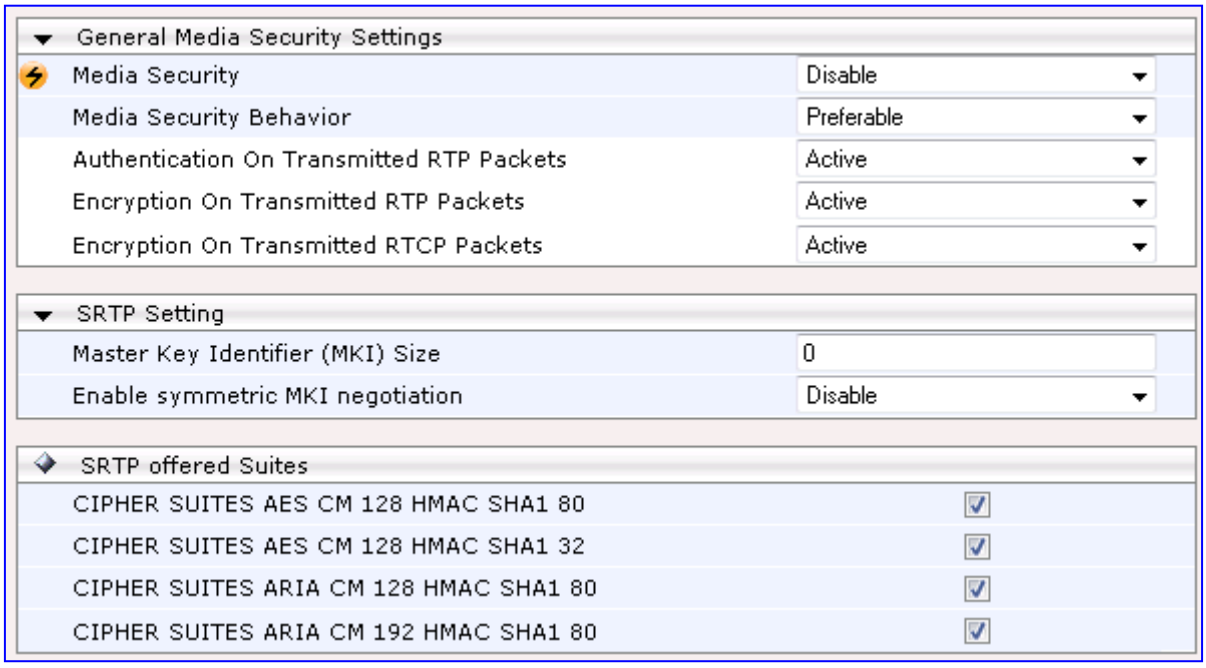
Parameter	Description
	Groups table.
IPv4 Interface Name **[CpMediaRealm_IPv4IF]**	Assigns an IPv4 interface to the Media Realm. **Note:** The name of this interface must be identical (i.e., case-sensitive etc.) as configured in the Multiple Interface table (InterfaceTable parameter). For the VoIP WAN IP address, you must enter the string "WAN" (case-sensitive).
Port Range Start **[CpMediaRealm_PortRangeStart]**	Defines the starting port for the range of Media interface UDP ports. **Notes:** ▪ You must either configure all media realms with port ranges or without (not some with and some without). ▪ The available UDP port range is calculated using the BaseUDPport parameter: ✓ BaseUDPport to BaseUDPport + 329*10 ▪ Port ranges over 60,000 must not be used. ▪ Ranges of Media Realm ports must not overlap.
Number of Media Session Legs **[CpMediaRealm_MediaSessionLeg]**	Defines the number of media sessions associated with the range of ports. This is the number of media sessions available in the port range. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10.
Port Range End **[CpMediaRealm_PortRangeEnd]**	Read-only field displaying the ending port for the range of Media interface UDP ports. This field is calculated by adding the 'Media Session Leg' field (multiplied by the port chunk size) to the 'Port Range Start' field. A value appears once a row has been successfully added to the table.
Trans Rate Ratio **[CpMediaRealm_TransRateRatio]**	**Note:** This field will be supported in the next applicable release.
Is Default **[CpMediaRealm_IsDefault]**	Defines the Media Realm as the default Media Realm. This default Media Realm is used when no Media Realm is configured for an IP Group or SRD for a specific call. ▪ **[0]** No (default) ▪ **[1]** Yes **Notes:** ▪ This parameter can be set to Yes (1) for only **one** defined Media Realm. ▪ If this parameter is not configured, then the first Media Realm in the table is used as default. ▪ If the table is not configured, then the default Media Realm includes all the configured media interfaces.

13.10 Configuring Media Security

The Media Security page allows you to configure media security. For a detailed description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

➢ **To configure media security:**

1. Open the Media Security page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Media Security**).



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

13.11 Configuring Quality of Experience Parameters per Media Realm

For each Media Realm, you can configure Quality of Experience (QoE). The QoE feature enables you to monitor and analyze media and signaling traffic, allowing you to detect problems causing service degradation. The device saves call information and statistics at call start, call end, or specific changes in the call. The information is stored as call records on an external server. The device connects to the server using TLS over TCP (as a client).

For each Media Realm you can specify the call parameters to monitor and configure the upper and lower thresholds, that when exceeded, the device reports the changes in these parameters to the monitoring server. The device can monitor the following parameters:

- Loss
- MOS
- Jitter
- Delay

At any given time during an active call, each of these parameters can be in one of the following states according to its value in the last RTCP / RTCP XR packet:

- Gray - indicates that the value is unknown
- Green - indicates good call quality
- Yellow - indicates medium call quality
- Red - indicates poor call quality

The mapping between the values of the parameters and the color is according to the configured threshold for these parameters, per Media Realm. The call itself also has a state (color), which is the worst-state color of all the monitored parameters. Every time a color of a parameter changes, a report is sent to the external server. In addition to this, a report is sent at the end of each call.

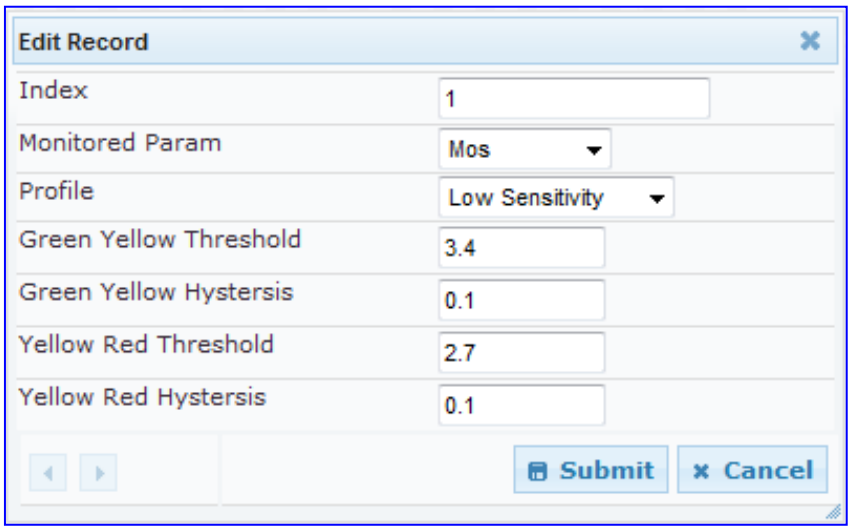
Notes:

- The QoE feature is available only if the device is installed with the relevant Software Upgrade Key.
- To configure the address of the AudioCodes Session Experience Manager (SEM) server to where the device reports the QoE, see 'Configuring Server for Media Quality of Experience' on page 188.
- You can also use the QOERules *ini* file parameter to configure QoE per Media Realm.

➢ **To configure Quality of Experience per Media Realm:**

1. Open the Media Realm Table page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Media Realm Configuration**).
2. Select the Media Realm for which you want to configure Quality of Experience, and then click the **Quality Of Experience** link; the Quality Of Experience page appears:
3. Click the **Add** button; the Add Record dialog box appears:

Figure 13-9: Add Record Dialog Box for QoE



The figure above shows value thresholds for the MOS parameter, which are assigned using pre-configured values of the Low Sensitivity profile. In this example setting, if the MOS value changes by 0.1 (hysteresis) to 3.3 or 3.5, the device sends a report to the SEM indicating this change. If the value changes to 3.3, it sends a yellow state (i.e., medium quality); if the value changes to 3.5, it sends a green state.

4. Configure the parameters as required. See the table below for a description of each parameter.
5. Click **Submit** to apply your settings.

Table 13-4: Quality of Experience for Media Realm Parameter Descriptions

Parameter	Description
Index **[QOERules_RuleIndex]**	Defines the table index entry. Up to four entries can be configured per Media Realm.
Monitored Param **[QOERules_MonitoredParam]**	Defines the parameter to monitor and report. ▪ Mos (default) ▪ Delay ▪ PacketLoss ▪ Jitter

Parameter	Description
Profile **[QOERules_Profile]**	Defines the pre-configured threshold profile to use. ▪ No Profile = No profile is used and you need to define the thresholds in the parameters described below. ▪ Low Sensitivity = Automatically sets the thresholds to low sensitivity values. Therefore, reporting is done only if changes in parameters' values is significant. ▪ Default Sensitivity = Automatically sets the thresholds to a medium sensitivity. ▪ High Sensitivity = Automatically sets the thresholds to high sensitivity values. Therefore, reporting is done for small fluctuations in parameters' values.
Green Yellow Threshold **[QOERules_GreenYellowThreshold]**	Defines the parameter threshold values between green (good quality) and yellow (medium quality) states.
Green Yellow Hystersis **[QOERules_GreenYellowHystersis]**	Defines the hysteresis (fluctuation) for the green-yellow threshold. When the threshold is exceeded by this hysteresis value, the device sends a report to the SEM indicating this change.
Yellow Red Threshold **[QOERules_YellowRedThreshold]**	Defines the parameter threshold values between yellow (medium quality) and red (poor quality). When this threshold is exceeded, the device sends a report to the SEM indicating this change.
Yellow Red Hystersis **[QOERules_YellowRedHystersis]**	Defines the hysteresis (fluctuation) for the yellow-red threshold. When the threshold is exceeded by this hystersis value, the device sends a report to the SEM indicating this change.

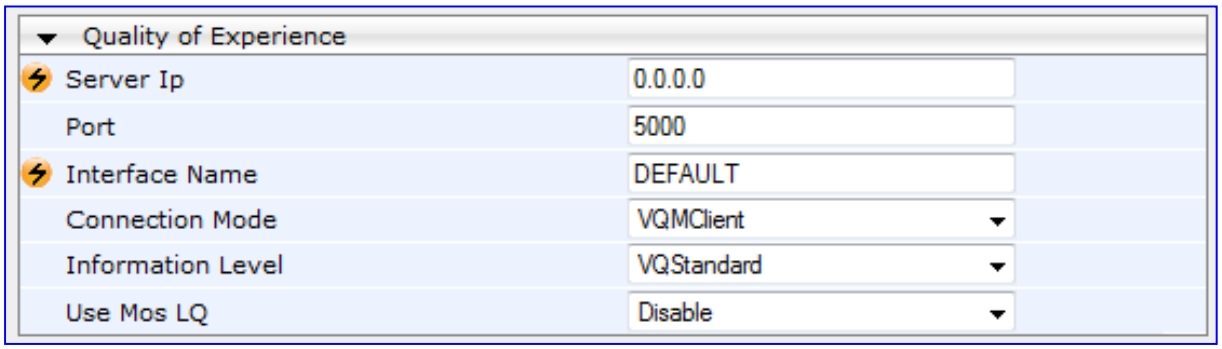
13.12 Configuring Server for Media Quality of Experience

The device can be configured to report voice (media) quality of experience to AudioCodes Session Experience Manager (SEM) server, a plug-in for AudioCodes EMS. The reports include real-time metrics of the quality of the actual call experience and processed by the SEM.

➢ **To configure QoE reporting of media:**

1. Open the Media Quality of Experience page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Media Quality of Experience**).

Figure 13-10: Media Quality of Experience Page



2. Configure the parameters as required
   - 'Server Ip' (QOEServerIP) - defines the IP address of the SEM server
   - 'Port' (QOEPort) - defines the port of the SEM server
   - 'Interface Name' (QOEInterfaceName) - defines the device's IP network interface on which the SEM reports are sent
   - 'Use Mos LQ' (QOEUseMosLQ) - defines the reported MOS type (listening or conversational)
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Notes:

- To support this feature, the device must be installed with the relevant Software Upgrade Feature Key.
- To configure the parameters to report and their thresholds per Media Realm, see 'Quality of Experience per Media Realm' on page 185.
- For information on the SEM server, refer to the *EMS User's Manual*.

14 Services

This section describes configuration for various supported services.

14.1 Routing Based on LDAP Active Directory Queries

The device supports Lightweight Directory Access Protocol (LDAP), enabling call routing decisions based on information stored on a third-party LDAP server (or Microsoft's Active Directory™ enterprise directory server). This feature enables the usage of a single common, popular database to manage and maintain information regarding user's availability, presence, and location.

Note: The LDAP feature is supported only by the GW/IP2IP application.

14.1.1 Configuring the LDAP Server

The basic LDAP mechanism is described below:

- **Connection:** The device connects and binds to the remote LDAP server either during the service's initialization (at device start-up) or whenever the LDAP server's IP address and port is changed. Service makes 10 attempts to connect and bind to the remote LDAP server with a timeout of 20 seconds between attempts. If connection fails, the service remains in disconnected state until either the LDAP server's IP address or port is changed.

  If connection to the LDAP server later fails, the service attempts to reconnect, as described previously. The SNMP alarm acLDAPLostConnection is sent when connection is broken. Upon successful reconnection, the alarm is cleared.

  Binding to the LDAP server can be anonymous or not. For anonymous binding, the LDAPBindDN and LDAPPassword parameters must not be defined or set to an empty string.

  The address of the LDAP server can be a DNS name (using the LDAPServerName parameter) or an IP address (using the LDAPServerIP parameter).
- **Search:** To run a search using the LDAP service, the path to the directory's subtree where the search is to be done must be defined (using the LDAPSearchDN parameter). In addition, the search key (known as "filter" in LDAP references), which defines the exact DN to be found and one or more attributes whose values should be returned, must be defined. The device supports up to 20 LDAP search requests.

  If connection to the LDAP server is disrupted during the search, all search requests are dropped and an alarm indicating a failed status is sent to client applications.
- **CLI:** The LDAP CLI is located in the directory IPNetworking\OpenLdap. The following commands can be used:
  - LdapSTatus - displays connection status
  - LdapSearch - searches an LDAP server
  - LDapOpen - opens connection to the LDAP server using parameters provided in configuration file
  - LDapSetDebugmode - sets the LdapDebugLevelMode parameter
  - LDapGetDebugmode – gets the LdapDebugLevelMode parameter value

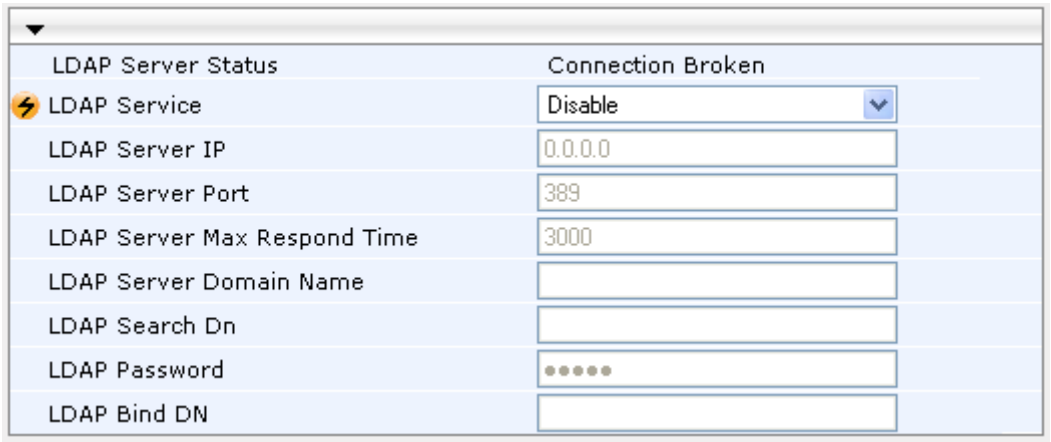
Relevant parameters: LDAPServiceEnable; LDAPServerIP; LDAPServerDomainName; LDAPServerPort; LDAPPassword; LDAPBindDN; LDAPSearchDN; LDAPDebugMode; LDAPServerMaxRespondTime.

The LDAP Settings page is used for configuring the LDAP server parameters. For a full description of these parameters, see 'Configuration Parameters Reference' on page 752.

➢ **To configure the LDAP server parameters:**

1. Open the LDAP Settings page (**Configuration** tab > **VoIP** menu > **Services** submenu > **LDAP Settings**).

Figure 11: LDAP Settings Page



The read-only 'LDAP Server Status' field displays one of the following possibilities:

- "Not Applicable"
- "Connection Broken"
- "Connecting"
- "Connected"

2. Configure the parameters as required.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see "Saving Configuration" on page 680.

14.1.2 AD-Based Tel-to-IP Routing in Microsoft Lync

Typically, enterprises wishing to deploy Microsoft® Lync™ Server 2010 (formerly known as Office Communication Server 2007) are faced with a complex, call routing dial plan when migrating users from their existing PBX or IP PBX to the Lync Server 2010 platform. As more and more end-users migrate to the new voice system, dialing plan management and PBX link capacity can be adversely impacted. To resolve this issue, enterprises can employ Microsoft's Active Directory (AD), which provides a central database to manage and maintain information regarding user's availability, presence, and location.

The device supports Tel-to-IP call routing decisions based on information stored on the AD. Based on queries sent to the AD, the device can route the Tel call to one of the following IP domains:

- Private - private telephone line for Lync users (in addition to the primary telephone line)
- Lync client (formally OCS) - users connected to Lync Server 2010 through the Mediation Server
- PBX or IP PBX - users not yet migrated to Lync Server 2010
- Mobile – mobile telephone number

14.1.2.1 Querying the AD and Routing Priority

The device queries the AD using the initial destination number (i.e., called number). The query can return up to four user phone numbers, each pertaining to one of the IP domains (i.e., private number, Lync / OCS number, PBX / IP PBX number, and mobile number). The configuration parameters listed in the table below are used to configure the query attribute keys that defines the AD attribute that you wish to query in the AD:

Parameters for Configuring Query Attribute Key

Parameter	Queried User Domain (Attribute) in AD	Query Result Example
MSLDAPPBXNumAttributeName	PBX or IP PBX number (e.g., "telephoneNumber" - default)	telephoneNumber= +3233554447
MSLDAPOCSNumAttributeName	Mediation Server / Lync client number (e.g., "msRTCSIP-line")	msRTCSIP-line=john.smith@company.com
MSLDAPMobileNumAttributeName	Mobile number (e.g., "mobile")	Mobile=+3247647156
MSLDAPPrivateNumAttributeName	Any attribute (e.g., "msRTCSIP-PrivateLine") **Note:** Used only if set to same value as Primary or Secondary key.	msRTCSIP-PrivateLine= +3233554480
MSLDAPPrimaryKey	Primary Key query search instead of PBX key - can be any AD attribute	msRTCSIP-PrivateLine= +3233554480
MSLDAPSecondaryKey	Secondary Key query key search if Primary Key fails - can be any attribute	-

The process for querying the AD and subsequent routing based on the query results is as follows:

1. Queries the AD:
   - **If Primary Key is not configured:** Queries AD for the destination number in the "PBX" attribute name, configured by the MSLDAPPBXNumAttributeName parameter.
   - **If Primary Key is configured:**
     - If Primary Key does not equal private attribute value: Queries AD for the destination number in an AD attribute configured by the MSLDAPPrimaryKey parameter. For example, if you set this parameter to "mobile", the query searches for a user whose "mobile" attribute value equals the original dialed destination number.
     - If Primary Key or Secondary Key value equals the Private Key value: Queries AD for the destination number in the private attribute name, configured by the MSLDAPPrivateNumAttributeName parameter. (For example, the private key will be used if both the MSLDAPPrivateNumAttributeName and MSLDAPPrimaryKey parameters are set to "msRTCSIP-PrivateLine").
2. If the query is not found:
   - **If Secondary Key is configured:** Does a second query for the destination number using a second AD attribute key name, configured by the MSLDAPSecondaryKey parameter.

- **If Secondary Key is not configured:** Routes the call to the original dialed destination number according to the routing rule matching the "LDAP_ERR" destination prefix number value, or rejects the call with a SIP 404 "Not Found" response.

3. If the query is found: The AD returns up to four attributes - Lync / OCS, PBX / IP PBX, private (only if it equals Primary or Secondary key), and mobile.
4. The device adds unique prefix keywords to the query results in order to identify the query type (i.e., IP domain). These prefixes are used as the prefix destination number value in the Outbound IP Routing table to denote the IP domains:
   - "PRIVATE" (PRIVATE:<private_number>): used to match a routing rule based on query results of the private number (MSLDAPPrivateNumAttributeName)
   - "OCS" (OCS:<Lync_number>): used to match a routing rule based on query results of the Lync client number (MSLDAPOCSNumAttributeName)
   - "PBX" (PBX:<PBX_number>): used to match a routing rule based on query results of the PBX / IP PBX number (MSLDAPPBXNumAttributeName)
   - "MOBILE" (MOBILE:<mobile_number>): used to match a routing rule based on query results of the mobile number (MSLDAPMobileNumAttributeName)
   - "LDAP_ERR": used to match a routing rule based on a failed query result when no attribute is found in the AD

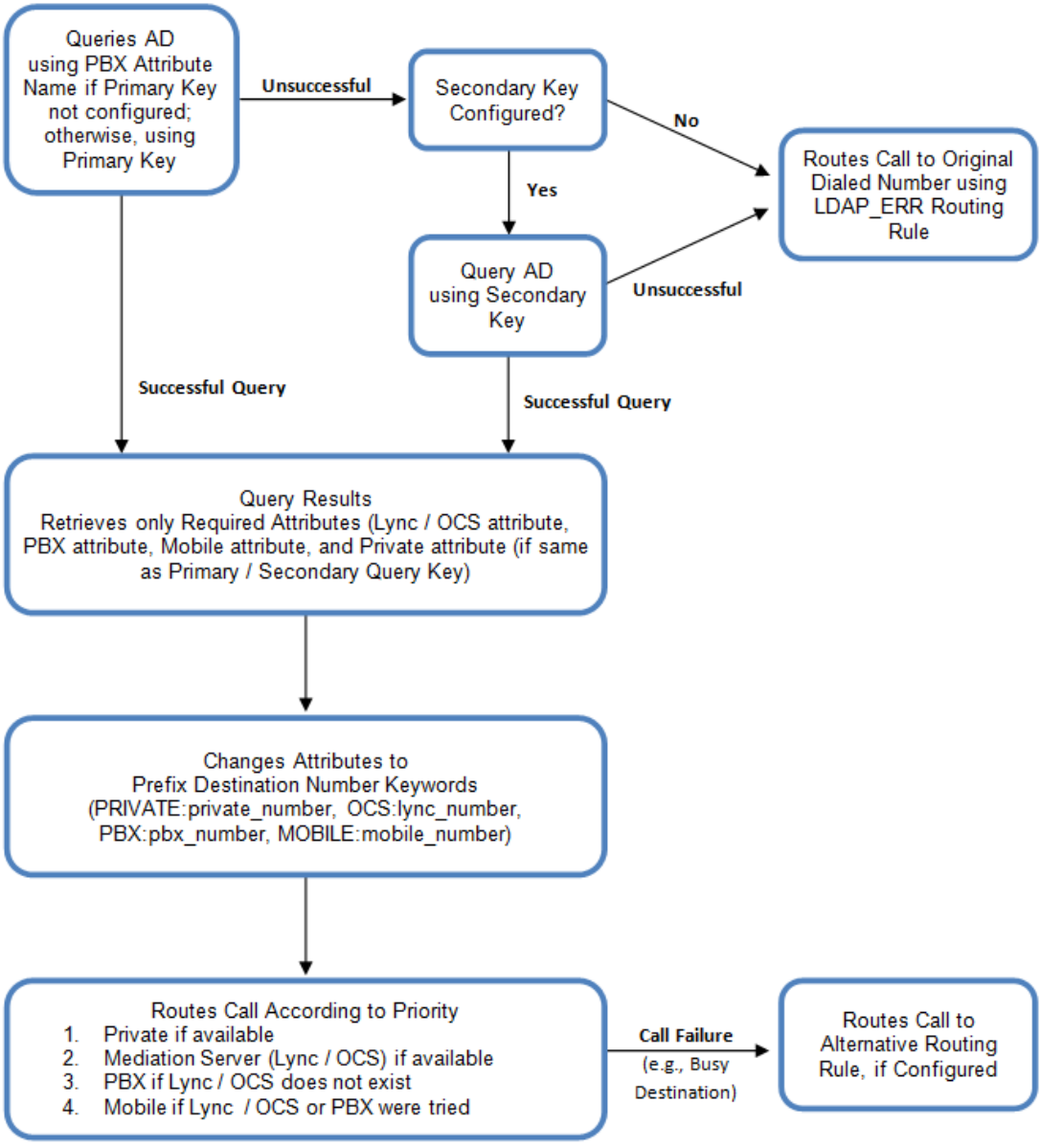
Note: These prefixes are involved only in the routing and manipulation processes; they are not used as the final destination number.

5. The device uses the Outbound IP Routing table to route the call based on the LDAP query result. The device routes the call according to the following priority:
   1. **Private line:** If the query is done for the private attribute and it's found, then the device routes the call according to this attribute.
   2. **Mediation Server SIP address (Lync / OCS):** If the private attribute does not exist or is not queried, then the device routes the call to the Mediation Server (which then routes the call to the Lync client).
   3. **PBX / IP PBX:** If the Lync / OCS client is not found in the AD, it routes the call to the PBX / IP PBX.
   4. **Mobile number:** If the Lync / OCS client (or Mediation Server) is unavailable (e.g., SIP response 404 "Not Found" upon INVITE sent to Lync client), and the PBX / IP PBX is also unavailable, then the device routes the call to the user's mobile number (if exists in the AD).
   5. **Alternative route:** If the call routing to all the above fails (e.g., due to unavailable destination - call busy), the device can route the call to an alternative destination if an alternative routing rule is configured.
   6. **"Redundant" route:** If the query failed (i.e., no attribute found in the AD), the device uses the routing rule matching the "LDAP_ERR" prefix destination number value.

Note: For Enterprises implementing a PBX / IP PBX system, but yet to migrate to Lync Server 2010, if the PBX / IP PBX system is unavailable or has failed, the device uses the AD query result for the user's mobile phone number, routing the call through the PSTN to the mobile destination.

The flowchart below summarizes the device's process for querying the AD and routing the call based on the query results:

Figure 12: LDAP Query Flowchart



14.1.2.2 Configuring AD-Based Routing Rules

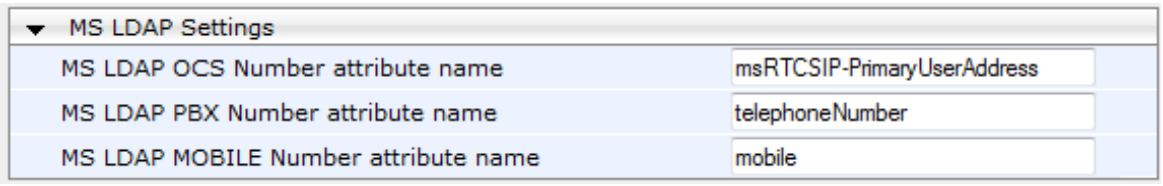
The procedure below describes how to configure Tel-to-IP routing based on LDAP queries.

➢ **To configure LDAP-based Tel-to-IP routing for Lync Server 2010:**

1. Configure the LDAP server parameters, as described in 'Configuring the LDAP Server' on page 189.
2. Configure the AD attribute names used in the LDAP query:

a. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Advanced Parameters**).

Figure 13: LDAP Parameters for Microsoft Lync Server 2010



b. Configure the LDAP attribute names as desired.

3. Configure AD-based Tel-to-IP routing rules:

a. Open the Outbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** > **Tel to IP Routing**). For more information, see "Configuring Outbound IP Routing Table" on page 291.

b. Configure query-result routing rules for each IP domain (private, PBX / IP PBX, Lync / OCS clients, and mobile), using the "LDAP" keywords for the prefix destination number (case-sensitive):

- **PRIVATE:** Private number
- **OCS:** Lync / OCS client number
- **PBX:** PBX / IP PBX number
- **MOBILE:** Mobile number
- **LDAP_ERR:** LDAP query failure

c. Configure a routing rule for routing the initial Tel call to the LDAP server, using the value "LDAP" for denoting the IP address of the LDAP server.

Note: The routing rule for sending the initial incoming Tel call to the LDAP server must be configured in a row located below the query result rules. This is done to prevent the looping of calls to the LDAP server. The value used for denoting the matching prefix destination number is typically "*" (i.e., all dialed destination numbers).

d. For alternative routing, enable the alternative routing mechanism and configure corresponding SIP reasons for alternative routing. For this feature, alternative routing always starts again from the top of the table (first routing rule entry) and not from the next row.

The figure below shows an example for configuring AD-based routing rules in the Outbound IP Routing Table:

Figure 14: Active Directory-based Routing Rules in Outbound IP Routing Table

Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Port
PRIVATE:			10.33.45.60	
PBX:			10.33.45.65	
OCS:			10.33.45.68	
MOBILE:			10.33.45.100	
LDAP_ERR:			10.33.45.80	
*			LDAP	
*			10.33.45.72	

- **Rule #1:** Sends call to private telephone line (at 10.33.45.60) upon successful AD query result for the private attribute.
- **Rule #2:** Sends call to IP PBX (at 10.33.45.65) upon successful AD query result for the PBX attribute.
- **Rule #3:** Sends call to Lync client (i.e., Mediation Server at 10.33.45.68) upon successful AD query result for the Lync attribute.
- **Rule #4:** Sends call to user's mobile phone number (to PSTN through the device's IP address at 10.33.45.100) upon successful AD query result for the Mobile attribute.
- **Rule #5:** Sends call to IP address of device (10.33.45.80) if AD query failure (e.g., no response from LDAP server or attribute not found).
- **Rule #6:** Sends query for original destination number of received Tel call to the LDAP server.
- **Rule #7:** Alternative routing rule that sends the call of original dialed number to IP destination 10.33.45.72. This rule is applied in any of the following cases
  - LDAP functionality is disabled.
  - LDAP query is successful but call fails (due to, for example, busy line) to all the relevant attribute destinations (private, Lync, PBX, and mobile), and a relevant Tel-to-IP Release Reason has been configured.

Once the device receives the incoming Tel call, the first rule that it uses is rule #6, which queries the AD server. When the AD replies, the device searches the table from the first rule down for the matching destination phone prefix (i.e., "PRIVATE:, "PBX:", "OCS:", "MOBILE:", and "LDAP_ERR:"), and then sends the call to the appropriate destination.

14.2 Least Cost Routing

This section provides a description of the device's least cost routing (LCR) feature and how to configure it.

Note: The LCR feature is applicable only to the GW/IP2IP application.

14.2.1 Overview

The LCR feature enables the device to choose the outbound IP destination routing rule based on lowest call cost. This is useful in that it enables service providers to optimize routing costs for customers. For example, you may wish to define different call costs for local and international calls, or different call costs for weekends and weekdays (specifying even the time of call). The device sends the calculated cost of the call to a Syslog server (as Information messages), thereby enabling billing by third-party vendors.

LCR is implemented by defining Cost Groups and assigning them to routing rules in the Outbound IP Routing table. The device searches this routing table for matching routing rules, and then selects the rule with the lowest call cost. If two routing rules have identical costs, then the rule appearing higher up in the table is used (i.e., first-matched rule). If a selected route is unavailable, the device selects the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched rules with Cost Groups. This is determined according to the settings of the Default Cost parameter in the Routing Rule Groups table.

The Cost Group defines a fixed connection cost (*connection cost*) and a charge per minute (*minute cost*). Cost Groups can also be configured with time segments (*time bands*), which define connection cost and minute cost based on specific days of the week and time of day (e.g., from Saturday through Sunday, between 6:00 and 18:00). If multiple time bands are configured per Cost Group and a call spans multiple time bands, the call cost is calculated using only the time band in which the call was initially established.

In addition to Cost Groups, the device can calculate the call cost using an optional, user-defined average call duration value. The logic in using this option is that a Cost Group may be cheap if the call duration is short, but due to its high minute cost, may prove very expensive if the duration is lengthy. Thus, together with Cost Groups, the device can use this option to determine least cost routing. The device calculates the Cost Group call cost as follows: Total Call Cost = Connection Cost + (Minute Cost * Average Call Duration).

The below table shows an example of call cost when taking into consideration call duration. This example shows four defined Cost Groups and the total call cost if the average call duration is 10 minutes:

Table 14-1: Call Cost Comparison between Cost Groups for different Call Durations

Cost Group	Connection Cost	Minute Cost	Total Call Cost per Duration	
			1 Minute	10 Minutes
A	1	6	7	61
B	0	10	10	100
C	0.3	8	8.3	80.3
D	6	1	7	**16**

If four matching routing rules are located in the routing table and each one is assigned a different Cost Group as listed in the table above, then the rule assigned Cost Group "D" is selected. Note that for one minute, Cost Groups "A" and "D" are identical, but due to the average call duration, Cost Group "D" is cheaper. Therefore, average call duration is an important factor in determining the cheapest routing role.

Below are a few examples of how you can implement LCR:

- **Example 1:** This example uses two different Cost Groups for routing local calls and international calls:

  Two Cost Groups are configured as shown below:

Table 14-2: Configured Cost Groups for Local and International Calls

Cost Group	Connection Cost	Minute Cost
1. "Local Calls"	2	1
2. "International Calls"	6	3

The Cost Groups are assigned to routing rules for local and international calls in the Outbound IP Routing table:

Table 14-3: Cost Groups Assigned to Outbound IP Routing Rules for Local and International Calls

Routing Index	Dest Phone Prefix	Destination IP	Cost Group ID
1	2000	x.x.x.x	1 "Local Calls"
2	00	x.x.x.x	2 "International Calls"

- **Example 2:** This example shows how the device determines the cheapest routing rule in the Outbound IP Routing table:

  The Default Cost parameter (global) in the Routing Rule Groups table is set to **Min**, meaning that if the device locates other matching LCR routing rules (with Cost Groups assigned), the routing rule without a Cost Group is considered the lowest cost route.

- The following Cost Groups are configured:

Table 14-4: Configured Cost Groups

Cost Group	Connection Cost	Minute Cost
1. "A"	2	1
2. "B"	6	3

- The Cost Groups are assigned to routing rules in the Outbound IP Routing table:

Table 14-5: Cost Groups Assigned to Outbound IP Routing Rules

Routing Index	Dest Phone Prefix	Destination IP	Cost Group ID
1	201	x.x.x.x	"A'
2	201	x.x.x.x	"B"
3	201	x.x.x.x	0
4	201	x.x.x.x	"B"

The device calculates the optimal route in the following index order: 3, 1, 2, and then 4, due to the following logic:

- Index 1 - Cost Group "A" has the lowest connection cost and minute cost
- Index 2 - Cost Group "B" takes precedence over Index 4 entry based on the first-matched method rule
- Index 3 - no Cost Group is assigned, but as the Default Cost parameter is set to **Min**, it is selected as the cheapest route
- Index 4 - Cost Group "B" is only second-matched rule (Index 1 is the first)

- **Example 3:** This example shows how the cost of a call is calculated if the call spans over multiple time bands:

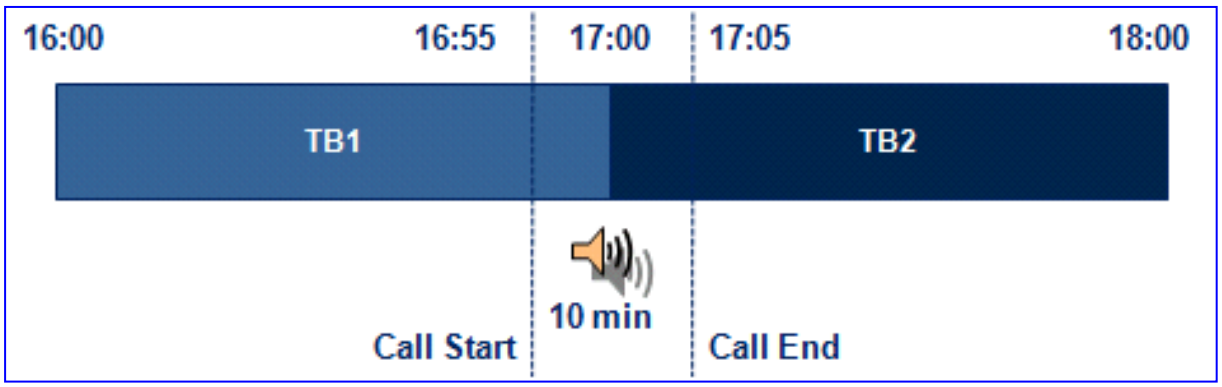
  Assume a Cost Group, "CG Local" is configured with two time bands, as shown below:

Table 14-6: Cost Group with Multiple Time Bands

Cost Group	Time Band	Start Time	End Time	Connection Cost	Minute Cost
CG Local	TB1	16:00	17:00	2	1
	TB2	17:00	18:00	7	2

Assume that the call duration is 10 minutes, occurring between 16:55 and 17:05. In other words, the first 5 minutes occurs in time band "TB1" and the next 5 minutes occurs in "TB2", as shown below:

Figure 14-15: LCR using Multiple Time Bands (Example)



The device calculates the call using the time band in which the call was initially established, regardless of whether the call spans over additional time bands:

Total call cost = "TB1" Connection Cost + ("TB1" Minute Cost x call duration) = 2 + 1 x 10 min = 12

14.2.2 Configuring LCR

The following main steps need to be done to configure LCR:

1. Enable the LCR feature and configure the average call duration and default call connection cost - see 'Enabling the LCR Feature' on page 200.
2. Configure Cost Groups - see 'Configuring Cost Groups' on page 201.
3. Configure Time Bands for a Cost Group - see 'Configuring Time Bands for Cost Groups' on page 202.
4. Assign Cost Groups to outbound IP routing rules - see 'Assigning Cost Groups to Routing Rules' on page 204.

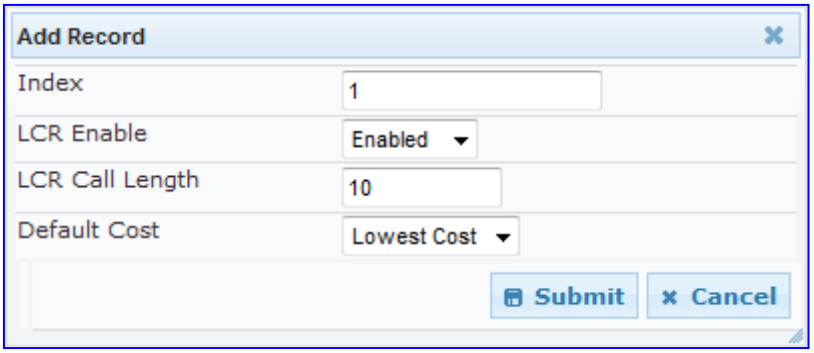
14.2.2.1 Enabling the LCR Feature

The procedure below describes how to enable the LCR feature. This also includes configuring the average call duration and default call cost for routing rules that are not assigned Cost Groups in the Outbound IP Routing table.

➢ **To enable LCR:**

1. Open the Routing Rule Groups Table page (**Configuration** tab > **VoIP** menu > **Services** submenu > **Least Cost Routing** > **Routing Rule Groups Table**).
2. Click the **Add** button; the Add Record dialog box appears:

Figure 14-16: Routing Rule Groups Table - Add Record



3. Configure the parameters as required. For a description of the parameters, see the table below.
4. Click **Submit**; the entry is added to the Routing Rule Groups table.

Table 14-7: Routing Rule Groups Table Description

Parameter	Description
Index **[RoutingRuleGroups_Index]**	Defines the table index entry. **Note:** Only one index entry can be configured.
LCR Enable **[RoutingRuleGroups_LCREnable]**	Enables the LCR feature: ▪ **[0]** Disable (default) ▪ **[1]** Enable
LCR Call Length **[RoutingRuleGroups_LCRAverageCallLength]**	Defines the average call duration (in minutes) and is used to calculate the variable portion of the call cost. This is useful, for example, when the average call duration spans over multiple time bands. The LCR is calculated as follows: cost = call connect cost + (minute cost * average call duration) The valid value range is 0-65533. the default is 1. For example, assume the following Cost Groups: ▪ "Weekend A": call connection cost is 1 and charge per minute is 6. Therefore, a call of 1 minute cost 7 units. ▪ "Weekend_ B": call connection cost is 6 and charge per minute is 1. Therefore, a call of 1 minute cost 7 units. Therefore, for calls under one minute, "Weekend A" carries the lower cost. However, if the average call duration is more than one minute, then "Weekend B" carries the lower cost.
Default Cost **[RoutingRuleGroups_LCRDefaultCost]**	Determines whether routing rules in the Outbound IP Routing table without an assigned Cost Group are considered a higher cost or lower cost route compared to other matched routing rules

Parameter	Description
	that are assigned Cost Groups. ▪ **[0]** Min = If the device locates other matching LCR routing rules, this routing rule is considered the lowest cost route and therefore, it is selected as the route to use (default.) ▪ **[1]** Max = If the device locates other matching LCR routing rules, this routing rule is considered as the highest cost route and therefore, is not used or used only if the other cheaper routes are unavailable. **Note:** If more than one valid routing rule without a defined Cost Group exists, the device selects the first-matched rule.

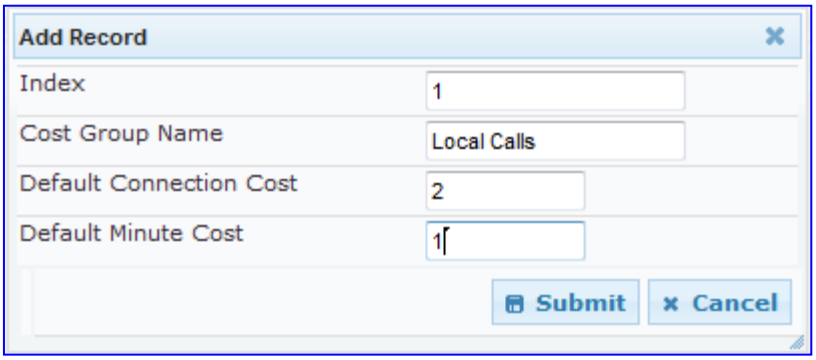
14.2.2.2 Configuring Cost Groups

The procedure below describes how to configure Cost Groups. Cost Groups are defined with a fixed call connection cost and a call rate (charge per minute). Once configured, you can configure Time Bands for each Cost Group. Up to 10 Cost Groups can be configured.

➢ **To configure Cost Groups:**

1. Open the Cost Group Table page (**Configuration** tab > **VoIP** menu > **Services** submenu > **Least Cost Routing** > **Cost Group Table**).
2. Click the **Add** button; the Add Record dialog box appears:

Figure 14-17: Cost Groups - Add Record



3. Configure the parameters as required. For a description of the parameters, see the table below.
4. Click **Submit**; the entry is added to the Cost Group table.

Table 14-8: Cost Group Table Description

Parameter	Description
Index **[CostGroupTable_Index]**	Defines the table index entry.
Cost Group Name **[CostGroupTable_CostGroupName]**	Defines an arbitrary name for the Cost Group. The valid value is a string of up to 30 characters. **Note:** Each Cost Group must have a unique name.
Default Connect Cost **[CostGroupTable_Defa**	Defines the call connection cost (added as a fixed charge to the call) for a call outside the time bands.

Parameter	Description
ultConnectionCost]	The valid value range is 0-65533. The default is 0. **Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default connection cost is used.
Default Time Cost **[CostGroupTable_DefaultMinuteCost]**	Defines the call charge per minute for a call outside the time bands. The valid value range is 0-65533. The default is 0. **Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default charge per minute is used.

14.2.2.3 Configuring Time Bands for Cost Groups

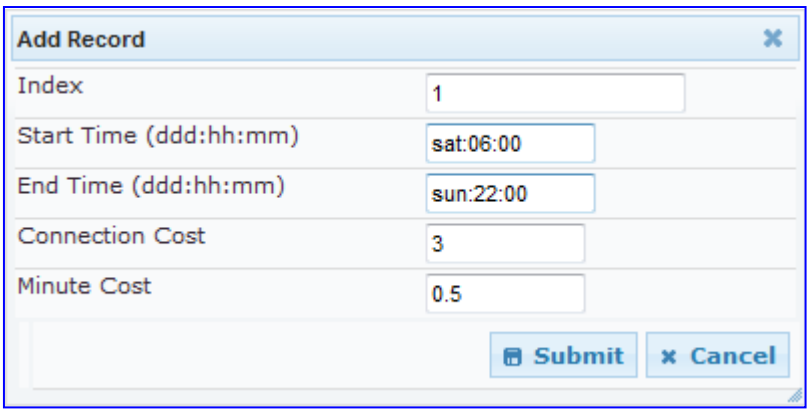
The procedure below describes how to configure Time Bands for a Cost Group. The time band defines the day and time range for which the time band is applicable (e.g., from Saturday 05:00 to Sunday 24:00) as well as the fixed call connection charge and call rate per minute for this interval. Up to 70 time bands can be configured, and up to 21 time bands can be assigned to each Cost Group.

Note: You cannot define overlapping time bands.

➢ **To configure Time Bands for a Cost Group:**

1. Open the Cost Group Table page (**Configuration** tab > **VoIP** menu > **Services** submenu > **Least Cost Routing** > **Cost Group Table**).
2. Select a Cost Group for which you want to assign Time Bands, and then click the **Time Band** link located below the table; the Time Band table for the selected Cost Group appears.
3. Click the **Add** button; the Add Record dialog box appears:

Figure 14-18: Time Bands - Add Record



4. Configure the parameters as required. For a description of the parameters, see the table below.
5. Click **Submit**; the entry is added to the Time Band table for the relevant Cost Group.

Table 14-9: Time Band Table Description

Parameter	Description
Index **[CostGroupTimebands_TimebandIndex]**	Defines the table index entry.
Start Time **[CostGroupTimebands_StartTime]**	Defines the day and time of day from when this time band is applicable. The format is ddd:hh:mm (e.g., sun:06:00), where: ▪ *ddd* is the day (i.e., sun, mon, tue, wed, thu, fri, or sat) ▪ *hh* and *mm* denote the time of day, where *hh* is the hour (00-23) and *mm* the minutes (00-59)
End Time **[CostGroupTimebands_EndTime]**	Defines the day and time of day until when this time band is applicable. For a description of the valid values, see the parameter above.
Connection Cost **[CostGroupTimebands_ConnectionCost]**	Defines the call connection cost during this time band. This is added as a fixed charge to the call. The valid value range is 0-65533. The default is 0.
Minute Cost **[CostGroupTimebands_MinuteCost]**	Defines the call cost per minute charge during this timeband. The valid value range is 0-65533. The default is 0.

14.2.2.4 Assigning Cost Groups to Routing Rules

Once you have configured your Cost Groups, you need to assign them to routing rules in the Outbound IP Routing table. For more information, see 'Configuring Outbound IP Routing Table' on page 291.

15 Enabling Applications

The device supports the following main applications:

- Stand-Alone Survivability (SAS) application
- Session Border Control (SBC) application
- IP to IP (IP2IP) application

The procedure below describes how to enable these applications. Once an application is enabled, the Web GUI provides menus and parameter fields relevant to the application.

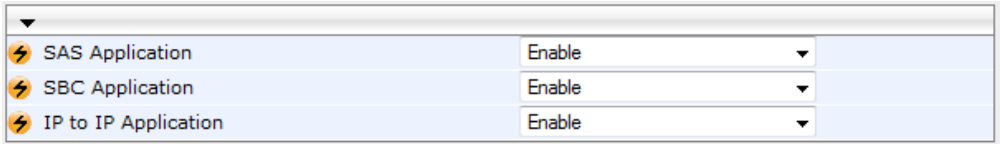
Notes:

- This page displays the application only if the device is installed with the relevant Software Upgrade Key supporting the application (see 'Loading Software Upgrade Key' on page 696).
- For configuring the SAS application, see 'Stand-Alone Survivability (SAS) Application' on page 473.
- For configuring the SBC application, see Session Border Controller on page 411.
- For an overview of the IP2IP application and configuration examples, see IP-to-IP Routing Application on page 391.
- For enabling an application, a device reset is required.
- The Gateway and IP-to-IP applications are depicted in the Web interface as "GW" and "IP2IP" respectively.

➢ **To enable an application:**

1. Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** submenu > **Applications Enabling**).

Figure 15-1: Applications Enabling Page



2. From the relevant application drop-down list, select **Enable**.
3. Save (burn) the changes to the device's flash memory with a device reset (see 'Saving Configuration' on page 680).

Reader's Notes

16 Control Network

This section describes configuration of the network at the SIP control level.

16.1 Configuring SRD Table

The SRD Settings page allows you to configure up to 32 signaling routing domains (SRD). An SRD is configured with a unique name and assigned a Media Realm (defined in the Media Realm table - see 'Configuring Media Realms' on page 181). In addition, other SBC attributes such as media anchoring and user registration can also be configured. Once configured, you can use the SRDs as follows:

- Associate it with a SIP Interface (see 'Configuring SIP Interface Table' on page 210)
- Associate it with an IP Group (see Configuring IP Groups on page 212)
- Associate it with a Proxy Set (see Configuring Proxy Sets Table on page 219)
- Apply an Admission Control rule to it (see Configuring Admission Control Table on page 446)
- Define it as a Classification rule for the incoming SIP request (see Configuring Classification Table on page 452)
- Use it as a destination IP-to-IP routing rule (see 'Configuring IP-to-IP Routing Table' on page 457)

Therefore, an SRD is a set of definitions together creating multiple, virtual multi-service IP gateways:

- Multiple and different SIP signaling interfaces (SRD associated with a SIP Interface) and RTP media (associated with a Media Realm) for multiple Layer-3 networks. Due to the B2BUA nature of the SBC application, different interfaces can be assigned to each leg of the call, and between the LAN and WAN side.
- Can operate with multiple gateway customers that may reside either in the same or in different Layer-3 networks as the device. This allows separation of signaling traffic between different customers. In such a scenario, the device is configured with multiple SRD's.

Typically, one SRD is defined for each group of SIP UAs (e.g. proxies, IP phones, application servers, gateways, softswitches) that communicate with each other. This provides these entities with VoIP services that reside on the same Layer-3 network (must be able to communicate without traversing NAT devices and must not have overlapping IP addresses). One SRD is generally configured for the LAN and one for the WAN. Routing from one SRD to another is possible, whereby each routing destination (IP Group or destination address) indicates the SRD to which it belongs.

The SRD Settings page also displays the IP Groups, Proxy Sets, and SIP Interfaces associated with a selected SRD index.

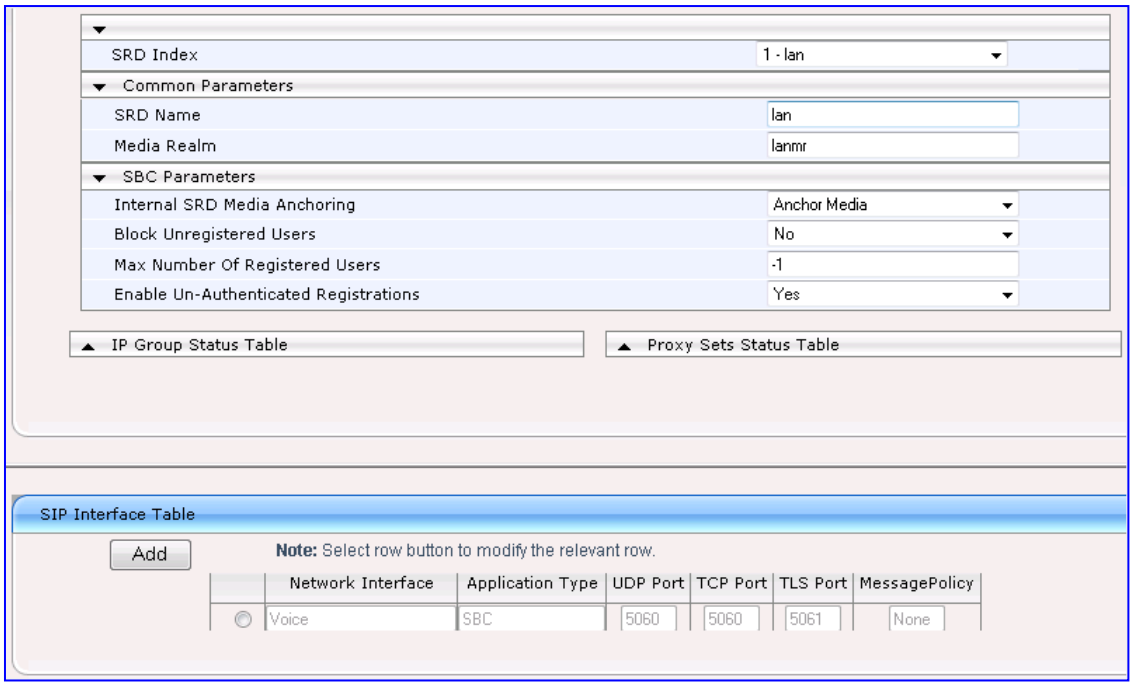
Notes:

- For a detailed description of SRD's, see 'Multiple SIP Signaling/Media Interfaces Environment' on page 225.
- The SRD table can also be configured using the *ini* file table parameter SRD.

➢ **To configure SRDs:**

1. Open the SRD Settings page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **SRD Table**).

Figure 16-1: SRD Settings Page



2. From the 'SRD Index' drop-down list, select an index for the SRD, and then configure it according to the table below.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Note: The SRD Settings page also allows you to define a SIP Interface in the SIP Interface table, instead of navigating to the SIP Interface Table page as described in 'Configuring SIP Interface Table' on page 210.

Table 16-1: SRD Table Parameters

Parameter	Description
SRD Name **[SRD_Name]**	Mandatory descriptive name of the SRD. The valid value can be a string of up to 21 characters.
Media Realm **[SRD_MediaRealm]**	Defines the Media Realm associated with the SRD. The entered string value must be identical (including case-sensitive) to the Media Realm name as defined in the Media Realm table. The valid value is a string of up to 40 characters. **Notes:** ▪ If the Media Realm is later deleted from the Media Realm table, then this value becomes invalid in the SRD table. ▪ For configuring Media Realms, see 'Configuring Media Realms' on page 181.

Parameter	Description
Internal SRD Media Anchoring **[SRD_IntraSRDMediaAnchoring]**	Determines whether the device performs media anchoring or not on media for the SRD. ▪ **[0]** Anchor Media (default) = RTP traverses the device and each leg uses a different coder or coder parameters. ▪ **[1]** Don't Anchor Media = The RTP packet flow does not traverse the device; instead, the two SIP UA's establish a direct RTP/SRTP (media) flow between one another. **Notes:** ▪ When No Media Anchoring is enabled: ✓ The device does not perform manipulation on SDP data (offer/answer transactions) such as ports, IP address, and coders. ✓ Opening voice channels and allocation of IP media ports are not required. ▪ When two UA's pertain to the same SRD and this parameter is set to **[1],** and one of the UA's is defined as a foreign user (example, “follow me service”) located on the WAN while the other UA is located on the LAN, then calls between these two UA's can’t be established until this parameter is set to 0, as the device doesn’t interfere in the SIP signaling. In other words, parameters such as IP addresses are not manipulated for calls between LAN and WAN (although required). ▪ When the global parameter SBCDirectMedia is disabled, you cannot enable No Media Anchoring for two UA's pertaining to separate SRDs; No Media Anchoring can only be enable for two UA's pertaining to the same SRD. ▪ For more information on media handling, see SBC Media Handling on page 424.
Block Unregistered Users **[SRD_BlockUnRegUsers]**	Determines whether the device blocks (rejects) incoming calls (INVITE requests) from unregistered users (pertaining to USER-type IP Groups) for the SRD. ▪ **[0]** No = Calls from unregistered users are not blocked (default). ▪ **[1]** Yes = Blocks calls from unregistered users. **Note:** When the call is blocked, the device sends a SIP 500 "Server Internal Error" response to the remote end.
Max Number of Registered Users **[SRD_MaxNumOfRegUsers]**	Maximum number of users belonging to this SRD that can register with the device. By default, no limitation exists for registered users
Enable Un-Authenticated Registrations **[SRD_EnableUnAuthenticatedRegistrations]**	Determines whether the device blocks REGISTER requests from new users (i.e., users not registered in the device's registration database) when the destination IP Group is of type USER. ▪ **[0]** No = The device sends REGISTER requests to the SIP proxy server and only if authenticated by the server does the device add the user registration to its database. ▪ **[1]** Yes = The device adds REGISTER requests to its database even if the requests are not authenticated by a SIP proxy (default).

16.2 Configuring SIP Interface Table

The SIP Interface Table page allows you to configure up to 32 SIP signaling interfaces, referred to as *SIP Interfaces*. A SIP Interface consists of a combination of ports (UDP, TCP, and TLS), associated with a specific IP address (IPv4) - LAN or WAN - , and for a specific application (i.e., SAS, Gateway\IP2IP, and SBC). Once defined, the SIP Interface can then be associated with an SRD (in the SRD Settings page - see 'Configuring SRD Table' on page 207).

SIP Interfaces can be used for the following:

- Implementing SIP signaling interfaces for each call leg (i.e., each SIP UA communicates with a specific SRD).
- Implementing different SIP signaling ports (listening UDP, TCP, and TLS, and the UDP source ports) for a single interface or for multiple interfaces.
- Differentiating between applications (i.e., SAS, Gateway\IP2IP, and SBC) by creating SIP Interfaces per application.
- Separating signaling traffic between networks (e.g., different customers) to use different routing tables, manipulations, SIP definitions, and so on.

Notes:

- The SIP Interface table also appears in the SRD Settings page, allowing you to add SIP Interfaces there as well.
- For more information on SIP interfaces, see 'Multiple SIP Signaling/Media Interfaces Environment' on page 225.
- The SIP Interface table can also be configured using the *ini* file table parameter SIPInterface.

➢ **To configure the SIP Interface table:**

1. Open the SIP Interface Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **SIP Interface Table**).

Figure 16-2: SIP Interface Table Page

Index	Network Interface	Application Type	UDP Port	TCP Port	TLS Port	SRD	MessagePolicy
1	voice	GW\IP2IP	5060	5060	5061	0	1

2. Add an entry and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 16-2: SIP Interface Table Parameters

Parameter	Description
Network Interface **[SIPInterface_NetworkInterface]**	Defines the Control-type IP network interface that you want to associate with the SIP Interface. This value string must be identical (including case-sensitive) to that configured in the 'Interface Name' in the Multiple Interface table (see 'Configuring IP Interface Settings' on page 117). The default is "Not Configured". **Notes:** ▪ SIP Interfaces that are assigned to a specific SRD must be defined with the same network interface. For example, if you

Parameter	Description
	define three SIP Interfaces for SRD ID #8, all these SIP Interfaces must be defined with the same network interface (e.g., "SIP1"). ▪ To create a SIP interface on the WAN interface, enter the string "WAN". This WAN interface is selected in the Multiple Interface table (or use the WANInterfaceName parameter), which is the WAN interface address as defined in WAN Access Settings on page 567. If VLANs are defined for the WAN interface and one of the VLANs is selected as the VoIP WAN interface, then the defined SIP Interface uses this interface.
Application Type **[SIPInterface_ApplicationType]**	Defines the application type associated with the SIP Interface. ▪ **[0]** GW/IP2IP (default) = IP-to-IP routing application and regular gateway functionality ▪ **[1]** SAS = Stand-Alone Survivability (SAS) application ▪ **[2]** SBC = SBC application
UDP Port **[SIPInterface_UDPPort]**	Defines the listening and source UDP port. The valid range is 1 to 65534. The default is 5060. **Notes:** ▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
TCP Port **[SIPInterface_TCPPort]**	Defines the listening TCP port. The valid range is 1 to 65534. The default is 5060. **Notes:** ▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
TLS Port **[SIPInterface_TLSPort]**	Defines the listening TLS port. The valid range is 1 to 65534. The default is 5061. **Notes:** ▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
SRD **[SIPInterface_SRD]**	Defines the SRD ID associated with the SIP Interface. The default SRD is 0. **Notes:** ▪ Each SRD can be associated with up to three SIP Interfaces, where each SIP Interface pertains to a different Application Type (GW/IP2IP, SAS, and SBC). ▪ SIP Interfaces that are assigned to a specific SRD must be defined with the same network interface. For example, if you define three SIP Interfaces for SRD ID #8, all these SIP Interfaces must be defined with the same network interface (e.g., "SIP1"). ▪ To configure SRDs, see 'Configuring SRD Table' on page 207.

Parameter	Description
Message Policy **[SIPInterface_MessagePolicy]**	Assigns a SIP message policy to the SIP interface. **Note:** To configure SIP message policies, see Configuring SIP Message Policy Rules.

16.3 Configuring IP Groups

The IP Group Table page allows you to create up to 32 logical IP entities called *IP Groups*. An IP Group is an entity with a set of definitions such as a Proxy Set ID (see 'Configuring Proxy Sets Table' on page 219), which represents the IP address of the IP Group.

IP Groups provide the following uses:

- SIP dialog registration and authentication (digest user/password) of a specific IP Group (*Served* IP Group, e.g., corporate IP-PBX) with another IP Group (*Serving* IP Group, e.g., ITSP). This is configured in the Account table (see 'Configuring Account Table' on page 237).
- Call routing rules:
  - Outgoing IP calls (IP-to-IP or Tel-to-IP): used to identify the source of the call and used as the destination for the outgoing IP call (defined in the Outbound IP Routing Table). For Tel-to-IP calls, the IP Group (*Serving IP Group*) can be used as the IP destination to where all SIP dialogs that are initiated from a Trunk Group are sent (defined in 'Configuring Trunk Group Settings' on page 273).
  - Incoming IP calls (IP-to-IP or IP-to-Tel): used to identify the source of the IP call
  - Number Manipulation rules to IP: used to associate the rule with a specific calls identified by IP Group
- For the SBC application, IP Groups are used to classify incoming SIP dialog-initiating requests (e.g., INVITE messages) to a source IP Group, based on Proxy Set ID (defined in Configuring Classification Table on page 452). This occurs if the database search for a registered user is unsuccessful. The classification process locates a Proxy Set ID (associated with the SIP dialog request's IP address) in the Proxy Set table, and then locates a match with an IP Group that is associated with this Proxy Set in the IP Group table. This classification is enabled using the parameter Classify By Proxy Set.

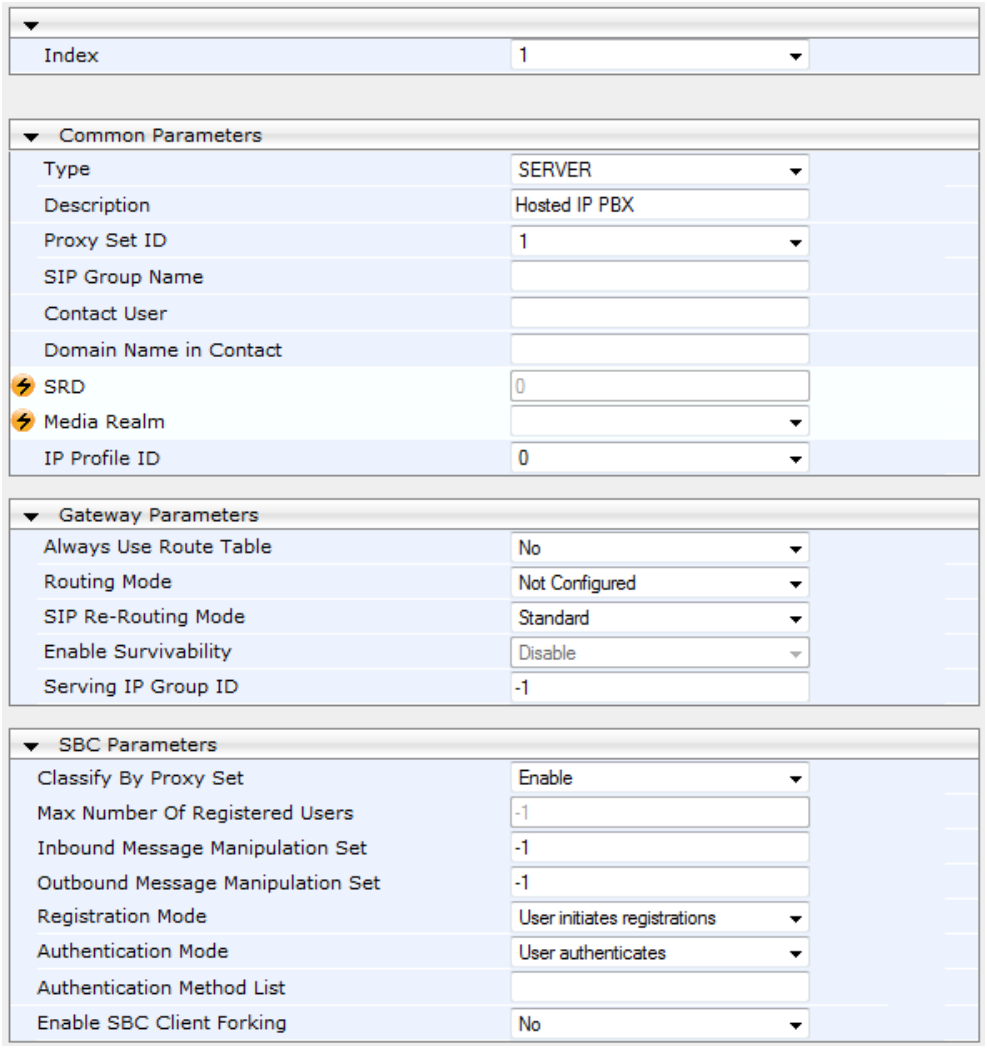
Notes:

- When operating with multiple IP Groups, the default Proxy server must not be used (i.e., the parameter IsProxyUsed must be set to 0).
- If different SRDs are configured in the IP Group and Proxy Set tables, the SRD defined for the Proxy Set takes precedence.
- You cannot modify IP Group index 0. This IP Group is set to default values and is used by the device when IP Groups are not implemented.
- You can also configure the IP Groups table using the *ini* file table parameter IPGroup (see 'Configuration Parameters Reference' on page 752).

➢ **To configure IP Groups:**

1. Open the IP Group Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **IP Group Table**).

Figure 16-3: IP Group Table



2. Configure the IP group parameters according to the table below.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' **on page 680.**

Table 16-3: IP Group Parameters

Parameter	Description
Common Parameters	
Type **[IPGroup**_Type]	The IP Group can be defined as one of the following types: ▪ **[0]** SERVER = used when the destination address (configured by the Proxy Set) of the IP Group (e.g., ITSP, Proxy, IP-PBX, or Application server) is known. ▪ **[1]** USER = represents a group of users (such as IP phones and softphones) where their location is dynamically

Parameter	Description
	obtained by the device when REGISTER requests and responses traverse (or are terminated) by the device. These users are considered remote (far-end) users. Typically, this IP Group is configured with a Serving IP Group that represents an IP-PBX, Application or Proxy server that serves this USER-type IP Group. Each SIP request sent by a user of this IP Group is proxied to the Serving IP Group. For registrations, the device updates its internal database with the AOR and contacts of the users. Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group. The device forwards these responses directly to the SIP users. To route a call to a registered user, a rule must be configured in the Outbound IP Routing Table table (see Configuring Outbound IP Routing Table on page 291). The device searches the dynamic database (by using the request URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained from this entry, and a SIP request is sent to the destination.The device also supports NAT traversal for the SIP clients that are behind NAT. In this case, the device must be defined with a global IP address. ▪ **[2]** GATEWAY = This is applicable only to the SBC application in scenarios where the device receives requests to and from a gateway representing multiple users. This IP Group type is necessary as the other IP Group types are not suitable: ✓ The IP Group cannot be defined as a SERVER since its destination address is unknown during configuration. ✓ The IP Group cannot be defined as a USER since the SIP Contact header of the incoming REGISTER does not represent a specific user. The Request-URI user part can change and therefore, the device is unable to identify an already registered user and therefore, adds an additional record to the database. The IP address of the "GATEWAY" IP Group is obtained dynamically from the host part of the Contact header in the REGISTER request received from the IP Group. Therefore, routing to this IP Group is possible only once a REGISTER request is received. If a REGISTER refresh request arrives, the device updates the new location (i.e., IP address) of the IP Group. If the REGISTER fails, no update is performed. If an UN-REGISTER request arrives, the IP address associated with the IP Group is deleted and therefore, no routing to the IP Group is done. **Note:** This field is available only if the SBC or IP-to-IP application is enabled.
Description **[IPGroup_Description]**	Brief string description of the IP Group. The value range is a string of up to 29 characters. The default is an empty field.
Proxy Set ID **[IPGroup_ProxySetId]**	The Proxy Set ID (defined in 'Configuring Proxy Sets Table' on page 219) associated with the IP Group. All INVITE messages destined to this IP Group are sent to the IP address associated with the Proxy Set. **Notes:**

Parameter	Description
	▪ Proxy Set ID 0 must **not** be selected; this is the device's default Proxy. ▪ The Proxy Set is applicable only to SERVER-type IP Groups.
SIP Group Name **[IPGroup_SIPGroupName]**	The SIP Request-URI host name used in INVITE and REGISTER messages sent to the IP Group, or the host name in the From header of INVITE messages received from the IP Group. If not specified, the value of the global parameter, ProxyName (see 'Configuring Proxy and Registration Parameters' on page 239) is used instead. The value range is a string of up to 100 characters. The default is an empty field. **Note:** If the IP Group is of type USER, this parameter is used internally as a host name in the Request-URI for Tel-to-IP initiated calls. For example, if an incoming call from the device's T1 trunk is routed to a USER-type IP Group, the device first creates the Request-URI (<destination_number>@<SIP Group Name>), and then it searches the user's internal database for a match.
Contact User **[IPGroup_ContactUser]**	Defines the user part for the From, To, and Contact headers of SIP REGISTER messages, and the user part for the Contact header of INVITE messages that are received from the IP Group and forwarded by the device to another IP Group. **Notes:** ▪ This parameter is applicable only to SERVER-type IP Groups. ▪ This parameter is overridden by the 'Contact User' parameter in the 'Account' table (see 'Configuring Account Table' on page 237).
Contact Name **[IPGroup_ContactName]**	Defines the host name (string) that the device uses in the SIP message's Via and Contact headers. This is typically used to define an FQDN as the host name. The device uses this string for Via and Contact headers in outgoing INVITE messages to a specific IP Group, and the Contact header in SIP 18x and 200 OK responses for incoming INVITE messages from a specific IP Group. The Inbound IP Routing table can be used to identify the source IP Group from where the INVITE message was received. If this parameter is not configured (default), these headers are populated with the device's dotted-decimal IP address of the network interface on which the message is sent. **Note:** To ensure proper device handling, this parameter should be a valid FQDN.
SRD **[IPGroup**_SRD]	The SRD (defined in Configuring SRD Table on page 207) associated with the IP Group. The default is 0. **Note:** For this parameter to take effect, a device reset is required.
Media Realm **[IPGroup**_MediaRealm]	Assigns a Media Realm to the IP Group. The entered string value must be identical (including case-sensitive) to the Media Realm name as defined in the Media Realm table.

Parameter	Description
	Notes: ▪ For this parameter to take effect, a device reset is required. ▪ If the Media Realm is later deleted from the Media Realm table, then this value becomes invalid. ▪ For configuring Media Realms, see Configuring Media Realms on page 181.
IP Profile ID **[IPGroup_ProfileId]**	The IP Profile (defined in to 'Configuring IP Profile Settings' on page 248) that you want assigned to this IP Group. The default is 0.
Gateway Parameters	
Always Use Route Table **[IPGroup_AlwaysUseRouteTable]**	Determines the Request-URI host name in outgoing INVITE messages. ▪ **[0]** No (default). ▪ **[1]** Yes = The device uses the IP address (or domain name) defined in the Outbound IP Routing Table (see 'Configuring the Outbound IP Routing Table' on page 291) as the Request-URI host name in outgoing INVITE messages instead of the value entered in the 'SIP Group Name' field. **Note:** This parameter is applicable only to SERVER-type IP Groups.
Routing Mode **[IPGroup**_RoutingMode]	Defines the routing mode for outgoing SIP INVITE messages. ▪ **[-1**] Not Configured = The routing is according to the selected Serving IP Group. If no Serving IP Group is selected, the device routes the call according to the Outbound IP Routing Table (see Configuring Outbound IP Routing Table on page 291). (Default) ▪ **[0]** Routing Table = The device routes the call according to the Outbound IP Routing Table'. ▪ **[1]** Serving IP Group = The device sends the SIP INVITE to the selected Serving IP Group. If no Serving IP Group is selected, the default IP Group is used. If the Proxy server(s) associated with the destination IP Group is not alive, the device uses the Outbound IP Routing Table (if the parameter IsFallbackUsed is set 1, i.e., fallback enabled - see Configuring Proxy and Registration Parameters on page 239). ▪ **[2]** Request-URI = The device sends the SIP INVITE to the IP address according to the received SIP Request-URI host name. **Notes:** ▪ This parameter is applicable only to the IP-to-IP application. ▪ This parameter is applicable only to SERVER-type IP Groups.
SIP Re-Routing Mode **[IPGroup_SIPReRoutingMode]**	Determines the routing mode after a call redirection (i.e., a 3xx SIP response is received) or transfer (i.e., a SIP REFER request is received). ▪ **[0]** Standard = INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI, according to the Refer-To header in the REFER message or Contact header in the 3xx response (default). ▪ **[1]** Proxy = Sends a new INVITE to the Proxy.

Parameter	Description
	Note: Applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0. ▪ **[2]** Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination. **Notes:** ▪ When this parameter is set to **[1]** and the INVITE sent to the Proxy fails, the device re-routes the call according to the Standard mode **[0].** ▪ When this parameter is set to **[2]** and the INVITE fails, the device re-routes the call according to the Standard mode **[0].** If DNS resolution fails, the device attempts to route the call to the Proxy. If routing to the Proxy also fails, the Redirect / Transfer request is rejected. ▪ When this parameter is set to **[2],** the XferPrefix parameter can be used to define different routing rules for redirected calls. ▪ This parameter is ignored if the parameter AlwaysSendToProxy is set to 1.
Enable Survivability **[IPGroup_EnableSurvivability]**	Determines how the device handles registration messages and whether Survivability mode is enabled for USER-type IP Groups. ▪ **[0]** Disable (default). ▪ **[1]** Enable if Necessary = Survivability mode is enabled only if the Serving IP Group is unavailable. The device saves in its Registration database the registration messages sent by the clients (e.g., IP phones) belonging to the USER-type IP Group. If communication with the Serving IP Group (e.g., IP-PBX) fails, the USER-type IP Group enters into Survivability mode in which the device uses its database for routing calls between the clients of the USER-type IP Group. In Survivability mode, the RTP packets between the clients always traverse through the device, and new registrations can also be processed. When the Serving IP Group is available again, the device returns to normal mode, sending INVITE and REGISTER messages to the Serving IP Group. ▪ **[2]** Always Enable = Survivability mode is always enabled. The communication with the Serving IP Group is always considered as failed. The device uses its database for routing calls between the clients of the USER-type IP Group. ▪ **[3]** Always Terminate Register = The registration messages received from the clients are saved in the device's registration database without forwarding them to the proxy server. Upon receipt of the registration message, the device returns a SIP 200 OK to the client. **Notes:** ▪ This parameter is applicable only to the IP-to-IP application. ▪ This parameter is applicable only to USER-type IP Groups.
Serving IP Group ID **[IPGroup_ServingIPGroup]**	If configured, INVITE messages initiated from the IP Group are sent to this Serving IP Group (range 1 to 9). In other words, the INVITEs are sent to the address defined for the Proxy Set

Parameter	Description
	associated with this Serving IP Group. The Request-URI host name in the INVITE messages are set to the value of the 'SIP Group Name' parameter defined for the Serving IP Group. **Notes:** ▪ This parameter is applicable only to the IP-to-IP application. ▪ If the parameter PreferRouteTable is set to 1, the routing rules in the Outbound IP Routing Table take precedence over this 'Serving IP Group ID' parameter. ▪ If this parameter is not configured, the INVITE messages are sent to the default Proxy or according to the Outbound IP Routing Table'.
SBC Parameters	
Classify By Proxy Set **[IPGroup**_ClassifyByProxySet]	Determines whether the incoming INVITE is classified to an IP Group according to the Proxy Set. ▪ **[0]** Disable ▪ **[1]** Enable (default) This classification occurs only if classification according to the device's database fails for locating whether the INVITE arrived from a registered user. The classification proceeds with checking whether the INVITE's IP address (if host names, then according to the dynamically resolved IP address list) is defined in the IP Group's Proxy Set ID (in the Proxy Set table). If the IP address is listed, then the INVITE is assigned to this IP Group. **Notes:** ▪ This parameter is applicable only to SERVER-type IP Groups. ▪ This classification is not relevant in cases where multiple IP Groups use the same Proxy Set.
Max Number Of Registered Users **[IPGroup**_MaxNumOfRegUsers]	Maximum number of users belonging to this IP Group that can register with the device. By default, no limitation exists for registered users. **Note:** This field is applicable only to USER-type IP Groups.
Inbound Message Manipulation Set **[IPGroup**_InboundManSet]	Message Manipulation Set (rule) that you want to assign to this IP Group for SIP message manipulation rule on the inbound message. The Message Manipulation rules are configured using the MessageManipulations parameter (see Configuring Message Manipulations on page 464).
Outbound Message Manipulation Set **[IPGroup**_OutboundManSet]	Message Manipulation Set (rule) that you want to assign to this IP Group for SIP message manipulation on the outbound message. The Message Manipulation rules are configured using the MessageManipulations parameter (see Configuring Message Manipulations on page 464).
Registration Mode **[IPGroup**_RegistrationMode]	Defines the registration mode for an IP Group: ▪ **[0]** User initiates registrations (default) ▪ **[1]** SBC initiate registrations (works only with User Info file) = Used when the device serves as a client (e.g., with an IP PBX). ▪ **[2]** No registrations needed = The device adds users to its database in active state.

Parameter	Description
Authentication Mode **[PGroup**_AuthenticationMode]	Defines the authentication mode. ▪ **[0]** User Authenticates (default) = The device does not handle the authentication, but simply passes the authentication messages between the SIP user agents. ▪ **[1]** SBC Authenticates (as client) = The device authenticates as a client. It receives the 401/407 response from the proxy requesting for authentication. The device sends the proxy the authorization credentials (i.e., user name and password) according to one of the following: 1) account defined in the Account table (only if authenticating SERVER-type IP Group), 2) global username and password parameters (only if authenticating SERVER-type IP Group), 3) User Information file, or 4) sends request to users requesting credentials (only if authenticating USER-type IP Group). ▪ **[2]** SBC Authenticates (as server) = The device authenticates as a server (using the User Information file).
Authentication Method List **[IPGroup**_MethodList]	Defines SIP methods that the device must challenge. Multiple entries are separated by the forward slash "/". If none are defined (default), no methods are challenged.
Enable SBC Client Forking **[IPGroup**_EnableSBCClientForking]	Enables call forking for USER-type IP Groups. ▪ **[0]** No (default) ▪ **[1]** Yes = The device forks INVITE messages (to up to five separate SIP outgoing legs).

16.4 Configuring Proxy Sets Table

The Proxy Sets Table page allows you to define *Proxy Sets*. A Proxy Set is a group of Proxy servers defined by IP address or fully qualified domain name (FQDN). You can define up to 32 Proxy Sets, each with a unique ID number and up to five Proxy server addresses. For each Proxy server address you can define the transport type (i.e., UDP, TCP, or TLS). In addition, Proxy load balancing and redundancy mechanisms can be applied per Proxy Set (if a Proxy Set contains more than one Proxy address).

Proxy Sets can later be assigned to IP Groups of type SERVER (see 'Configuring IP Groups' on page 212). When the device sends an INVITE message to an IP Group, it is sent to the IP address or domain name defined for the Proxy Set that is associated with the IP Group. In other words, the Proxy Set represents the **destination** of the call. Typically, for IP-to-IP call routing, at least two Proxy Sets are defined for call destination – one for each leg (IP Group) of the call (i.e., both directions). For example, one Proxy Set for the Internet Telephony Service provider (ITSP) interfacing with one 'leg' of the device and another Proxy Set for the second SIP entity (e.g., ITSP) interfacing with the other 'leg' of the device.

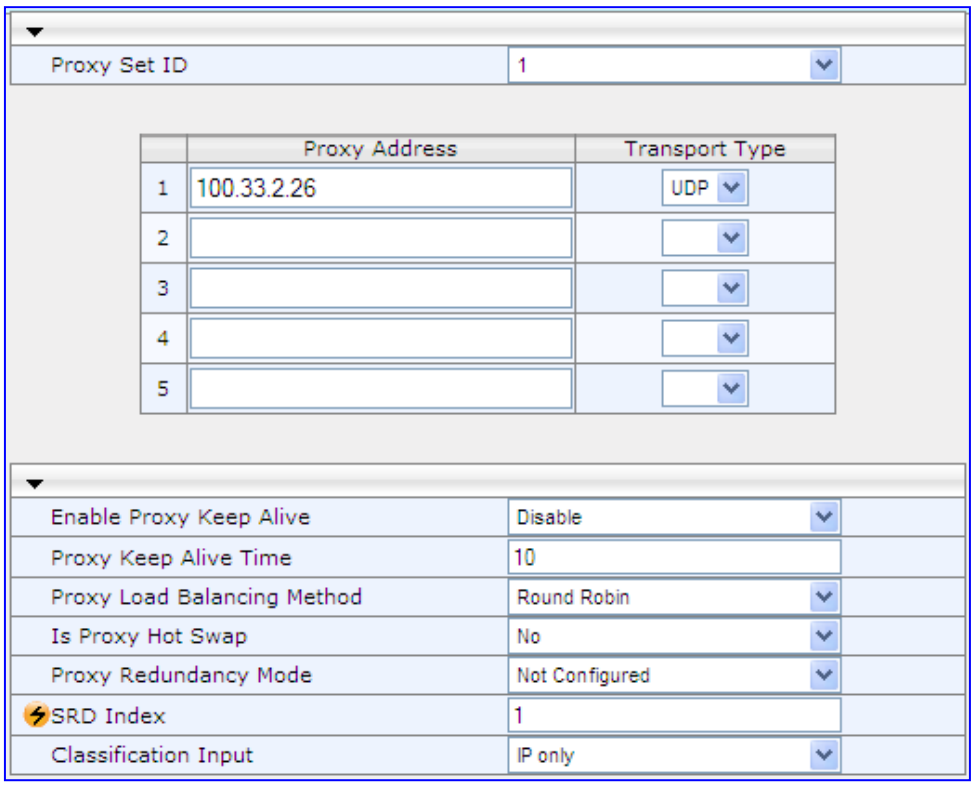
Notes:

- You can also configure the Proxy Sets table using two complementary *ini* file table parameters (see 'Configuration Parameters Reference' on page 752):
  - ProxyIP: used for creating a Proxy Set ID defined with IP addresses.
  - ProxySet: used for defining various attributes for the Proxy Set ID.
- Proxy Sets can be assigned only to SERVER-type IP Groups.
- Each IP Group can be classified according to its Proxy Set ID, if in the IP Group table the parameter ClassifyByProxySet is enabled.

➢ **To add Proxy servers:**

1. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **Proxy Sets Table**).

Figure 16-4: Proxy Sets Table Page



2. From the 'Proxy Set ID' drop-down list, select an ID for the desired group.
3. Configure the Proxy parameters according to the following table.
4. Click Subm**it to** apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 16-4: Proxy Sets Table Parameters

Parameter	Description
Web: Proxy Set ID EMS: Index **[ProxySet_Index]**	The Proxy Set identification number. The valid range is 0 to 31. The Proxy Set ID 0 is used as the default Proxy Set. **Note:** Although not recommended, you can use both default Proxy

Parameter	Description
	Set (ID 0) and IP Groups for call routing. For example, in the Trunk Group Settings page (see 'Configuring Trunk Group Settings' on page 273) you can configure a Serving IP Group to where you want to route specific Trunk Group channels, and all other device channels then use the default Proxy Set. You can also use IP Groups in the Outbound IP Routing Table (see 'Configuring the Outbound IP Routing Table' on page 291) to configure the default Proxy Set if the parameter PreferRouteTable is set to 1. To summarize, if the default Proxy Set is used, the INVITE message is sent according to the following preferences: ▪ To the Trunk Group's Serving IP Group ID, as defined in the Trunk Group Settings table. ▪ According to the Outbound IP Routing Table if the parameter PreferRouteTable is set to 1. ▪ To the default Proxy. Typically, when IP Groups are used, there is no need to use the default Proxy, and all routing and registration rules can be configured using IP Groups and the Account tables (see 'Configuring Account Table' on page 237).
Proxy Address **[ProxyIp_IpAddress]**	The IP address (and optionally port number) of the Proxy server. Up to five IP addresses can be configured per Proxy Set. Enter the IP address as an FQDN or in dotted-decimal notation (e.g., 201.10.8.1). You can also specify the selected port in the format: <IP address>:<port>. If you enable Proxy Redundancy (by setting the parameter EnableProxyKeepAlive to 1 or 2), the device can operate with multiple Proxy servers. If there is no response from the first (*primary*) Proxy defined in the list, the device attempts to communicate with the other (*redundant*) Proxies in the list. When a redundant Proxy is located, the device either continues operating with it until the next failure occurs or reverts to the primary Proxy (refer to the parameter ProxyRedundancyMode). If none of the Proxy servers respond, the device goes over the list again. The device also provides real-time switching (Hot-Swap mode) between the primary and redundant proxies (refer to the parameter IsProxyHotSwap). If the first Proxy doesn't respond to the INVITE message, the same INVITE message is immediately sent to the next Proxy in the list. The same logic applies to REGISTER messages (if RegistrarIP is not defined). **Notes:** ▪ If EnableProxyKeepAlive is set to 1 or 2, the device monitors the connection with the Proxies by using keep-alive messages (OPTIONS or REGISTER). ▪ To use Proxy Redundancy, you must specify one or more redundant Proxies. ▪ When a port number is specified (e.g., domain.com:5080), DNS NAPTR/SRV queries aren't performed, even if ProxyDNSQueryType is set to 1 or 2.
Transport Type **[ProxyIp_TransportType]**	The transport type per Proxy server. ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS

Parameter	Description
	▪ **[-1]** = Undefined **Note:** If no transport type is selected, the value of the global parameter SIPTransportType is used (see 'Configuring SIP General Parameters' on page 235).
Web/EMS: Enable Proxy Keep Alive **[ProxySet_EnableProxyKeepAlive]**	Determines whether Keep-Alive with the Proxy is enabled or disabled. This parameter is configured per Proxy Set. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Using Options = Enables Keep-Alive with Proxy using SIP OPTIONS messages. ▪ **[2]** Using Register = Enables Keep-Alive with Proxy using SIP REGISTER messages. If set to 'Using Options', the SIP OPTIONS message is sent every user-defined interval (configured by the parameter ProxyKeepAliveTime). If set to 'Using Register', the SIP REGISTER message is sent every user-defined interval (configured by the RegistrationTime parameter for the GW/IP2IP application or by the SBCProxyRegistrationTime parameter for SBC application). Any response from the Proxy, either success (200 OK) or failure (4xx response) is considered as if the Proxy is communicating correctly. **Notes:** ▪ For Survivability mode for USER-type IP Groups, this parameter must be enabled (1 or 2). ▪ This parameter must be set to 'Using Options' when Proxy redundancy is used. ▪ When this parameter is set to 'Using Register', the homing redundancy mode is disabled. ▪ When the active proxy doesn't respond to INVITE messages sent by the device, the proxy is tagged as 'offline'. The behavior is similar to a Keep-Alive (OPTIONS or REGISTER) failure. ▪ If this parameter is enabled and the proxy uses the TCP/TLS transport type, you can enable CRLF Keep-Alive mechanism, using the UsePingPongKeepAlive parameter.
Web: Proxy Keep Alive Time EMS: Keep Alive Time **[ProxySet_ProxyKeepAliveTime]**	Defines the Proxy keep-alive time interval (in seconds) between Keep-Alive messages. This parameter is configured per Proxy Set. The valid range is 5 to 2,000,000. The default value is 60. **Note:** This parameter is applicable only if the parameter EnableProxyKeepAlive is set to 1 (OPTIONS). When the parameter EnableProxyKeepAlive is set to 2 (REGISTER), the time interval between Keep-Alive messages is determined by the parameter RegistrationTime for the GW/IP2IP application or by the SBCProxyRegistrationTime parameter for SBC application.
Web: Proxy Load Balancing Method EMS: Load Balancing Method **[ProxySet_ProxyLoadBalancingMethod]**	Enables the Proxy Load Balancing mechanism per Proxy Set ID. ▪ **[0]** Disable = Load Balancing is disabled (default) ▪ **[1]** Round Robin ▪ **[2]** Random Weights When the Round Robin algorithm is used, a list of all possible Proxy IP addresses is compiled. This list includes all IP addresses per Proxy Set, after necessary DNS resolutions (including NAPTR and SRV, if configured). After this list is compiled, the Proxy Keep-Alive mechanism (according to parameters EnableProxyKeepAlive and ProxyKeepAliveTime) tags each entry as 'offline' or 'online'. Load balancing is only performed on Proxy servers that are tagged as

Parameter	Description
	'online'. All outgoing messages are equally distributed across the list of IP addresses. REGISTER messages are also distributed unless a RegistrarIP is configured. The IP addresses list is refreshed according to ProxyIPListRefreshTime. If a change in the order of the entries in the list occurs, all load statistics are erased and balancing starts over again. When the Random Weights algorithm is used, the outgoing requests are not distributed equally among the Proxies. The weights are received from the DNS server by using SRV records. The device sends the requests in such a fashion that each Proxy receives a percentage of the requests according to its' assigned weight. A single FQDN should be configured as a Proxy IP address. The Random Weights Load Balancing is not used in the following scenarios: ▪ The Proxy Set includes more than one Proxy IP address. ▪ The only Proxy defined is an IP address and not an FQDN. ▪ SRV is not enabled (DNSQueryType). ▪ The SRV response includes several records with a different Priority value.
Web/EMS: Is Proxy Hot-Swap **[ProxySet_IsProxyHotSwap]**	Enables the Proxy Hot-Swap redundancy mode per Proxy Set. ▪ **[0]** No (default) ▪ **[1]** Yes If Proxy Hot-Swap is enabled, the SIP INVITE/REGISTER message is initially sent to the first Proxy/Registrar server. If there is no response from the first Proxy/Registrar server after a specific number of retransmissions (configured by the parameter HotSwapRtx), the message is resent to the next redundant Proxy/Registrar server.
Web/EMS: Redundancy Mode **[ProxySet_ProxyRedundancyMode]**	Determines whether the device switches back to the primary Proxy after using a redundant Proxy (per this Proxy Set). ▪ **[-1]** = Not configured – the “global” parameter ProxyRedundancyMode applies (default). ▪ **[0]** Parking = The device continues operating with a redundant (now active) Proxy until the next failure, after which it operates with the next redundant Proxy. ▪ **[1]** Homing = The device always attempts to operate with the primary Proxy server (i.e., switches back to the primary Proxy whenever it's available). **Notes:** ▪ To use the Proxy Redundancy mechanism, you need to enable the keep-alive with Proxy option, by setting the parameter EnableProxyKeepAlive to 1 or 2. ▪ If this parameter is configured, then the global parameter is ignored.
Web/EMS: SRD Index **[ProxySet**_ProxySet_SRD]	The SRD (defined in Configuring SRD Table on page 207) associated with the Proxy Set ID. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If no SRD is defined for this parameter, by default, SRD ID #0 is associated with the Proxy Set.

Parameter	Description
Web/EMS: Classification Input **[ClassificationInput]**	Classifies an IP call to a Proxy Set, based on either its IP address, or based on its IP address, port, and transport type: ▪ **[0]** Compare only IP = IP call classified to Proxy Set according to IP address only (default). ▪ **[1]** Compare IP, port and transport type = IP call classified to Proxy Set according to IP address, port, and transport type.

16.5 Configuring NAT Translation per IP Interface

The NAT Translation table defines network address translation (NAT) rules for translating source IP addresses per VoIP interface (SIP control and RTP media traffic) into NAT IP addresses (*public* or *global*). This allows, for example, the separation of VoIP traffic between different ISTP's, and topology hiding of internal IP addresses to the "public" network. Each IP interface (configured in the Multiple Interface table - InterfaceTable parameter) can be associated with a NAT rule in this table, translating the source IP address and port of the outgoing packet into the NAT address (IP address and port range).

The device's priority method for performing NAT is as follows (not relevant for the SBC application):

a. Uses an external STUN server (STUNServerPrimaryIP parameter) to assign a NAT address to all interfaces.

b. Uses the StaticNATIP parameter to define one NAT IP address for all interfaces.

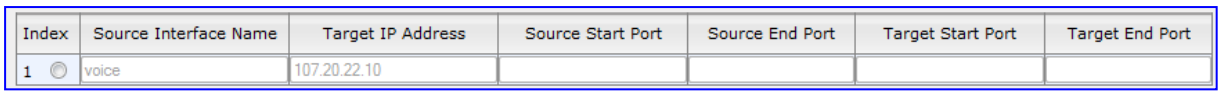
c. Uses the NATTranslation parameter to define NAT per interface.

If NAT is not configured (by any of the above-mentioned methods), the device sends the packet according to its IP address defined in the Multiple Interface table.

➢ **To configure NAT translation rules:**

1. Open the NAT Translation Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **NAT Translation Table**).

Figure 16-5: NAT Translation Table Page



2. Configure the parameters according to the table below.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 16-5: NAT Translation Table Parameters

Parameter	Description
Index **[NATTranslation_Index]**	Defines the table index entry. This table can include up to 32 entries.
Source Interface Name **[NATTranslation_SourceIPInterfaceName]**	Defines the name of the IP interface, as appears in the Multiple Interface table.
Target IP Address **[NATTranslation_TargetIPAddress]**	Defines the global IP address.

Parameter	Description
Source Start Port **[NATTranslation_SourceStartPort]**	Defines the optional starting port range (1-65536) of the IP interface. If no ports are required, leave this field blank.
Source End Port **[NATTranslation_SourceEndPort]**	Defines the optional ending port range (1-65536) of the IP interface. If no ports are required, leave this field blank.
Target Start Port **[NATTranslation_TargetStartPort]**	Defines the optional starting port range (1-65536) of the global address. If no ports are required, leave this field blank.
Target End Port **[NATTranslation_TargetEndPort]**	Defines the optional ending port range (1-65536) of the global address. If no ports are required, leave this field blank.

16.6 Multiple SIP Signaling and Media Interfaces using SRDs

The device supports the configuration of multiple, logical SIP signaling interfaces and media (RTP) interfaces. Multiple SIP and media interfaces allows you to:

- Separate SIP and media traffic between different applications (i.e., SAS, Gateway\IP-to-IP, and SBC)
- Separate SIP and media traffic between different Layer-3 networks (e.g., when operating with multiple ITSPs - separation of signaling traffic between different customers). This separation allows you to use different routing rules, manipulations, SIP definitions, etc. per network (customer). This is also applicable for networks residing in the same or in different Layer-3 networks as the device. In such a scenario, the device is configured with multiple SRDs.
- Due to the back-to-back user agent (B2BUA) nature of the SBC application, different interfaces can be assigned to each leg of the call and between the LAN and WAN.
- Implement different SIP signaling ports (listening UDP, TCP, and TLS, and the UDP source ports) for single or multiple interfaces.
- Only one signaling interface per application type is allowed per SRD. An SRD can be associated with many SIP interfaces which are based on one Layer-3 interface, with different ports.

Multiple SIP and RTP interfaces are implemented using SRDs (Signaling Routing Domains). An SRD is a set of definitions of IP interfaces, device resources, SIP behaviors and other definitions that together create (from the IP user's perspective), multiple, virtual multi-service gateways, from one physical device.

An SRD is composed of the following main entities:

- **Media Realm:** A Media Realm is a range of UDP ports associated with a specific Media-type IP interface (defined in the Multiple Interface table in 'Configuring IP Interface Settings' on page 117). You can configure multiple Media Realms (each with a specified UDP port range) for a specific media IP interface, thereby allowing you to divide a media IP interface (RTP traffic) into a pool of media realms. Media Realms are configured in the Media Realm table (see 'Configuring Media Realms' on page 181). Once configured, you can assign Media Realms to an SRDs (and/or IP Groups).
- **SIP Interface:** A SIP Interface is a combination of UDP, TCP, and/or TLS ports associated with a specific Control-type IP interface (defined in the Multiple Interface table). Therefore, a SIP Interface represents a SIP signaling interface. SIP Interfaces are configured n the SIP Interface table (see 'Configuring SIP Interface Table' on page 210) where they are assigned to SRDs:

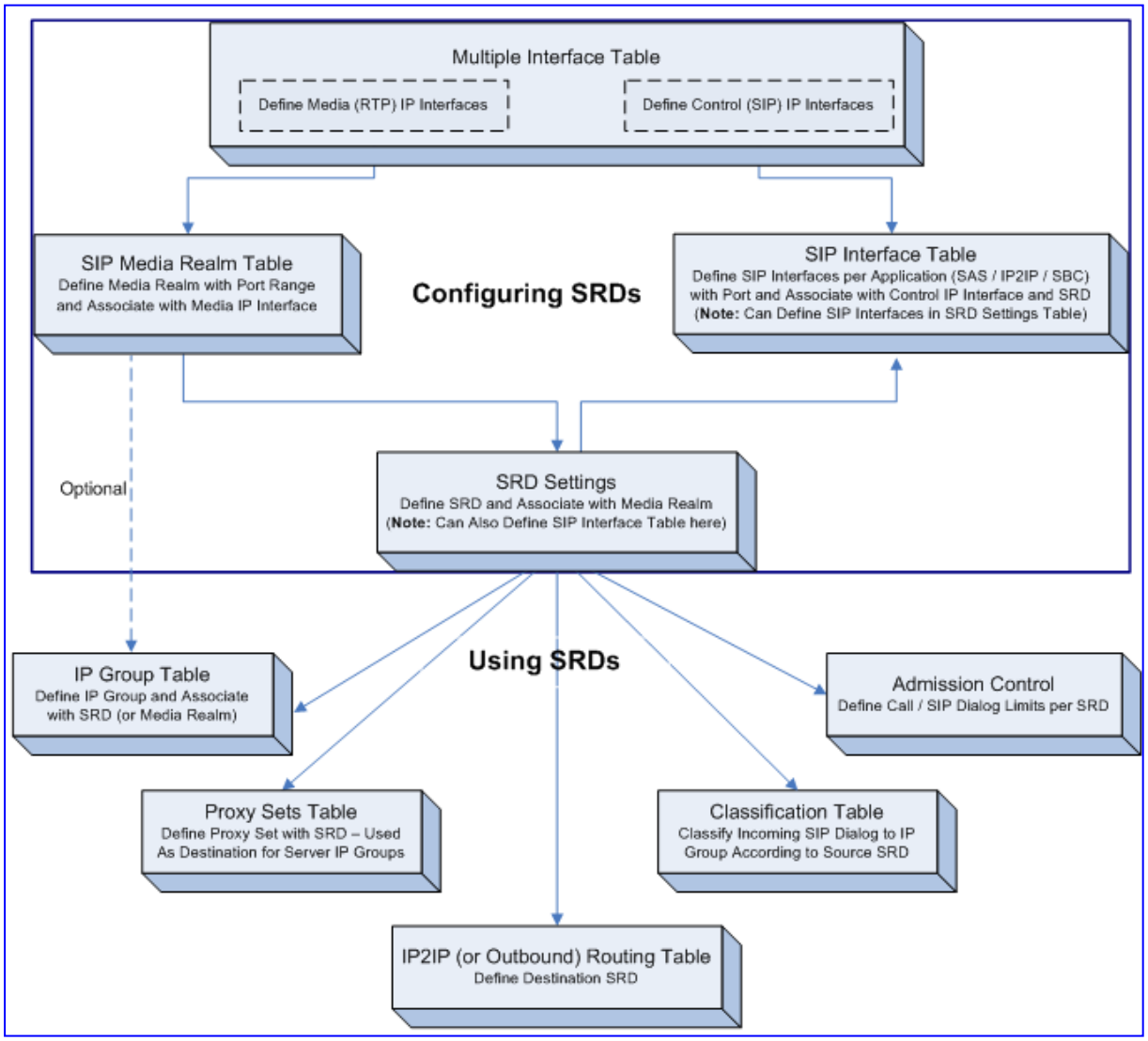
- Each SIP Interface is defined with a unique signaling port (i.e., no two SIP Interfaces can share the same port - no overlapping).
- SIP Interfaces assigned to a specific SRD ID must all be defined with the same network interface (from the Multiple Interface table). For example, if you define three SIP Interfaces for SRD ID #8, all these SIP Interfaces must be defined with the same network interface (e.g., "SIP1").
- Each SIP Interface assigned to a specific SRD ID must be defined with a different application type (i.e., SAS, Gateway\IP-to-IP, and SBC). Therefore, up to three SIP Interfaces can be assigned to a specific SRD.

Once configured, you can use an SRD as follows:

- Use it in classification rules for identifying incoming SIP requests (see Configuring Classification Table on page 452)
- Associate it with an IP Group (see Configuring IP Groups on page 212).
- Associate it with a Proxy Set (see Configuring Proxy Sets Table on page 219).
- Define it as a destination SRD for IP-to-IP routing rules (see 'Configuring IP-to-IP Routing Table' on page 457). Routing from one SRD to another is possible, where each routing destination (IP Group or destination address) indicates the SRD to which it belongs.

- Apply an Admission Control rule to it (see Configuring Admission Control Table on page 446).

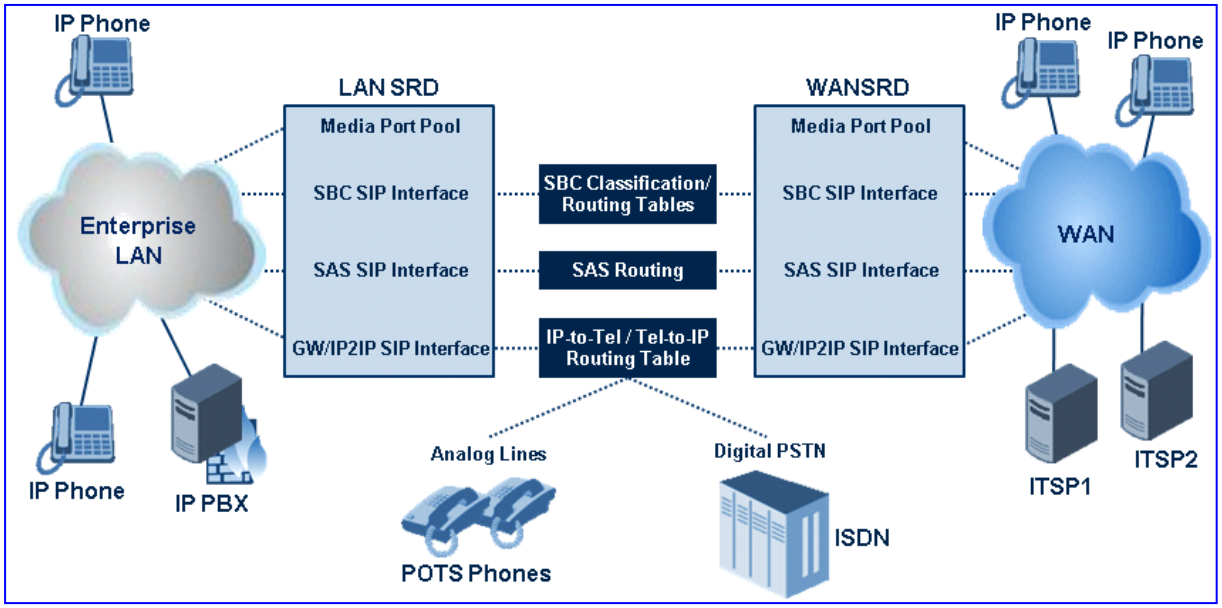
Figure 16-6: Configuring SRDs and Assignment



Typically, an SRD is defined per group of SIP UAs (e.g., proxies, IP phones, application servers, gateways, softswitches) that communicate with each other. This provides these entities with VoIP services that reside on the same Layer-3 network (must be able to communicate without traversing NAT devices and must not have overlapping IP addresses). Typically, one SRD is configured for the LAN and one SRD is configured for the WAN.

The figure below illustrates two SRD's - one for LAN and one for WAN. Each application (i.e., SAS, Gateway\IP2IP, and SBC) pertains to the same SRD, but each has its own SIP interface.

Figure 16-7: SIP Interfaces per Application using Multiple SRDs (Example)

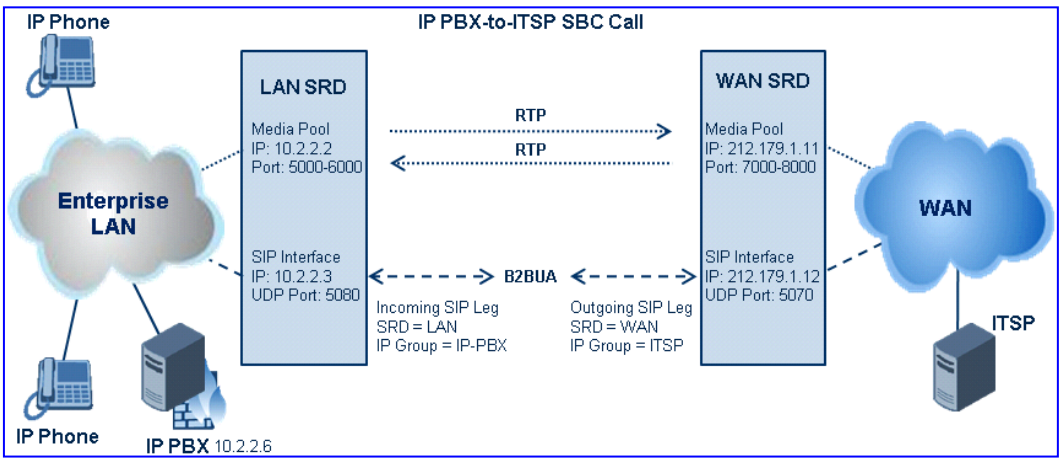


The figure below illustrates the SBC call flow between an enterprises LAN (IP PBX) and an ITSP (WAN) implementing different interfaces (IP addresses and ports) for RTP packets and SIP signaling. In addition, for each leg (LAN and WAN side), different interfaces are used.

The example uses the following IP addresses:

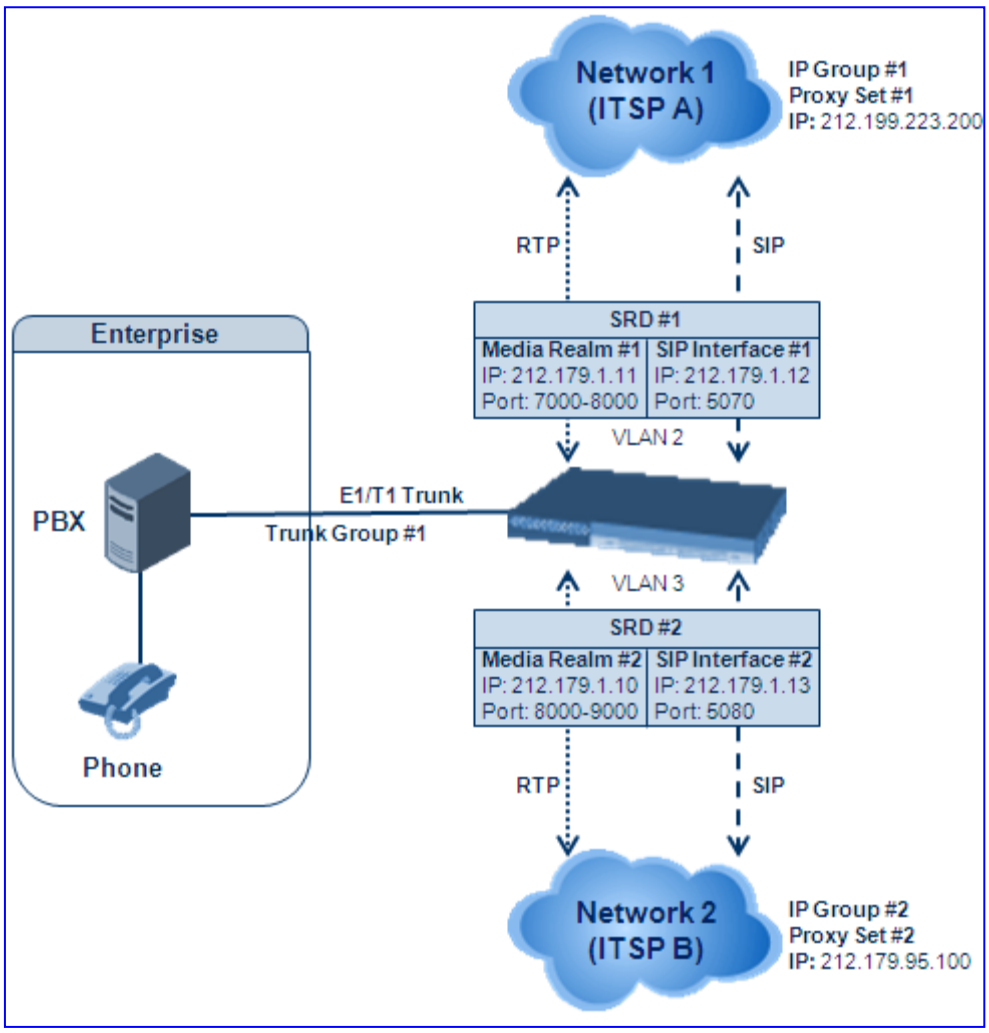
- IP-PBX: 10.2.2.6
- LAN MSBG: 10.2.2.3
- WAN MSBG: 212.179.1.12
- ITSP: 212.179.1.13
- MSBG LAN Media: 10.2.2.2:5000-6000
- MSBG WAN Media: 212.179.1.11:7000-8000

Figure 16-8: Back-to-Back SBC Call Flow (RTP and Signaling)



Below provides an example for configuring multiple SIP signaling and RTP interfaces. In this example, the device serves as the interface between the enterprise's PBX (connected using an E1/T1 trunk) and two ITSP's, as shown in the figure below:

Figure 16-9: Multiple SIP Signaling/RTP Interfaces Example

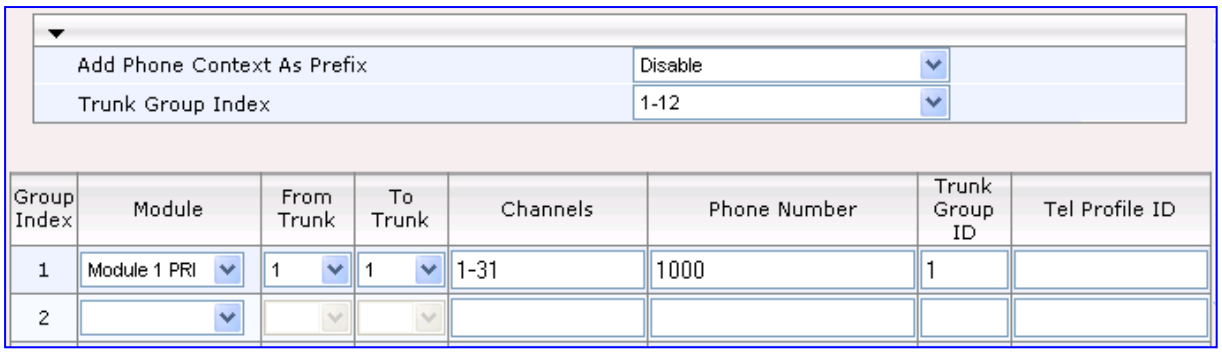


Note that only the steps specific to multiple SIP signaling/RTP configuration are described in detail in the procedure below.

➢ **To configure multiple SIP signaling and RTP interfaces:**

1. Configure Trunk Group ID #1 in the Trunk Group Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Trunk Group** > **Trunk Group**), as shown in the figure below:

Figure 16-10: Configuring Trunk Group 1



2. Configure the trunk in the Trunk Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Trunk Group** > **Trunk Group Settings**).
3. Configure the IP interfaces in the Multiple Interface table (**Configuration** tab > **VoIP** menu > **Network** submenu > **IP Settings**):

Figure 16-11: Defining IP Interfaces (Only Relevant Fields are Shown)

Index	Application Type	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP + Media + Control	192.168.0.2	24	192.168.0.1	1	Voice
1	Media	212.179.1.11	16	0.0.0.0	2	Media1
2	Media	212.179.1.10	16	0.0.0.0	3	Media2
3	Control	212.179.1.12	16	0.0.0.0	2	SIP1
4	Control	212.179.1.13	16	0.0.0.0	3	SIP2

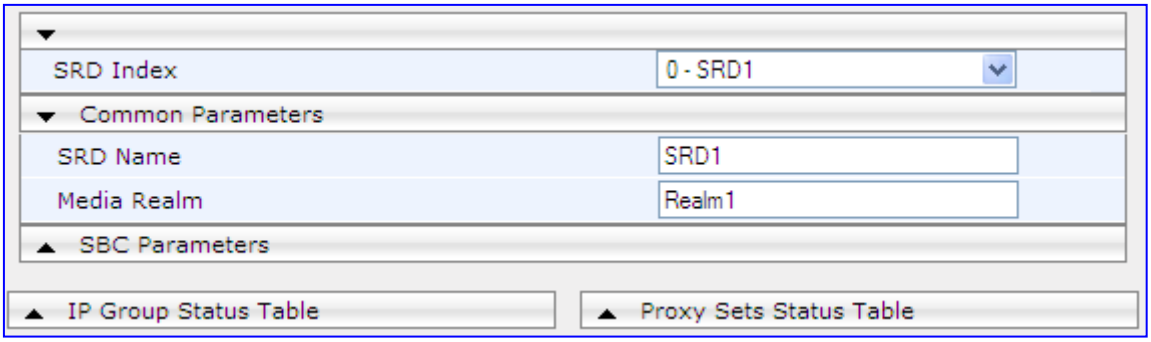
4. Configure Media Realms in the Media Realm table (**Configuration** tab > **VoIP** menu > **Media** submenu > **Media Realm Configuration**):

Figure 16-12: Defining Media Realms

Index	Media Realm Name	IPv4 Interface Name	Port Range Start	Number Of Media Session Legs	Port Range End
1	Realm1	Media1	7000	101	8000
2	Realm2	Media2	8020	20	8210

5. Configure SRDs in the SRD table (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **SRD Table**):
   - SRD1 associated with media realm "Realm1".
   - SRD2 associated with media realm "Realm2".

Figure 16-13: Defining SRDs



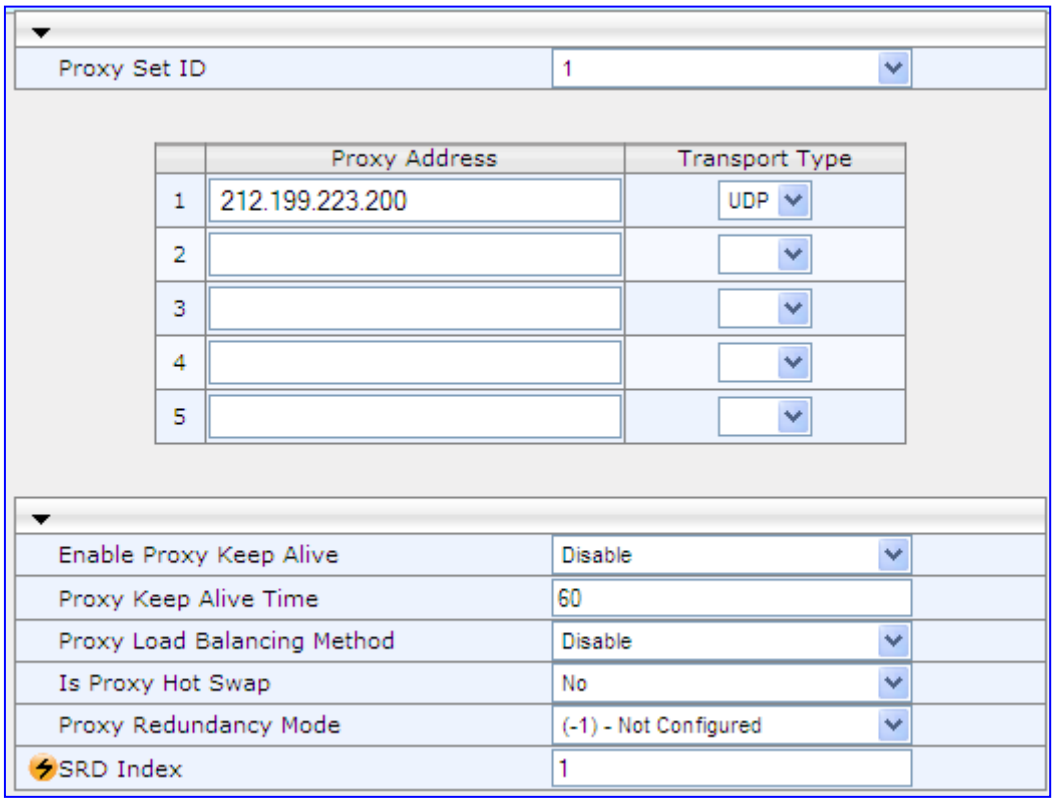
6. Configure the SIP Interfaces in the SIP Interface Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **SIP Interface Table**):

Figure 16-14: Defining SIP Interfaces

Index	Network Interface	Application Type	UDP Port	TCP Port	TLS Port	SRD
1	SIP1	GW\IP2IP	5070	5070	5071	1
2	SIP2	GW\IP2IP	5080	5080	5081	2

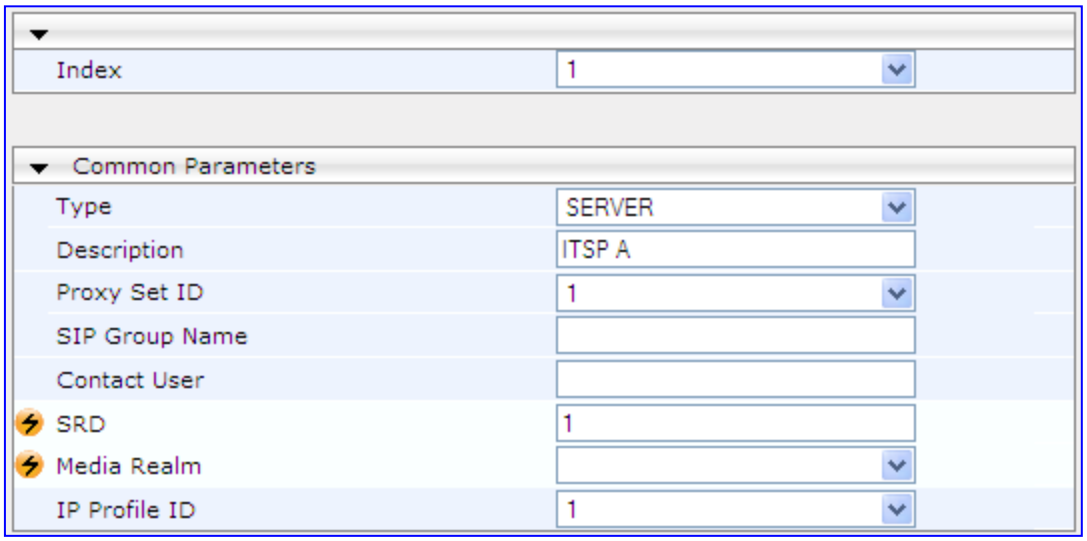
7. Configure Proxy Sets in the 'Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **Proxy Sets Table**). The figure below configures ITSP A. Do the same for ITSP B but for Proxy Set 2 with IP address 212.179.95.100 and SRD 2.

Figure 16-15: Defining Proxy Set



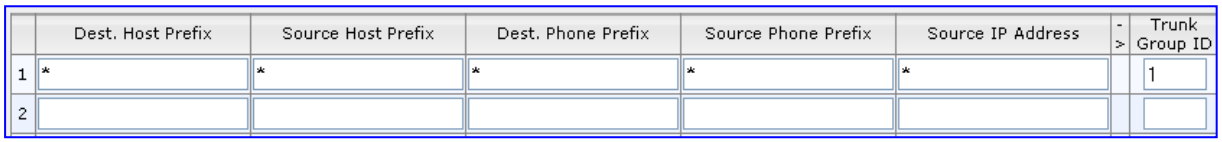
8. Configure IP Groups in the IP Group Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **IP Group Table**). The figure below configures IP Group for ITSP A. Do the same for ITSP B but for Index 2 with SRD 1 and Media Realm to "Realm2".

Figure 16-16: Defining IP Groups



9. Configure IP-to-Trunk Group routing in the Inbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **IP to Trunk Group Routing**):

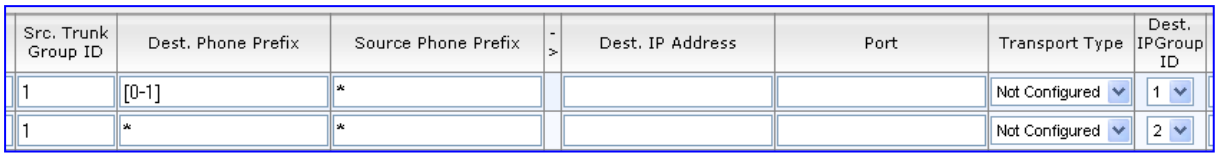
Figure 16-17: Defining IP-to-Trunk Group Routing



	Dest. Host Prefix	Source Host Prefix	Dest. Phone Prefix	Source Phone Prefix	Source IP Address	->	Trunk Group ID
1	*	*	*	*	*		1
2							

10. Configure Trunk Group-to-IP routing in the Outbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **Tel to IP Routing**):

Figure 16-18: Defining Trunk Group to IP Group Routing



Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Port	Transport Type	Dest. IPGroup ID
1	[0-1]	*				Not Configured	1
1	*	*				Not Configured	2

Reader's Notes

17 SIP Definitions

This section describes configuration of SIP parameters.

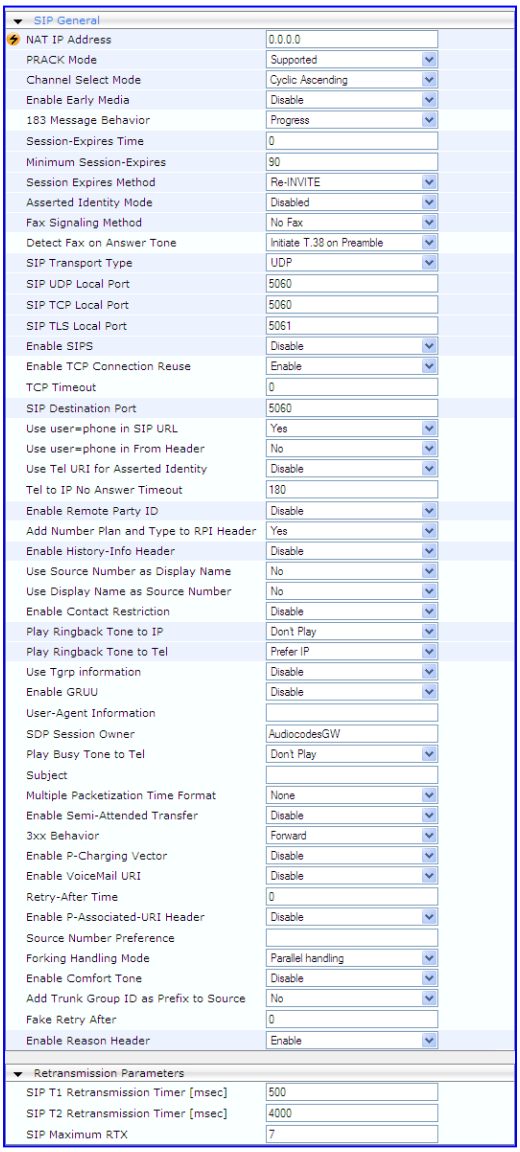
17.1 Configuring SIP General Parameters

The SIP General Parameters page is used to configure general SIP parameters. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

➢ **To configure general SIP parameters:**

1. Open the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **General Parameters**).

Figure 17-1: SIP General Parameters



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

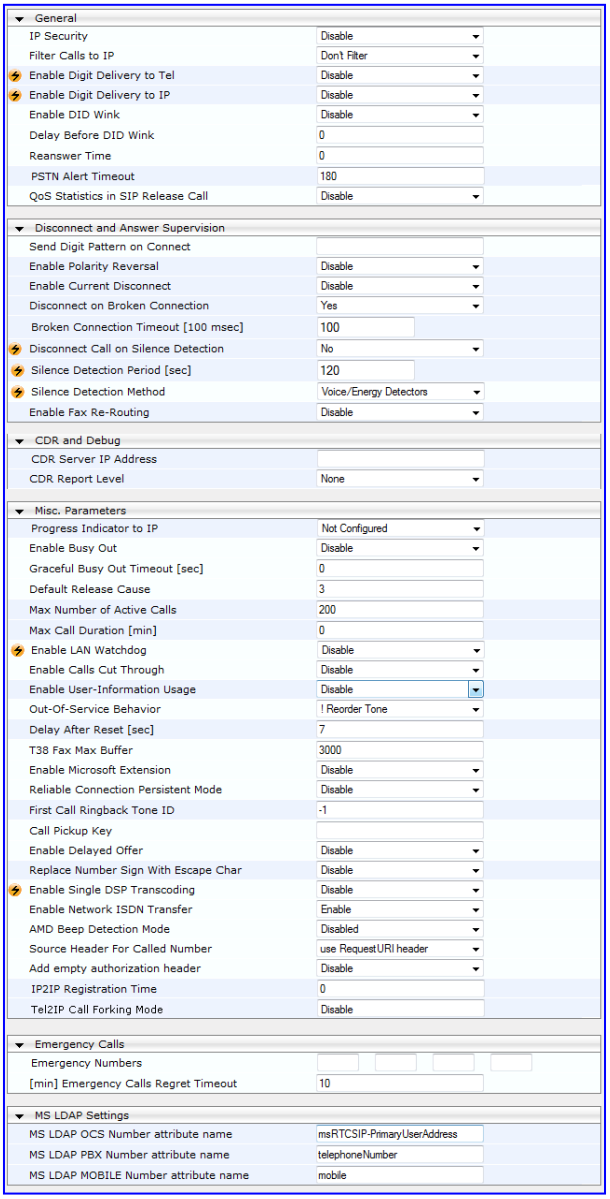
17.2 Configuring Advanced Parameters

The Advanced Parameters page allows you to configure advanced SIP control parameters. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

➢ **To configure advanced general protocol parameters:**

1. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Advanced Parameters**).

Figure 17-2: Advanced Parameters Parameters



2. Configure the parameters as required.
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

17.3 Configuring Account Table

The Account Table page allows you to define up to 32 *Accounts* per Trunk Group (*Served Trunk Group*) or source IP Group (Served IP Group). This is used for registration and/or digest authentication (user name and password) to a destination IP address (*Serving IP Group*). The Account table can be used, for example, to register to an ITSP on behalf of an IP-PBX to which the device is connected. The registrations are sent to the Proxy Set ID (see 'Configuring Proxy Sets Table' on page 219) associated with these Serving IP Groups.

A Trunk Group or source IP Group can register to more than one Serving IP Group (e.g., ITSP's). This can be achieved by configuring multiple entries in the Account table with the same Served Trunk Group or Served IP Group, but with different Serving IP Groups, user name/password, host name, and contact user values.

When using the Account table to register a Trunk Group (to a proxy server), if all trunks pertaining to the Trunk Group are down, the device un-registers the trunks. If any trunk belonging to the Trunk Group is returned to service, the device registers them again. This ensures, for example, that the Proxy does not send INVITEs to trunks that are out of service.

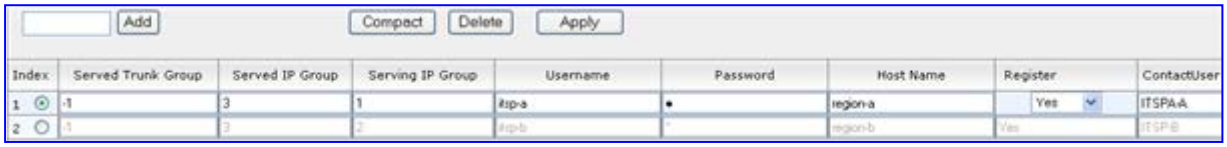
Notes:

- For viewing Account registration status, see 'Viewing Registration Status' on page 724.
- You can also configure the Account table using the *ini* file table parameter Account (see 'Configuration Parameters Reference' on page 752).

➢ **To configure Accounts:**

1. Open the Account Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Account Table**).

Figure 17-3: Account Table Page



2. To add an Account, in the 'Add' field, enter the desired table row index, and then click **Add**. A new row appears.
3. Configure the Account parameters according to the table below.
4. Click the **Apply** button to save your changes.
5. To save the changes, see 'Saving Configuration' on page 680.
6. To perform registration, click the **Register** button; to unregister, click **Unregister**. The registration method for each Trunk Group is according to the setting of the 'Registration Mode' parameter in the Trunk Group Settings page.

Note: For a description of the Web interface's table command buttons (e.g., **Duplicate** and **Delete**), see 'Working with Tables' on page 62.

Table 17-1: Account Table Parameters Description

Parameter	Description
Served Trunk Group **[Account_ServedTrunkGroup]**	The Trunk Group ID for which you want to register and/or authenticate to a destination IP Group (i.e., Serving IP Group). For Tel-to-IP calls, the Served Trunk Group is the source Trunk Group from where the call originated. For IP-to-Tel calls, the Served Trunk Group is the 'Trunk Group ID' defined in the Inbound IP Routing Table (see 'Configuring the Inbound IP Routing Table' on page 299). For defining Trunk Groups, see Configuring the Trunk Group Table on page 271. **Note:** For the IP2IP application, this parameter must be set to -1 (i.e., no trunk).
Served IP Group **[Account**_ServedIPGroup]	The Source IP Group (e.g., IP-PBX) for which registration and/or authentication is performed. **Note:** This field is applicable only when the IP2IP application is enabled.
Serving IP Group **[Account_ServingIPGroup]**	The destination IP Group ID (defined in 'Configuring IP Groups' on page 212) to where the REGISTER requests (if enabled) are sent or authentication is performed. The actual destination to where the REGISTER requests are sent is the IP address defined for the Proxy Set ID (see 'Configuring Proxy Sets Table' on page 219) associated with the IP Group. This occurs only in the following conditions: ▪ The parameter 'Registration Mode' is set to 'Per Account' in the Trunk Group Settings table (see 'Configuring Trunk Group Settings' on page 273). ▪ The parameter 'Register' in this table is set to 1. In addition, for a SIP call that is identified by both the Served Trunk Group/Served IP Group and Serving IP Group, the username and password for digest authentication defined in this table is used. For Tel-to-IP calls, the Serving IP Group is the destination IP Group defined in the Trunk Group Settings table or Outbound IP Routing Table (see 'Configuring the Outbound IP Routing Table' on page 291). For IP-to-Tel calls, the Serving IP Group is the 'Source IP Group ID' defined in the Inbound IP Routing Table (see 'Configuring the Inbound IP Routing Table' on page 299). **Note:** If no match is found in this table for incoming or outgoing calls, the username and password defined in the Authentication table for FXS interfaces (see Configuring Authentication on page 355) or by the global parameters UserName and Password (in the 'Proxy & Registration page) are used.
Username **[Account_Username]**	Digest MD5 Authentication user name (up to 50 characters).
Password **[Account_Password]**	Digest MD5 Authentication password (up to 50 characters). **Note:** After you click the **Apply** button, this password is displayed as an asterisk (*).
Host Name **[Account_HostName]**	Defines the Address of Record (AOR) host name. It appears in REGISTER From/To headers as ContactUser@HostName. For successful registrations, this HostName is also included in the INVITE request's From header URI. If not configured or if registration fails, the 'SIP Group Name' parameter from the 'IP

Parameter	Description
	Group' table is used instead. This parameter can be up to 49 characters.

Register
[Account_Register]
	Enables registration.
	▪ **[0]** No = Don't register
	▪ **[1]** Yes = Enables registration
	When enabled, the device sends REGISTER requests to the Serving IP Group. In addition, to activate registration, you also need to set the parameter 'Registration Mode' to 'Per Account' in the Trunk Group Settings table for the specific Trunk Group. The Host Name (i.e., host name in SIP From/To headers) and Contact User (user in From/To and Contact headers) are taken from this table upon a successful registration. See the example below:
	`REGISTER sip:xyz SIP/2.0`
	`Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac1397582418`
	`From: <sip:ContactUser@HostName>;tag=1c1397576231`
	`To: <sip: ContactUser@HostName >`
	`Call-ID: 139756895726120002256@10.33.37.78`
	`CSeq: 1 REGISTER`
	`Contact: <sip:ContactUser@10.33.37.78>;expires=3600`
	`Expires: 3600`
	`User-Agent: Sip-Gateway/v.6.00A.008.002`
	`Content-Length: 0`
	Notes:
	▪ The Trunk Group account registration is not affected by the parameter IsRegisterNeeded.
	▪ For the IP2IP application, you can configure this table so that a specific IP Group can register to multiple ITSP's. This is performed by defining several rows in this table containing the same Served IP Group, but with different Serving IP Groups, user/password, Host Name and Contact User parameters.
	▪ If registration to an IP Group(s) fails for **all** the accounts defined in this table for a specific Trunk Group, and if this Trunk Group includes all the channels in the Trunk Group, the Trunk Group is set to Out-Of-Service if the parameter OOSOnRegistrationFail is set to 1 (see 'Proxy & Registration Parameters' on page 239).

Contact User **[Account_ContactUser]**	Defines the AOR user name. It appears in REGISTER From/To headers as ContactUser@HostName, and in INVITE/200 OK Contact headers as ContactUser@<device's IP address>. If not configured, the 'Contact User' parameter in the IP Group Table page is used instead. **Note:** If registration fails, then the user part in the INVITE Contact header contains the source party number.
Application Type **[Account_ApplicationType]**	Defines the application type: ▪ **[0]** GW/IP2IP = Gateway and IP-to-IP application (default) ▪ **[2]** SBC = SBC application

17.4 Configuring Proxy and Registration Parameters

The Proxy & Registration page allows you to configure the Proxy server and registration parameters. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

Note: To view whether the device or its endpoints have registered to a SIP Registrar/Proxy server, see 'Viewing Registration Status' on page 724.

➢ **To configure the Proxy and registration parameters:**

1. Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Proxy & Registration**).

Figure 17-4: Proxy & Registration Page

2. Configure the parameters as required.
3. Click Submit to apply your changes.

4. Click the **Register** or **Un-Register** buttons to save your changes and register/unregister the device to a Proxy/Registrar. Instead of registering the entire device, you can register specific entities (FXS/FXO endpoints, Trunk Groups, BRI endpoints, and Accounts), by using the **Register** button located on the page in which these entities are configured.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Click the **Proxy Set Table** button to Open the Proxy Sets Table page to configure groups of proxy addresses. Alternatively, you can open this page from the **Proxy Sets Table** page item (see 'Configuring Proxy Sets Table' on page 219 for a description of this page).

Reader's Notes

18 Coders and Profiles

This section describes configuration of the coders and SIP profiles parameters.

18.1 Configuring Coders

The Coders page allows you to configure up to 10 voice coders for the device to use. Each coder can be configured with packetization time (ptime), rate, payload type, and silence suppression.

The first coder in the table has the highest priority and is used by the device whenever possible. If the remote side cannot use the first coder, the device attempts to use the next coder in the table, and so on.

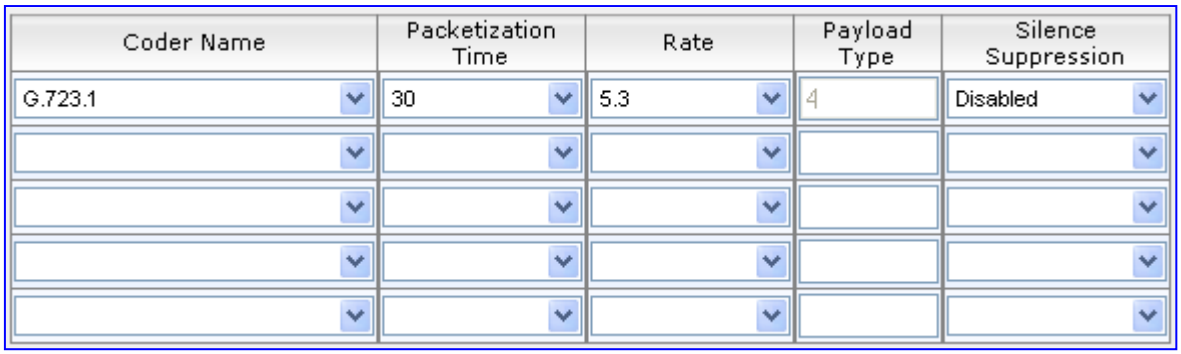
Notes:

- For a list of supported coders and for configuring coders using the *ini* file, refer to the *ini* file parameter table CodersGroup, described in 'Configuration Parameters Reference' on page 752.
- Each voice coder can appear only once in the table.
- If packetization time and/or rate are not specified, the default value is applied.
- Only the packetization time of the first coder in the coder list is declared in INVITE/200 OK SDP, even if multiple coders are defined.
- The device always uses the packetization time requested by the remote side for sending RTP packets.
- For G.729, it's also possible to select silence suppression without adaptations.
- If the coder G.729 is selected with silence suppression is disabled, the device includes 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is enabled or set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception to this logic is when the remote gateway is a Cisco device (IsCiscoSCEMode).
- For defining groups of coders, which can be assigned to Tel and IP Profiles, see 'Configuring Coder Groups' on page 244.
- For information on V.152 and implementation of T.38 and VBD coders, see 'Supporting V.152 Implementation' on page 162.

➢ **To configure the device's coders:**

1. Open the Coders page (**Configuration** tab > **VoIP** menu > **Coders And Profiles** submenu > **Coders**).

Figure 18-1: Coders Page



2. From the 'Coder Name' drop-down list, select the required coder.
3. From the 'Packetization Time' drop-down list, select the packetization time (in msec) for the selected coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
4. From the 'Rate' drop-down list, select the bit rate (in kbps) for the selected coder.
5. In the 'Payload Type' field, if the payload type (i.e., format of the RTP payload) for the selected coder is dynamic, enter a value from 0 to 120 (payload types of 'well-known' coders cannot be modified).
6. From the 'Silence Suppression' drop-down list, enable or disable the silence suppression option for the selected coder.
7. Repeat steps 2 through 6 for the next optional coders.
8. Click Subm**it to** apply your changes.
9. To save the changes to flash memory, see 'Saving Configuration' on page 680.

18.2 Configuring Coder Groups

The Coder Group Settings page allows you to define up to four groups of coders (termed*Coder Groups*). For each Coder Group, you can define up to 10 coders, configured with packetization time (ptime), rate, payload type, and silence suppression. The first coder in the Coder Group table is the highest priority coder and is used by the device whenever possible. If the remote side cannot use the first coder, the device attempts to use the next coder, and so on.

Coder Groups can be used as follows:

- Assigned to Tel Profiles in the Tel Profiles table (see Configuring Tel Profiles on page 246). This is applicable only to the GW/IP2IP application.
- Assigned to IP Profiles in the IP Profiles table (see 'Configuring IP Profiles' on page 248). In this setup, they can be used as Extension coders and Allowed coders for the SBC application.

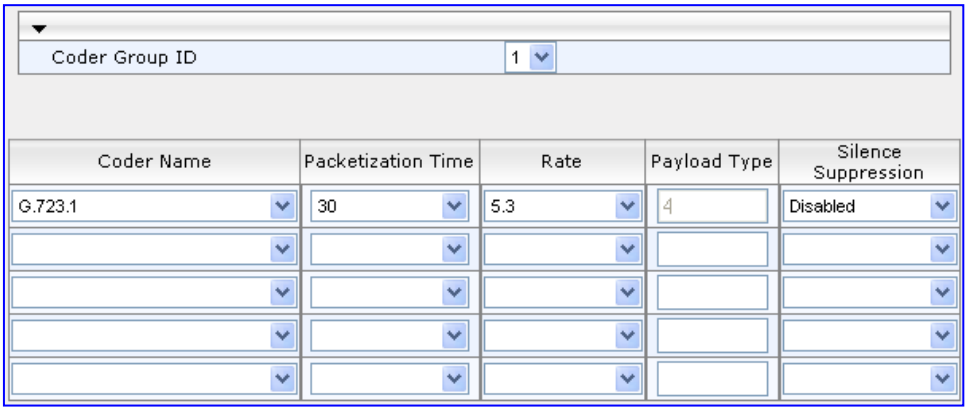
Notes:

- Each voice coder can appear only once per Coder Group.
- For a list of supported coders and for configuring coders using the *ini* file, refer to the *ini* file parameter table CodersGroup, described in 'Configuration Parameters Reference' on page 752.
- For information on coders, refer to the notes in 'Configuring Coders' on page 243.

➢ **To configure Coder Groups:**

1. Open the Coder Group Settings page (**Configuration** tab > **VoIP** menu > **Coders And Profiles** submenu > **Coders Group Settings**).

Figure 18-2: Coder Group Settings Page



2. From the 'Coder Group ID' drop-down list, select a Coder Group ID.
3. From the 'Coder Name' drop-down list, select the first coder for the Coder Group.
4. From the 'Packetization Time' drop-down list, select the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
5. From the 'Rate' drop-down list, select the bit rate (in kbps) for the coder you selected.
6. In the 'Payload Type' field, if the payload type (i.e., format of the RTP payload) for the coder you selected is dynamic, enter a value from 0 to 120 (payload types of 'well-known' coders cannot be modified).
7. From the 'Silence Suppression' drop-down list, enable or disable the silence suppression option for the coder you selected.
8. Repeat steps 3 through 7 for the next coders (optional).
9. Repeat steps 2 through 8 for the next coder group (optional).
10. Click Submit to apply your changes.
11. To save the changes to flash memory, see 'Saving Configuration' on page 680.

18.3 Configuring Tel Profile

The Tel Profile Settings page allows you to define up to nine SIP profiles for Tel calls (termed *Tel Profiles*). Each Tel Profile contains a set of parameters for configuring various behaviors, for example, used coder, silence suppression support, and echo canceler. Once configured, Tel Profiles can then be assigned to specific trunks (channels). For example, specific channels can be assigned a Tel Profile that must use the G.711 coder. Thus, implementing Tel Profiles provides high-level adaptation when connected to a variety of equipment and protocols (at both Tel and IP sides), each of which may require different system behavior.

The Tel Profiles are assigned to the device's channels in the Trunk Group Table (see Configuring the Trunk Group Table on page 271)).

Note: You can also configure Tel Profiles using the *ini* file table parameter TelProfile (see 'Configuration Parameters Reference' on page 752).

➢ **To configure Tel Profiles:**

1. Open the Tel Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders And Profiles** submenu > **Tel Profile Settings**).

Figure 18-3: Tel Profile Settings Page

2. From the 'Profile ID' drop-down list, select the Tel Profile index.
3. In the 'Profile Name' field, enter an arbitrary name that enables you to easily identify the Tel Profile.
4. From the 'Profile Preference' drop-down list, select the priority of the Tel Profile, where **1** is the lowest priority and **20** the highest. If both IP and Tel profiles apply to the same call, the coders and other common parameters (noted by an asterisk in the description of the parameter TelProfile) of the preferred Profile are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied.
   Note: If the coder lists of both IP and Tel Profiles apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.
5. Configure the parameters as required. For more information on each parameter, refer to the description of the "global" parameter.
6. From the 'Coder Group' drop-down list, select the Coder Group (see 'Configuring Coder Groups' on page 244) or the device's default coder (see 'Configuring Coders' on page 243) to which you want to assign the Tel Profile.
7. Click Subm**it to** apply your changes.
8. To save the changes to flash memory, see 'Saving Configuration' on page 680.

18.4 Configuring IP Profiles

The IP Profile Settings page allows you to define up to nine SIP profiles for IP calls (termed *IP Profile*). Each IP Profile contains a set of parameters for configuring various behaviors, for example, used coder, echo canceller support, and jitter buffer. Once configured, different IP Profiles can be assigned to specific inbound and outbound calls. For example, specific calls can be assigned an IP Profile that must use the G.711 coder. Thus, implementing IP Profiles provides high-level adaptation when connected to a variety of equipment and protocols (at both Tel and IP sides), each of which may require different system behavior.

The IP Profiles can be used in the following tables:

- Outbound IP Routing Table - see 'Configuring Outbound IP Routing Table' on page 291
- Inbound IP Routing Table - see 'Configuring Inbound IP Routing Table' on page 299
- IP Group table - see 'Configuring IP Groups' on page 212

The IP Profile Settings page conveniently groups parameters according to application to which they pertain:

- Common Parameters - parameters common to all application types
- Gateway Parameters - parameters applicable to the GW (gateway) application
- SBC Parameters - parameters applicable to the SBC application

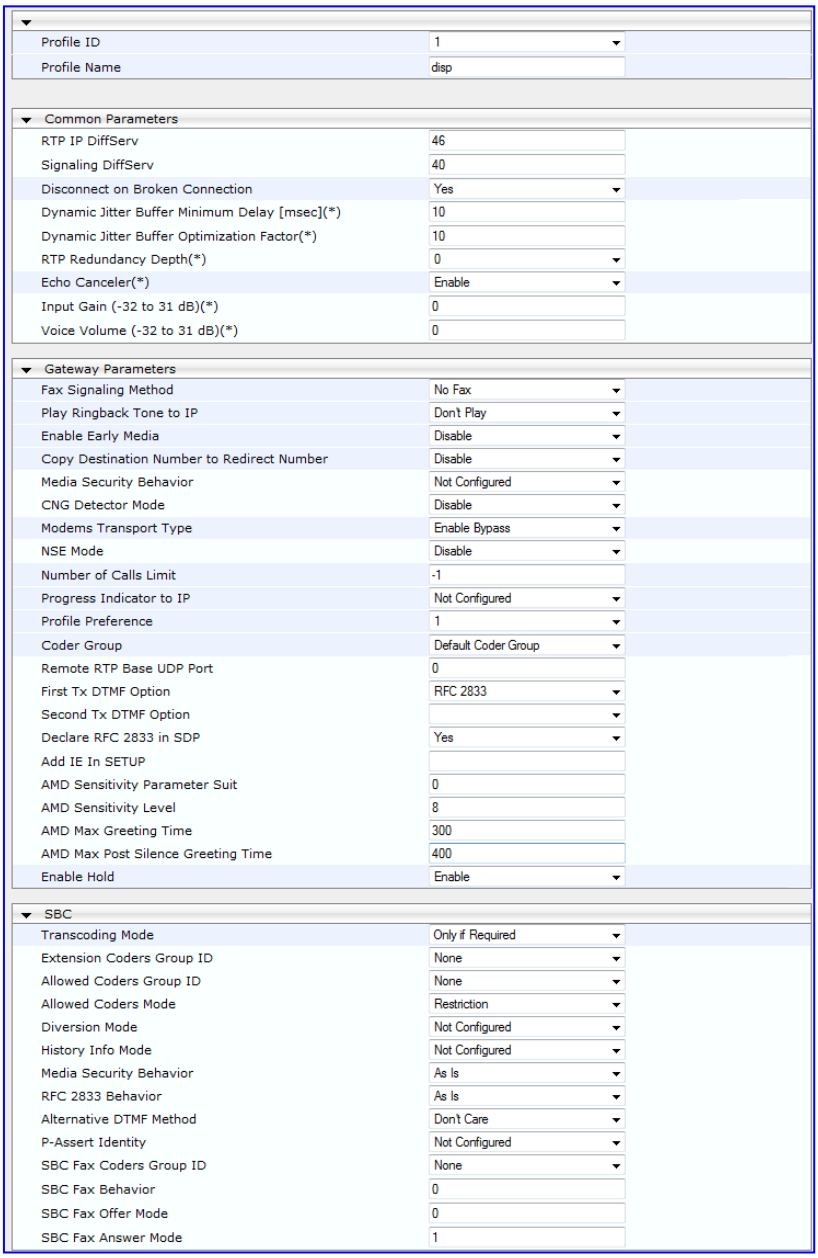
Notes:

- For a detailed description of each IP Profile parameter, refer to its corresponding "global" parameter (configured as an individual parameter).
- IP Profiles can also be implemented when operating with a Proxy server (when the AlwaysUseRouteTable parameter is set to 1).
- You can use IP Profiles in the IP Group table, Outbound IP Routing table, and Inbound IP Routing table. The device selects the IP Profile as follows:
  1) If different IP Profiles (not default) are assigned to these tables, the device uses the IP Profile with the highest preference level (as set in the 'Profile Preference' field). If they have the same preference level, the device uses the IP Profile assigned to the IP Group table.
  2) If different IP Profiles are assigned to these tables and one table is set to the default IP Profile, the device uses the IP Profile that is not the default.
- You can also configure IP Profiles using the *ini* file table parameter IPProfile (see 'Configuration Parameters Reference' on page 752).

➢ To configure IP Profiles:

1. Open the IP Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders And Profiles** submenu > **IP Profile Settings**).

Figure 18-4: IP Profile Settings Page



2. From the 'Profile ID' drop-down list, select the IP Profile index.
3. In the 'Profile Name' field, enter an arbitrary name that allows you to easily identify the IP Profile.
4. From the 'Profile Preference' drop-down list, select the priority of the IP Profile, where '1' is the lowest priority and '20' is the highest. If both IP and Tel profiles apply to the same call, the coders and other common parameters (noted by an asterisk) of the preferred Profile are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied.
   Note: If the coder lists of both IP and Tel Profiles apply to the same call, only the

coders common to both are used. The order of the coders is determined by the preference.

5. Configure the parameters as required.
6. From the 'Coder Group' drop-down list, select the coder group that you want to assign to the IP Profile. You can select the device's default coders (see 'Configuring Coders' on page 243), or one of the coder groups you defined in the Coder Group Settings page (see 'Configuring Coder Groups' on page 244).
7. Click **Submit** to apply your changes.
8. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19 GW and IP to IP

This section describes configuration for the GW/IP2IP applications.

Note: The "GW" and "IP2IP" applications refer to the Gateway and IP-to-IP applications respectively.

19.1 Digital PSTN

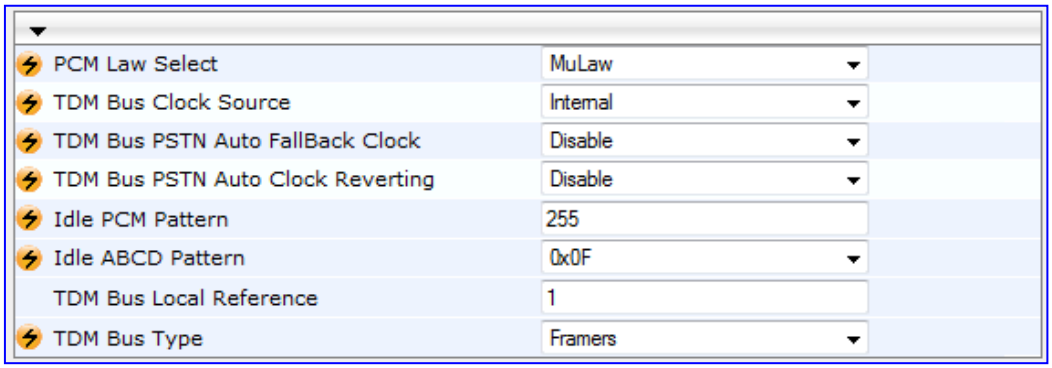
This section describes configuration of the public switched telephone network (PSTN) parameters.

19.1.1 Configuring TDM Bus Settings

The TDM Bus Settings page allows you to configure the device's Time-Division Multiplexing (TDM) bus settings. For a description of these parameters, see 'Configuration Parameters Reference' on page 752.

➢ **To configure the TDM Bus settings:**

1. Open the TDM Bus Settings page (**Configuration** tab > **VoIP** menu > **TDM** submenu > **TDM Bus Settings**).



2. Configure the parameters as required.
3. Click **Submit** to apply your changes.
4. Save the changes to flash memory (see 'Saving Configuration' on page 680).

19.1.2 Configuring CAS State Machines

The CAS State Machine page allows you to modify various timers and other basic parameters to define the initialization of the CAS state machine without changing the state machine itself (no compilation is required). The change doesn't affect the state machine itself, but rather the configuration.

The CAS table used can be chosen in two ways (using the parameter CasChannelIndex):

- Single CAS table per trunk
- Different CAS table per group of B-Channels in a trunk

Note: The CAS state machine can only be configured using the Web-based management tool.

➢ **To modify the CAS state machine parameters:**

1. Open the CAS State Machine page (**Configuration** tab > **VoIP** menu > **PSTN** submenu > **CAS State Machines**).

Figure 19-1: CAS State Machine Page

2. Ensure that the trunk is inactive. The trunk number displayed in the 'Related Trunks' field must be green. If it is red (indicating that the trunk is active), click the trunk number to open the Trunk Settings page (see 'Configuring Trunk Settings' on page 254), select the required Trunk number icon, and then click **Stop Trunk**.
3. In the CAS State Machine page, modify the required parameters according to the table below.
4. Once you have completed the configuration, activate the trunk if required in the Trunk Settings page, by clicking the trunk number in the 'Related Trunks' field, and in the Trunk Settings page, select the required Trunk number icon, and then click **Apply Trunk Settings**.
5. Click **Submit**, and then reset the device (see 'Resetting the Device' on page 677).

Notes:

- Don't modify the default values unless you fully understand the implications of the changes and know the default values. Every change affects the configuration of the state machine parameters and the call process related to the trunk you are using with this state machine.
- You can modify CAS state machine parameters only if the following conditions are met:
  1) Trunks are inactive (stopped), i.e., the 'Related Trunks' field displays the trunk number in green.
  2) State machine is not in use or is in reset, or when it is not related to any trunk. If it is related to a trunk, you must delete the trunk or de-activate (*Stop*) the trunk.
- Field values displaying '-1' indicate CAS default values. In other words, CAS state machine values are used.
- The modification of the CAS state machine occurs at the CAS application initialization only for non-default values (-1).
- For more information on the CAS Protocol table, refer to the *Product Reference Manual*.

Table 19-1: CAS State Machine Parameters Description

Parameter	Description
Generate Digit On Time **[CasStateMachineGenerateDigitOnTime]**	Generates digit on-time (in msec). The value must be a positive value. The default value is -1 (use value from CAS state machine).
Generate Inter Digit Time **[CasStateMachineGenerateInterDigitTime]**	Generates digit off-time (in msec). The value must be a positive value. The default value is -1 (use value from CAS state machine).
DTMF Max Detection Time **[CasStateMachineDTMFMaxOnDetectionTime]**	Detects digit maximum on time (according to DSP detection information event) in msec units. The value must be a positive value. The default value is -1 (use value from CAS state machine).
DTMF Min Detection Time **[CasStateMachineDTMFMinOnDetectionTime]**	Detects digit minimum on time (according to DSP detection information event) in msec units. The digit time length must be longer than this value to receive a detection. Any number may be used, but the value must be less than CasStateMachineDTMFMaxOnDetectionTime. The value must be a positive value. The default value is -1 (use value from CAS state machine).
MAX Incoming Address Digits **[CasStateMachineMaxNumOfIncomingAddressDigits]**	Defines the limitation for the maximum address digits that need to be collected. After reaching this number of digits, the collection of address digits is stopped. The value must be an integer. The default value is -1 (use value from CAS state machine).
MAX Incoming ANI Digits **[CasStateMachineMaxNumOfIncomingANIDigits]**	Defines the limitation for the maximum ANI digits that need to be collected. After reaching this number of digits, the collection of ANI digits is stopped. The value must be an integer. The default value is -1 (use value from CAS state machine).
Collet ANI **[CasStateMachineCollectANI]**	In some cases, when the state machine handles the ANI collection (not related to MFCR2), you can control the state machine to collect ANI or discard ANI. ▪ **[0]** No = Don't collect ANI. ▪ **[1]** Yes = Collect ANI. ▪ **[-1]** Default = Default value - use value from CAS state machine.
Digit Signaling System **[CasStateMachineDigitSignalingSystem]**	Defines which Signaling System to use in both directions (detection\generation). ▪ **[0]** DTMF = Uses DTMF signaling. ▪ **[1]** MF = Uses MF signaling (default). ▪ **[-1]** Default = Default value - use value from CAS state machine.

19.1.3 Configuring Trunk Settings

The Trunk Settings page allows you to configure the device's trunks. This includes selecting the PSTN protocol and configuring related parameters.

Some parameters can be configured when the trunk is in service, while others require you to take the trunk out of service (by clicking the **Stop** button). Once you have "stopped" a trunk, all calls are dropped and no new calls can be made on that trunk.

You can also deactivate a trunk (by clicking the **Deactivate** Deactivate button) for maintenance. Deactivation temporarily disconnects (logically) the trunk from the PSTN network. Upon trunk deactivation, the device generates an AIS alarm on that trunk to the far-end (as a result, an RAI alarm signal may be received by the device). A subsequent trunk activation (by clicking the **Activate** Activate button), reconnects the trunk to the PSTN network and clears the AIS alarm. Trunk deactivation is typically used for maintenance such as checking the trunk's physical integrity.

For a description of the trunk parameters, see 'PSTN Parameters' on page 896.

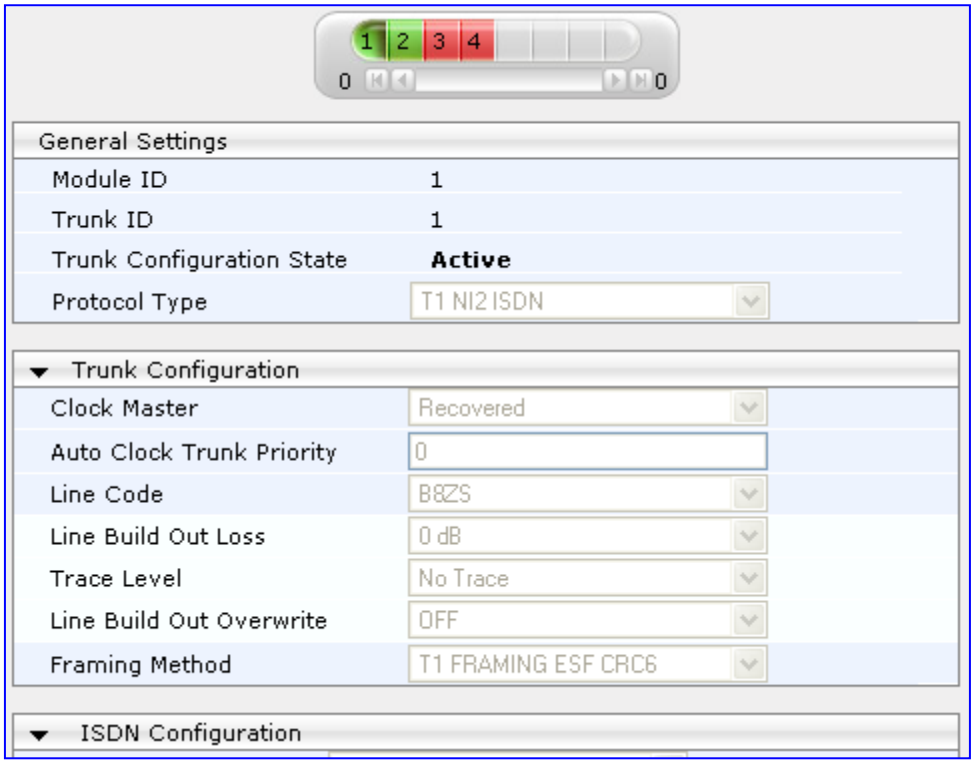
Notes:

- During trunk deactivation, trunk configuration cannot be performed.
- A stopped trunk cannot also be activated and a trunk cannot be deactivated if it has been stopped.

➢ **To configure the trunks:**

1. Open the Trunk Settings page (**Configuration** tab > **VoIP** menu > **PSTN** submenu > **Trunk Settings**).

Figure 19-2: Trunk Settings Page

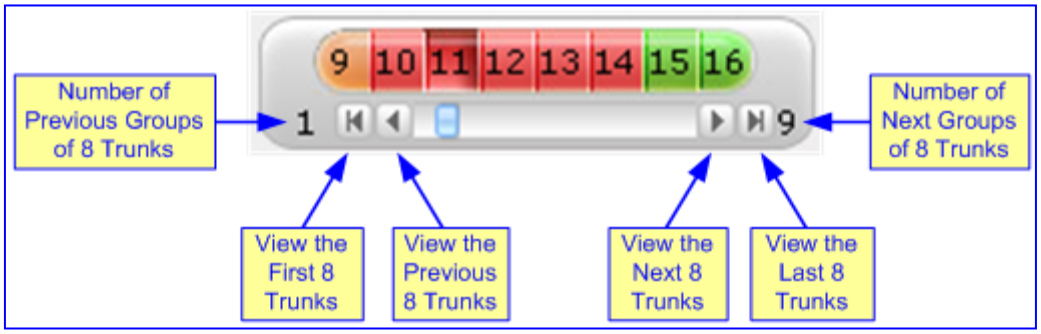


On the top of the page, a bar with Trunk number icons displays the status of each trunk, according to the following color codes:

- **Grey:** Disabled
- **Green:** Active
- **Yellow:** RAI alarm (also appears when you deactivate a Trunk by clicking the **Deactivate** button)
- **Red:** LOS/LOF alarm
- **Blue:** AIS alarm
- **Orange:** D-channel alarm (ISDN only)

2. Select the trunk that you want to configure by clicking the desired Trunk number icon. The bar initially displays the first eight trunk number icons (i.e., trunks 1 through 8). To scroll through the trunk number icons (i.e., view the next/last or previous/first group of eight trunks), refer to the figure below:

Figure 19-3: Trunk Scroll Bar (Used Only as an Example)



Note: If the Trunk scroll bar displays all available trunks, the scroll bar buttons are unavailable.

After you have selected a trunk, the following is displayed:

- The read-only 'Module ID' field displays the module number to which the trunk belongs.
- The read-only 'Trunk ID' field displays the selected trunk number.
- The read-only 'Trunk Configuration State' displays the state of the trunk ('Active' or 'Inactive').
- The displayed parameters pertain to the selected trunk only.

3. Click the **Stop Trunk** button (located at the bottom of the page) to take the trunk out of service so that you can configure the currently grayed out (unavailable) parameters. (Skip this step if you want to configure parameters that are available when the trunk is active). The stopped trunk is indicated by the following:
   - The 'Trunk Configuration State' field displays 'Inactive'.
   - The **Stop Trunk** button is replaced by the **Apply Trunk Settings** button.

     When all trunks are stopped, the **Apply to All Trunks** button also appears.
   - All the parameters are available and can be modified.
4. Configure the trunk parameters as required.

5. Click the **Apply Trunk Settings** button to apply the changes to the selected trunk (or click **Apply to All Trunks** to apply the changes to all trunks); the **Stop Trunk** button replaces **Apply Trunk Settings** and the 'Trunk Configuration State' displays 'Active'.
6. To save the changes to flash memory, see 'Saving Configuration' on page 680.
7. To reset the device, see 'Resetting the Device' on page 677.

Notes:

- If the 'Protocol Type' field displays 'NONE' (i.e., no protocol type is selected) and no other trunks have been configured, after selecting a PRI protocol type, you must reset the device.
- The displayed parameters depend on the protocol selected.
- All PRI trunks of the device must be of the same line type (i.e., E1 or T1). However, different variants of the same line type can be configured on different trunks, for example, E1 Euro ISDN and E1 CAS (subject to the constraints in the device's Release Notes).
- BRI trunks can operate with E1 or T1 trunks.
- If the protocol type is CAS, you can assign or modify a dial plan (in the 'Dial Plan' field) and perform this without stopping the trunk.
- If the trunk can't be stopped because it provides the device's clock (assuming the device is synchronized with the E1/T1 clock), assign a different E1/T1 trunk to provide the device's clock or enable 'TDM Bus PSTN Auto Clock' in the TDM Bus Settings page (see Configuring TDM Bus Settings on page 251).
- To delete a previously configured trunk, set the parameter 'Protocol Type' to 'None'.

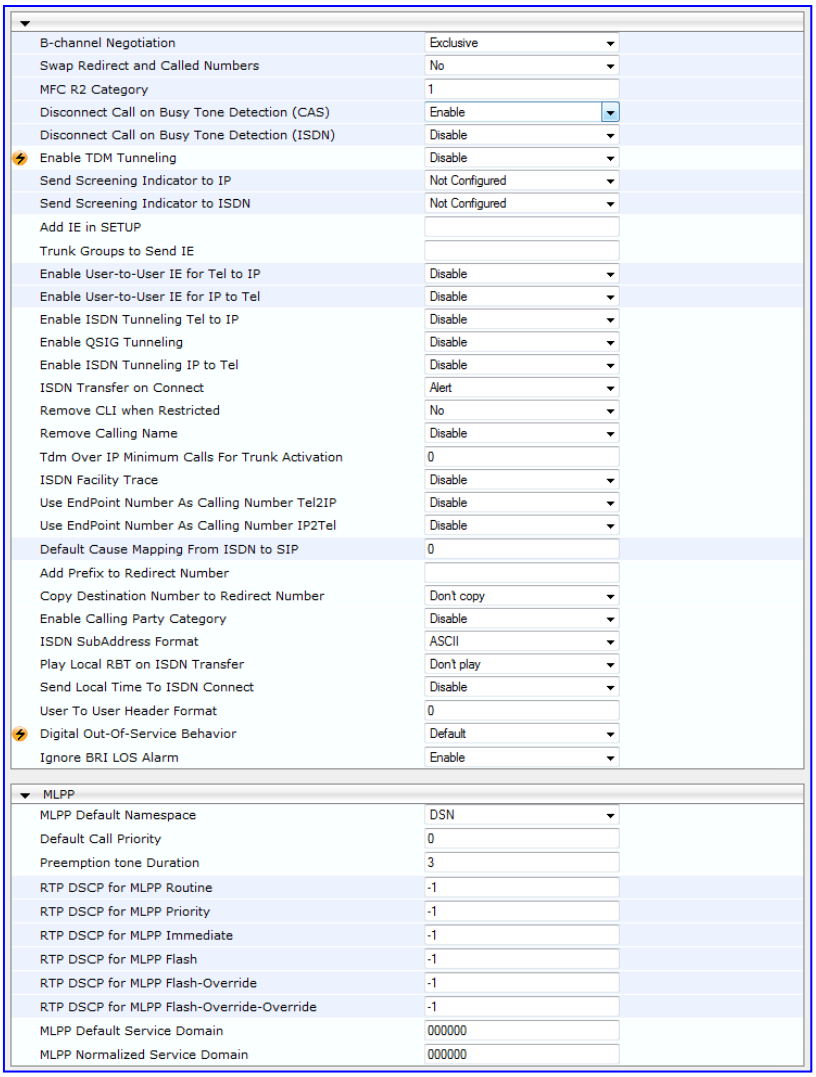
19.1.4 Configuring Digital Gateway Parameters

The Digital Gateway Parameters page allows you to configure miscellaneous digital parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 752.

➢ **To configure the digital gateway parameters:**

1. Open the Digital Gateway Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Digital Gateway** submenu > **Digital Gateway Parameters**).

Figure 19-4: Digital Gateway Parameters Page



2. Configure the parameters as required.
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.1.5 Tunneling Applications

This section discusses the device's support for VoIP tunneling applications.

19.1.5.1 TDM Tunneling

The device's TDM Tunneling feature allows you to tunnel groups of digital trunk spans or timeslots (B-channels) over the IP network. TDM Tunneling utilizes the device's internal routing (without Proxy control) capabilities to receive voice and data streams from TDM (E1/T1/J1/) spans or individual timeslots, convert them into packets, and then transmit them over the IP network (using point-to-point or point-to-multipoint device distributions). A device opposite it (or several devices when point-to-multipoint distribution is used) converts the IP packets back into TDM traffic. Each timeslot can be targeted to any other timeslot within a trunk in the opposite device.

When TDM Tunneling is enabled (the parameter EnableTDMoverIP is set to '1') on the originating device, the originating device automatically initiates SIP calls from all enabled B-channels belonging to the E1/T1/J1 spans that are configured with the protocol type 'Transparent' (for ISDN trunks) or 'Raw CAS' (for CAS trunks). The called number of each call is the internal phone number of the B-channel from where the call originates. The Inbound IP Routing Table is used to define the destination IP address of the terminating device. The terminating device automatically answers these calls if its E1/T1 protocol type is set to 'Transparent' (ProtocolType = 5) or 'Raw CAS' (ProtocolType = 3 for T1 and 9 for E1) and the parameter ChannelSelectMode is set to 0 (By Phone Number).

Note: It's possible to configure both devices to also operate in symmetric mode. To do so, set EnableTDMOverIP to 1 and configure the Outbound IP Routing Table in both devices. In this mode, each device (after it's reset) initiates calls to the second device. The first call for each B-channel is answered by the second device.

The device continuously monitors the established connections. If for some reason, one or more calls are released, the device automatically re-establishes these 'broken' connections. In addition, when a failure in a physical trunk or in the IP network occurs, the device re-establishes the tunneling connections when the network is restored.

Note: It's recommended to use the keep-alive mechanism for each connection, by activating the 'session expires' timeout and using Re-INVITE messages.

The device supports the configuration (TDMoIPInitiateInviteTime and TDMoIPInviteRetryTime parameters) of the following timers for the TDM-over-IP tunneling application:

- Time between successive INVITEs sent from the same E1/T1 trunk.
- Time between call release and the new INVITE that is sent on the same channel. The call can be released if the device receives a 4xx or 5xx response.

By utilizing the 'Profiles' mechanism (see 'Coders and Profiles' on page 243), you can configure the TDM Tunneling feature to choose different settings based on a timeslot or groups of timeslots. For example, you can use low-bit-rate vocoders to transport voice and 'Transparent' coder to transport data (e.g., for D-channel). You can also use Profiles to assign ToS (for DiffServ) per source - a timeslot carrying data or signaling is assigned a higher priority value than a timeslot carrying voice.

For tunneling of E1/T1 CAS trunks, set the protocol type to 'Raw CAS' (ProtocolType = 3 / 9) and enable RFC 2833 CAS relay mode ('CAS Transport Type' parameter is set to 'CAS RFC2833 Relay').

Note: For TDM over IP, the parameter CallerIDTransportType must be set to '0' (disabled), i.e., transparent.

Below is an example of *ini* files for two devices implementing TDM Tunneling for four E1 spans. Note that in this example both devices are dedicated to TDM tunneling.

Terminating Side:

```
EnableTDMOverIP = 1
;E1_TRANSPARENT_31
ProtocolType_0 = 5
ProtocolType_1 = 5
ProtocolType_2 = 5
ProtocolType_3 = 5
[PREFIX]
PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix,
PREFIX_ProfileId, PREFIX_MeteringCode, PREFIX_DestPort,
PREFIX_SrcIPGroupID, PREFIX_DestHostPrefix, PREFIX_DestIPGroupID,
PREFIX_SrcHostPrefix, PREFIX_TransportType,
PREFIX_SrcTrunkGroupID, PREFIX_DestSRD, PREFIX_CostGroup,
PREFIX_ForkingGroup;
Prefix 1 = *,10.8.24.12;
[\PREFIX]

;IP address of the device in the opposite
;location
;Channel selection by Phone number.
ChannelSelectMode = 0
;Profiles can be used do define different coders per B-channels
;such as Transparent
;coder for B-channels (timeslot 16) that carries PRI ;signaling.
[TrunkGroup]
FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum,
TrunkGroup_FirstTrunkId, TrunkGroup_LastTrunkId,
TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel,
TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId,
TrunkGroup_Module;
TrunkGroup 1 = 0,0,0,1,31,1000,1;
TrunkGroup 1 = 0,1,1,1,31,2000,1;
TrunkGroup 1 = 0,2,2,1,31,3000,1;
TrunkGroup 1 = 0,3,3,1,31,4000,1;
TrunkGroup 1 = 0,0,0,16,16,7000,2;
TrunkGroup 1 = 0,1,1,16,16,7001,2;
TrunkGroup 1 = 0,2,2,16,16,7002,2;
TrunkGroup 1 = 0,3,3,16,16,7003,2;
[/TrunkGroup]
[ CodersGroup0 ]
FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime,
CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce;
CodersGroup0 0 = g7231;
CodersGroup0 1 = Transparent;
```

```
[ \CodersGroup0 ]
[TelProfile]
FORMAT TelProfile_Index = TelProfile_ProfileName,
TelProfile_TelPreference, TelProfile_CodersGroupID,
TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay,
TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ,
TelProfile_SigIPDiffServ, TelProfile_DtmfVolume,
TelProfile_InputGain, TelProfile_VoiceVolume,
TelProfile_EnableReversePolarity,
TelProfile_EnableCurrentDisconnect,
TelProfile_EnableDigitDelivery, TelProfile_EnableEC,
TelProfile_MWIAnalog, TelProfile_MWIDisplay,
TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia,
TelProfile_ProgressIndicator2IP;
TelProfile 1 = voice,$$,1,$$,$$,$$,$$,$$,$$,$$;
TelProfile 2 = data,$$,2,$$,$$,$$,$$,$$,$$,$$;
[\TelProfile]
```

Originating Side:

```
;E1_TRANSPARENT_31
ProtocolType_0 = 5
ProtocolType_1 = 5
ProtocolType_2 = 5
ProtocolType_3 = 5
;Channel selection by Phone number.
ChannelSelectMode = 0
[TrunkGroup]
FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum,
TrunkGroup_FirstTrunkId, TrunkGroup_LastTrunkId,
TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel,
TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId,
TrunkGroup_Module;
TrunkGroup 0 = 0,0,0,1,31,1000,1;
TrunkGroup 0 = 0,1,1,1,31,2000,1;
TrunkGroup 0 = 0,2,2,1,31,3000,1;
TrunkGroup 0 = 0,3,1,31,4000,1;
TrunkGroup 0 = 0,0,0,16,16,7000,2;
TrunkGroup 0 = 0,1,1,16,16,7001,2;
TrunkGroup 0 = 0,2,2,16,16,7002,2;
TrunkGroup 0 = 0,3,3,16,16,7003,2;
[\TrunkGroup]
[ CodersGroup0 ]
FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime,
CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce;
CodersGroup0 0 = g7231;
CodersGroup0 1 = Transparent;
[ \CodersGroup0 ]
[TelProfile]
FORMAT TelProfile_Index = TelProfile_ProfileName,
TelProfile_TelPreference, TelProfile_CodersGroupID,
TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay,
TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ,
TelProfile_SigIPDiffServ, TelProfile_DtmfVolume,
TelProfile_InputGain, TelProfile_VoiceVolume,
TelProfile_EnableReversePolarity,
TelProfile_EnableCurrentDisconnect,
TelProfile_EnableDigitDelivery, TelProfile_EnableEC,
```

```
TelProfile_MWIAnalog, TelProfile_MWIDisplay,
TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia,
TelProfile_ProgressIndicator2IP;
TelProfile_1 = voice,$$,1,$$,$$,$$,$$,$$,$$,$$
TelProfile_2 = data,$$,2,$$,$$,$$,$$,$$,$$,$$
[\TelProfile]
```

19.1.5.1.1 DSP Pattern Detector

For TDM tunneling applications, you can use the DSP pattern detector feature to initiate the echo canceller at call start. The device can be configured to support detection of a specific one-byte idle data pattern transmitted over digital E1/T1 timeslots. The device can be configured to detect up to four different one-byte data patterns. When the defined idle data pattern is detected, the channel resets its echo canceller.

The following parameters must be configured:

- EnableDSPIPMDetectors = 1
- EnablePatternDetector = 1
- PDThreshold - Pattern Detector Threshold, which defines the number of consecutive patterns to trigger the pattern detection event. For example: PDThreshold = 5
- PDPattern - Detection Pattern, which defines the patterns that can be detected by the Pattern Detector. For example: PDPattern = 84, 85, 212, 213 (for idle patterns: 54, 55, D4 and D5)

19.1.5.2 QSIG Tunneling

The device supports QSIG tunneling over SIP, according to IETF Internet-Draft draft-elwell-sipping-qsig-tunnel-03 ("Tunnelling of QSIG over SIP") and ECMA-355/ISO/IEC 22535. This is applicable to all ISDN variants. QSIG tunneling can be applied to all calls or to specific calls using IP Profiles.

QSIG tunneling sends all QSIG messages as raw data in corresponding SIP messages using a dedicated message body. This is used, for example, to enable two QSIG subscribers connected to the same or different QSIG PBX to communicate with each other over an IP network. Tunneling is supported in both directions (Tel-to-IP and IP-to-Tel).

The term tunneling means that messages are transferred 'as is' to the remote side without being converted (QSIG > SIP > QSIG). The advantage of tunneling over QSIG-to-SIP interworking is that by using interworking, QSIG functionality can only be partially achieved. When tunneling is used, all QSIG capabilities are supported and the tunneling medium (the SIP network) does not need to process these messages.

QSIG messages are transferred in SIP messages in a separate Multipurpose Internet Mail Extensions (MIME) body. Therefore, if a message contains more than one body (e.g., SDP and QSIG), multipart MIME must be used. The Content-Type of the QSIG tunneled message is 'application/QSIG'. The device also adds a Content-Disposition header in the following format:

```
Content-Disposition: signal; handling=required.
```

QSIG tunneling is done as follows:

- **Call setup (originating device):** The QSIG Setup request is encapsulated in the SIP INVITE message without being altered. After the SIP INVITE request is sent, the device does not encapsulate the subsequent QSIG message until a SIP 200 OK response is received. If the originating device receives a 4xx, 5xx, or 6xx response, it disconnects the QSIG call with a 'no route to destination' cause.

- **Call setup (terminating device):** After the terminating device receives a SIP INVITE request with a 'Content-Type: application/QSIG', it sends the encapsulated QSIG Setup message to the Tel side and sends a 200 OK response (no 1xx response is sent) to IP. The 200 OK response includes an encapsulated QSIG Call Proceeding message (without waiting for a Call Proceeding message from the Tel side). If tunneling is disabled and the incoming INVITE includes a QSIG body, a 415 response is sent.
- **Mid-call communication:** After the SIP connection is established, all QSIG messages are encapsulated in SIP INFO messages.
- **Call tear-down:** The SIP connection is terminated once the QSIG call is complete. The Release Complete message is encapsulated in the SIP BYE message that terminates the session.

➢ **To enable QSIG tunneling:**

1. Set the EnableQSIGTunneling parameter to 1 on the originating and terminating devices.
2. Configure the QSIGTunnelingMode parameter for defining the format of encapsulated QSIG message data in the SIP message MIME body (0 for ASCII presentation; 1 for binary encoding).
3. Set the ISDNDuplicateQ931BuffMode parameter to 128 to duplicate all messages.
4. Set the ISDNInCallsBehavior parameter to 4096.
5. Set the ISDNRxOverlap parameter to 0 for tunneling of QSIG overlap-dialed digits (see below for description).

The configuration of the ISDNInCallsBehavior and ISDNRxOverlap parameters allows tunneling of QSIG overlap-dialed digits (Tel to IP). In this configuration, the device **delays** the sending of the QSIG Setup Ack message upon receipt of the QSIG Setup message. Instead, the device sends the Setup Ack message to QSIG only when it receives the SIP INFO message with Setup Ack encapsulated in its MIME body. The PBX sends QSIG Information messages (to complete the Called Party Number) only after it receives the Setup Ack. The device relays these Information messages encapsulated in SIP INFO messages to the remote party.

19.1.6 Advanced PSTN Configuration

This section describes various advanced PSTN configurations.

19.1.6.1 Release Reason Mapping

This section describes the available mapping mechanisms of SIP responses to Q.850 Release Causes and vice versa. The existing mapping of ISDN Release Causes to SIP Responses is described in 'Fixed Mapping of ISDN Release Reason to SIP Response' on page 263 and 'Fixed Mapping of SIP Response to ISDN Release Reason' on page 265. To override this hard-coded mapping and flexibly map SIP responses to ISDN Release Causes, use the *ini* file (CauseMapISDN2SIP and CauseMapSIP2ISDN, as described in 'ISDN and CAS Interworking Parameters' on page 912) or the Web interface (see 'Configuring Release Cause Mapping' on page 288).

It is also possible to map the less commonly used SIP responses to a single default ISDN Release Cause. Use the parameter DefaultCauseMapISDN2IP (described in 'ISDN and CAS Interworking Parameters' on page 912) to define a default ISDN Cause that is always used except when the following Release Causes are received: Normal Call Clearing (16), User Busy (17), No User Responding (18) or No Answer from User (19). This mechanism is only available for Tel-to-IP calls.

19.1.6.1.1 Reason Header

The device supports the Reason header according to RFC 3326. The Reason header conveys information describing the disconnection cause of a call:

- **Sending Reason header:** If a call is disconnected from the Tel side (ISDN), the Reason header is set to the received Q.850 cause in the appropriate message (BYE/CANCEL/final failure response) and sent to the SIP side. If the call is disconnected because of a SIP reason, the Reason header is set to the appropriate SIP response.
- **Receiving Reason header:** If a call is disconnected from the IP side and the SIP message includes the Reason header, it is sent to the Tel side according to the following logic:
  - If the Reason header includes a Q.850 cause, it is sent as is.
  - If the Reason header includes a SIP response:
    - If the message is a final response, the response status code is translated to Q.850 format and passed to ISDN.
    - If the message isn't a final response, it is translated to a Q.850 cause.
  - When the Reason header is received twice (i.e., SIP Reason and Q.850), the Q.850 takes precedence over the SIP reason and is sent to the Tel side.

19.1.6.1.2 Fixed Mapping of ISDN Release Reason to SIP Response

The following table describes the mapping of ISDN release reason to SIP response.

Table 19-2: Mapping of ISDN Release Reason to SIP Response

ISDN Release Reason	Description	SIP Response	Description
1	Unallocated number	404	Not found
2	No route to network	404	Not found
3	No route to destination	404	Not found
6	Channel unacceptable	406*	Not acceptable
7	Call awarded and being delivered in an established channel	500	Server internal error
16	Normal call clearing	-*	BYE
17	User busy	486	Busy here
18	No user responding	408	Request timeout
19	No answer from the user	480	Temporarily unavailable
21	Call rejected	403	Forbidden
22	Number changed w/o diagnostic	410	Gone
26	Non-selected user clearing	404	Not found
27	Destination out of order	502	Bad gateway
28	Address incomplete	484	Address incomplete
29	Facility rejected	501	Not implemented
30	Response to status enquiry	501*	Not implemented

ISDN Release Reason	Description	SIP Response	Description
31	Normal unspecified	480	Temporarily unavailable
34	No circuit available	503	Service unavailable
38	Network out of order	503	Service unavailable
41	Temporary failure	503	Service unavailable
42	Switching equipment congestion	503	Service unavailable
43	Access information discarded	502*	Bad gateway
44	Requested channel not available	503*	Service unavailable
47	Resource unavailable	503	Service unavailable
49	QoS unavailable	503*	Service unavailable
50	Facility not subscribed	503*	Service unavailable
55	Incoming calls barred within CUG	403	Forbidden
57	Bearer capability not authorized	403	Forbidden
58	Bearer capability not presently available	503	Service unavailable
63	Service/option not available	503*	Service unavailable
65	Bearer capability not implemented	501	Not implemented
66	Channel type not implemented	480*	Temporarily unavailable
69	Requested facility not implemented	503*	Service unavailable
70	Only restricted digital information bearer capability is available	503*	Service unavailable
79	Service or option not implemented	501	Not implemented
81	Invalid call reference value	502*	Bad gateway
82	Identified channel does not exist	502*	Bad gateway
83	Suspended call exists, but this call identity does not	503*	Service unavailable
84	Call identity in use	503*	Service unavailable
85	No call suspended	503*	Service unavailable
86	Call having the requested call identity has been cleared	408*	Request timeout
87	User not member of CUG	503	Service unavailable
88	Incompatible destination	503	Service unavailable
91	Invalid transit network selection	502*	Bad gateway
95	Invalid message	503	Service unavailable
96	Mandatory information element is missing	409*	Conflict
97	Message type non-existent or not implemented	480*	Temporarily not available

ISDN Release Reason	Description	SIP Response	Description
98	Message not compatible with call state or message type non-existent or not implemented	409*	Conflict
99	Information element non-existent or not implemented	480*	Not found
100	Invalid information elements contents	501*	Not implemented
101	Message not compatible with call state	503*	Service unavailable
102	Recovery of timer expiry	408	Request timeout
111	Protocol error	500	Server internal error
127	Interworking unspecified	500	Server internal error

***** Messages and responses were created because the 'ISUP to SIP Mapping' draft doesn't specify their cause code mapping.

19.1.6.1.3 Fixed Mapping of SIP Response to ISDN Release Reason

The following table describes the mapping of SIP response to ISDN release reason.

Table 19-3: Mapping of SIP Response to ISDN Release Reason

SIP Response	Description	ISDN Release Reason	Description
400*	Bad request	31	Normal, unspecified
401	Unauthorized	21	Call rejected
402	Payment required	21	Call rejected
403	Forbidden	21	Call rejected
404	Not found	1	Unallocated number
405	Method not allowed	63	Service/option unavailable
406	Not acceptable	79	Service/option not implemented
407	Proxy authentication required	21	Call rejected
408	Request timeout	102	Recovery on timer expiry
409	Conflict	41	Temporary failure
410	Gone	22	Number changed w/o diagnostic
411	Length required	127	Interworking
413	Request entity too long	127	Interworking
414	Request URI too long	127	Interworking
415	Unsupported media type	79	Service/option not implemented
420	Bad extension	127	Interworking
480	Temporarily unavailable	18	No user responding

SIP Response	Description	ISDN Release Reason	Description
481*	Call leg/transaction doesn't exist	127	Interworking
482*	Loop detected	127	Interworking
483	Too many hops	127	Interworking
484	Address incomplete	28	Invalid number format
485	Ambiguous	1	Unallocated number
486	Busy here	17	User busy
488	Not acceptable here	31	Normal, unspecified
500	Server internal error	41	Temporary failure
501	Not implemented	38	Network out of order
502	Bad gateway	38	Network out of order
503	Service unavailable	41	Temporary failure
504	Server timeout	102	Recovery on timer expiry
505*	Version not supported	127	Interworking
600	Busy everywhere	17	User busy
603	Decline	21	Call rejected
604	Does not exist anywhere	1	Unallocated number
606*	Not acceptable	38	Network out of order

***** Messages and responses were created because the 'ISUP to SIP Mapping' draft does not specify their cause code mapping.

19.1.6.2 ISDN Overlap Dialing

Overlap dialing is a dialing scheme used by several ISDN variants to send and/or receive called number digits one after the other (or several at a time). This is in contrast to en-bloc dialing in which a complete number is sent in one message. ISDN overlap dialing is applicaable to PRI and BRI.

The device supports the following ISDN overlap dialing methods:

- Collects ISDN called party number digits and then sends the SIP INVITE to the IP side with the complete destination number (see 'Collecting ISDN Digits and Sending Complete Number in SIP' on page 266)
- Interworks ISDN overlap dialing with SIP, according to RFC 3578 (see 'Interworking ISDN Overlap Dialing with SIP According to RFC 3578' on page 267)

19.1.6.2.1 Collecting ISDN Digits and Sending Complete Number in SIP

The device can support an overlap dialing mode whereby the device collects the called party number digits from ISDN Q.931 Information messages or DTMF signals, and then sends a SIP INVITE message to the IP side containing the complete destination number.

ISDN overlap dialing for incoming ISDN calls can be configured for the entire device or per E1/T1 trunk. This is configured using the global, ISDNRxOverlap parameter or the ISDNRxOverlap_x parameter (where *x* denotes the trunk number), respectively.

By default (see the ISDNINCallsBehavior parameter), the device plays a dial tone to the ISDN user side when it receives an empty called number from the ISDN. In this scenario, the device includes the Progress Indicator in the SetupAck ISDN message that it sends to the ISDN side.

The device can also mute in-band DTMF detection until it receives the complete destination number from the ISDN. This is configured using the MuteDTMFInOverlap parameter. The Information digits can be sent in-band in the voice stream, or out-of-band using Q.931 Information messages. If Q.931 Information messages are used, the DTMF in-band detector must be disabled. Note that when at least one digit is received in the ISDN Setup message, the device stops playing a dial tone.

The device stops collecting digits (from the ISDN) upon the following scenarios:

- The device receives a Sending Complete IE in the ISDN Setup or Information messages, indicating no more digits.
- The timeout between received digits expires (configured by the TimeBetweenDigits parameter).
- The maximum number of received digits has been reached (configured by the MaxDigits parameter).
- A match is found with the defined digit map (configured by the DigitMapping parameter).

Relevant parameters (described in 'PSTN Parameters' on page 896):

- ISDNRxOverlap_x = 1 (can be configured per trunk)
- TimeBetweenDigits
- MaxDigits
- MuteDTMFInOverlap
- DigitMapping

For configuring ISDN overlap dialing using the Web interface, see 'Configuring Trunk Settings' on page 254.

19.1.6.2.2 Interworking ISDN Overlap Dialing with SIP According to RFC 3578

The device supports the interworking of ISDN overlap dialing to SIP and vice versa, according to RFC 3578.

- **Interworking ISDN overlap dialing to SIP (Tel to IP):** The device sends collected digits each time it receives them (initially from the ISDN Setup message and then from subsequent Q.931 Information messages) to the IP side, using subsequent SIP INVITE messages. You can also define the minimum number of overlap digits to collect before sending the first SIP message (INVITE) for routing the call, using the MinOverlapDigitsForRouting parameter.
- **Interworking SIP to ISDN overlap dialing (IP to Tel):** For each received SIP INVITE pertaining to the same dialog session, the device sends an ISDN Setup message (and subsequent Q.931 Information messages) with the collected digits to the Tel side. For all subsequent INVITEs received, the device sends a SIP 484 "Address Incomplete" response to the IP in order to maintain the current dialog session and to receive additional digits from subsequent INVITEs.

Relevant parameters (described in 'PSTN Parameters' on page 896):

- ISDNRxOverlap = 2
- ISDNTxOverlap
- ISDNOutCallsBehavior = 2
- MinOverlapDigitsForRouting
- TimeBetweenDigits
- MaxDigits

- DigitMapping
- MuteDTMFInOverlap

For configuring ISDN overlap dialing using the Web interface, see 'Configuring Trunk Settings' on page 254.

19.1.6.3 ISDN Non-Facility Associated Signaling (NFAS)

In regular T1 ISDN trunks, a single 64 kbps channel carries signaling for the other 23 B-channels of that particular T1 trunk. This channel is called the D-channel and usually resides on timeslot # 24. The ISDN Non-Facility Associated Signaling (NFAS) feature enables the use of a single D-channel to control multiple PRI interfaces.

With NFAS it is possible to define a group of T1 trunks, called an NFAS group, in which a single D-channel carries ISDN signaling messages for the entire group. The NFAS group's B-channels are used to carry traffic such as voice or data. The NFAS mechanism also enables definition of a backup D-channel on a different T1 trunk, to be used if the primary D-channel fails.

The device supports up to 12 NFAS groups. Each group can comprise up to 10 T1 trunks and each group must contain different T1 trunks. Each T1 trunk is called an 'NFAS member'. The T1 trunk whose D-channel is used for signaling is called the 'Primary NFAS Trunk'. The T1 trunk whose D-channel is used for backup signaling is called the 'Backup NFAS Trunk'. The primary and backup trunks each carry 23 B-channels while all other NFAS trunks each carry 24 B-channels.

The NFAS group is identified by an NFAS GroupID number (possible values are 1 to 12). To assign a number of T1 trunks to the same NFAS group, use the CLI command isdn-nfas-group-number = groupID .

The parameter 'DchConfig_x = Trunk_type' defines the type of NFAS trunk. Trunk_type is set to 0 for the primary trunk, to 1 for the backup trunk, and to 2 for an ordinary NFAS trunk. 'x' denotes the physical trunk ID (0 to the maximum number of trunks). You can also use the Web interface (see 'Configuring Trunk Settings' on page 254).

For example, to assign the first four T1 trunks to NFAS group #1, in which trunk #0 is the primary trunk and trunk #1 is the backup trunk, use the following configuration:

```
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0             ;Primary T1 trunk
DchConfig_1 = 1             ;Backup T1 trunk
DchConfig_2 = 2             ;24 B-channel NFAS trunk
DchConfig_3 = 2             ;24 B-channel NFAS trunk
```

The NFAS parameters are described in 'PSTN Parameters' on page 896.

19.1.6.3.1 NFAS Interface ID

Several ISDN switches require an additional configuration parameter per T1 trunk that is called 'Interface Identifier'. In NFAS T1 trunks, the Interface Identifier is sent explicitly in Q.931 Setup / Channel Identification IE for all NFAS trunks, except for the B-channels of the Primary trunk (see note below).

The Interface ID can be defined per member (T1 trunk) of the NFAS group, and must be coordinated with the configuration of the Switch. The default value of the Interface ID is identical to the number of the physical T1 trunk (0 for the first trunk, 1 for the second T1 trunk, and so on, up to the maximum number of trunks).

To define an explicit Interface ID for a T1 trunk (that is different from the default), use the following parameters:

- ISDNIBehavior_x = 512 (x = 0 to the maximum number of trunks identifying the device's physical trunk)
- ISDNNFASInterfaceID_x = ID (x = 0 to 255)

Notes:

- Usually the Interface Identifier is included in the Q.931 Setup/Channel Identification IE only on T1 trunks that doesn't contain the D-channel. Calls initiated on B-channels of the Primary T1 trunk, by default, don't contain the Interface Identifier. Setting the parameter ISDNIBehavior_x to 2048' forces the inclusion of the Channel Identifier parameter also for the Primary trunk.
- The parameter ISDNNFASInterfaceID_x = ID can define the 'Interface ID' for any Primary T1 trunk, even if the T1 trunk is not a part of an NFAS group. However, to include the Interface Identifier in Q.931 Setup/Channel Identification IE configure ISDNIBehavior_x = 2048 in the *ini* file.

19.1.6.3.2 Working with DMS-100 Switches

The DMS-100 switch requires the following NFAS Interface ID definitions:

- InterfaceID #0 for the Primary trunk
- InterfaceID #1 for the Backup trunk
- InterfaceID #2 for a 24 B-channel T1 trunk
- InterfaceID #3 for a 24 B-channel T1 trunk, and so on for subsequent T1 trunks

For example, if four T1 trunks on a device are configured as a single NFAS group with Primary and Backup T1 trunks that is used with a DMS-100 switch, the following parameters should be used:

```
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0    ;Primary T1 trunk
DchConfig_1 = 1    ;Backup T1 trunk
DchConfig_2 = 2    ;B-Channel NFAS trunk
DchConfig_3 = 2    ;B-channel NFAS trunk
```

If there is no NFAS Backup trunk, the following configuration should be used:

```
ISDNNFASInterfaceID_0 = 0
ISDNNFASInterfaceID_1 = 2
ISDNNFASInterfaceID_2 = 3
ISDNNFASInterfaceID_3 = 4
ISDNIBehavior = 512   ;This parameter should be added because of
;ISDNNFASInterfaceID coniguration above
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0    ;Primary T1 trunk
DchConfig_1 = 2    ;B-Channel NFAS trunk
DchConfig_2 = 2    ;B-Channel NFAS trunk
DchConfig_3 = 2    ;B-channel NFAS trunk
```

19.1.6.3.3 Creating an NFAS-Related Trunk Configuration

The procedures for creating and deleting an NFAS group must be performed in the correct order, as described below.

➢ **To create an NFAS Group:**

1. If there's a backup ('secondary') trunk for this group, it must be configured first.
2. Configure the primary trunk before configuring any NFAS ('slave') trunk.
3. Configure NFAS ('slave') trunks.

➢ **To stop / delete an NFAS Group:**

1. Stop or delete (by setting ProtocolType to 0, i.e., 'None') all NFAS ('slave') trunks.
2. Stop or delete (by setting ProtocolType to 0, i.e., 'None') the backup trunk if a backup trunk exists.
3. Stop or delete (by setting ProtocolType to 0, i.e., 'None') the primary trunk.

Notes:

- All trunks in the group must be configured with the same values for trunk parameters TerminationSide, ProtocolType, FramingMethod, and LineCode.
- After stopping or deleting the backup trunk, delete the group and then reconfigure it.

19.1.6.4 Redirect Number and Calling Name (Display)

The following tables define the device's redirect number and calling name (Display) support for various ISDN variants according to NT (Network Termination) / TE (Termination Equipment) interface direction:

Table 19-4: Calling Name (Display)

NT/TE Interface	DMS-100	NI-2	4/5ESS	Euro ISDN	QSIG
NT-to-TE	Yes	Yes	No	Yes	Yes
TE-to-NT	Yes	Yes	No	No	Yes

Table 19-5: Redirect Number

NT/TE Interface	DMS-100	NI-2	4/5ESS	Euro ISDN	QSIG
NT-to-TE	Yes	Yes	Yes	Yes	Yes
TE-to-NT	Yes	Yes	Yes	Yes*	Yes

* When using ETSI DivertingLegInformation2 in a Facility IE (not Redirecting Number IE).

19.2 Trunk Group

This section describes the configuration of the device's channels, which entails assigning them numbers and Trunk Group IDs.

19.2.1 Configuring Trunk Group Table

The Trunk Group Table page allows you to define up to 120 Trunk Groups. A Trunk Group is a logical group of physical trunks and channels, and is assigned an ID. The Trunk Group can include multiple trunks and ranges of channels.

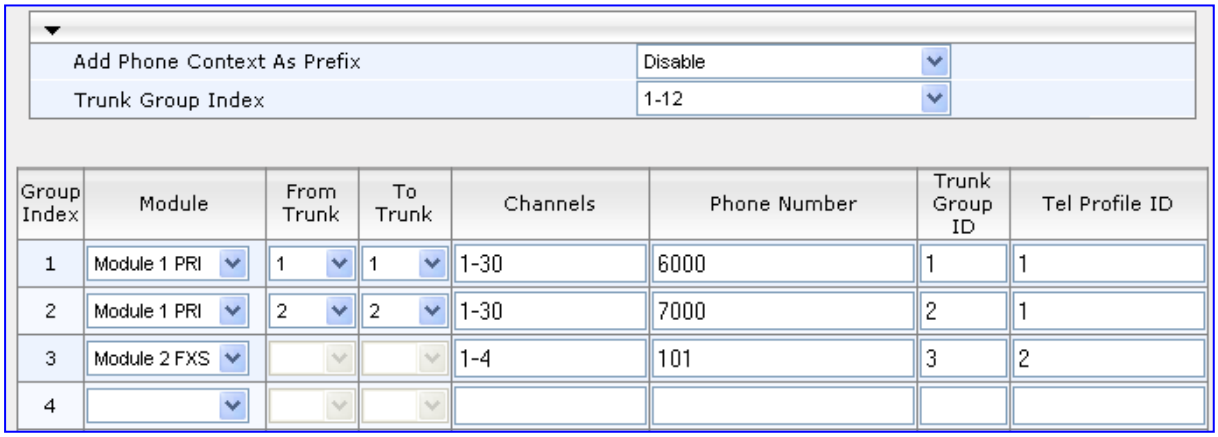
To enable and activate the channels of the device, Trunk Groups need to be defined and with telephone numbers. Channels that are not defined in this table are disabled. The Trunk Groups are later used for routing IP-to-Tel and Tel-to-IP calls.

Note: You can also configure Trunk Groups using the *ini* file table parameter TrunkGroup_x to (see 'Number Manipulation Parameters' on page 962).

➢ **To configure the Trunk Group Table:**

1. Open the Trunk Group Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Trunk Group** > **Trunk Group**).

Figure 19-5: Trunk Group Table Page



2. Configure the Trunk Group according to the table below**.**
3. Click **Submit** to apply your changes.
4. To save the changes to the flash memory, see 'Saving Configuration' on page 680.
5. To register the Trunk Groups, click the **Register** button. To unregister the Trunk Groups, click **Unregister**. The registration method for each Trunk Group is according to the setting of the 'Registration Mode' parameter in the Trunk Group Settings page.

Table 19-6: Trunk Group Table Parameters

Parameter	Description
Module **[TrunkGroup**_Module]	Defines the module (i.e., FXS, FXO, PRI, or BRI) for which you want to define the Trunk Group.

Parameter	Description
From Trunk **[TrunkGroup_FirstTrunkId]**	Defines the starting physical Trunk number in the Trunk Group. The number of listed Trunks depends on the device's hardware configuration. **Note:** This parameter is applicable only to PRI and BRI modules.
To Trunk **[TrunkGroup_LastTrunkId]**	Defines the ending physical Trunk number in the Trunk Group. The number of listed Trunks depends on the device's hardware configuration. **Note:** This parameter is applicable only to PRI and BRI modules.
Channels **[TrunkGroup_FirstBChannel]**, **[TrunkGroup_LastBChannel]**	Defines the device's channels/ports (analog module) or Trunk B-channels (digital module). To enable channels, enter the channel numbers. You can enter a range of channels by using the syntax *n-m*, where *n* represents the lower channel number and *m* the higher channel number. For example, "1-4" specifies channels 1 through 4. **Notes:** ▪ The number of defined channels must not exceed the maximum number of the Trunk's B-channels. ▪ To represent all the Trunk's B-channels, enter a single asterisk (*).
Phone Number **[TrunkGroup_FirstPhoneNumber]**	Defines the telephone number that is assigned to the channel. This value can include up to 50 characters. For a range of channels, enter only the first telephone number. Subsequent channels are assigned the next consecutive telephone number. For example, if you enter 400 for channels 1 to 4, then channel 1 is assigned phone number 400, channel 2 is assigned phone number 401, and so on. These numbers are also used for channel allocation for IP-to-Tel calls if the Trunk Group's 'Channel Select Mode' is set to 'By Dest Phone Number'. **Notes:** ▪ If this field includes alphabetical characters and the phone number is defined for a range of channels (e.g., 1-4), then the phone number must end with a number (e.g., 'user1'). ▪ This field is optional for BRI/PRI interfaces. The logical numbers defined in this field are used when an incoming PSTN/PBX call doesn't contain the calling number or called number (the latter being determined by the ReplaceEmptyDstWithPortNumber parameter). These numbers are used to replace them.

Parameter	Description
Trunk Group ID **[TrunkGroup_TrunkGroupNum]**	Defines the Trunk Group ID (0-119) for the corresponding channels. The same Trunk Group ID can be assigned to more than one group of channels. The Trunk Group ID is used to define a group of common channel behavior that are used for routing IP-to-Tel calls. If an IP-to-Tel call is assigned to a Trunk Group, the IP call is routed to the channel(s) pertaining to that Trunk Group ID. **Notes:** ▪ Once you have defined a Trunk Group, you must configure the parameter PSTNPrefix (Inbound IP Routing Table) to assign incoming IP calls to the appropriate Trunk Group. If you do not configure this, calls cannot be established. ▪ You can define the method for which calls are assigned to channels within Trunk Groups, using the parameter TrunkGroupSettings.
Tel Profile ID **[TrunkGroup_ProfileId]**	Assigns a Tel Profile ID assigned to the channels pertaining to the Trunk Group. **Note:** For configuring Tel Profiles, refer to the parameter TelProfile.

19.2.2 Configuring Trunk Group Settings

The Trunk Group Settings page allows you to configure the settings of up to 120 Trunk Groups. These Trunk Groups are configured in the Trunk Group Table page (see Configuring Trunk Group Table on page 271).

This page allows you to select the method for which IP-to-Tel calls are assigned to channels within each Trunk Group. If no method is selected for a specific Trunk Group, the setting of the global parameter, ChannelSelectMode takes effect. In addition, this page defines the method for registering Trunk Groups to selected Serving IP Group IDs (if defined).

Note: You can also configure the Trunk Group Settings table using the *ini* file table parameter TrunkGroupSettings (see 'Number Manipulation Parameters' on page 962).

➢ **To configure the Trunk Group Settings table:**

1. Open the Trunk Group Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP s**ubmenu > **Trunk Group** submenu > **Trunk Group Settings**).

Figure 19-6: Trunk Group Settings Page

2. From the 'Index' drop-down list, select the range of entries that you want to edit.
3. Configure the Trunk Group according to the table below.
4. Click **Submit** to apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

An example is shown below of a REGISTER message for registering endpoint "101" using registration Per Endpoint mode. The "SipGroupName" in the Request-URI is defined in the IP Group table (see 'Configuring IP Groups' on page 212).

```
REGISTER sip:SipGroupName SIP/2.0
Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac862428454
From: <sip:101@GatewayName>;tag=1c862422082
To: <sip:101@GatewayName>
Call-ID: 990797706251200023825@10.33.37.78
CSeq: 3 REGISTER
Contact: <sip:101@10.33.37.78>;expires=3600
Expires: 3600
User-Agent: Sip-Gateway/v.6.00A.008.002
Content-Length: 0
```

Table 19-7: Trunk Group Settings Parameters

Parameter	Description
Trunk Group ID **[TrunkGroupSettings_TrunkGroupId]**	The Trunk Group ID that you want to configure.

Parameter	Description
Channel Select Mode **[TrunkGroupSettings_ChannelSelectMode]**	The method for which IP-to-Tel calls are assigned to channels pertaining to a Trunk Group. For a detailed description of this parameter, refer to the global parameter ChannelSelectMode. ▪ **[0]** By Dest Phone Number. ▪ **[1]** Cyclic Ascending (default) ▪ **[2]** Ascending ▪ **[3]** Cyclic Descending ▪ **[4]** Descending ▪ **[5]** Dest Number + Cyclic Ascending ▪ **[6]** By Source Phone Number ▪ **[7]** Trunk Cyclic Ascending (applicable only to digital interfaces) ▪ **[8]** Trunk & Channel Cyclic Ascending (applicable only to digital interfaces) ▪ **[9]** Ring to Hunt Group (applicable only to FXS interfaces) ▪ **[10]** Select Trunk by Supplementary Services Table (applicable only to BRI interfaces) ▪ **[11]** Dest Number + Ascending **Note:** For a detailed description of these options, refer to the "global" ChannelSelectMode parameter.
Registration Mode **[TrunkGroupSettings_RegistrationMode]**	Registration method for the Trunk Group: ▪ **[1]** Per Gateway = Single registration for the entire device (default). This mode is applicable only if a default Proxy or Registrar IP are configured, and Registration is enabled (i.e., parameter IsRegisterUsed is set to 1). In this mode, the SIP URI user part in the From, To, and Contact headers is set to the value of the global registration parameter GWRegistrationName or username if GWRegistrationName is not configured. ▪ **[0]** Per Endpoint = Each channel in the Trunk Group registers individually. The registrations are sent to the ServingIPGroupID if defined in the table, otherwise to the default Proxy, and if no default Proxy, then to the Registrar IP. ▪ **[4]** Don't Register = No registrations are sent by endpoints pertaining to the Trunk Group. For example, if the device is configured globally to register all its endpoints (using the parameter ChannelSelectMode), you can exclude some endpoints from being registered by assigning them to a Trunk Group and configuring the Trunk Group registration mode to 'Don't Register'. ▪ **[5]** Per Account = Registrations are sent (or not) to an IP Group, according to the settings in the Account table (see 'Configuring Account Table' on page 237). **Notes:** ▪ To enable Trunk Group registrations, configure the global parameter IsRegisterNeeded to 1. This is unnecessary for 'Per Account' registration mode. ▪ If no mode is selected, the registration is performed according to the global registration parameter ChannelSelectMode. ▪ If the device is configured globally (ChannelSelectMode) to register Per Endpoint, and channels group comprising four channels is configured to register Per Gateway, the device registers all channels except the first four channels. The

Parameter	Description
	channels Group of these four channels sends a single registration request.
Serving IP Group ID **[TrunkGroupSettings_ServingIPGroup]**	The Serving IP Group ID to where INVITE messages initiated by this Trunk Group's endpoints are sent. The actual destination to where these INVITE messages are sent is according to the Proxy Set ID (see 'Configuring Proxy Sets Table' on page 219) associated with this Serving IP Group. The Request-URI host name in the INVITE and REGISTER messages (except for 'Per Account' registration modes) is set to the value of the field 'SIP Group Name' defined in the IP Group table (see 'Configuring IP Groups' on page 212). If no Serving IP Group ID is selected, the INVITE messages are sent to the default Proxy or according to the Outbound IP Routing Table (see 'Configuring Outbound IP Routing Table' on page 291). **Note:** If the parameter PreferRouteTable is set to 1 (see 'Configuring Proxy and Registration Parameters' on page 239), the routing rules in the Outbound IP Routing Table prevail over the selected Serving IP Group ID.
Gateway Name **[TrunkGroupSettings_GatewayName]**	The host name used in the SIP From header in INVITE messages, and as a host name in From/To headers in REGISTER requests. If not configured, the global parameter SIPGatewayName is used instead.
Contact User **[TrunkGroupSettings_ContactUser]**	The user part in the SIP Contact URI in INVITE messages, and as a user part in From, To, and Contact headers in REGISTER requests. This is applicable only if the field 'Registration Mode' is set to 'Per Account', and the Registration through the Account table is successful. **Notes:** ▪ If registration fails, then the user part in the INVITE Contact header contains the source party number. ▪ The Contact User' parameter in the 'Account Table page overrides this parameter.

19.3 Manipulation

This section describes the configuration of number / name manipulation rules and various SIP to non-SIP mapping.

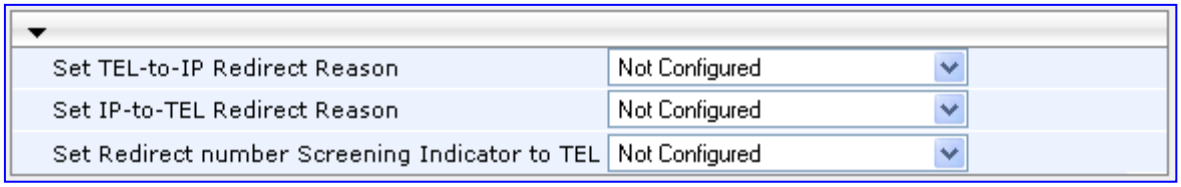
19.3.1 Configuring General Settings

The General Settings page allows you to configure general manipulation parameters. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

➢ **To configure the general manipulation parameters:**

1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu >**General Settings**).

Figure 19-7: General Settings Page



2. Configure the parameters as required.
3. Click **Submit** to apply your changes.

19.3.2 Configuring Number Manipulation Tables

The device provides number manipulation tables for incoming (IP-to-Tel) and outgoing (Tel-to-IP) calls. These tables are used to modify the destination and source telephone numbers so that the calls can be routed correctly. The number manipulation tables include the following:

- **Tel-to-IP calls:**
  - Destination Phone Number Manipulation Table for Tel-to-IP Calls table (NumberMapTel2IP *ini* file parameter) - up to 120 entries
  - Source Phone Number Manipulation Table for Tel-to-IP Calls table (SourceNumberMapTel2IP *ini* file parameter) - up to 120 entries
- **IP-to-Tel calls:**
  - Destination Phone Number Manipulation Table for IP-to-Tel Calls table (NumberMapIP2Tel *ini* file parameter) - up to 100 entries
  - Source Phone Number Manipulation Table for IP-to-Tel Calls table (SourceNumberMapIP2Tel *ini* file parameter) - up to 120 entries

The manipulation rules can be applied to incoming calls that match one or any combination of the following characteristics:

- Source Trunk Group
- Source IP Group
- Destination (called) number prefix or suffix
- Source (calling) number prefix or suffix
- Source IP address

The device manipulates the number in the following order:

1. Strips digits from the left of the number.
2. Strips digits from the right of the number.
3. Retains the defined number of digits.
4. Adds the defined prefix.
5. Adds the defined suffix.

The device searches a matching manipulation rule starting from the first entry (i.e., top of the table). In other words, a rule at the top of the table takes precedence over a rule defined lower down in the table. Therefore, define more specific rules above more generic rules. For example, if you enter 551 in Index 1 and 55 in Index 2, the device applies rule 1 to numbers that start with 551 and applies rule 2 to numbers that start with 550, 552, 553, and so on until 559. However, if you enter 55 in Index 1 and 551 in Index 2, the device applies rule 1 to all numbers that start with 55, including numbers that start with 551.

You can perform a second "round" (additional) of destination (NumberMapIP2Tel parameter) and source (SourceNumberMapIP2Tel parameter) number manipulations for IP-to-Tel calls on an already manipulated number. The initial and additional number manipulation rules are both configured in these tables. The additional manipulation is performed on the initially manipulated number. Therefore, for complex number manipulation schemes, you only need to configure relatively few manipulation rules in these tables (that would otherwise require many rules). This feature is enabled using the following parameters:

- PerformAdditionalIP2TELSourceManipulation for source number manipulation
- PerformAdditionalIP2TELDestinationManipulation for destination number manipulation

Telephone number manipulation can be useful, for example, for doing the following:

- Stripping or adding dialing plan digits from or to the number, respectively. For example, a user may need to first dial 9 before dialing the phone number to indicate an external line. This number 9 can then be removed by number manipulation before the call is setup.
- Allowing or blocking Caller ID information according to destination or source prefixes. For more information on Caller ID, see Configuring Caller Display Information on page 357.
- For digital modules only: Assigning Numbering Plan Indicator (NPI) and Type of Numbering (TON) to IP-to-Tel calls. The device can use a single global setting for NPI/TON classification or it can use the setting in the manipulation tables on a call-by-call basis.

Notes:

- Number manipulation can occur before or after a routing decision is made. For example, you can route a call to a specific Trunk Group according to its original number, and then you can remove or add a prefix to that number before it is routed. To determine when number manipulation is performed, configure the 'IP to Tel Routing Mode' parameter (RouteModeIP2Tel) described in 'Configuring Inbound IP Routing Table' on page 299, and 'Tel to IP Routing Mode' parameter (RouteModeTel2IP) described in 'Configuring Outbound IP Routing Table' on page 291.
- For configuring number manipulation using *ini* file table parameters NumberMapIP2Tel, NumberMapTel2IP, SourceNumberMapIP2Tel, and SourceNumberMapTel2IP, see 'Number Manipulation Parameters' on page 962.

➢ **To configure number manipulation rules:**

1. Open the required 'Number Manipulation page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu > **Dest Number IP->Tel**, **Dest Number Tel->IP**, **Source Number IP->Tel**, or **Source Number Tel->IP**); the relevant Manipulation table page is displayed (e.g., 'Source Phone Number Manipulation Table for Tel→IP Calls page).

Figure 19-8: Source Phone Number Manipulation Table for Tel-to-IP Calls

Index	Source Trunk Group	Source IP Group	Destination Prefix	Source Prefix	Stripped Digits From Left
1	-1	2	03	201	0
2	0	0		1001	4
3	-1	-1	*	123451001#	0
4	-1	-1	*	[30-40]x	0
5	-1	-1	[6,7,8]	2001	5

Stripped Digits From Right	Prefix to Add	Suffix to Add	Number of Digits to Leave	Presentation
0	971		255	Allowed
0	5	23	255	Restricted
0		8	4	Not Configured
1	2		255	Not Configured
0	3		255	Not Configured

The previous figure shows an example of the use of manipulation rules for Tel-to-IP source phone number manipulation:

- **Index 1:** When the destination number has the prefix 03 (e.g., 035000), source number prefix 201 (e.g., 20155), and from source IP Group ID 2, the source number is changed to, for example, 97120155.
- **Index 2:** When the source number has prefix 1001 (e.g., 1001876), it is changed to 587623.
- **Index 3:** When the source number has prefix 123451001 (e.g., 1234510012001), it is changed to 20018.
- **Index 4:** When the source number has prefix from 30 to 40 and a digit (e.g., 3122), it is changed to 2312.
- **Index 5:** When the destination number has the prefix 6, 7, or 8 (e.g., 85262146), source number prefix 2001, it is changed to 3146.

2. Configure the Number Manipulation table according to the table below.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 19-8: Number Manipulation Parameters Description

Parameter	Description
Source Trunk Group	The source Trunk Group ID for Tel-to-IP calls. To denote all Trunk Groups, leave this field empty. **Notes:** ▪ The value -1 indicates that this field is ignored in the rule. ▪ This parameter is available only in the 'Source Phone Number Manipulation Table for Tel -> IP Calls' and 'Destination Phone Number Manipulation Table for Tel -> IP Calls pages.

Parameter	Description
	▪ For IP-to-IP call routing, this parameter is not required (i.e., leave the field empty).
Source IP Group	The IP Group from where the IP-to-IP call originated. Typically, this IP Group of an incoming INVITE is determined/classified using the Inbound IP Routing Table'. If not used (i.e., any IP Group), simply leave the field empty. **Notes:** ▪ The value -1 indicates that this field is ignored in the rule. ▪ This parameter is available only in the 'Source Phone Number Manipulation Table for Tel -> IP Calls' and 'Destination Phone Number Manipulation Table for Tel -> IP Calls pages. ▪ If this Source IP Group has a Serving IP Group, then all calls originating from this Source IP Group are sent to the Serving IP Group. In this scenario, this table is used only if the parameter PreferRouteTable is set to 1.
Web: Destination Prefix EMS: Prefix	Destination (called) telephone number prefix and/or suffix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. For a description of notations that you can use to represent single and multiple numbers (ranges), see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 374.
Web/EMS: Source Prefix	Source (calling) telephone number prefix and/or suffix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. For a description of notations that you can use to represent single and multiple numbers (ranges), see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 374.
Web/EMS: Source IP Address	Source IP address of the caller (obtained from the Contact header in the INVITE message). **Notes:** ▪ This parameter is applicable only to the Number Manipulation tables for IP-to-Tel calls. ▪ The source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 to 10.8.8.99. ▪ The source IP address can include the asterisk (*) wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all IP addresses between 10.8.8.0 and 10.8.8.255.
Web: Stripped Digits From Left EMS: Number Of Stripped Digits	Number of digits to remove from the left of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234.
Web: Stripped Digits From Right EMS: Number Of Stripped Digits	Number of digits to remove from the right of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 5551.
Web: Prefix to Add EMS: Prefix/Suffix To Add	The number or string that you want added to the front of the telephone number. For example, if you enter '9' and the phone number is 1234, the new number is 91234.
Web: Suffix to Add EMS: Prefix/Suffix To Add	The number or string that you want added to the end of the telephone number. For example, if you enter '00' and the phone number is 1234, the new number is 123400.

Parameter	Description
Web/EMS: Number of Digits to Leave	The number of digits that you want to retain from the right of the phone number. For example, if you enter '4' and the phone number is 00165751234, then the new number is 1234.
Web: NPI EMS: Number Plan	The Numbering Plan Indicator (NPI) assigned to this entry. ▪ **[0]** Unknown (default) ▪ **[9]** Private ▪ **[1]** E.164 Public ▪ **[-1**] Not Configured = value received from PSTN/IP is used **Notes:** ▪ This parameter is applicable only to Number Manipulation tables for IP-to-Tel calls. ▪ For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 287
Web: TON EMS: Number Type	The Type of Number (TON) assigned to this entry. ▪ If you selected 'Unknown' for the NPI, you can select Unknown **[0].** ▪ If you selected 'Private' for the NPI, you can select Unknown **[0],** Level 2 Regional **[1],** Level 1 Regional **[2],** PISN Specific **[3]** or Level 0 Regional (Local) **[4].** ▪ If you selected 'E.164 Public' for the NPI, you can select Unknown **[0],** International **[1],** National **[2],** Network Specific **[3],** Subscriber **[4]** or Abbreviated **[6].** **Notes:** ▪ This parameter is applicable only to Number Manipulation tables for IP-to-Tel calls. ▪ The default is 'Unknown'.
Web: Presentation EMS: Is Presentation Restricted	Determines whether Caller ID is permitted: ▪ Not Configured = Privacy is determined according to the Caller ID table (see 'Configuring Caller Display Information' on page 357). ▪ **[0]** Allowed = Sends Caller ID information when a call is made using these destination/source prefixes. ▪ **[1]** Restricted = Restricts Caller ID information for these prefixes. **Notes:** ▪ This field is applicable only to Number Manipulation tables for source number manipulation. ▪ If 'Presentation' is set to 'Restricted' and the AssertedIdMode parameter is set to 'P-Asserted', the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.

19.3.3 Configuring Redirect Number IP to Tel

The Redirect Number IP > Tel page allows you to configure IP-to-Tel redirect number manipulation rules. This feature allows you to manipulate the value of the received SIP Diversion, Resource-Priority, or History-Info headers, which is then added to the Redirecting Number Information Element (IE) in the ISDN Setup message that is sent to the Tel side.

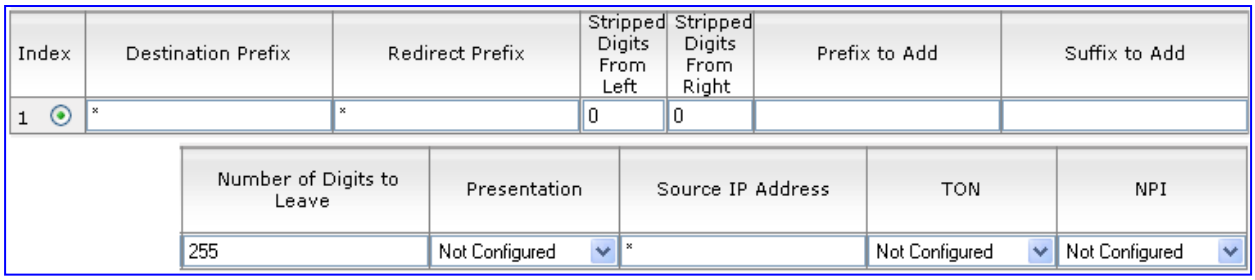
Notes:

- You can also configure the Redirect Number IP to Tel table using the *ini* file parameter RedirectNumberMapIp2Tel (see 'Number Manipulation Parameters' on page 962).
- If the characteristics Destination Prefix, Redirect Prefix, and/or Source Address match the incoming SIP message, manipulation is performed according to the configured manipulation rule.
- The manipulation rules are done in the following order: Stripped Digits From Left, Stripped Digits From Right, Number of Digits to Leave, Prefix to Add, and then Suffix to Add.
- The Destination Number and Redirect Prefix parameters are used before any manipulation has been done on them.

➢ **To configure Redirect Number IP-to-Tel manipulation rules:**

1. Open the Redirect Number IP > Tel page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu > **Redirect Number IP > Tel**).

Figure 19-9: Redirect Number IP to Tel Page



2. Configure the rules according to the table below.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 19-9: Redirect Number IP to Tel Parameters Description

Parameter	Description
Web/EMS: Destination Prefix	Destination (called) telephone number prefix. An asterisk (*) represents any number.
Web/EMS: Redirect Prefix	Redirect telephone number prefix. An asterisk (*) represents any number.
Web: Stripped Digits From Left EMS: Remove From Left	Number of digits to remove from the left of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234.
Web: Stripped Digits From Right EMS: Remove From Right	Number of digits to remove from the right of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 5551.
Web/EMS: Prefix to Add	The number or string that you want added to the front of the telephone number. For example, if you enter '9' and the phone number is 1234, the new number is 91234.
Web/EMS: Suffix to Add	The number or string that you want added to the end of the telephone number. For example, if you enter '00' and the phone number is 1234, the new number is 123400.

Parameter	Description
Web/EMS: Number of Digits to Leave	The number of digits that you want to retain from the right of the phone number.
Web: Presentation EMS: Is Presentation Restricted	Determines whether Caller ID is permitted: ▪ Not Configured = Privacy is determined according to the Caller ID table (see 'Configuring Caller Display Information' on page 357). ▪ **[0]** Allowed = Sends Caller ID information when a call is made using these destination / source prefixes. ▪ **[1]** Restricted = Restricts Caller ID information for these prefixes. **Notes:** ▪ If 'Presentation' is set to 'Restricted' and the AssertedIdMode parameter is set to 'P-Asserted', the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.
Web/EMS: Source IP Address	Source IP address of the caller (obtained from the Contact header in the INVITE message). **Note:** The source IP address can include the following wildcards: ▪ "x": represents single digits. For example, 10.8.8.xx denotes all addresses between 10.8.8.10 and 10.8.8.99. ▪ "*": represents any number between 0 and 255. For example, 10.8.8.* denotes all addresses between 10.8.8.0 and 10.8.8.255.
Web: TON EMS: Number Type	The Type of Number (TON) assigned to this entry. The default is 'Unknown' **[0]**. ▪ If you select 'Unknown' for the NPI, you can select Unknown **[0]**. ▪ If you select 'Private' for the NPI, you can select Unknown **[0]**, International **[1]**, National **[2]**, Network Specific **[3]** or Subscriber **[4]**. ▪ If you select 'E.164 Public' for the NPI, you can select Unknown **[0]**, International **[1]**, National **[2]**, Network Specific **[3]**, Subscriber **[4]** or Abbreviated **[6]**.
Web: NPI EMS: Number Plan	The Numbering Plan Indicator (NPI) assigned to this entry. ▪ **[0]** Unknown (default) ▪ **[9]** Private ▪ **[1]** E.164 Public ▪ **[-1]** Not Configured = value received from PSTN/IP is used **Note:** For more information on available NPI/TON values, see 'Numbering Plans and Type of Number' on page 287.

19.3.4 Configuring Redirect Number Tel to IP

The Redirect Number Tel > IP page allows you to configure Tel-to-IP Redirect Number manipulation rules. This feature manipulates the prefix of the redirect number received from the PSTN for the outgoing SIP Diversion, Resource-Priority, or History-Info header that is sent to IP.

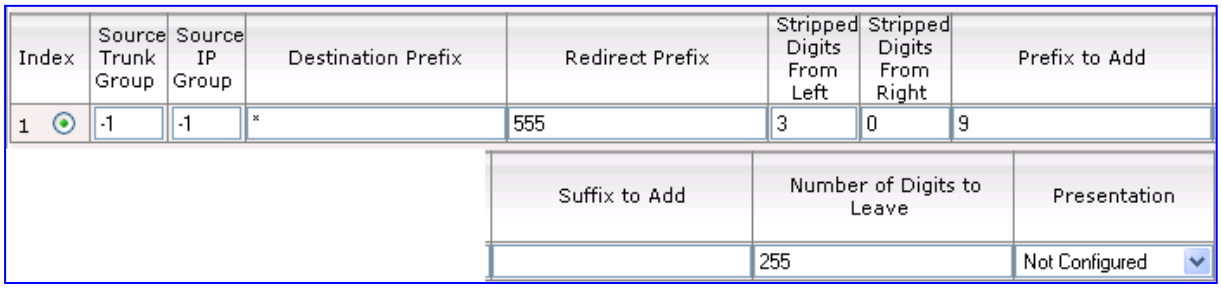
Notes:

- Redirect Tel-to-IP manipulation is not done if the device copies the received destination number to the outgoing SIP redirect number, as enabled by the CopyDest2RedirectNumber parameter.
- You can also configure the Redirect Number Tel to IP table using the *ini* file parameter RedirectNumberMapTel2Ip (see 'Number Manipulation Parameters' on page 962).
- If the characteristics Destination Prefix, Redirect Prefix, and/or Source Address match the incoming SIP message, manipulation is performed according to the configured manipulation rule.
- The manipulation rules are executed in the following order: Stripped Digits From Left, Stripped Digits From Right, Number of Digits to Leave, Prefix to Add, and then Suffix to Add.
- The Destination Number and Redirect Prefix parameters are used before any manipulation has been done on them.

➢ **To configure redirect Tel-to-IP manipulation rules:**

1. Open the Redirect Number Tel > IP page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu > **Redirect Number Tel > IP**).

Figure 19-10: Redirect Number Tel to IP Page



The figure below shows an example configuration in which the redirect prefix "555" is manipulated. According to the configured rule, if for example the number 5551234 is received, after manipulation the device sends the number to IP as 91234.

2. Configure the redirect number Tel to IP rules according to the table below.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 19-10: Redirect Number Tel to IP Parameters Description

Parameter	Description
Source Trunk Group	The Trunk Group from where the Tel call is received. To denote any Trunk Group, leave this field empty. **Notes:** ▪ The value -1 indicates that this field is ignored in the rule. ▪ For IP-to-IP call routing, this parameter is not required (i.e., leave the field empty).
Source IP Group	The IP Group from where the IP-to-IP call originated. Typically, the IP Group of an incoming INVITE is determined/classified using the Inbound IP Routing Table'. If not used (i.e., any IP Group), simply leave the field empty.

Parameter	Description
	Notes: ▪ The value -1 indicates that it is ignored in the rule. ▪ This parameter is applicable only to the IP-to-IP application.
Web/EMS: Destination Prefix	Destination (called) telephone number prefix. An asterisk (*) represents any number.
Web/EMS: Redirect Prefix	Redirect telephone number prefix. An asterisk (*) represents any number.
Web: Stripped Digits From Left EMS: Remove From Left	Number of digits to remove from the left of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234.
Web: Stripped Digits From Right EMS: Remove From Right	Number of digits to remove from the right of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 5551.
Web/EMS: Prefix to Add	The number or string that you want added to the front of the telephone number. For example, if you enter '9' and the phone number is 1234, the new number is 91234.
Web/EMS: Suffix to Add	The number or string that you want added to the end of the telephone number. For example, if you enter '00' and the phone number is 1234, the new number is 123400.
Web/EMS: Number of Digits to Leave	The number of digits that you want to retain from the right of the phone number.
Web: Presentation EMS: Is Presentation Restricted	Determines whether Caller ID is permitted: ▪ Not Configured = Privacy is determined according to the Caller ID table (see 'Configuring Caller Display Information' on page 357). ▪ **[0]** Allowed = Sends Caller ID information when a call is made using these destination/source prefixes. ▪ **[1]** Restricted = Restricts Caller ID information for these prefixes. **Note:** If 'Presentation' is set to 'Restricted' and the AssertedIdMode parameter is set to 'P-Asserted', then the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.

19.3.5 Mapping NPI/TON to SIP Phone-Context

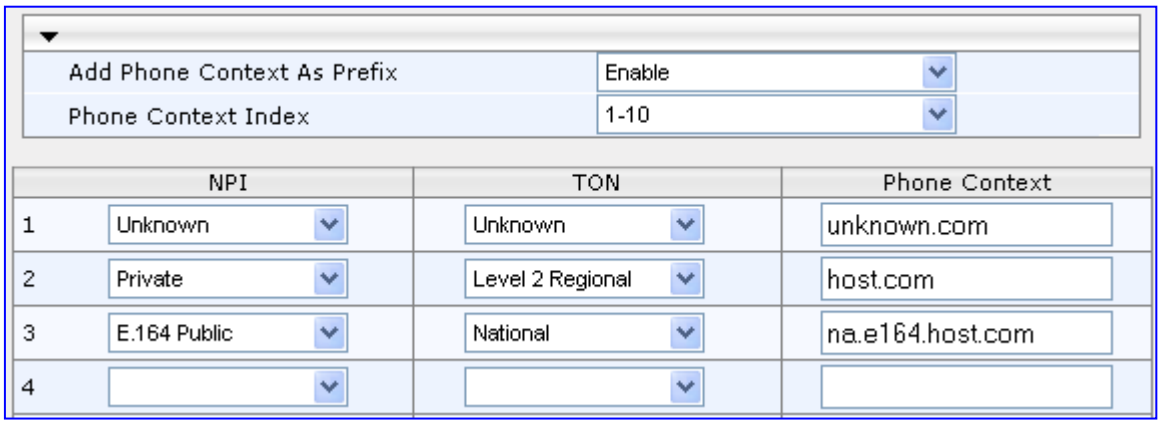
The Phone-Context Table page allows you to map Numbering Plan Indication (NPI) and Type of Number (TON) to the SIP Phone-Context parameter. When a call is received from the ISDN/Tel, the NPI and TON are compared against the table and the matching Phone-Context value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a Phone-Context attribute is received. The Phone-Context parameter appears in the standard SIP headers where a phone number is used (Request-URI, To, From, Diversion).

For example, for a Tel-to-IP call with NPI/TON set as E164 National (values 1/2), the device sends the outgoing SIP INVITE URI with the following settings: “sip:12365432;phone-context= na.e.164.nt.com“. This is configured for entry 3 in the figure below. In the opposite direction (IP-to-Tel call), if the incoming INVITE contains this Phone-Context (e.g. "phone-context= na.e.164.nt.com"), the NPI/TON of the called number in the outgoing SETUP message is changed to E164 National.

➢ **To configure the Phone-Context tables:**

1. Open the Phone Context Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu > **Phone Context**).

Figure 19-11: Phone Context Table Page



2. Configure the Phone Context table according to the table below.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Notes:

- Several rows with the same NPI-TON or Phone-Context are allowed. In such a scenario, a Tel-to-IP call uses the first match.
- You can also configure the Phone Context table using the *ini* file table parameter PhoneContext (see 'Number Manipulation Parameters' on page 962).

Table 19-11: Phone-Context Parameters Description

Parameter	Description
Add Phone Context As Prefix **[AddPhoneContextAsPrefix]**	Determines whether the received Phone-Context parameter is added as a prefix to the outgoing ISDN SETUP message (digital interfaces) with Called and Calling numbers. ▪ **[0]** Disable (default) ▪ **[1]** Enable
NPI	Select the Number Plan assigned to this entry. ▪ **[0]** Unknown (default) ▪ **[1]** E.164 Public ▪ **[9]** Private For a detailed list of the available NPI/TON values, see Numbering Plans and Type of Number on page 287.
TON	Select the Type of Number assigned to this entry. ▪ If you selected Unknown as the NPI, you can select Unknown **[0]**. ▪ If you selected Private as the NPI, you can select one of the following: ✓ **[0]** Unknown ✓ **[1]** Level 2 Regional ✓ **[2]** Level 1 Regional ✓ **[3]** PSTN Specific

Parameter	Description
	✓ **[4]** Level 0 Regional (Local) ▪ If you selected E.164 Public as the NPI, you can select one of the following: ✓ **[0]** Unknown ✓ **[1]** International ✓ **[2]** National ✓ **[3]** Network Specific ✓ **[4]** Subscriber ✓ **[6]** Abbreviated
Phone Context	The Phone-Context SIP URI parameter.

19.3.6 Numbering Plans and Type of Number

The IP-to-Tel destination or source number manipulation tables allow you to classify numbers by their Numbering Plan Indication (NPI) and Type of Number (TON). The device supports all NPI/TON classifications used in the standard. The list of ISDN ETSI NPI/TON values is shown in the following table:

Table 19-12: NPI/TON Values for ISDN ETSI

NPI	TON	Description
Unknown **[0]**	Unknown **[0]**	A valid classification, but one that has no information about the numbering plan.
E.164 Public **[1]**	Unknown **[0]**	A public number in E.164 format, but no information on what kind of E.164 number.
	International **[1]**	A public number in complete international E.164 format, e.g., 16135551234.
	National **[2]**	A public number in complete national E.164 format, e.g., 6135551234.
	Subscriber **[4]**	A public number in complete E.164 format representing a local subscriber, e.g., 5551234.
Private **[9]**	Unknown **[0]**	A private number, but with no further information about the numbering plan.
	Level 2 Regional **[1]**	
	Level 1 Regional **[2]**	A private number with a location, e.g., 3932200.
	PISN Specific **[3]**	
	Level 0 Regional (local) **[4]**	A private local extension number, e.g., 2200.

For NI-2 and DMS-100 ISDN variants, the valid combinations of TON and NPI for calling and called numbers include (Plan/Type):

- 0/0 - Unknown/Unknown
- 1/1 - International number in ISDN/Telephony numbering plan
- 1/2 - National number in ISDN/Telephony numbering plan
- 1/4 - Subscriber (local) number in ISDN/Telephony numbering plan
- 9/4 - Subscriber (local) number in Private numbering plan

19.3.7 Configuring Release Cause Mapping

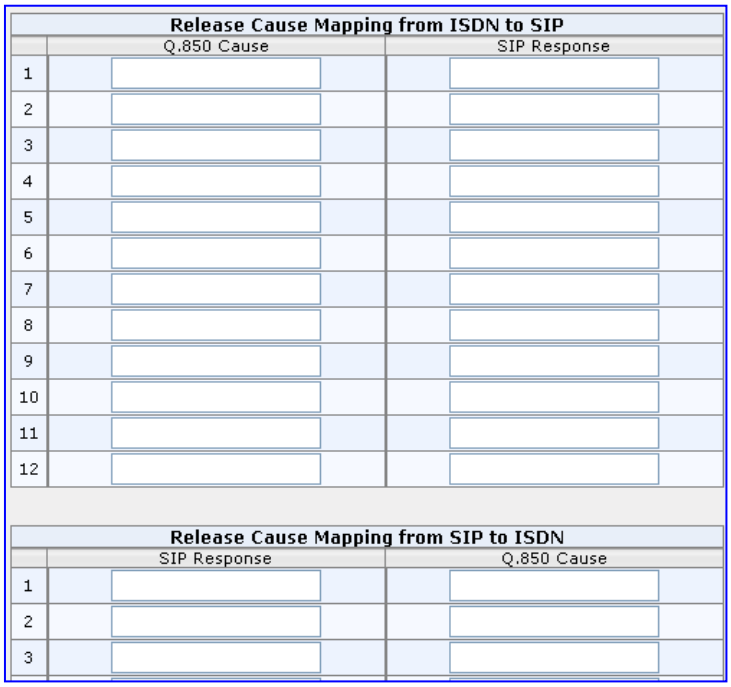
The Release Cause Mapping page consists of two groups that allow the device to map up to 12 different SIP Response Codes to ITU-T Q.850 Release Cause Codes and vice versa, thereby overriding the hard-coded mapping mechanism (described in 'Release Reason Mapping' on page 262).

Note: You can also configure SIP Responses-Q.850 Release Causes mapping using the *ini* file table parameters CauseMapISDN2SIP and CauseMapSIP2ISDN (see 'ISDN and CAS Interworking-Related Parameters' on page 912).

➢ **To configure Release Cause Mapping:**

1. Open the Release Cause Mapping page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu > **Release Cause Mapping**).

Figure 19-12: Release Cause Mapping Page



2. In the 'Release Cause Mapping from ISDN to SIP' group, map different Q.850 Release Causes to SIP Responses.
3. In the 'Release Cause Mapping from SIP to ISDN' group, map different SIP Responses to Q.850 Release Causes.
4. Click **Submit** to apply your changes.

19.3.8 SIP Calling Name Manipulations

You can configure manipulation rules for manipulating the calling name (i.e., caller ID) in the SIP message. This can include modifying or removing the calling name. SIP calling name manipulation is applicable to Tel-to-IP and IP-to-Tel calls.

For example, assume that an incoming SIP INVITE message includes the following header:

```
P-Asserted-Identity: "company:john" sip:6666@78.97.79.104
```

Using the IP-to-Tel calling name manipulation, the text, "company" can be changed to "worker" in the outgoing INVITE, as shown below:

```
P-Asserted-Identity: "worker:john" sip:996666@10.13.83.10
```

To manipulate the calling name received in the SIP message, use the following patrameters:

- For IP-to-Tel calls, use the CallingNameMapIp2Tel ini file parameter
- For Tel-to-IP calls, use the CallingNameMapTel2Ip ini file parameter

19.3.9 SIP Message Manipulation

You can manipulate SIP messages using the Message Manipulations table. This can be configured using the MessageManipulations ini file parameter.

Once you have defined SIP message manipulation rules, you can assign them to inbound and outbound SIP messages:

- For manipulation on all inbound SIP INVITE messages, the Manipulation Set ID is selected (and enabled) using the "global" parameter, GWInboundManipulationSet.
- For manipulation on outbound SIP INVITE messages, the Manipulation Set ID is selected (and enabled) using the following logic:
  - **e.** According to the settings of the Outbound Message Manipulation Set parameter of the destination IP Group table. In other words, manipulation can be done per destination IP Group. If this parameter is not configured, see b below.
  - **f.** According to the settings of the "global" parameter, GWOutboundManipulationSet. If this parameter is also not configured, no manipulation is done.

Note: For the IP-to-IP application, the outgoing message is re-created and SIP headers not relevant to the outgoing SIP session (e.g., Referred-By) are not included in the outgoing message. Therefore, if required, manipulations on such headers should be handled in inbound manipulation).

19.3.10 Manipulating Number Prefix

The device supports a notation for adding a prefix where part of the prefix is first extracted from a user-defined location in the original destination or source number. This notation is entered in the 'Prefix to Add' field in the Number Manipulation tables (see 'Manipulation' on page 277):

x**[n,**l]y...

where,

- *x* = any number of characters/digits to add at the beginning of the number (i.e. first digits in the prefix).
- ***[n,l]*** = defines the location in the original destination or source number where the digits *y* are added:
  - *n* = location (number of digits counted from the left of the number) of a specific string in the original destination or source number.
  - *l* = number of digits that this string includes.
- y = prefix to add at the specified location.

For example, assume that you want to manipulate an incoming IP call with destination number +5492028888888 (area code 202 and phone number 8888888) to the number 0202158888888. To perform such a manipulation, the following configuration is required in the Number Manipulation table:

1. The following notation is used in the 'Prefix to Add' field:

   0**[5,**3]15

   where,

   - 0 is the number to add at the beginning of the original destination number.
   - **[5,**3] denotes a string that is located after (and including) the fifth character (i.e., the first '2' in the example) of the original destination number, and its length being three digits (i.e., the area code 202, in the example).
   - 15 is the number to add immediately after the string denoted by **[5,**3] - in other words, 15 is added after (i.e. to the right of) the digits 202.
2. The first seven digits from the left are removed from the original number, by entering "7" in the 'Stripped Digits From Left' field.

Figure 19-13: Prefix to Add Field with Notation

Index	Destination Prefix	Source Prefix	Source IP Address	Stripped Digits From Left	Stripped Digits From Right	Prefix to Add
1	+5492028888888	*	*	7	0	0[5,3]15

In this configuration, the following manipulation process occurs: 1) the prefix is calculated, 020215 in the example; 2) the first seven digits from the left are removed from the original number, in the example, the number is changed to 8888888; 3) the prefix that was previously calculated is then added.

19.4 Routing

This section describes the configuration of call routing rules.

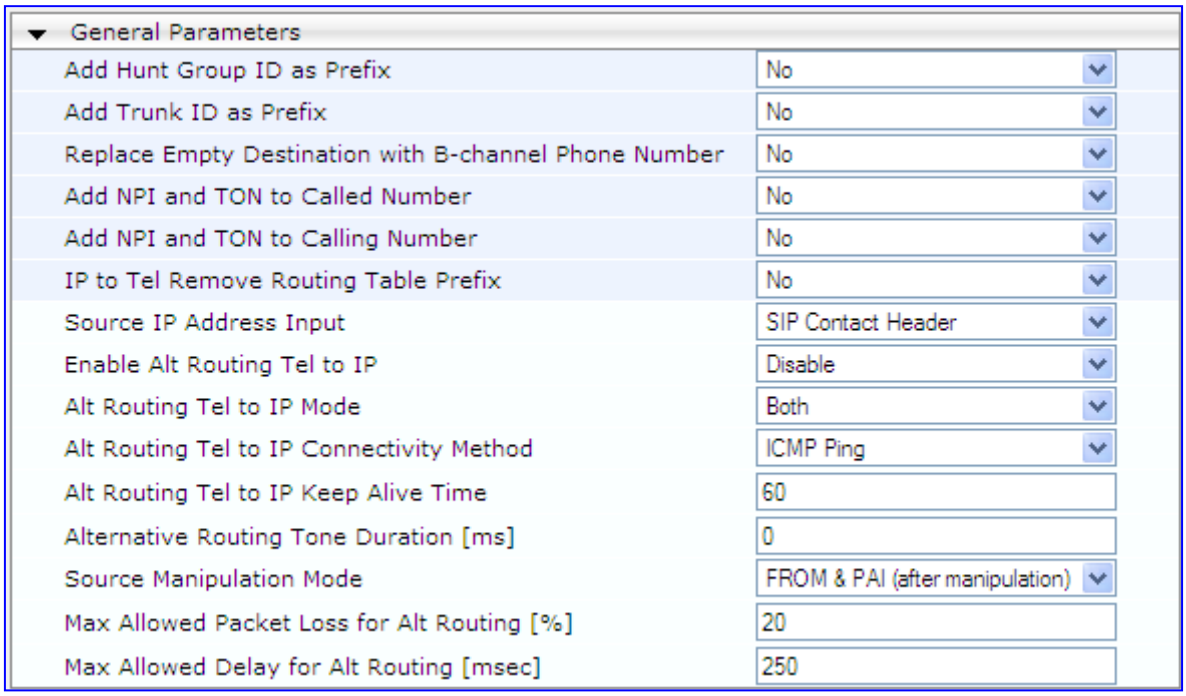
19.4.1 Configuring General Routing Parameters

The Routing General Parameters page allows you to configure general routing parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 752.

➢ **To configure general routing parameters:**

1. Open the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **General Parameters**).

Figure 19-14: Routing General Parameters Page



2. Configure the parameters as required.
3. Click **Submit** to apply your changes.

19.4.2 Configuring Outbound IP Routing Table

The Outbound IP Routing Table page allows you to configure up to 180 Tel-to-IP/outbound IP call routing rules. The device uses these rules to route calls from the Tel or IP to user-defined IP destinations.

The Outbound IP Routing Table table provides two main areas for defining a routing rule:

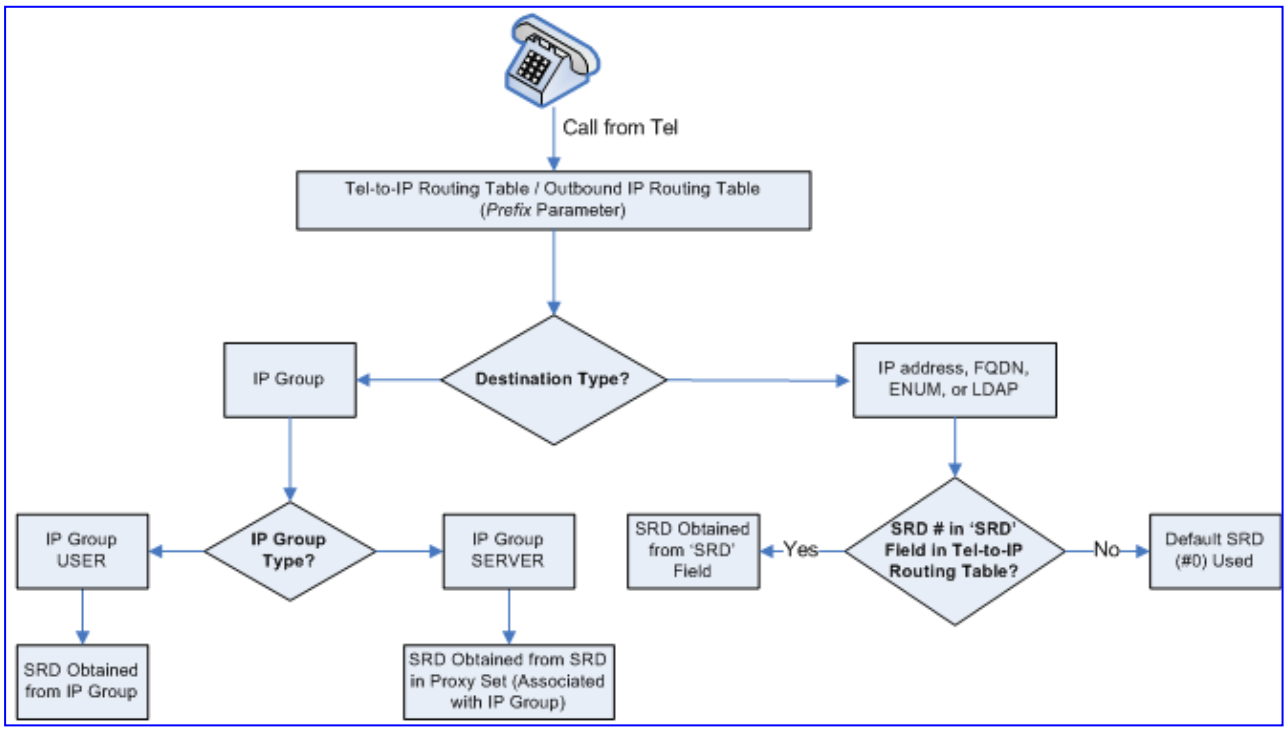
- **Matching Characteristics:** User-defined characteristics of the incoming call. If the call characteristics match a table entry, the routing rule is used to route the call to the specified destination. One or more characteristics can be defined for the rule:
  - Source IP Group (to which the call belongs)
  - Source and destination Request-URI host name prefix
  - Source Trunk Group (from where the call is received)

- Source (calling) and destination (called) telephone number prefix and suffix
- Source and destination Request-URI host name prefix

- **Destination:** User-defined IP destination. If the call matches the characteristics, the device routes the call to this destination. If the number dialed does not match the characteristics, the call is not made. The destination can be any of the following:
  - IP address
  - Fully Qualified Domain Name (FQDN)
  - E.164 Number Mapping (ENUM)
  - Lightweight Directory Access Protocol (LDAP) - for a description, see 'Routing Based on LDAP Active Directory Queries' on page 189
  - IP Group - the call is routed to the Proxy Set (IP address) or SRD associated with the IP Group (defined in 'Configuring IP Groups' on page 212). If the device is configured with multiple SRDs, you can also indicate (in the table's 'Dest. SRD' field) the destination SRD for routing to one of the following destination types - IP address, FQDN, ENUM, or LDAP. If the SRD is not selected, then the default SRD–0–is used. In scenarios where routing is to an IP Group, the destination SRD is obtained from the SRD defined for that IP Group (in the IP Group table). The specified destination SRD determines the:
  - Destination SIP interface (SIP port and control IP interface) - important when using multiple SIP control VLANs
  - Media Realm (port and IP interface for media / RTP voice)
  - Other SRD-related interfaces and features on which the call is routed

  Since each call must have a destination IP Group (even in cases where the destination type is not to an IP Group), in cases when the IP Group is not specified, the SRD's default IP Group is used (the first defined IP Group that belongs to the SRD).

Figure 19-15: Locating SRD



Notes: When using a proxy server, you do not need to configure this table unless you require one of the following:

- Fallback(alternative) routing if communication is lost with the proxy server.
- IP security, whereby the device routes only received calls whose source IP addresses are defined in this table. IP security is enabled using the SecureCallFromIP parameter.
- Filter Calls to IP feature: the device checks this table before a call is routed to the proxy server. However, if the number is not allowed, i.e., the number does not exist in the table or a Call Restriction (see below) routing rule is applied, the call is released.
- Obtain different SIP URI host names (per called number).
- Assign IP Profiles to calls.

For this table to take precedence over a proxy for routing calls, you need to set the parameter PreferRouteTable to 1. The device checks the 'Destination IP Address' field in this table for a match with the outgoing call; a proxy is used only if a match is not found.

In addition to basic outbound IP routing, supports the following features:

- **Least cost routing (LCR):** If the LCR feature is enabled, the device searches the routing table for matching routing rules and then selects the one with the lowest call cost. The call cost of the routing rule is done by assigning it a Cost Group. For configuring Cost Groups, see 'Least Cost Routing' on page 196. If two routing rules have identical costs, then the rule appearing higher up in the table (i.e., first-matched rule) is used. If a selected route is unavailable, the device uses the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched routing rules with Cost Groups, according to the settings of the LCR parameter, LCRDefaultCost (see 'Enabling the LCR Feature' on page 200).
- **Call forking:** If the Tel-to-IP Call Forking feature is enabled, the device can send a Tel call to multiple IP destinations. An incoming Tel call with multiple matched routing rules (e.g., all with the same source prefix numbers) can be sent (forked) to multiple IP destinations if the rules are defined with a Forking Group in the table. The call is established with the first IP destination that answers the call.
- **Call Restriction:** Rejects calls whose routing rule is associated with the destination IP address of 0.0.0.0.
- **Always Use Routing Table feature:** Even if a proxy server is used, the SIP Request-URI host name in the sent INVITE message is obtained from this table. Using this feature, you can assign a different SIP URI host name for different called and/or calling numbers. This feature is enabled using the AlwaysUseRouteTable parameter.
- **Assign IP Profiles:** IP Profiles can be assigned to destination addresses (also when a proxy is used).
- **Alternative Routing (when a proxy isn't used):** An alternative IP destination can be configured for a specific call. To associate an alternative IP address to a called telephone number prefix, assign it with an additional entry (with a different IP address), or use an FQDN that resolves into two IP addresses. For more information on alternative routing, see "Alternative Routing for IP-to-Tel Calls" on page 307.

Notes:

- Outbound IP routing can be performed before or after number manipulation. This is configured using the RouteModeTel2IP parameter, as described below.
- You can also configure this table using the *ini* file table parameter Prefix (see 'Number Manipulation Parameters' on page 962).

➢ **To configure outbound IP routing rules:**

1. Open the Outbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **Tel to IP Routing**).

Figure 19-16: Outbound IP Routing Table Page

	Src. IPGroupID	Src. Host Prefix	Dest Host Prefix	Src. Hunt Group ID	Dest. Phone Prefix	Source Phone Prefix
1	-1			*	10	100
2	-1			*	10	100
3	-1			*	20	*
4	-1			1	[5,7-9]	*
5	-1			*	00	*
6	2	domain.com		*	*	*
7	-1			*	100	*
8	-1			*	100	*
9	-1			*	100	*
10	-1					

->	Dest. IP Address	Port	Transport Type	Dest. IPGroup ID	Dest. SRD	IP Profile ID	Status	Charge Code	Cost Group ID	Forking Group
	10.33.45.63		Not Configured	-1	-1	1	n/a		Weekend	-1
	10.33.45.50		Not Configured	-1	-1	1	n/a		Weekend_B	-1
			Not Configured	1	-1	0	n/a		None	-1
	domain.com		Not Configured	-1	-1	0	n/a		None	-1
	0.0.0.0		Not Configured	-1	-1	0	n/a		None	-1
	10.343.45.65		Not Configured	-1	-1	0	n/a		None	-1
	10.33.45.68		Not Configured	-1	-1	0	n/a		None	1
	10.33.45.67		Not Configured	-1	-1	0	n/a		None	2
	domain.com		Not Configured	-1	-1	0	n/a		None	1
			Not Configured	-1					None	-1

The figure above displays the following outbound IP routing rules:

- **Rule 1 and Rule 2:** For both rules, the called phone number prefix is 10, the caller's phone number prefix is 100, and the call is assigned IP Profile ID 1. However, Rule 1 is assigned a cheaper Cost Group than Rule 2, and therefore, the call is sent to the destination IP address (10.33.45.63) associated with Rule 1.
- **Rule 3:** For all callers (*), if the called phone number prefix is 20, the call is sent to the destination according to IP Group 1 (which in turn is associated with a Proxy Set ID providing the IP address).
- **Rule 4:** If the called phone number prefix is 5, 7, 8, or 9 and the caller belongs to Trunk Group ID 1, the call is sent to domain.com.
- **Rule 5:** For all callers (*), if the called phone number prefix is 00, the call is rejected (discarded).
- **Rule 6:** If an incoming IP call pertaining to Source IP Group 2 with domain.com as source host prefix in its SIP Request-URI, the IP call is sent to IP address 10.33.45.65.
- **Rule 7, Rule 8, and Rule 9:** For all callers (*), if the called phone number prefix is 100, the call is sent to Rule 7 and 9 (belonging to Forking Group "1"). If their destinations are unavailable and alternative routing is enabled, the call is sent to Rule 8 (Forking Group "2").

2. From the 'Routing Index' drop-down list, select the range of entries that you want to add.
3. Configure the routing rules according to the table below.
4. Click Subm**it to** apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 19-13: Outbound IP Routing Table Parameters

Parameter	Description
Web/EMS: Tel to IP Routing Mode **[RouteModeTel2IP]**	Determines whether to route received calls to an IP destination before or after manipulation of the destination number. ▪ **[0]** Route calls before manipulation = Calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied. **Notes:** ▪ This parameter is not applicable if outbound proxy routing is used. ▪ For number manipulation, see 'Configuring Number Manipulation Tables' on page 277.
Web: Src. IPGroupID EMS: Source IP Group ID	Defines the IP Group from where the incoming IP call is received. Typically, the IP Group of an incoming INVITE is determined according to the Inbound IP Routing Table'. **Notes:** ▪ This parameter is applicable only for IP-to-IP routing. ▪ To denote all IP Groups, leave this field empty. ▪ If this IP Group has a Serving IP Group, then all calls from this IP Group are sent to the Serving IP Group. In such a scenario, this routing table is used only if the parameter PreferRouteTable is set to 1.

Parameter	Description
Web: Src. Host Prefix EMS: Source Host Prefix	Defines the prefix of the SIP Request-URI host name in the From header of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Notes:** ▪ To denote any prefix, use the asterisk (*) symbol. ▪ This parameter is applicable only for IP-to-IP routing.
Web: Dest. Host Prefix EMS: Destination Host Prefix	Defines the SIP Request-URI host name prefix of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Notes:** ▪ To denote any prefix, use the asterisk (*) symbol. ▪ This parameter is applicable only for IP-to-IP routing.
Web: Src. Trunk Group ID EMS: Source Trunk Group ID	Defines the Trunk Group from where the call is received. **Notes:** ▪ To denote any Trunk Group, use the asterisk (*) symbol. ▪ This parameter is applicable only for the GW application.
Web: Dest. Phone Prefix EMS: Destination Phone Prefix	Defines the prefix and/or suffix of the called (destination) telephone number. The suffix is enclosed in parenthesis after the suffix value. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. For a description of notations that you can use to represent single and multiple numbers (ranges), see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 374. The number can include up to 50 digits. **Note:** To denote any prefix, enter the asterisk (*) symbol.
Web/EMS: Source Phone Prefix	Defines the prefix and/or suffix of the calling (source) telephone number. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. For a description of notations that you can use to represent single and multiple numbers (ranges), see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 374. The number can include up to 50 digits. **Note:** To denote any prefix, enter the asterisk (*) symbol.
All calls matching all or any combination of the above characteristics are sent to the IP destination defined below. **Note:** For alternative routing, additional entries of the same prefix can be configured.	
Web: Dest. IP Address EMS: Address	Defines the IP address (in dotted-decimal notation or FQDN) to where the call must be sent. If an FQDN is used (e.g., domain.com), DNS resolution is done according to the DNSQueryType parameter. **Notes:** ▪ If you defined a destination IP Group (below), then this IP address is not used for routing and therefore, not required. ▪ To reject calls, enter 0.0.0.0. For example, if you want to prohibit International calls, then in the 'Dest Phone Prefix' field, enter 00 and in the 'Dest IP Address' field, enter 0.0.0.0. ▪ For routing calls between phones connected to the device (i.e., local routing), enter the device's IP address. ▪ When the device's IP address is unknown (e.g., when DHCP is used), enter IP address 127.0.0.1. ▪ When using domain names, you must enter the DNS server's IP address or alternatively, define these names in the 'Internal DNS Table' (see 'Configuring the Internal DNS Table' on page 137).

Parameter	Description
	▪ If the string 'ENUM' is specified for the destination IP address, an ENUM query containing the destination phone number is sent to the DNS server. The ENUM reply includes a SIP URI used as the Request-URI in the outgoing INVITE and for routing (if a proxy is not used). ▪ The IP address can include the following wildcards: ✓ "x": represents single digits. For example, 10.8.8.xx denotes all addresses between 10.8.8.10 and 10.8.8.99. ✓ "*": represents any number between 0 and 255. For example, 10.8.8.* denotes all addresses between 10.8.8.0 and 10.8.8.255.
Web: Port EMS: Destination Port	Defines the destination port to where you want to route the call.
Web/EMS: Transport Type	Defines the transport layer type for sending the IP call: ▪ **[-1]** Not Configured ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to Not Configured (-1), the transport type defined by the SIPTransportType parameter is used.
Web: Dest IP Group ID EMS: Destination IP Group ID	Defines the IP Group to where you want to route the call. The SIP INVITE message is sent to the IP address defined for the Proxy Set ID associated with the IP Group. **Notes:** ▪ If you select an IP Group, you do not need to configure a destination IP address. However, if both parameters are configured in this table, the INVITE message is sent only to the IP Group (and not the defined IP address). ▪ If the destination IP Group is of type USER, the device searches for a match between the Request-URI (of the received INVITE) to an AOR registration record in the device's database. The INVITE is then sent to the IP address of the registered contact. ▪ If the parameter AlwaysUseRouteTable is set to 1 (see 'Configuring IP Groups' on page 212), then the Request-URI host name in the INVITE message is set to the value defined for the parameter 'Dest. IP Address' (above); otherwise, if no IP address is defined, it is set to the value of the parameter 'SIP Group Name' (defined in the IP Group table). ▪ This parameter is used as the 'Serving IP Group' in the Account table for acquiring authentication user/password for this call (see 'Configuring Account Table' on page 237). ▪ For defining Proxy Set ID's, see 'Configuring Proxy Sets Table' on page 219.
Dest SRD	Defines the SRD to where you want to route the call. The actual destination is defined by the Proxy Set associated with the SRD. This allows you to route the call to a specific SIP Media Realm and SIP Interface. To configure SRD's, see Configuring SRD Table on page 207.
IP Profile ID	Assigns an IP Profile to this IP destination call. This allows you to assign numerous configuration attributes (e.g., voice codes) per routing rule. To configure IP Profiles, see 'Configuring IP Profiles' on page 248.
Status	Displays the Quality of Service of the destination IP address: ▪ "n/a" = Alternative Routing feature is disabled

Parameter	Description
	▪ "OK" = IP route is available ▪ "Ping Error" = No ping to IP destination; route is unavailable ▪ "QoS Low" = Poor QoS of IP destination; route is unavailable ▪ "DNS Error" = No DNS resolution (only when domain name is used instead of an IP address)
Web/EMS: Charge Code	Assigns a Charge Code to the routing rule. To configure Charge Codes, see Configuring Charge Codes Table on page 353. **Note:** This parameter is applicable only to FXS interfaces.
Cost Group ID	Assigns a Cost Group to the routing rule for determining the cost of the call. To configure Cost Groups, see 'Configuring Cost Groups' on page 201.
Forking Group	Defines a forking group ID for the routing rule. This enables forking of incoming Tel calls to two or more IP destinations. The device sends simultaneous INVITE messages and handles multiple SIP dialogs until one of the calls is answered. When a call is answered, the other calls are dropped. If all matched routing rules belong to the same Forking Group number, the device sends an INVITE to all the destinations belonging to this group and according to the following logic: ▪ If matched routing rules belong to different Forking Groups, the device sends the call to the Forking Group of the first matched routing rule. If the call cannot be established with any of the destinations associated with this Forking Group and alternative routing is enabled, the device forks the call to the Forking Group of the next matched routing rules as long as the Forking Group is defined with a **higher** number than the previous Forking Group. For example: ▪ Table index entries 1 and 2 are defined with Forking Group "1", and index entries 3 and 4 with Forking Group "2": The device first sends the call according to index entries 1 and 2, and if unavailable and alternative routing is enabled, sends the call according to index entries 3 and 4. ▪ Table index entry 1 is defined with Forking Group "2", and index entries 2, 3, and 4 with Forking Group "1": The device sends the call according to index entry 1 only and ignores the other index entries even if the destination is unavailable and alternative routing is enabled. This is because the subsequent index entries are defined with a Forking Group number that is lower than that of index entry 1. ▪ Table index entry 1 is defined with Forking Group "1", index entry 2 with Forking Group "2", and index entries 3 and 4 with Forking Group "1": The device first sends the call according to index entries 1, 3, and 4 (all belonging to Forking Group "1"), and if the destination is unavailable and alternative routing is enabled, the device sends the call according to index entry 2. ▪ Table index entry 1 is defined with Forking Group "1", index entry 2 with Forking Group "3", index entry 3 with Forking Group "2", and index entry 4 with Forking Group "1": The device first sends the call according to index entries 1 and 4 (all belonging to Forking Group "1"), and if the destination is unavailable and alternative routing is enabled, the device sends the call according to index entry 2 (Forking Group "3"). Even if index entry 2 is unavailable and alternative routing is enabled, the device ignores index entry 3 because it belongs to a Forking Group that is lower than index entry 2. **Notes:** ▪ To enable Tel-to-IP call forking, you must set the 'Tel2IP Call Forking Mode' (*Tel2IPCallForkingMode*) parameter to **Enable**. ▪ You can implement Forking Groups when the destination is an LDAP

Parameter	Description
	server or a domain name using DNS. In such scenarios, the INVITE is sent to all the queried LDAP or resolved IP addresses respectively. You can also use LDAP routing rules with standard routing rules for Forking Groups.

19.4.3 Configuring Inbound IP Routing Table

The Inbound IP Routing Table page allows you to configure up to 24 inbound call routing rules:

- For IP-to-IP routing: identifying IP-to-IP calls and assigning them to IP Groups (referred to as Source IP Groups). These IP-to-IP calls, now pertaining to an IP Group, can later be routed to an outbound destination IP Group (see Configuring Outbound IP Routing Table on page 291).
- For IP-to-Tel routing: routing incoming IP calls to Trunk Groups. The specific channel pertaining to the Trunk Group to which the call is routed is determined according to the Trunk Group's channel selection mode. The channel selection mode can be defined per Trunk Group (see 'Configuring Trunk Group Settings' on page 273), or for all Trunk Groups using the global parameter ChannelSelectMode.

This table provides two main areas for defining a routing rule:

- **Matching Characteristics:** user-defined characteristics of the incoming IP call are defined in this area. If the characteristics match a table entry, the rule is used to route the call. One or more characteristics can be defined for the rule:
  - Source and destination Request-URI host name prefix
  - Source (calling) and destination (called) telephone number prefix and suffix
  - Source IP address (from where the call is received)
- **Destination:** user-defined destination. If the call matches the characteristics, the device routes the call to the defined destination:
  - Trunk Group
  - Source IP Group

The device also supports the following alternative routing methods if the Trunk Group used in the routing rule is unavailable:

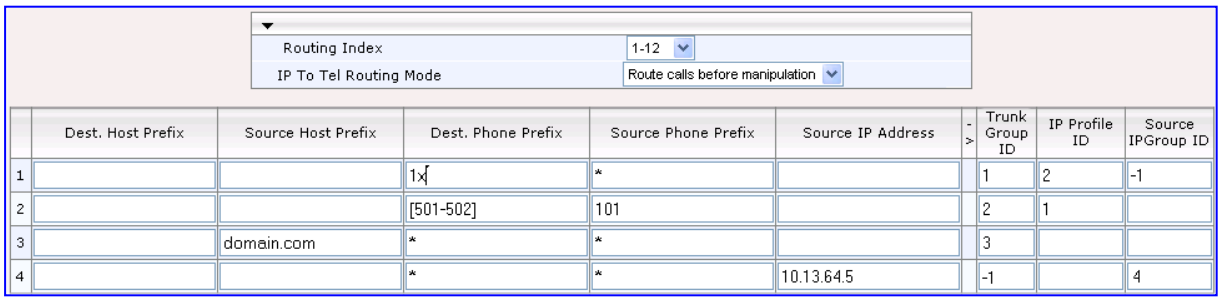
- When a call release reason that you have configured for alternative IP-to-Tel routing is received for a specific IP-to-Tel call, the device re-routes the call to an alternative Trunk Group. This is done by configuring an additional routing rule with the same call characteristics but different Trunk Group ID. For more information on IP-to-Tel alternative routing, see "Alternative Routing to Trunk upon Q.931 Call Release Cause Code" on page 307.
- The device can re-route (i.e., call redirection) IP-to-Tel calls to an alternative IP destination, using SIP 3xx responses. For more information, see "Alternative Routing to IP Destinations upon Busy Trunk" on page 308.

Note: You can also configure the Inbound IP Routing Table using the *ini* file table parameter PSTNPrefix (see 'Number Manipulation Parameters' on page 962).

➢ **To configure inbound IP routing rules:**

1. Open the Inbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **IP to Trunk Group Routing**).

Figure 19-17: Inbound IP Routing Table



The previous figure displays the following configured routing rules:

- **Rule 1:** If the incoming IP call destination phone prefix is between 10 and 19, the call is assigned settings configured for IP Profile ID 2 and routed to Trunk Group ID 1.
- **Rule 2:** If the incoming IP call destination phone prefix is between 501 and 502, and source phone prefix is 101, the call is assigned settings configured for IP Profile ID 1 and routed to Trunk Group ID 2.
- **Rule 3:** If the incoming IP call has a From URI host prefix as domain.com, the call is routed to Trunk Group ID 3.
- **Rule 4:** If the incoming IP call has IP address 10.13.64.5 in the INVITE's Contact header, the call is identified as an IP-to-IP call and assigned to Source IP Group 4. This call is routed according to the outbound IP routing rules for this Source IP Group configured in the Outbound IP Routing Table'.

2. From the 'Routing Index' drop-down list, select the range of entries that you want to add.
3. Configure the inbound IP routing rule according to the table below.
4. Click Submit to apply your changes.
5. To save the changes so they are available after a power failure, see 'Saving Configuration' on page 680.

Table 19-14: Inbound IP Routing Table Description

Parameter	Description
IP to Tel Routing Mode **[RouteModeIP2Tel]**	Determines whether to route the incoming IP call before or after manipulation of destination number (configured in 'Configuring Number Manipulation Tables' on page 277). ▪ **[0]** Route calls before manipulation = Incoming IP calls are routed before number manipulation (default). ▪ **[1]** Route calls after manipulation = Incoming IP calls are routed after number manipulation are applied.
Dest. Host Prefix	The Request-URI host name prefix of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Note:** The asterisk (*) wildcard can be used to denote any prefix.

Parameter	Description
Source Host Prefix	The From URI host name prefix of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Notes:** ▪ The asterisk (*) wildcard can be used to denote any prefix. ▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of this parameter is compared to the P-Asserted-Identity URI host name (and not the From header).
Dest. Phone Prefix	Defines the prefix or suffix of the called (destined) telephone number. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. For a description of notations that you can use to represent single and multiple numbers (ranges), see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 374. The prefix can include up to 49 digits.
Source Phone Prefix	Defines the prefix or suffix of the calling (source) telephone number. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. For a description of notations that you can use to represent single and multiple numbers (ranges), see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 374. The prefix can include up to 49 digits.
Source IP Address	The source IP address of the incoming IP call (obtained from the Contact header in the INVITE message) that can be used for routing decisions. **Notes:** ▪ You can configure from where the source IP address is obtained, using the parameter SourceIPAddressInput. ▪ The source IP address can include the following wildcards: ✓ "x": denotes single digits. For example, 10.8.8.xx represents all the addresses between 10.8.8.10 and 10.8.8.99. ✓ "*": denotes any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255.
Calls matching all or any combination of the above characteristics are sent to the Trunk Group ID or assigned to the source IP Group for IP-to-IP routing defined below. **Note:** For alternative routing, additional entries of the same characteristics can be configured.	
Trunk Group ID	For IP-to-Tel calls: The Trunk Group to which the incoming SIP call is assigned if it matches all or any combination of the parameters described above. For IP-to-IP calls: Identifies the call as an IP-to-IP call when this parameter is set to -1.
IP Profile ID	The IP Profile (configured in 'Configuring IP Profiles' on page 248) to assign to the call.
Source IP Group ID	For IP-to-Tel calls: The IP Group associated with the incoming IP call. This is the IP Group from where the INVITE message originated. This IP Group can later be used as the 'Serving IP Group' in the Account table for obtaining authentication user name/password for this call (see 'Configuring Account Table' on page 237). For IP-to-IP calls: The IP Group you want to assign the incoming IP call. This IP Group can later be used for outbound IP routing and as the 'Serving IP Group' in the Account table for obtaining authentication user name/password for this call (see Configuring Account Table on page 237).

19.4.4 Mapping PSTN Release Cause to SIP Response

The device's FXO interface interoperates between the SIP network and the PSTN/PBX. This interoperability includes the mapping of PSTN/PBX Call Progress Tones to SIP 4xx or 5xx responses for IP-to-Tel calls. The converse is also true - for Tel-to-IP calls, the SIP 4xx or 5xx responses are mapped to tones played to the PSTN/PBX.

When establishing an IP-to-Tel call, the following rules are applied:

- If the remote party (PSTN/PBX) is busy and the FXO device detects a Busy tone, it sends a SIP 486 Busy response to IP. If it detects a Reorder tone, it sends a SIP 404 Not Found (no route to destination) to IP. In both cases, the call is released. Note that if the parameter DisconnectOnBusyTone is set to 0, the FXO device ignores the detection of Busy/Reorder tones and doesn't release the call.
- For all other FXS/FXO release types (caused when there are no free channels in the specific Trunk Group), or when an appropriate rule for routing the call to a Trunk Group doesn't exist, or if the phone number isn't found), the device sends a SIP response (to IP) according to the parameter DefaultReleaseCause. This parameter defines Q.931 release causes. Its default value is '3', which is mapped to the SIP 404 response. By changing its value to '34', the SIP 503 response is sent. Other causes can be used as well.

19.4.5 IP Destinations Connectivity Feature

You can configure the device to check the integrity of the connectivity to IP destinations of Tel-to-IP routing rules in the Outbound IP Routing table. The IP Connectivity feature can be used for the Alternative Routing feature, whereby the device attempts to re-route calls from unavailable Tel-to-IP routing destinations to available ones (see 'Alternative Routing Based on IP Connectivity' on page 303).

The device supports the following methods for checking the connectivity of IP destinations:

- **Network Connectivity:** The device checks the network connectivity of the IP destination using one of the following methods (depending on the settings of the AltRoutingTel2IPConnMethod parameter):
  - **Ping:** The device periodically (every seven seconds) pings the IP destination.
  - **SIP OPTIONS:** The device sends "keep-alive" SIP OPTIONS messages to the IP destination. If the device receives a SIP 200 OK in response, it considers the destination as available. If the destination does not respond to the OPTIONS message, then it is considered unavailable. You can configure the time interval for sending these OPTIONS messages, using the AltRoutingTel2IPKeepAliveTime parameter.
- **Quality of Service (QoS):** You can enable the device to check the QoS of IP destinations. The device measures the QoS according to RTCP statistics of previously established calls with the IP destination. The RTCP includes packet delay (in milliseconds) and packet loss (in percentage). If these measured statistics exceed a user-defined threshold, the destination is considered unavailable. Note that if call statistics is not received within two minutes, the QoS data is reset. These thresholds are configured using the following parameters:
  - IPConnQoSMaxAllowedPL – defines the threshold value for packet loss, after which the IP destination is considered unavailable
  - IPConnQoSMaxAllowedDelay - defines the threshold value for packet delay, after which the IP destination is considered unavailable
- **DNS Resolution:** When a host name (FQDN) is used (instead of an IP address) for the IP destination, it is resolved into an IP address by a DNS server. The device checks network connectivity and QoS of the resolved IP address. If the DNS host

name is unresolved, the device considers the connectivity of the IP destination as unavailable.

You can view the connectivity status of IP destinations in the following Web interface pages:

- **Outbound IP Routing Table:** The connectivity status of the IP destination per routing rule is displayed in the 'Status' column. For more information, see "Configuring Outbound IP Routing Table" on page 291.
- **IP Connectivity:** This page displays a more informative connectivity status of the IP destinations used in Tel-to-IP routing rules in the Outbound IP Routing table. For viewing this page, see "Viewing IP Connectivity" on page 725.

19.4.6 Alternative Routing for Tel-to-IP Calls

The device supports various alternative Tel-to-IP call routing methods, as described in this section.

19.4.6.1 Alternative Routing Based on IP Connectivity

You can configure the device to do alternative Tel-to-IP call routing based on IP connectivity. When the connectivity state of an IP destination is unavailable, the device attempts to re-route the Tel-to-IP call to an alternative IP destination. It does this by searching for the next call matching rule (e.g., phone number prefix) in the Outbound IP Routing table.

Notes:

- Alternative routing based on IP connectivity is applicable only when a proxy server is not used.
- As the device searches the Outbound IP Routing table for a matching rule starting from the top, you must configure the main routing rule above the alternative routing rules.
- You can configure up to two alternative routing rules.
- For configuring Tel-to-IP routing rules in the Outbound IP Routing table, see "Configuring Outbound IP Routing Table" on page 291.

The device searches for an alternative IP destination when any of the following connectivity states are detected with the IP destination of the initial Tel-to-IP routing rule:

- No response received from a ping or from SIP OPTIONS messages. This depends on the chosen method for checking IP connectivity.
- Poor QoS according to the configured thresholds for packet loss and delay.
- Unresolved DNS, if the configured IP destination is a domain name (or FQDN). If the domain name is resolved into two IP addresses, the timeout for INVITE re-transmissions can be configured using the HotSwapRtx parameter. For example, if you set this parameter to 3, the device attempts up to three times to route the call to the first IP address and if unsuccessful, it attempts up to three times to re-route it to the second resolved IP address.

The connectivity status of the IP destination is displayed in the 'Status' column of the Outbound IP Routing table per routing rule. If it displays a status other than "ok", then the device considers the IP destination as unavailable and attempts to re-route the call to an alternative destination. For more information on the IP connectivity methods and on viewing IP connectivity status, see 'IP Destinations Connectivity Feature' on page 302.

The table below shows an example of alternative routing where the device uses an available alternative routing rule in the Outbound IP Routing table to re-route the initial, unsuccessful Tel-to-IP call routing destination.

Table 19-15: Alternative Routing based on IP Connectivity Example

	Destination Phone Prefix	IP Destination	IP Connectivity Status	Rule Used?
Main Route	40	10.33.45.68	"No Connectivity"	No
Alternative Route #1	40	10.33.45.70	"QoS Low"	No
Alternative Route #2	40	10.33.45.72	"ok"	Yes

For analog interfaces: The device plays a tone to the Tel endpoint whenever an alternative route is used. This tone is played for a user-defined time, configured using the AltRoutingToneDuration parameter.

The steps for configuring alternative Tel-to-IP routing based on IP connectivity is summarized below.

➢ **To configure alternative Tel-to-IP routing based on IP connectivity:**

1. Add alternative Tel-to-IP routing rules for specific calls in the Outbound IP Routing table.
2. Enable alternative routing based on IP connectivity, by setting the AltRoutingTel2IPEnable parameter to 1.
3. Determine the IP connectivity reason for triggering alternative routing, using the AltRoutingTel2IPMode parameter:
   - Ping or SIP OPTIONS failure
   - Poor QoS
   - Ping or SIP OPTIONS failure, poor QoS, or unresolved DNS

19.4.6.2 Alternative Routing Based on SIP Responses

You can configure the device to do alternative routing based on the received SIP response. If the SIP response code reflects an error (i.e., 4xx, 5xx, or 6xx) and you have configured this specific response code as a trigger for alternative routing, then the device attempts to re-route the call to an alternative destination.

You can configure up to five SIP response codes for triggering alternative routing. This is done in the Reasons for Alternative Routing table (AltRouteCauseTel2IP parameter), explained later in this section.

Typically, the device performs alternative routing when there is no response at all to an INVITE message after a user-defined number of INVITE re-transmissions, configured using the SIPMaxRtx parameter. In such a scenario, the device issues itself the SIP response code 408 "Request Timeout". If this release code is defined in the Reasons for Alternative Routing table, then alternative routing is done.

Depending on configuration, the alternative routing is done using one of the following entities:

- **Outbound IP Routing Rules:** You can configure up to two alternative routing rules in the table. If the initial, main routing rule destination is unavailable, the device searches the table (starting from the top) for the next call matching rule (e.g., destination phone number), and if available attempts to re-route the call to the IP destination configured for this alternative routing rule. The table below shows an example of alternative routing where the device uses the first available alternative routing rule to re-route the Tel-to-IP call.

Table 19-16: Alternative Routing based on SIP Response Code Example

	Destination Phone Prefix	IP Destination	SIP Response	Rule Used?
Main Route	40	10.33.45.68	408 Request Timeout	No
Alternative Route #1	40	10.33.45.70	486 Busy Here	No
Alternative Route #2	40	10.33.45.72	200 OK	Yes

- **Proxy Sets:** Proxy Sets are used for SERVER-type IP Groups (e.g., an IP PBX) and define the actual IP destination (IP address or FQDN) of the server. As you can define up to five IP destinations per Proxy Set, the device supports proxy redundancy which works together with the alternative routing feature. If the destination of a routing rule in the Outbound IP Routing table is an IP Group, the device routes the call to the IP destination configured for the Proxy Set associated with the IP Group. If the first IP destination of the Proxy Set is unavailable, the device attempts to re-route the call to the next proxy destination, and so on until an available IP destination is located. To enable the Proxy Redundancy feature, set the IsProxyHotSwap parameter to 1 (per Proxy Set) and set the EnableProxyKeepAlive to 1.

  When the Proxy Redundancy feature is enabled, the device continually monitors the connection with the proxies by using keep-alive messages (SIP OPTIONS). The device sends these messages every user-defined interval (ProxyKeepAliveTime parameter). Any response from the proxy, either success (200 OK) or failure (4xx response) is considered as if the proxy is communicating. If there is no response from the first (primary) proxy after a user-defined number of re-transmissions (re-INVITEs) configured using the HotSwapRtx parameter, the device attempts to communicate (using the same INVITE) with the next configured (redundant) proxy in the list, and so on until an available redundant proxy is located. The device's behavior can then be one of the following, depending on the ProxyRedundancyMode parameter setting:

  - The device continues operating with the redundant proxy (now active) until the next failure occurs, after which it switches to the next redundant proxy. This is referred to as *Parking* mode.
  - The device always attempts to operate with the primary proxy. In other words, it switches back to the primary proxy whenever it's available again. This is referred to as *Homing* mode.

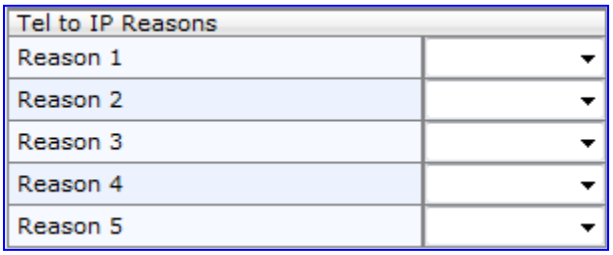
  If none of the proxy servers respond, the device goes over the list again.

The steps for configuring alternative Tel-to-IP routing based on SIP response codes are summarized below.

➢ **To configure alternative Tel-to-IP routing based on SIP response codes:**

1. Enable alternative routing based on SIP responses, by setting the RedundantRoutingMode parameter to 1 (for using the Outbound IP Routing table) or 2 (for using the Proxy Set).
2. If you are using the Outbound IP Routing table, configure alternative routing rules with the same call matching characteristics, but with different IP destinations. If you are using the Proxy Set, configure redundant proxies.
3. Define SIP response codes (call failure reasons) that invoke alternative Tel-to-IP routing:
   a. Open the Reasons for Alternative Routing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **Alternative Routing Reasons**).

Figure 19-18: Tel to IP Reasons - Reasons for Alternative Routing Page



   b. Under the 'Tel to IP Reasons' group, select up to five different SIP response codes (call failure reasons) that invoke alternative Tel-to-IP routing, and then click **Submit**.

19.4.6.3 PSTN Fallback

The PSTN Fallback feature enables the device to re-route a Tel-to-IP call to the legacy PSTN using one of its trunks if the IP destination is unavailable. For example, if poor voice quality is detected over the IP network, the device attempts to re-route the call to the PSTN.

The steps for configuring alternative Tel-to-IP routing to the legacy PSTN are summarized below.

➢ **To configure alternative Tel-to-IP routing to the legacy PSTN:**

1. Configure an alternative routing rule in the Outbound IP Routing table with the same call matching characteristics (e.g., phone number destination), but where the destination is the IP address of the device itself.
2. Configure an IP-to-Tel routing rule in the Inbound IP Routing table to route calls received from the device (i.e., its IP address) to a specific Trunk Group connected to the PSTN. This configuration is necessary as the re-routed call is now considered an IP-to-Tel call. For configuring IP-to-Tel routing rules, see "Configuring the Inbound IP Routing Table" on page 299.

Note: The PSTN Fallback feature is applicable only to digital interfaces (e.g., E1 / T1 trunks).

19.4.7 Alternative Routing for IP-to-Tel Calls

The device supports alternative IP-to-Tel call routing, as described in this section.

19.4.7.1 Alternative Routing to Trunk upon Q.931 Call Release Cause Code

You can configure the device to do alternative IP-to-Tel call routing based on the received ISDN Q.931 cause code. If an IP-to-Tel call is rejected or disconnected on the Tel side as a result of a specific ISDN Q.931 release cause code that is listed in the Reasons for Alternative Routing table (AltRouteCauseIP2Tel), the device searches for an alternative IP-to-Tel routing rule in the Inbound IP Routing table and sends it to the alternative Trunk Group. For example, you can enable alternative IP-to-Tel routing for scenarios where the initial Tel destination is busy and a Q.931 Cause Code No. 17 is issued (or for other call releases that issue the default Cause Code No. 3).

You can also configure a default release cause code that the device issues itself upon the following scenarios:

- The device initiates a call release whose cause is unknown
- No free channels (i.e., busy) in the Trunk Group
- No appropriate routing rule located in the Inbound IP Routing table to the Trunk Group
- Phone number is not found in the Inbound IP Routing table

By default, it is set to Cause Code No. 3 (No Route to Destination). This default cause code can be changed using the DefaultReleaseCause parameter. To enable alternative routing using this cause code, you need to define this cause code in the AltRouteCauseIP2Tel table.

Notes:

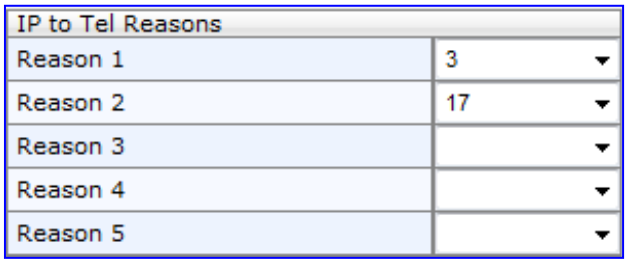
- You can configure up to two alternative routing rules in the Inbound IP Routing table.
- If a Trunk is disconnected or not synchronized, the device issues itself the internal Cause Code No. 27. This cause code is mapped (by default) to SIP 502.
- The default release cause is described in the Q.931 notation and is translated to corresponding SIP 40x or 50x values (e.g., Cause Code No. 3 to SIP 404, and Cause Code No. 34 to SIP 503).
- For analog interfaces: For information on mapping PSTN release causes to SIP responses, see “Mapping PSTN Release Cause to SIP Response” on page 302.
- For mapping SIP-to-Q.931 and Q.931-to-SIP release causes, see “Configuring Release Cause Mapping” on page 288.
- For a list of SIP responses-Q.931 release cause mapping, see “Release Reason Mapping” on page 262.
- For configuring IP-to-Tel routing rules in the Inbound IP Routing table, see "Configuring Inbound IP Routing Table" on page 299.

The steps for configuring alternative Trunk Group routing based on Q.931 cause codes are summarized below.

➢ **To configure alternative Trunk Group routing based on Q.931 cause codes:**

1. Set the RedundantRoutingMode parameter to 1 so that the device uses the Inbound IP Routing table for alternative routing.
2. Configure alternative routing rules in the Inbound IP Routing table with the same call matching characteristics, but with different Trunk Group destinations.
3. Configure up to five Q.931 cause codes that invoke alternative IP-to-Tel routing:
   a. Open the Reasons for Alternative Routing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **Alternative Routing Reasons**).

Figure 19-19: IP to Tel Reasons - Reasons for Alternative Routing Page



   b. Under the 'IP to Tel Reasons' group, select the desired Q.931 cause codes.
   c. Click **Submit** to apply your changes.

19.4.7.2 Alternative Routing to IP Destination upon Busy Trunk

You can configure the device to forward (i.e., call redirection) IP-to-Tel calls to an alternative IP destination using SIP 3xx responses. This can be done upon the following scenarios:

- For digital interfaces: Trunk Group has no free channels (i.e., “busy”).
- For analog interfaces: Unavailable FXS / FXO Trunk Group. This feature can be used, for example, to forward the call to another FXS / FXO device.

This feature is configured per Trunk Group and is configured in the Forward on Busy Trunk Destination page, as described later in this section.

The alternative destination can be defined as a host name (IP address with optional port and transport type), or as a SIP Request-URI user name and host part (i.e., user@host). For example, the below configuration forwards IP-to-Tel calls to destination user “112” at host IP address 10.13.4.12, port 5060, using transport protocol TCP, if Trunk Group ID 2 is unavailable:

```
ForwardOnBusyTrunkDest 1 = 2, 112@10.13.4.12:5060;transport=tcp;
```

When configured with user@host, the original destination number is replaced by the user part.

The device forwards calls using this table only if no alternative IP-to-Tel routing rule has been configured in the Inbound IP Routing table or alternative routing fails and one of the following reasons included in the SIP Diversion header of 3xx messages exists:

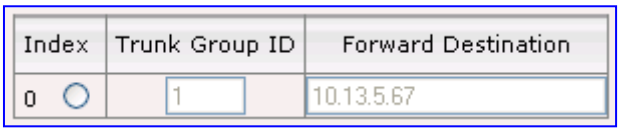
- For digital interfaces: “out-of-service” - all trunks are unavailable/disconnected
- "unavailable":
  - For digital interfaces: All trunks are busy or unavailable
  - For analog interfaces: All FXS / FXO lines pertaining to a Trunk Group are busy or unavailable

Note: You can also configure the Forward on Busy Trunk Destination table using the *ini* file parameter table ForwardOnBusyTrunkDest.

➢ To configure Forward on Busy Trunk Destination rules:

1. Open the Forward on Busy Trunk Destination page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **Forward on Busy Trunk**).

Figure 19-20: Forward on Busy Trunk Destination Page



The figure above displays a configuration that forwards IP-to-Tel calls destined for Trunk Group ID 1 to destination IP address 10.13.5.67 if the conditions mentioned earlier exist.

2. Configure the table as required, and then click **Submit** to apply your changes.

Save the changes to the device's flash memory with a device reset (see "Saving Configuration" on page 680).

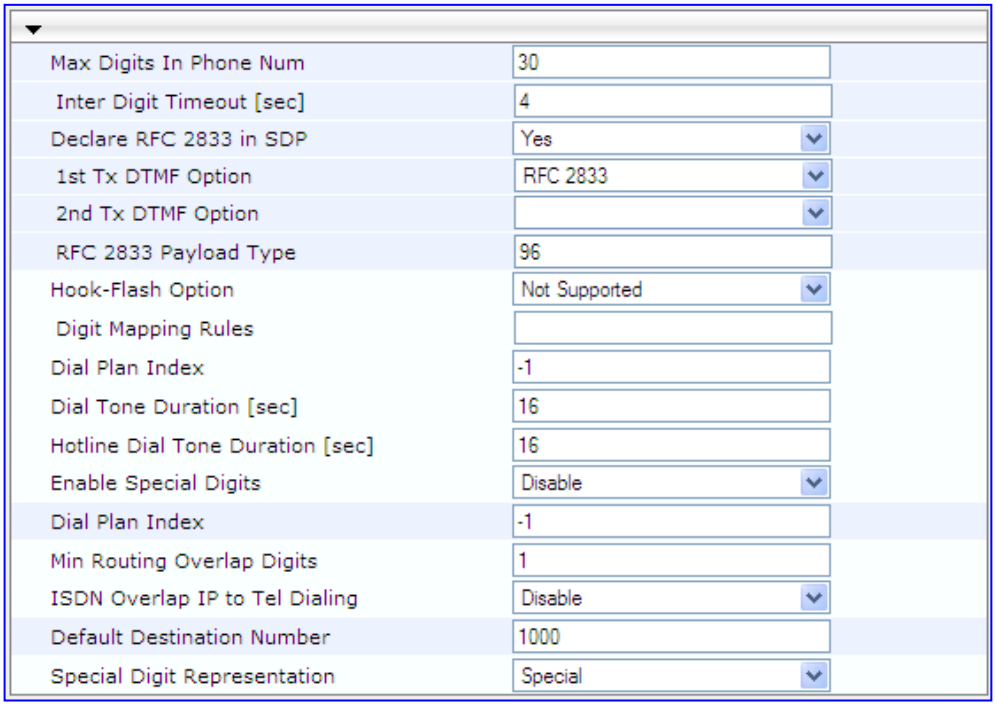
19.5 Configuring DTMF and Dialing

The DTMF & Dialing page is used to configure parameters associated with dual-tone multi-frequency (DTMF) and dialing. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

➢ **To configure the DTMF and dialing parameters:**

1. Open the DTMF & Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **DTMF & Supplementary** submenu > **DTMF & Dialing**).

Figure 19-21: DTMF & Dialing Page



2. Configure the parameters as required.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.6 Configuring Supplementary Services

The Supplementary Services page is used to configure parameters associated with supplementary services. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.

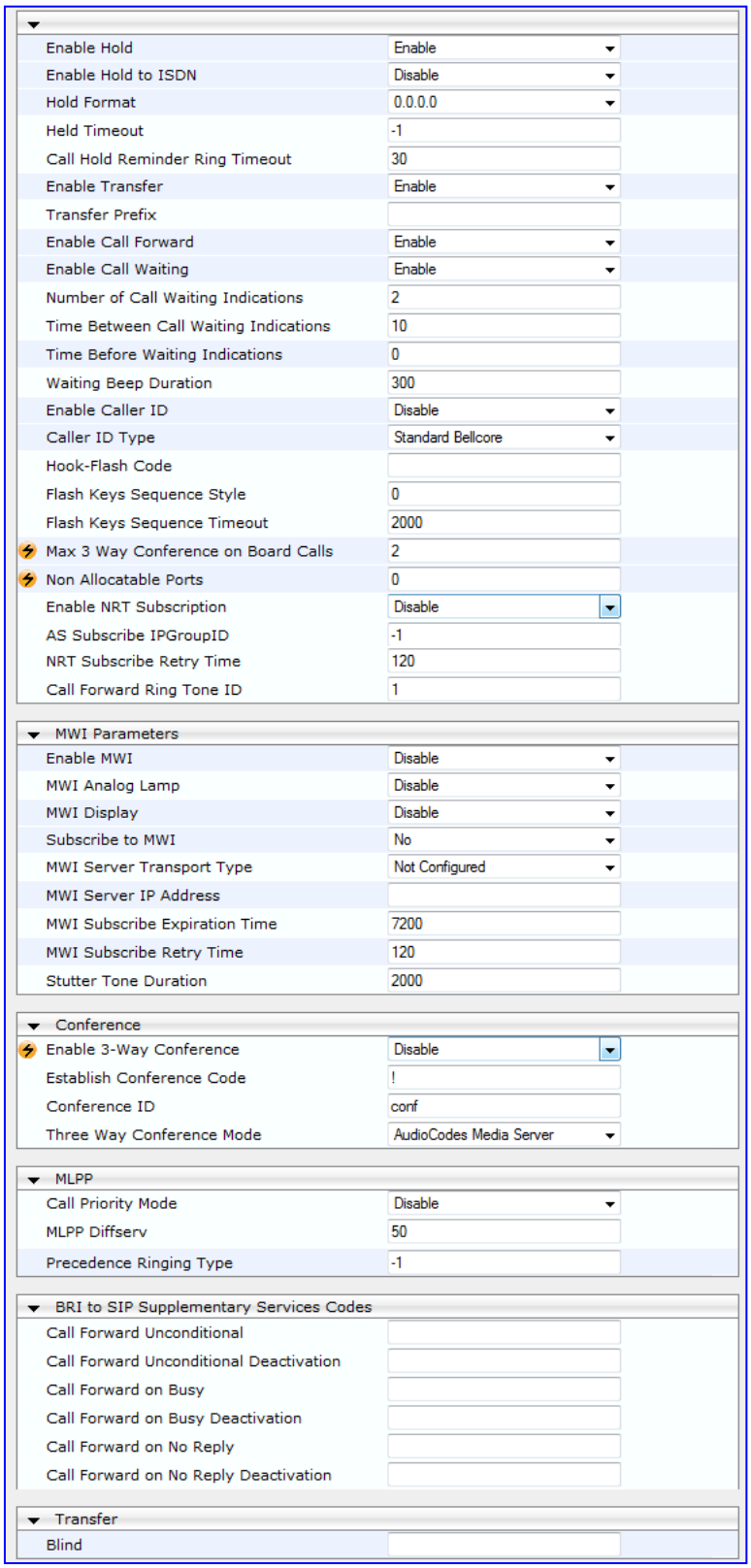
Notes:

- All call participants must support the specific supplementary service that is used.
- When working with certain application servers (such as BroadSoft's BroadWorks) in client server mode (the application server controls all supplementary services and keypad features by itself), the device's supplementary services must be disabled.

➢ **To configure supplementary services parameters:**

1. Open the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **DTMF & Supplementary** submenu > **Supplementary Services**).

Figure 19-22: Supplementary Services Page



2. Configure the parameters as required.
3. Click **Submit** to apply your changes, or click the **Subscribe to MWI** or **Unsubscribe to MWI** buttons to save your changes and to subscribe / unsubscribe to the MWI server.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.6.1 Call Hold and Retrieve

Initiating Call Hold and Retrieve:

- Active calls can be put on-hold by pressing the phone's hook-flash button.
- The party that initiates the hold is called the *holding* party; the other party is called the *held* party.
- After a successful Hold, the holding party hears a Dial tone (HELD_TONE defined in the device's Call Progress Tones file).
- Call retrieve can be performed only by the holding party while the call is held and active.
- The holding party performs the retrieve by pressing the telephone's hook-flash button.
- After a successful retrieve, the voice is connected again.
- Hold is performed by sending a Re-INVITE message with IP address 0.0.0.0 or a=sendonly in the SDP according to the parameter HoldFormat.
- The hold and retrieve functionalities are implemented by Re-INVITE messages. The IP address 0.0.0.0 as the connection IP address or the string 'a=inactive' in the received Re-INVITE SDP cause the device to enter Hold state and to play the Held tone (configured in the device) to the PBX/PSTN. If the string 'a=sendonly' is received in the SDP message, the device stops sending RTP packets, but continues to listen to the incoming RTP packets. Usually, the remote party plays, in this scenario, Music on Hold (MOH) and the device forwards the MOH to the held party.

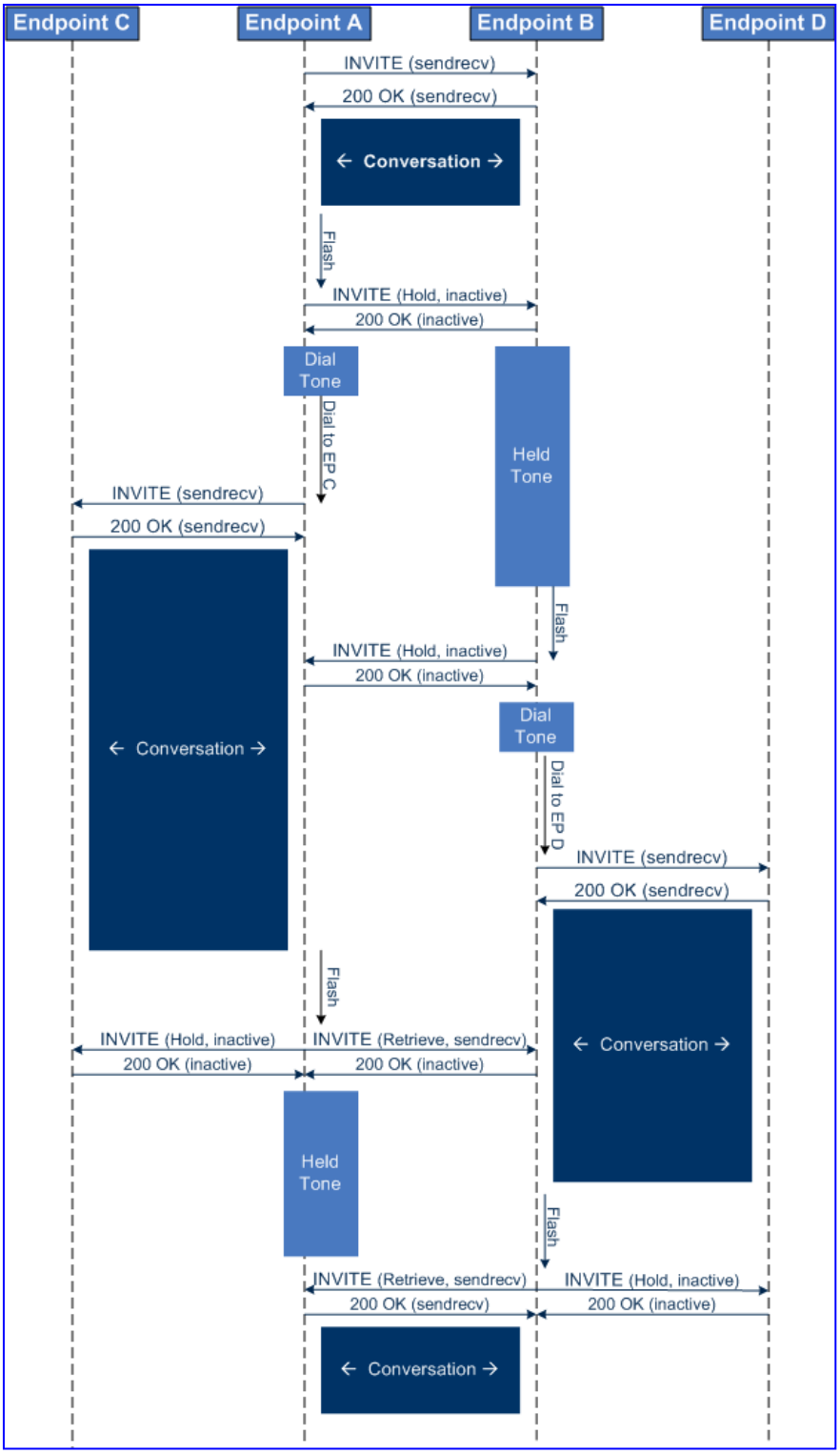
Receiving Hold/Retrieve:

- When an active call receives a Re-INVITE message with either the IP address 0.0.0.0 or the 'inactive' string in SDP, the device stops sending RTP and plays a local Held tone.
- When an active call receives a Re-INVITE message with the 'sendonly' string in SDP, the device stops sending RTP and listens to the remote party. In this mode, it is expected that on-hold music (or any other hold tone) is played (over IP) by the remote party.

You can also configure the device to keep a call on-hold for a user-defined time after which the call is disconnected, using the HeldTimeout parameter.

The device also supports "double call hold" for FXS interfaces where the called party, which has been placed on-hold by the calling party, can then place the calling party on hold as well and make a call to another destination. The flowchart below provides an example of this type of call hold:

Figure 19-23: Double Hold SIP Call Flow



The flowchart above describes the following "double" call-hold scenario:

1. A calls B and establishes a voice path.
2. A places B on hold; A hears a Dial tone and B hears a Held tone.
3. A calls C and establishes a voice path.
4. B places A on hold; B hears a Dial tone.
5. B calls D and establishes a voice path.
6. A ends call with C; A hears a Held tone.
7. B ends call with D.
8. B retrieves call with A.

Notes:

- If a party that is placed on hold (e.g., B in the above example) is called by another party (e.g., D), then the on-hold party receives a Call Waiting tone instead of the Held tone.
- While in a Double Hold state, placing the phone on-hook disconnects both calls (i.e. call transfer is not performed).

19.6.2 BRI Suspend and Resume

The device supports call suspend and resume services initiated by ISDN BRI phones connected to the device. During an ongoing call, the BRI phone user can suspend the call by typically pressing the phone's "P" button or a sequence of keys (depending on the phone), and then on-hooking the handset. To resume the call, the phone user typically presses the same keys or button again and then off-hooks the phone. During the suspended state, the device plays a Howler tone to the remote party. This service is also supported when instead of pressing the call park button(s), the phone cable is disconnected (suspending the call) and then reconnected again (resuming the call).

If the phone user does not resume the call within a user-defined interval (configured using the HeldTimeout parameter), the device releases the call.

Note: Only one call can be suspended per trunk. If another suspend request is received from a BRI phone while there is already a suspended call (even if done by another BRI phone connected to the same trunk), the device rejects this suspend request.

19.6.3 Consultation Feature

The device's Consultation feature allows you to place one number on hold and make a second call to another party.

- After holding a call (by pressing hook-flash), the holding party hears a dial tone and can then initiate a new call, which is called a Consultation call.
- While hearing a dial tone, or when dialing to the new destination (before dialing is complete), the user can retrieve the held call by pressing hook-flash.

- The held call can't be retrieved while Ringback tone is heard.
- After the Consultation call is connected, the user can toggle between the held and active call by pressing the hook-flash key.

Note: The Consultation feature is applicable only to FXS interfaces.

19.6.4 Call Transfer

The device supports the following call transfer types:

- Consultation Transfer (see 'Consultation Call Transfer' on page 315)
- Blind Transfer (see 'Blind Call Transfer' on page 316)

Notes:

- Call transfer is initiated by sending REFER with REPLACES.
- The device can receive and act upon receiving REFER with or without REPLACES.
- The device can receive and act upon receiving INVITE with REPLACES, in which case the old call is replaced by the new one.
- The INVITE with REPLACES can be used to implement Directed Call Pickup.

19.6.4.1 Consultation Call Transfer

The device supports Consultation Call Transfer (using the SIP REFER message and Replaces header). The common method to perform a consultation transfer is described in the following example, which assumes three call parties:

- Party A = transferring
- Party B = transferred
- Party C = transferred to

1. A Calls B.
2. B answers.
3. A presses the hook-flash button and places B on-hold (party B hears a hold tone).
4. A dials C.
5. After A completes dialing C, A can perform the transfer by on-hooking the A phone.
6. After the transfer is complete, B and C parties are engaged in a call.

   The transfer can be initiated at any of the following stages of the call between A and C:

   - Just after completing dialing C phone number - transfer from setup.
   - While hearing Ringback – transfer from alert.
   - While speaking to C - transfer from active.

The device also supports attended (consultation) call transfer for BRI phones (user side) connected to the device and using the Euro ISDN protocol. BRI call transfer is according to

ETSI TS 183 036, Section G.2 (Explicit Communication Transfer – ECT). Call transfer is enabled using the EnableTransfer and EnableHoldtoISDN parameters.

The Explicit Call Transfer (ECT, according to ETS-300-367, 368, 369) supplementary service is supported for BRI and PRI trunks. This service provides the served user who has two calls to ask the network to connect these two calls together and release its connection to both parties. The two calls can be incoming or outgoing calls. This service is similar to NI2 Two B-Channel Transfer (TBCT) Supplementary Service. The main difference is that in ECT one of the calls must be in HELD state. The ECT standard defines two methods - Implicit and Explicit. In implicit method, the two calls must be on the same trunk. BRI uses the implicit mechanism, and PRI the explicit mechanism.

19.6.4.2 Consultation Transfer for QSIG Path Replacement

The device can interwork consultation call transfer requests for ISDN QSIG-to-IP calls. When the device receives a request for a consultation call transfer from the PBX, the device sends a SIP REFER message with a Replaces header to the SIP UA to transfer it to another SIP UA. Once the two SIP UA parties are successfully connected, the device requests the PBX to disconnect the ISDN call, thereby freeing resources on the PBX.

For example, assume legacy PBX user "A" has two established calls connected through the device – one with remote SIP UA "B" and the other with SIP UA "C". In this scenario, user "A" initiates a consultation call transfer to connect "B" with "C". The device receives the consultation call transfer request from the PBX and then connects "B" with "C", by sending "B" a REFER message with a Replaces header (i.e., replace caller "A" with "C"). Upon receipt of a SIP NOTIFY 200 message in response to the REFER, the device sends a Q.931 DISCONNECT messages to the PBX, notifying the PBX that it can disconnect the ISDN calls (of user "A").

This feature is enabled by the QSIGPathReplacementMode parameter.

19.6.4.3 Blind Call Transfer

Blind call transfer is done (using SIP REFER messages) after a call is established between call parties A and B, and party A decides to immediately transfer the call to C without first speaking to C. The result of the transfer is a call between B and C (similar to consultation transfer, but skipping the consultation stage).

You can also use the ManipulateIP2PSTNReferTo parameter to manipulate the destination number according to the number received in the SIP Refer-To header. This is applicable to all types of blind transfers to the PSTN (e.g., TBCT, ECT, RLT, QSIG, FXO, and CAS). During blind transfer, the device initiates a new call to the PSTN and the destination number of this call can be manipulated if this parameter is enabled. The following is an example of such a blind transfer:

1. IP phone "A" calls PSTN phone "B", and the call is established.
2. "A" performs a blind transfer to PSTN phone "C". It does this as follows:
   a. "A" sends a SIP REFER message (with the phone number of "C" in the Refer-To header) to the device.
   b. The device sends a Q.931 Setup message to "C". This feature enables manipulating the called party number in this outgoing Setup message.

The manipulation is done as follows:

1. If you configure a value for the xferPrefix parameter, then this value (string) is added as a prefix to the number in the Refer-To header.
2. This called party number is then manipulated using the IP-to-Tel Destination Phone Number Manipulation table.
3. The source number of the transferred call is taken from the original call, according to its initial direction:
   - Tel-to-IP call: source number of the original call.

- IP-to-Tel call: destination number of the original call.
- If the UseReferredByForCallingNumber parameter is set to 1, the source number is taken from the SIP Referred-By header if included in the received SIP REFER message.

This source number can also be used as the value for the 'Source Prefix' field in the IP-to-Tel Destination Phone Number Manipulation table. The local IP address is used as the value for the 'Source IP Address' field.

Notes:

- Manipulation using the ManipulateIP2PSTNReferTo parameter does not affect IP-to-Trunk Group routing rules.
- Currently, the device does not support blind transfer for BRI interfaces.

19.6.5 Call Forward

For digital interfaces: The device supports Call Deflection (ETS-300-207-1) for Euro ISDN and QSIG (ETSI TS 102 393) for Network and User sides, which provides IP-ISDN interworking of call forwarding (call diversion) when the device receives a SIP 302 response.

Call forward performed by the SIP side: Upon receipt of a Facility message with Call Rerouting IE from the PSTN, the device initiates a SIP transfer process by sending a SIP 302 (including the Call Rerouting destination number) to the IP in response to the remote SIP entity's INVITE message. The device then responds with a Disconnect message to the PSTN side.

Call forward performed by the PSTN side: When the device sends the INVITE message to the remote SIP entity and receives a SIP 302 response, the device sends a Facility message with the same IE mentioned above to the PSTN, and waits for the PSTN side to disconnect the call. This is configured using the CallReroutingMode.

For analog interfaces: The following methods of call forwarding are supported:

- **Immediate:** incoming call is forwarded immediately and unconditionally.
- **Busy:** incoming call is forwarded if the endpoint is busy.
- **No Reply:** incoming call is forwarded if it isn't answered for a specified time.
- **On Busy or No Reply:** incoming call is forwarded if the port is busy or when calls are not answered after a specified time.
- **Do Not Disturb:** immediately reject incoming calls. Upon receiving a call for a Do Not Disturb, the 603 Decline SIP response code is sent.

Three forms of forwarding parties are available:

- **Served party:** party configured to forward the call (FXS device).
- **Originating party:** party that initiates the first call (FXS or FXO device).
- **Diverted party:** new destination of the forwarded call (FXS or FXO device).

The served party (FXS interface) can be configured through the Web interface (see 'Configuring Call Forward' on page 358) or *ini* file to activate one of the call forward modes. These modes are configurable per endpoint.

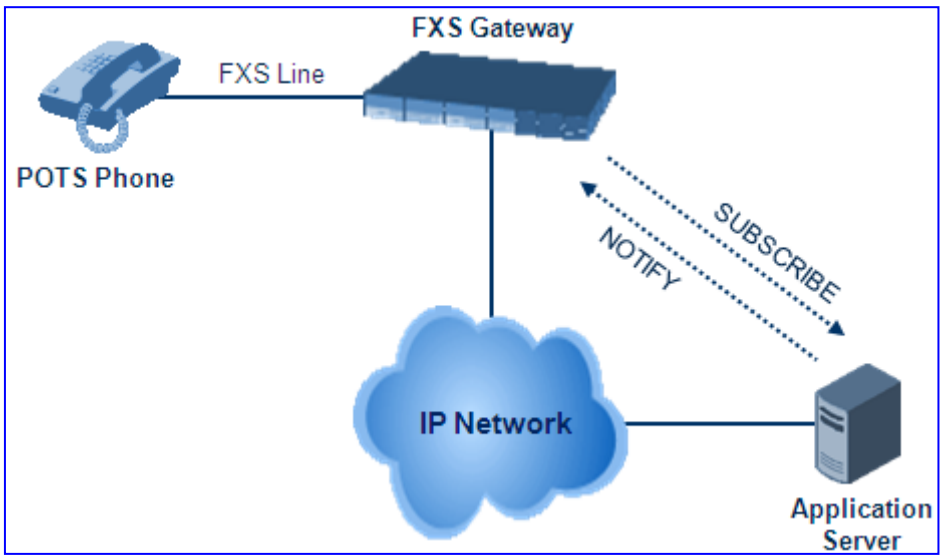
Notes:

- When call forward is initiated, the device sends a SIP 302 response with a contact that contains the phone number from the forward table and its corresponding IP address from the routing table (or when a proxy is used, the proxy's IP address).
- For receiving call forward, the device handles SIP 3xx responses for redirecting calls with a new contact.

19.6.5.1 Call Forward Reminder Ring

The device supports the Call Forward Reminder Ring feature for FXS interfaces, whereby the device's FXS endpoint emits a short ring burst (only if in **onhook** state) when a third-party Application Server (e.g., softswitch) forwards an incoming call to another destination. This is important in that it notifies (audibly) the FXS endpoint user that a call forwarding service is currently being performed.

Figure 19-24: Call Forward Reminder Ring



The device generates a Call Forward Reminder ring burst to the FXS endpoint each time it receives a SIP NOTIFY message with a "reminder ring" xml body. The NOTIFY request is sent from the Application Server to the device each time the Application Server forwards an incoming call. The service is cancelled when an UNSUBSCRIBE request is sent from the device, or when the Subscription time expires.

The Reminder Ring tone can be defined by using the parameter CallForwardRingToneID, which points to a ring tone defined in the Call Progress Tone file.

The following parameters are used to configure this feature:

- EnableNRTSubscription
- ASSubscribeIPGroupID
- NRTSubscribeRetryTime
- CallForwardRingToneID

19.6.5.2 Call Forward Reminder (Off-Hook) Special Dial Tone

The device plays a special dial tone (Stutter Dial tone - Tone Type #15) to a specific FXS endpoint when the phone is off-hooked and when a third-party Application server (AS), e.g., a softswitch is used to forward calls intended for the endpoint, to another destination. This is useful in that it reminds the FXS user of this service. This feature does not involve device subscription (SIP SUBSCRIBE) to the AS.

Activation/deactivation of the service is notified by the server. An unsolicited SIP NOTIFY request is sent from the AS to the device when the Call Forward service is activated or cancelled. Depending on this NOTIFY request, the device plays either the standard dial tone or the special dial tone for Call Forward.

For playing the special dial tone, the received SIP NOTIFY message must contain the following headers:

- **From and To:** contain the same information, indicating the specific endpoint
- **Event:** ua-profile
- **Content-Type:** "application/simservs+xml"
- Message body is the XML body and contains the “dial-tone-pattern” set to "special-condition-tone" (<ss:dial-tone-pattern>special-condition-tone</ss:dial-tone-pattern>), which is the special tone indication.

For cancelling the special dial tone and playing the regular dial tone, the received SIP NOTIFY message must contain the following headers:

- **From and To:** contain the same information, indicating the specific endpoint
- **Event:** ua-profile
- **Content-Type:** "application/simservs+xml"
- Message body is the XML body containing the “dial-tone-pattern” set to "standard-condition-tone" (<ss:dial-tone-pattern>standard-condition-tone</ss:dial-tone-pattern>), which is the regular dial tone indication.

Therefore, the special dial tone is valid until another SIP NOTIFY is received that instructs otherwise (as described above).

Note: if the MWI service is active, the MWI dial tone overrides this special Call Forward dial tone

19.6.5.3 BRI Call Forwarding

The device supports call forwarding (CF) services initiated by ISDN Basic BRI phones connected to it. Upon receipt of an ISDN Facility message for call forward from the BRI phone, the device sends a SIP INVITE to the softswitch with a user-defined code in the SIP To header, representing the reason for the call forward.

The codes for the call forward can be defined using the following parameters:

- SuppServCodeCFU - Call Forward Unconditional
- SuppServCodeCFUDeact - Call Forward Unconditional Deactivation
- SuppServCodeCFB - Call Forward on Busy
- SuppServCodeCFBDeact - Call Forward on Busy Deactivation

- SuppServCodeCFNR - Call Forward on No Reply
- SuppServCodeCFNRDeact - Call Forward on No Reply

Note: These codes must be defined according to the settings of the softswitch (i.e., the softswitch must recognize them).

Below is an example of an INVITE message sent by the device indicating an unconditional call forward ("*72") to extension number 100. This code is defined using the SuppServCodeCFU parameter.

```
INVITE sip:*72100@10.33.8.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.5:5060;branch=z9hG4bKWDSUKUHWFEXQSVOUVJGM
From: <sip:400@10.33.2.5;user=phone>;tag=DUOROSXSOYJJLNBFRQTG
To: <sip:*72100@10.33.8.53;user=phone>
Call-ID: GMNOVQRRXUUCYCQSFAHS@10.33.2.5
CSeq: 1 INVITE
Contact: <sip:400@10.33.2.5:5060>
Supported: em,100rel,timer,replaces
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE
User-Agent: Sip Message Generator V1.0.0.5
User-to-User: 31323334;pd=4
Content-Type: application/sdp
Content-Length: 155
```

19.6.6 Call Waiting

The Call Waiting feature enables FXS devices to accept an additional (second) call on busy endpoints. If an incoming IP call is designated to a busy port, the called party hears a call waiting tone (several configurable short beeps) and (for Bellcore and ETSI Caller IDs) can view the Caller ID string of the incoming call. The calling party hears a Call Waiting Ringback Tone. The called party can accept the new call using hook-flash, and can toggle between the two calls.

➢ **To enable call waiting:**

1. Set the parameter EnableCallWaiting to 1.
2. Set the parameter EnableHold to 1.
3. Define the Call Waiting indication and Call Waiting Ringback tones in the Call Progress Tones file. You can define up to four Call Waiting indication tones (refer to the FirstCallWaitingToneID parameter).
4. To configure the Call Waiting indication tone cadence, modify the following parameters: NumberOfWaitingIndications, WaitingBeepDuration and TimeBetweenWaitingIndications.
5. To configure a delay interval before a Call Waiting Indication is played to the currently busy port, use the parameter TimeBeforeWaitingIndication. This enables the caller to hang up before disturbing the called party with Call Waiting Indications. Applicable only to FXS modules.

Both the calling and called sides are supported by FXS interfaces; FXO interfaces support only the calling side.

To indicate Call Waiting, the device sends a 182 Call Queued response. The device identifies Call Waiting when a 182 Call Queued response is received.

Note: The Call Waiting feature is applicable only to FXS/FXO interfaces.

19.6.7 Message Waiting Indication

The device supports Message Waiting Indication (MWI) according to IETF RFC 3842, including SUBSCRIBE (to an MWI server).

Note: For more information on IP voice mail configuration, refer to the *IP Voice Mail CPE Configuration Guide*.

For analog interfaces: The FXS device can accept an MWI NOTIFY message that indicates waiting messages or that the MWI is cleared. Users are informed of these messages by a stutter dial tone. The stutter and confirmation tones are defined in the CPT file (refer to the Product Reference Manual). If the MWI display is configured, the number of waiting messages is also displayed. If the MWI lamp is configured, the phone's lamp (on a phone that is equipped with an MWI lamp) is lit. The device can subscribe to the MWI server per port (usually used on FXS) or per device (used on FXO).

To configure MWI, use the following parameters:

- EnableMWI
- MWIServerIP, or MWISubscribeIPGroupID and ProxySet
- MWIAnalogLamp
- MWIDisplay
- StutterToneDuration
- EnableMWISubscription
- MWIExpirationTime
- SubscribeRetryTime
- SubscriptionMode
- CallerIDType (determines the standard for detection of MWI signals)
- ETSIVMWITypeOneStandard
- BellcoreVMWITypeOneStandard
- VoiceMailInterface
- EnableVMURI

The device supports the following MWI features for its digital PSTN interfaces:

- For BRI interfaces: This feature provides support for MWI on BRI phones connected to the device and using the Euro ISDN BRI variant. When this feature is activated and a voice mail message is recorded to the mail box of a BRI extension, the softswitch sends a notification to the device. In turn, the device notifies the BRI extension and a red light flashes on the BRI extension's phone. Once the voice message is retrieved, the MWI light on the BRI extension turns off. This feature is configured by setting the VoiceMailInterface parameter to 8 ("ETSI") and enabled by the EnableMWI parameter.
- Euro-ISDN MWI: The device supports Euro-ISDN MWI for IP-to-Tel calls. The device interworks SIP MWI NOTIFY messages to Euro-ISDN Facility information element (IE) MWI messages. This is supported by setting the VoiceMailInterface parameter to 8.

- QSIG MWI: The device also supports the interworking of QSIG MWI to IP (in addition to interworking of SIP MWI NOTIFY to QSIG Facility MWI messages). This provides interworking between an ISDN PBX with voicemail capabilities and a softswitch, which requires information on the number of messages waiting for a specific user. This support is configured using the MWIInterrogationType parameter, which determines the device's handling of MWI Interrogation messages. The process for sending the MWI status upon request from a softswitch is as follows:
  1. The softswitch sends a SIP SUBSCRIBE message to the device.
  2. The device responds by sending an empty SIP NOTIFY to the softswitch, and then sending an ISDN Setup message with Facility IE containing an MWI Interrogation request to the PBX.
  3. The PBX responds by sending to the device an ISDN Connect message containing Facility IE with an MWI Interrogation result, which includes the number of voice messages waiting for the specific user.
  4. The device sends another SIP NOTIFY to the softswitch, containing this MWI information.
  5. The SIP NOTIFY messages are sent to the IP Group defined by the NotificationIPGroupID parameter.

  In addition, when a change in the status occurs (e.g., a new voice message is waiting or the user has retrieved a message from the voice mail), the PBX initiates an ISDN Setup message with Facility IE containing an MWI Activate request, which includes the new number of voice messages waiting for the user. The device forwards this information to the softswitch by sending a SIP NOTIFY.

  Depending on the PBX support, the MWIInterrogationType parameter can be configured to handle these MWI Interrogation messages in different ways. For example, some PBXs support only the MWI Activate request (and not MWI Interrogation request). Some support both these requests. Therefore, the device can be configured to disable this feature, or enable it with one of the following support:
  - Responds to MWI Activate requests from the PBX by sending SIP NOTIFY MWI messages (i.e., does not send MWI Interrogation messages).
  - Send MWI Interrogation message, but don't use its result. Instead, wait for MWI Activate requests from the PBX.
  - Send MWI Interrogation message, use its result, and use the MWI Activate requests.

19.6.8 Caller ID

This section discusses the device's Caller ID support.

Note: The Caller ID feature is applicable only to FXS/FXO interfaces.

19.6.8.1 Caller ID Detection / Generation on the Tel Side

By default, generation and detection of Caller ID to the Tel side is disabled. To enable Caller ID, set the parameter EnableCallerID to 1. When the Caller ID service is enabled:

- For FXS: the Caller ID signal is sent to the device's port
- For FXO: the Caller ID signal is detected

The configuration for Caller ID is described below:

- Use the parameter CallerIDType to define the Caller ID standard. Note that the Caller ID standard that is used on the PBX or phone must match the standard defined in the

device.

- Select the Bellcore caller ID sub standard using the parameter BellcoreCallerIDTypeOneSubStandard
- Select the ETSI FSK caller ID sub standard using the parameter ETSICallerIDTypeOneSubStandard
- Enable or disable (per port) the caller ID generation (for FXS) and detection (for FXO) using the 'Generate / Detect Caller ID to Tel' table (EnableCallerID). If a port isn't configured, its caller ID generation / detection are determined according to the global parameter EnableCallerID.
- EnableCallerIDTypeTwo: disables / enables the generation of Caller ID type 2 when the phone is off-hooked (used for call waiting).
- RingsBeforeCallerID: sets the number of rings before the device starts detection of caller ID (FXO only). By default, the device detects the caller ID signal between the first and second rings.
- AnalogCallerIDTimimgMode: determines the time period when a caller ID signal is generated (FXS only). By default, the caller ID is generated between the first two rings.
- PolarityReversalType: some Caller ID signals use reversal polarity and/or wink signals. In these scenarios, it is recommended to set PolarityReversalType to 1 (Hard) (FXS only).
- The Caller ID interworking can be changed using the parameters UseSourceNumberAsDisplayName and UseDisplayNameAsSourceNumber.

19.6.8.2 Debugging a Caller ID Detection on FXO

The procedure below describes debugging caller ID detection in FXO interfaces.

➢ **To debug a Caller ID detection on an FXO interface:**

1. Verify that the parameter EnableCallerID is set to 1.
2. Verify that the caller ID standard (and substandard) of the device matches the standard of the PBX (using the parameters CallerIDType, BellcoreCallerIDTypeOneSubStandard, and ETSICallerIDTypeOneSubStandard).
3. Define the number of rings before the device starts the detection of caller ID (using the parameter RingsBeforeCallerID).
4. Verify that the correct FXO coefficient type is selected (using the parameter CountryCoefficients), as the device is unable to recognize caller ID signals that are distorted.
5. Connect a phone to the analog line of the PBX (instead of to the device's FXO interface) and verify that it displays the caller ID.

If the above does not solve the problem, you need to record the caller ID signal (and send it to AudioCodes), as described below.

➢ **To record the caller ID signal using the debug recording mechanism:**

1. Access the FAE page (by appending "FAE" to the device's IP address in the Web browser's URL, for example, http://10.13.4.13/FAE).
2. Press the **Cmd Shell** link.
3. Enter the following commands:

```
dr
ait <IP address of PC to collect the debug traces sent from
the device>
```

```
AddChannelIdTrace ALL-WITH-PCM <port number, which starts from
0>
Start
```

4. Make a call to the FXO.
5. To stop the DR recording, at the CLI prompt, type **STOP**.

19.6.8.3 Caller ID on the IP Side

Caller ID is provided by the SIP From header containing the caller's name and "number", for example:

```
From: “David” <SIP:101@10.33.2.2>;tag=35dfsgasd45dg
```

If Caller ID is restricted (received from Tel or configured in the device), the From header is set to:

```
From: “anonymous” <anonymous@anonymous.invalid>; tag=35dfsgasd45dg
```

The P-Asserted (or P-Preferred) headers are used to present the originating party's caller ID even when the caller ID is restricted. These headers are used together with the Privacy header.

- If Caller ID is restricted:
  - The From header is set to “anonymous” <anonymous@anonymous.invalid>
  - The ‘Privacy: id’ header is included
  - The P-Asserted-Identity (or P-Preferred-Identity) header shows the caller ID
- If Caller ID is allowed:
  - The From header shows the caller ID
  - The ‘Privacy: none’ header is included
  - The P-Asserted-Identity (or P-Preferred-Identity) header shows the caller ID

In addition, the caller ID (and presentation) can be displayed in the Calling Remote-Party-ID header.

The ‘Caller Display Information’ table (CallerDisplayInfo) is used for the following:

- **FXS interfaces** - to define the caller ID (per port) that is sent to IP.
- **FXO interfaces** - to define the caller ID (per port) that is sent to IP if caller ID isn't detected on the Tel side, or when EnableCallerID = 0.
- **FXS and FXO interfaces** - to determine the presentation of the caller ID (allowed or restricted).
- **To maintain backward compatibility** - when the strings ‘Private’ or ‘Anonymous’ are set in the Caller ID/Name field, the caller ID is restricted and the value in the Presentation field is ignored.

The value of the ‘Presentation’ field that is defined in the ‘Caller Display Information’ table can be overridden by configuring the ‘Presentation’ parameter in the ‘Tel to IP Source Number Manipulation’ table. Therefore, this table can be used to set the presentation for specific calls according to Source / Destination prefixes.

The caller ID can be restricted/allowed (per port) using keypad features KeyCLIR and KeyCLIRDeact (FXS only).

AssertedIdMode defines the header that is used (in the generated INVITE request) to deliver the caller ID (P-Asserted-Identity or P-preferred-Identity). Use the parameter UseTelURIForAssertedID to determine the format of the URI in these headers (sip: or tel:).

The parameter EnableRPIheader enables Remote-Party-ID (RPI) headers for calling and called numbers for Tel-to-IP calls.

19.6.9 Three-Way Conferencing

The device supports three-way conference calls. These conference calls can also occur simultaneously. The device supports the following conference modes (configured by the parameter 3WayConferenceMode):

- **Conferencing controlled by an external AudioCodes Conference (media) server:** The Conference-initiating INVITE sent by the device uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. For this mode, the 3WayConferenceMode parameter is set to 0 (default.)
- **Conferencing controlled by an external, third-party Conference (media) server:** The Conference-initiating INVITE sent by the device uses only the ConferenceID as the Request-URI. The Conference server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the Conference server using this conference URI. For this mode, the 3WayConferenceMode parameter is set to 1.
- Local, on-board conferencing, whereby the conference is established on the device without the need for an external Conference server. This feature includes local mixing and transcoding of the 3-Way Call legs on the device, and even allowing multi-codec conference calls. The device sets up the call conference using its IP media channels. These channels are obtained from the IP media module (i.e., MPM module). Note that the MPM module(s) must be installed to support three-way conferencing. The device supports up to five simultaneous, on-board, three-way conference calls. For this mode, the 3WayConferenceMode parameter is set to 2.

Notes:

- Three-way conferencing using an external conference server is supported only by FXS interfaces.
- Instead of using the flash-hook button to establish a three-way conference call, you can dial a user-defined hook-flash code (e.g., "*1"), configured by the HookFlashCode parameter.
- Three-way conferencing is applicable only to FXS and BRI interfaces. Three-way conferencing support for the BRI phones connected to the device complies with ETS 300 185.

The following example demonstrates three-way conferencing. This example assumes that a telephone "A" connected to the device wants to establish a three-way conference call with two remote IP phones "B" and "C":

1. User A has an ongoing call with IP phone B.
2. User A places IP phone B on hold (by pressing the telephone's flash hook button, defined by the parameter HookFlashCode).
3. User A hears a dial tone, and then makes a call to IP phone C.
4. IP phone C answers the call.
5. User A can now establish a three-way conference call (between A, B and C) by pressing the flash-hook button, defined by the parameter ConferenceCode (e.g.,

regular flash-hook button or "*1").

To configure three-way conferencing, use the following parameters:

- Enable3WayConference
- ConferenceCode = '!' (default, which is the hook flash button)
- HookFlashCode
- 3WayConferenceMode (conference mode)
- FlashKeysSequenceStyle = 1 or 2 (makes a three-way call conference using the Flash button + 3)

19.6.10 Emergency E911 Phone Number Services

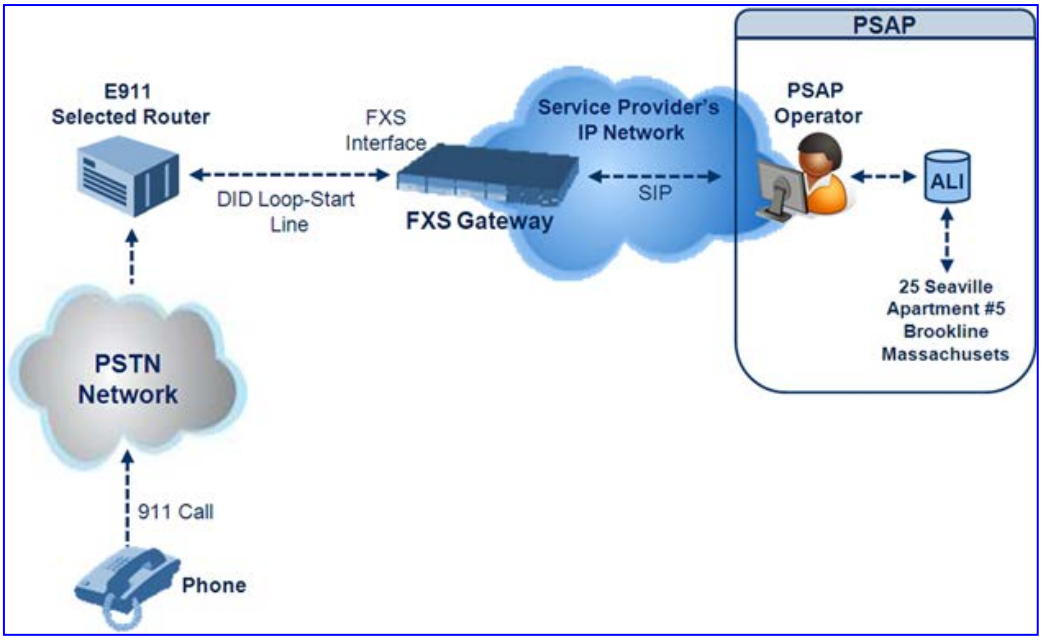
The device supports emergency phone number services. The device supports the North American emergency telephone number system known as Enhanced 911 (E911), according to the TR-TSY-000350 and Bellcore's GR-350-Jun2003 standards. The E911 emergency system automatically associates a physical address with the calling party's telephone number, and routes the call to the most appropriate (closest) Public Safety Answering Point (PSAP), allowing the PSAP to quickly dispatch emergency response (e.g., police) to the caller's location.

Typically, the dialed emergency number is routed to the appropriate PSAP by the telephone company's switch, known as a 911 Selective Router (or E911 tandem switch). If the PSAP receives calls from the telephone company on old-style digital trunks, they are specially formatted Multi-Frequency (MF) trunks that pass only the calling party's number (known as Automatic Number Identification - ANI). Once the PSAP receives the call, it searches for the physical address that is associated with the calling party's telephone number (in the Automatic Location Identification database - ALI).

19.6.10.1 FXS Device Emulating PSAP using DID Loop-Start Lines

The FXS device can be configured to emulate PSAP (using DID loop start lines), according to the Telcordia GR-350-CORE specification.

Figure 19-25: FXS Device Emulating PSAP using DID Loop-Start Lines



The call flow of an E911 call to the PSAP is as follows:

1. The E911 tandem switch seizes the line.
2. The FXS device detects the line seize, and then generates a wink signal (nominal 250 msec). The wink can be delayed by configuring the parameter DelayBeforeDIDWink to 200 (for 200 msec or a higher value).
3. The switch detects the wink and then sends the MF Spill digits with ANI and (optional) Pseudo-ANI (P ANI).
4. The FXS device collects the MF digits, and then sends a SIP INVITE message to the PSAP with all collected MF digits in the SIP From header as one string.
5. The FXS device generates a mid-call wink signal (two subsequent polarity reversals) toward the E911 tandem switch upon either detection of an RFC 2833 "hookflash" telephony event, or if a SIP INFO message with a "hooflash" body is received from the PSAP (see the example below). The duration of this "flashhook" wink signal is configured using the parameter FlashHookPeriod (usually 500 msec). Usually the wink signal is followed by DTMF digits sent by PSAP to perform call transfer. Another way to perform the call transfer is to use SIP REFER messages, as described below.
6. The FXS device supports call transfer initiated by the PSAP. If it receives a SIP REFER message with the Refer-To URI host part containing an IP address that is equal to the device's IP address, the FXS device generates a 500-msec wink signal (double polarity reversals), and then (after a user-defined interval configured by the parameter WaitForDialTime), plays DTMF digits according to the transfer number received in the SIP Refer-To header URI userpart.
7. When the call is answered by the PSAP operator, the PSAP sends a SIP 200 OK to the FXS device, and the FXS device then generates a polarity reversal signal to the E911 switch.
8. After the call is disconnected by the PSAP, the PSAP sends a SIP BYE to the FXS device, and the FXS device reverses the polarity of the line toward the tandem switch.

The following parameters need to be configured:

- EnableDIDWink = 1
- EnableReversalPolarity = 1
- PolarityReversalType = 1
- FlashHookPeriod = 500 (for 500 msec "hookflash" mid-call Wink)
- WinkTime = 250 (for 250 msec signalling Wink generated by the FXS device after it detects the line seizure)
- EnableTransfer = 1 (for call transfer)
- LineTransferMode = 1 (for call transfer)
- WaitforDialTime = 1000 (for call transfer)
- SwapTEl2IPCalled&CallingNumbers = 1
- DTMFDetectorEnable = 0
- MFR1DetectorEnable = 1
- DelayBeforeDIDWink = 200 (for 200 msec) - can be configured in the range from 0 (default) to 1000.

Note: Modification of the WinkTime and FlashHookPeriod parameters require a device reset.

The outgoing SIP INVITE message contains the following headers:

```
INVITE sip:Line@DomainName
From: <sip:*81977820#@sipgw>;tag=1c143
To: <sip:Line@DomainName>
```

Where:

- Line = as configured in the Endpoint Phone Number Table.
- SipGtw = configured using the SIPGatewayName parameter.
- From header/user part = calling party number as received from the MF spill.

The ANI and the pseudo-ANI numbers are sent to the PSAP either in the From and/or P-AssertedID SIP header.

Typically, the MF spills are sent from the E911 tandem switch to the PSAP, as shown in the table below:

Table 19-17: Dialed MF Digits Sent to PSAP

Digits of Calling Number	Dialed MF Digits
8 digits "nnnnnnnn" (ANI)	"KPnnnnnnnnST"
12 digits "nnnnnnnnnnnn" (ANI)	"KPnnnnnnnnnnnnSTP"
12 digits ANI and 10 digits PANI	"KPnnnnnnnnnnnnSTKPmmmmmmmmmmST"
two digits "nn"	"KPnnSTP"

The MF KP, ST, and STP digits are mapped as follows:

- * for KP
- # for ST
- B for STP

For example, if ANI and PANI are received, the SIP INVITE contains the following From header:

```
From: <sip:*nnnnnnnnnnnn#*mmmmmmmmmm#@10.2.3.4>;tag=1c14
```

Note: It is possible to remove the * and # characters, using the device's number manipulation rules.

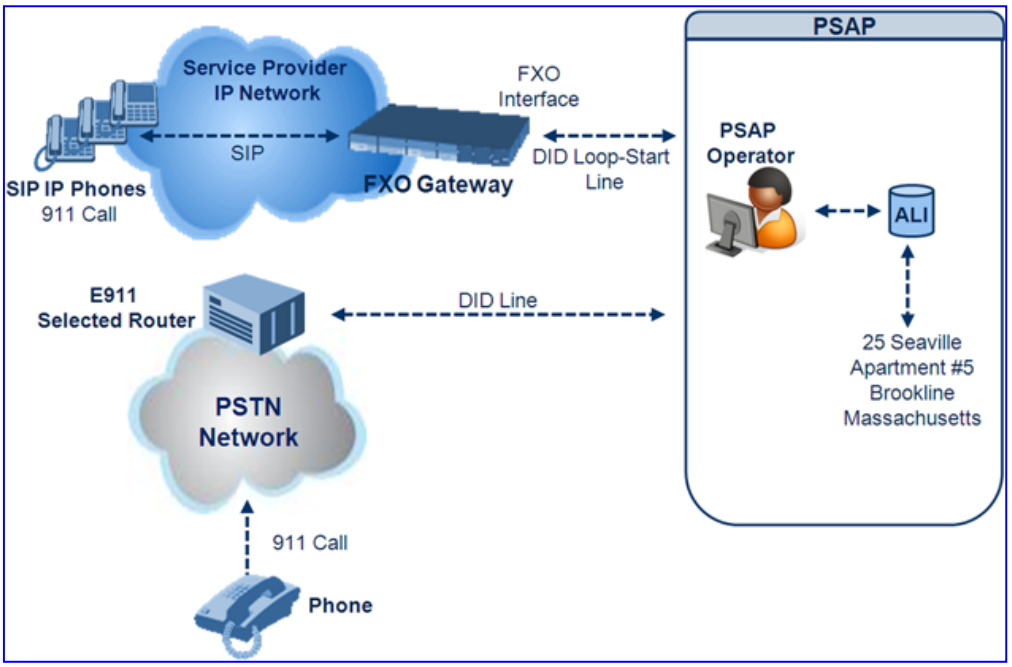
If the device receives the SIP INFO message below, it then generates a "hookflash" mid-call Wink signal:

```
INFO sip:4505656002@192.168.13.40:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.13.2:5060
From: port1vega1 <sip:06@192.168.13.2:5060>
To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-
1040067870294
Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2
CSeq:2 INFO
Content-Type: application/broadsoft
Content-Length: 17
event flashhook
```

19.6.10.2 FXO Device Interworking SIP E911 Calls from Service Provider's IP Network to PSAP DID Lines

The FXO device can interwork SIP emergency E911 calls from the Service Provider's IP network to the analog PSAP DID lines. The standards that define this interface include TR-TSY-000350 or Bellcore's GR-350-Jun2003. This protocol defines signaling between the E911 tandem switch (E911 Selective Router) and the PSAP, using analog loop-start lines. The FXO device can be implemented instead of an E911 switch, by connecting directly to the PSAP DID loop-start lines.

Figure 19-26: FXO Device Interfacing between E911 Switch and PSAP



When an IP phone subscriber dials 911, the device receives the SIP INVITE message and makes a call to the PSAP as follows:

1. The FXO device seizes the line.
2. PSAP sends a Wink signal (250 msec) to the device.
3. Upon receipt of the Wink signal, the device dials MF digits after a user-defined time (WaitForDialTime) containing the caller's ID (ANI) obtained from the SIP headers From or P-Asserted-Identity.
4. When the PSAP operator answers the call, the PSAP sends a polarity reversal to the device, and the device then sends a SIP 200 OK to the IP side.
5. After the PSAP operator disconnects the call, the PSAP reverses the polarity of the line, causing the device to send a SIP BYE to the IP side.
6. If, during active call state, the device receives a Wink signal (typically of 500 msec) from the PSAP, the device generates a SIP INFO message that includes a "hookflash" body, or sends RFC 2833 hookflash Telephony event (according to the HookFlashOption parameter).
7. Following the "hookflash" Wink signal, the PSAP sends DTMF digits. These digits are detected by the device and forwarded to the IP, using RFC 2833 telephony events (or inband, depending on the device's configuration). Typically, this Wink signal followed by the DTMF digits initiates a call transfer.

For supporting the E911 service, used the following configuration parameter settings:

- Enable911PSAP = 1 (also forces the EnableDIDWink and EnableReversalPolarity)

- HookFlashOption = 1 (generates the SIP INFO hookflash message) or 4 for RFC 2833 telephony event
- WinkTime = 700 (defines detection window of 50 to 750 msec for detection of both winks - 250 msec wink sent by the PSAP for starting the device's dialing; 500 msec wink during the call)
- IsTwoStageDial = 0
- EnableHold = 0
- EnableTransfer = 0
  - Use RFC 2833 DTMF relay:
    - RxDTMFOption = 3
    - TxDTMFOption = 4
    - RFC2833PayloadType = 101
- TimeToSampleAnalogLineVoltage = 100
- WaitForDialTime = 1000 (default is 1 sec)
- SetDefaultLinePolarityState = 0 (you need to verify that the RJ-11 two-wire cable is connected without crossing, Tip to Tip, Ring to Ring. Typically, the Tip line is positive compared to the Ring line.)

Note: If the two-wire cable is crossed, the SetDefaultLinePolarityState parameter must be set to 1.

The device expects to receive the ANI number in the From and/or P-Asserted-Identity SIP header. If the pseudo-ANI number exists, it should be sent as the display name in these headers.

Table 19-18: Dialed Number by Device Depending on Calling Number

Digits of Calling Number (ANI)	Digits of Displayed Number	Number Dialed MF Digits
8 "nnnnnnnn"	-	MF dialed "KPnnnnnnnnST"
12 "nnnnnnnnnnnn"	None	"KPnnnnnnnnnnnnSTP"
12 "nnnnnnnnnnnn"	10 "mmmmmmmmmm" (pANI)	"KPnnnnnnnnnnnnSTKPmmmmmmmmmmST"
2 "nn"	None	"KPnnSTP"
1 "n"	-	MF dialed "KPnST" For example: "From: <sip:8>@xyz.com>" generates device MF spill of KP 8 ST

Table **Notes:**

- For all other cases, a SIP 484 response is sent.
- KP is for .
- ST is for #.
- STP is for B.

The MF duration of all digits, except for the KP digit is 60 msec. The MF duration of the KP digit is 120 msec. The gap duration is 60 msec between any two MF digits.

Notes:

- Manipulation rules can be configured for the calling (ANI) and called number (but not on the "display" string), for example, to strip 00 from the ANI "00INXXYYYY".
- The called number, received as userpart of the Request URI ("301" in the example below), can be used to route incoming SIP calls to FXO specific ports, using the TrunkGroup and PSTNPrefix parameters.
- When the PSAP party off-hooks and then immediately on-hooks (i.e., the device detects wink), the device releases the call sending SIP response "403 Forbidden" and the release reason 21 (i.e., call rejected) "Reason: Q.850 ;cause=21" is sent. Using the cause mapping parameter, it is possible to change the 403 to any other SIP reason, for example, to 603.
- Sometimes a wink signal sent immediately after the FXO device seizes the line is not detected. To overcome this problem, configure the parameter TimeToSampleAnalogLineVoltage to 100 (instead of 1000 msec, which is the default value). The wink is then detected only after this timeout + 50 msec (minimum 150 msec).

Below are two examples for a) INVITE messages and b) INFO messages generated by hook-flash.

- Example (a): INVITE message with ANI = 333333444444 and pseudo-ANI = 0123456789:

```
INVITE sip:301@10.33.37.79;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac771627168
Max-Forwards: 70
From: "0123456789"
<sip:333333444444@audiocodes.com>;tag=1c771623824
To: <sip:301@10.33.37.79;user=phone>
Call-ID: 77162335841200014153@10.33.37.78
CSeq: 1 INVITE
Contact: <sip:101@10.33.37.78>
Supported: em,100rel,timer,replaces,path
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-FXO/v.6.00A.020.077
Privacy: none
P-Asserted-Identity: "0123456789" <sip:3333344444@audiocodes.com>
Content-Type: application/sdp
Content-Length: 253
v=0
o=AudiocodesGW 771609035 771608915 IN IP4 10.33.37.78
s=Phone-Call
c=IN IP4 10.33.37.78
t=0 0
m=audio 4000 RTP/AVP 8 0 101
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
```

```
a=sendrecv
```

- Example (b): The detection of a Wink signal generates the following SIP INFO message:

```
INFO sip:4505656002@192.168.13.40:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.13.2:5060
From: port1vega1 <sip:06@192.168.13.2:5060>
To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-
1040067870294
Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2
CSeq:2 INFO
Content-Type: application/broadsoft
Content-Length: 17
event flashhook
```

19.6.10.3 Pre-empting Existing Calls for E911 IP-to-Tel Calls

If the device receives an E911 call from the IP network destined to the Tel, and there are unavailable channels (e.g., all busy), the device terminates one of the calls (arbitrary) and then sends the E911 call to that channel. The preemption is done only on a channel pertaining to the same Trunk Group for which the E911 call was initially destined and if the channel select mode (configured by the ChannelSelectMode parameter) is set to other than “By Dest Number” (0).

The preemption is done only if the incoming IP-to-Tel call is identified as an emergency call. The device identifies emergency calls by one of the following:

- The destination number of the IP call matches one of the numbers defined by the EmergencyNumbers parameter. (For E911, you must defined this parameter with the value "911".)
- The incoming SIP INVITE message contains the “emergency” value in the Priority header.

This feature is enabled by setting the CallPriorityMode parameter to “Emergency” (2).

Notes:

- This feature is applicable to FXS/FXO, CAS, and ISDN interfaces.
- For FXO interfaces, the preemption is done only on existing IP-to-Tel calls. In other words, if all the current FXO channels are busy with calls that were answered by the FXO device (i.e., Tel-to-IP calls), new incoming emergency IP-to-Tel calls are rejected.

19.6.11 Enhanced 9-1-1 Support for Microsoft Lync Server 2010

The Enhanced 9-1-1 (E9-1-1) service is becoming the mandatory emergency service required in many countries around the world. The E9-1-1 service, based on its predecessor 911, enables emergency operators to pinpoint the location (granular location) of callers who dial the 9-1-1 emergency telephone number.

Today, most enterprises implement an IP-based infrastructure providing a Voice-over-Internet (VoIP) network with fixed and nomadic users, allowing connectivity anywhere with any device. This, together with an often deployed multi-line telephone system (MLTS), poses a challenge for E9-1-1 due to the difficulty in accurately locating the E9-1-1 caller.

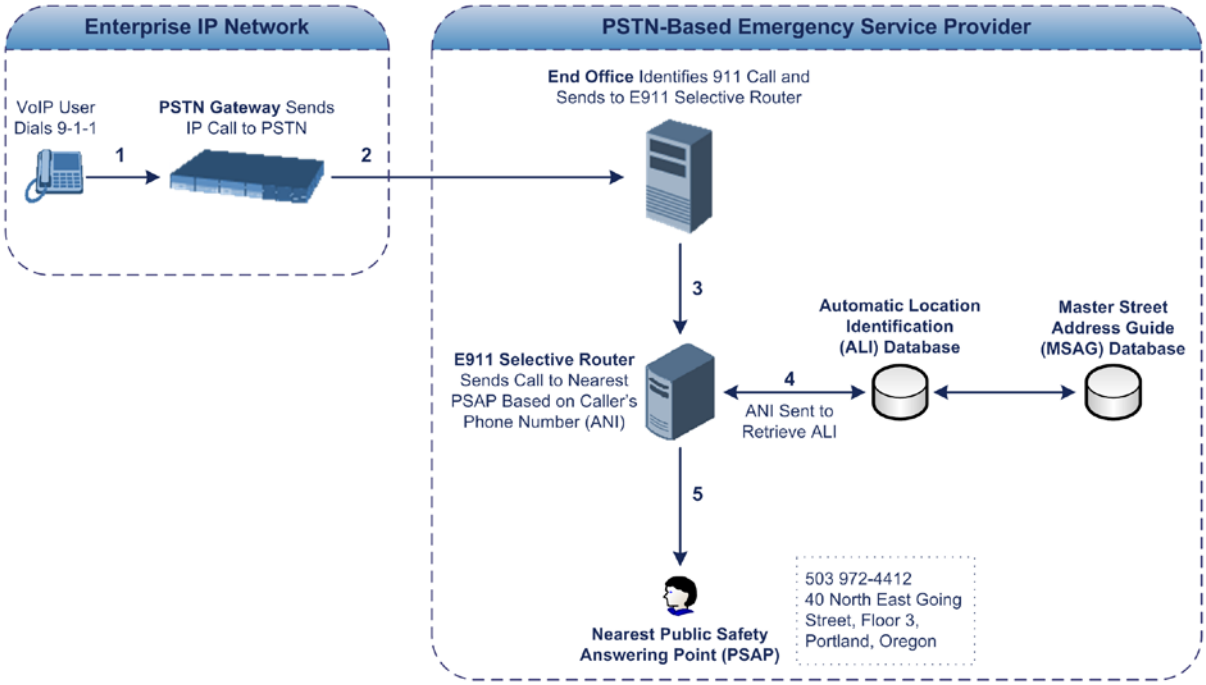
This section describes the E9-1-1 solution offered by Microsoft Lync Server 2010 (hereafter referred to as *Lync Server 2010*), and the deployed AudioCodes ELIN Gateway which provides the ISDN (or CAMA) connectivity to the PSTN-based E9-1-1 emergency providers. This document also describes the configuration of AudioCodes ELIN Gateway for interoperating between the Lync Server 2010 environment and the E9-1-1 emergency provider.

19.6.11.1 About E9-1-1 Services

Enhanced 9-1-1 (E9-1-1) is a national emergency service for many countries, enabling E9-1-1 operators to automatically identify the geographical location and phone number of a 911 caller. In E9-1-1, the 911 caller is routed to the nearest E9-1-1 operator, termed *public safety answering point* (PSAP) based on the location of the caller. Automatic identification of the caller's location and phone number reduces the time spent on requesting this information from the 911 caller. Therefore, the E9-1-1 service enables the PSAP to quickly dispatch the relevant emergency services (for example, fire department or police) to the caller's location. Even if the call prematurely disconnects, the operator has sufficient information to call back the 911 caller.

The figure below illustrates the routing of an E9-1-1 call to the PSAP:

Figure 27: Call Flow of E9-1-1 to PSTN-Based Emergency Services Provider



1. The VoIP user dials 9-1-1.
2. The call is eventually sent to the PSTN through a PSTN Gateway.

3. The PSTN identifies the call is an emergency call and sends it to an E9-1-1 Selective Router in the Emergency Services provider's network.
4. The E9-1-1 Selective Router determines the geographical location of the caller by requesting this information from an Automatic Location Identification (ALI) database based on the phone number or Automatic Number Identifier (ANI) of the 911 caller. Exact location information is also supplied by the Master Street Address Guide (MSAG) database, which is a companion database to the ALI database. Phone companies and public safety agencies collaborate beforehand to create master maps that match phone numbers, addresses and cross streets to their corresponding PSAP. This MSAG is the official record of valid streets (with exact spelling), street number ranges, and other address elements with which the service providers are required to update their ALI databases.
5. The E9-1-1 Selective Router sends the call to the appropriate PSAP based on the retrieved location information from the ALI.
6. The PSAP operator dispatches the relevant emergency services to the E9-1-1 caller.

19.6.11.2 Microsoft Lync Server 2010 and E9-1-1

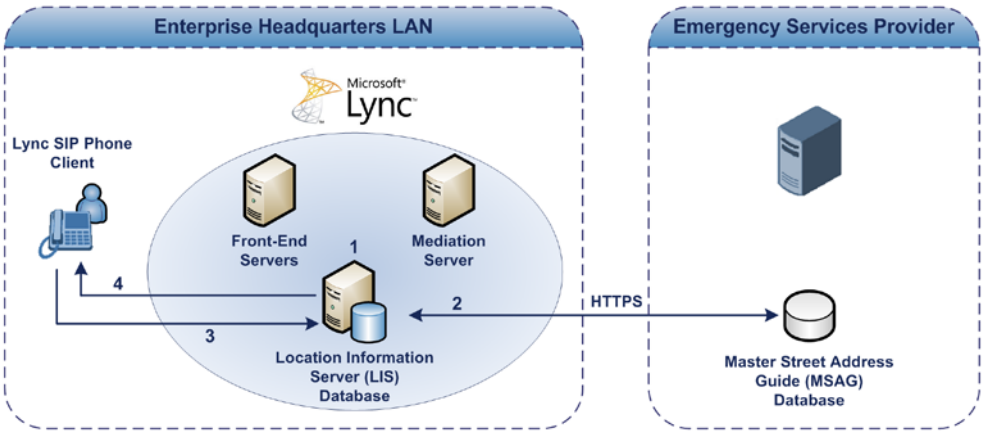
Microsoft Lync Server 2010 enables Enterprise voice users to access its unified communications platform from virtually anywhere and through many different devices. This, together with a deployed MLTS, poses a challenge for E9-1-1 due to the difficulty in accurately locating the E9-1-1 caller. However, Lync Server 2010 offers an innovative solution to solving Enterprises E9-1-1 location problems.

19.6.11.2.1 Gathering Location Information of Lync 2010 Clients for 911 Calls

When a Microsoft® Lync™ 2010 client (hereafter referred to as *Lync 2010 client*) is enabled for E9-1-1, the location data that is stored on the client is sent during an emergency call. This stored location information is acquired automatically from the Microsoft Location Information Server (LIS). The LIS stores the location of each network element in the enterprise. Immediately after the Lync 2010 client registration process or when the operating system detects a network connection change, each Lync 2010 client submits a request to the LIS for a location. If the LIS is able to resolve a location address for the client request, it returns the address in a location response. Each client then caches this information. When the Lync 2010 client dials 9-1-1, this location information is then included as part of the emergency call and used by the Emergency Services provider to route the call to the correct PSAP.

The gathering of location information in the Lync Server 2010 network is illustrated in the figure below:

Figure 28: Microsoft Lync Server 2010 Client Acquiring Location Information



1. The Administrator provisions the LIS database with the location of each network element in the Enterprise. The location is a civic address, which can include contextual in-building and company information. In other words, it associates a specific network entity (for example, a WAP) with a physical location in the Enterprise (for example, Floor 2, Wing A, and the Enterprise's street address). For more information on populating the LIS database, see Section 19.6.11.2.2 on page 336.
2. The Administrator validates addresses with the Emergency Services provider's MSAG –a companion database to the ALI database. This ensures that the civic address is valid as an official address (e.g., correct address spelling).
3. The Lync 2010 client initiates a location request to the LIS under the following circumstances:
   - Immediately after startup and registering the user with Lync Server 2010
   - Approximately every four hours after initial registration
   - Whenever a network connection change is detected (such as roaming to a new WAP)

   The Lync 2010 client includes in its location request the following known network connectivity information:
   - Always included:
     - IPv4 subnet
     - Media Access Control (MAC) address
   - Depends on network connectivity:
     - Wireless access point (WAP) Basic Service Set Identifier (BSSID)
     - Link Layer Discovery Protocol-Media Endpoint Discovery (LLDP-MED) chassis ID and port ID

   For a Lync 2010 client that moves inside the corporate network such as a soft phone on a laptop that connects wirelessly to the corporate network, Lync Server 2010 can determine which subnet the phone belongs to or which WAP / SSID is currently serving the soft-client.
4. The LIS queries the published locations for a location and if a match is found, returns the location information to the client. The matching order is as follows:
   - WAP BSSID
   - LLDP switch / port
   - LLDP switch
   - Subnet
   - MAC address

   This logic ensures that for any client that is connected by a wireless connection, a match is first attempted based on the hardware address of its connected access point. The logic is for the match to be based on the most detailed location. The subnet generally provides the least detail. If no match is found in the LIS for WAP BSSID, LLDP switch / port, LLDP switch, or subnet, the LIS proxies the MAC address to an integrated Simple Network Management Protocol (SNMP) scanning application. Using SNMP may benefit some organizations for the following reasons:
   - LLDP is not supported by Lync Server 2010 so this provides a mechanism for soft phones to acquire detailed location information.
   - Installed Layer-2 switches may not support LLDP.

If there is no match and the LIS cannot determine the location, the user may be prompted to manually enter the location. For example, the client may be located in an undefined subnet, at home, in a coffee shop or anywhere else outside the network. When a user manually provides a location, the location is mapped based on the MAC address of the default gateway of the client's network and stored on the client. When the client returns to any previously stored location, the client is automatically set to that location. A user can

also manually select any location stored in the local users table and manage existing entries.

19.6.11.2.2 Adding ELINs to the Location Information Server

As mentioned in the previous section, the Administrator needs to populate the Location Information Server (LIS) database with a network wire map, which maps the Enterprise's network elements to civic addresses. Once done, it can automatically locate clients within a network. You can add addresses individually to the LIS or in a batch using a comma-separated value (CSV) file containing the column formats listed in the table below.

Network Element	Columns
Wireless access point	<BSSID>,<Description>,<Location>,<**CompanyName**>,<HouseNumber>,<HouseNumberSuffix>,<PreDirectional>,…<StreetName>,<StreetSuffix>,<PostDirectional>,<City>,<State>,<PostalCode>,<Country>
Subnet	<Subnet>,<Description>,<Location>,<CompanyName>,<HouseNumber>,<HouseNumberSuffix>,<PreDirectional>,…<StreetName>,<StreetSuffix>,<PostDirectional>,<City>,<State>,<PostalCode>,<Country>
Port	<ChassisID>,<PortIDSubType>,<PortID>,<Description>,<Location>,<CompanyName>,<HouseNumber>,<HouseNumberSuffix>,…<PreDirectional>,<StreetName>,<StreetSuffix>,<PostDirectional>,<City>,<State>,<PostalCode>,<Country>
Switch	<ChassisID>,<Description>,<Location>,<CompanyName>,<HouseNumber>,<HouseNumberSuffix>,<PreDirectional>,…<StreetName>,<StreetSuffix>,<PostDirectional>,<City>,<State>,<PostalCode>,<Country>

For the ELIN number to be included in the SIP INVITE (XML-based PIDF-LO message) sent by the Mediation Server to the ELIN Gateway, the Administrator must add the ELIN number to the <CompanyName> column (shown in the table above in **bold** typeface). As the ELIN Gateway supports up to five ELINs per PIDF-LO, the <CompanyName> column can be populated with up to this number of ELINs, each separated by a semicolon. The digits of each ELIN can be separated by hyphens (xxx-xxx-xxx) or they can be adjacent (xxxxxxxxx).

When the ELIN Gateway receives the SIP INVITE, it extracts the ELINs from the NAM field in the PIDF-LO (e.g., <ca:NAM>1111-222-333; 1234567890 </ca:NAM>), which corresponds to the <CompanyName> column of the LIS.

If you do not populate the location database, and the Lync Server 2010 location policy, Location Required is set to **Yes** or **Disclaimer**, the user will be prompted to enter a location manually.

19.6.11.2.3 Passing Location Information to the PSTN Emergency Provider

When a Lync 2010 client, enabled for E9-1-1 emergency services, dials 9-1-1, the location data and callback information stored on the client is sent with the call through the Mediation Server to a PSTN-based Emergency Services provider. The Emergency Services provider then routes the call to the nearest and most appropriate PSAP based on the location information contained within the call.

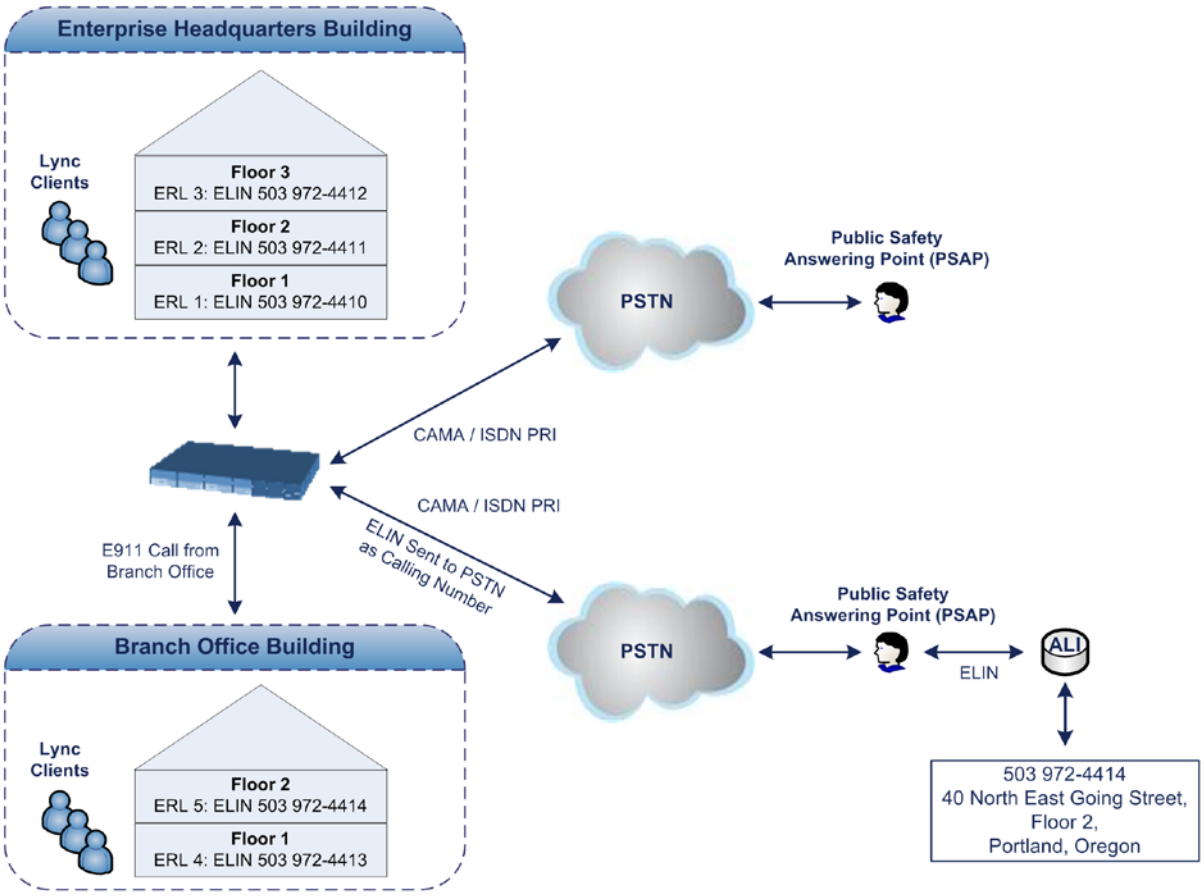
Lync Server 2010 passes the location information of the Lync 2010 client in an IETF-standard format–Presence Information Data Format - Location Object (PIDF-LO)–in a SIP INVITE message. However, this content cannot be sent on the PSTN network using ISDN PRI due to protocol limitations. To overcome this, Enterprises using PSTN Gateways can

divide their office space into Emergency Response Locations (ERLs) and assign a dedicated Emergency Location Identification Number (ELIN) to each ERL (or zone). When Lync Server 2010 sends a SIP INVITE message with the PIDF-LO to the PSTN Gateway, it can parse the content and translate the calling number to an appropriate ELIN. The PSTN Gateway then sends the call to the PSTN with the ELIN number as the calling number. This ELIN number is sent to the Emergency Services provider, which sends it on to the appropriate PSAP according to the ELIN address match in the ALI database lookup.

The ERL defines a specific location at a street address, for example, the floor number of the building at that address. The geographical size of an ERL is according to local or national regulations (for example, less than 7000 square feet per ERL). Typically, you would have an ERL for each floor of the building. The ELIN is used as the phone number for 911 callers within this ERL.

The figure below illustrates the use of ERLs and ELINs, with an E9-1-1 call from floor 2 at the branch office:

Figure 29: Implementing ERLs and ELINs for E9-1-1 in Microsoft Lync Server 2010



The table below shows an example of designating ERLs to physical areas (floors) in a building and associating each ERL with a unique ELIN.

ERL Number	Physical Area	IP Address	ELIN
1	Floor 1	10.13.124.xxx	503 972-4410
2	Floor 2	10.15.xxx.xxx	503 972-4411
3	Floor 3	10.18.xxx.xxx	503 972-4412

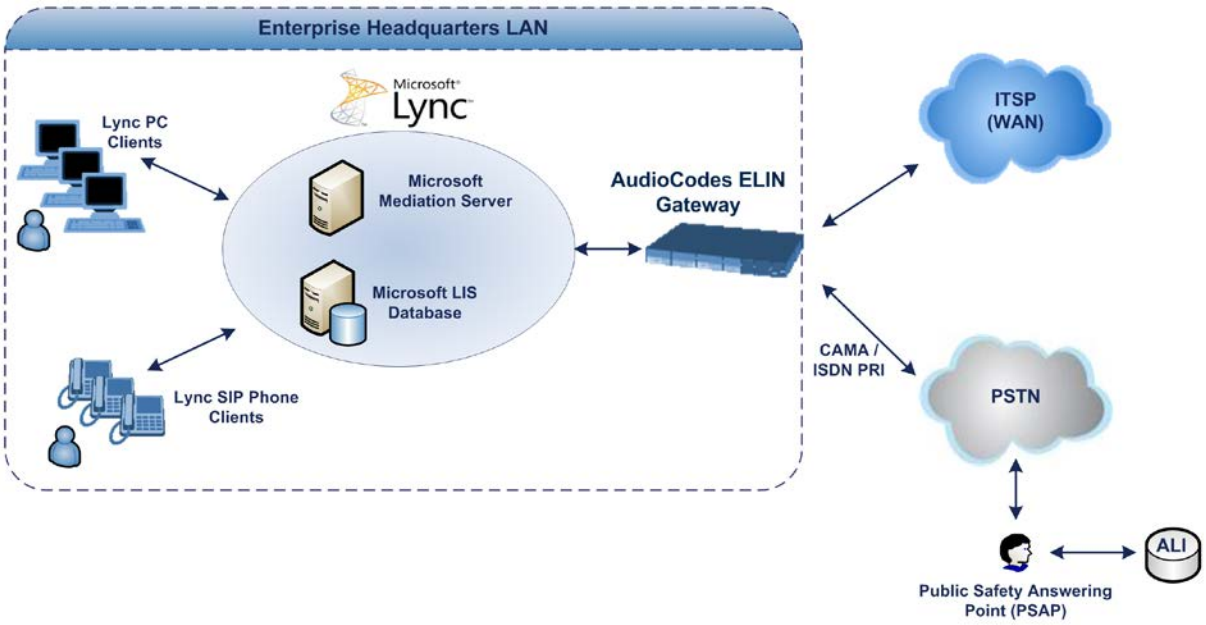
In the table above, a unique IP subnet is associated per ERL. This is useful if you implement different subnets between floors. Therefore, IP phones, for example, on a specific floor are in the same subnet and therefore, use the same ELIN when dialing 9-1-1.

19.6.11.3 AudioCodes ELIN Gateway for Lync Server 2010 E9-1-1 Calls to PSTN

The Microsoft Mediation Server sends the location information of the E9-1-1 caller in the XML-based PIDF-LO body contained in the SIP INVITE message. However, this content cannot be sent on the PSTN network using ISDN PRI due to protocol limitations. To solve this issue, Lync Server 2010 requires a PSTN Gateway (*ELIN Gateway*) to send the E9-1-1 call to the PSTN. When Lync Server 2010 sends the PIDF-LO to the PSTN Gateway, it parses the content and translates the calling number to an appropriate ELIN. This ensures that the call is routed to an appropriate PSAP, based on ELIN-address match lookup in the Emergency Services provider's ALI database.

The figure below illustrates an AudioCodes ELIN Gateway deployed in the Lync Server 2010 environment for handling E9-1-1 calls between the Enterprise and the PSTN.

Figure 30: AudioCodes ELIN Gateway for E9-1-1 in Microsoft Lync Server 2010 Environment



19.6.11.3.1 Detecting and Handling E9-1-1 Calls

The ELIN Gateway identifies E9-1-1 calls and translates their incoming E9-1-1 calling numbers into ELIN numbers, sent toward the PSAP. The ELIN Gateway handles the received E9-1-1 calls as follows:

1. The ELIN Gateway identifies E9-1-1 calls if the incoming SIP INVITE message contains a PIDF-LO XML message body. This is indicated in the SIP *Content-Type* header, as shown below:

```
Content-Type: application/pidf+xml
```

2. The ELIN Gateway extracts the ELIN number(s) from the "NAM" field in the XML message. The "NAM" field corresponds to the <CompanyName> column in the Location Information Server (LIS). The ELIN Gateway supports up to five ELIN numbers per XML message. The ELINs are separated by a semicolon. The digits of the ELIN number can be separated by hyphens (xxx-xxx-xxx) or they can be adjacent (xxxxxxxxx), as shown below:

```
<ca:NAM>1111-222-333; 1234567890 </ca:NAM>
```

3. The ELIN Gateway saves the *From* header value of the SIP INVITE message in its ELIN database table (**Call From** column). The ELIN table is used for PSAP callback, as discussed later in Section 19.6.11.3.3 on page 341. The ELIN table also stores the following information:
   - **ELIN:** ELIN number
   - **Time:** Time at which the original E9-1-1 call was terminated with the PSAP
   - **Count:** Number of E9-1-1 calls currently using this ELIN

   An example of the ELIN database table is shown below:

ELIN	Time	Count	Index	Call From
4257275678	22:11:52	0	2	4258359333
4257275999	22:11:57	0	3	4258359444
4257275615	22:12:03	0	0	4258359555
4257275616	22:11:45	0	1	4258359777

   The ELIN table stores this information for a user-defined period (see Section 19.6.11.4.2 on page 342), starting from when the E9-1-1 call, established with the PSAP, terminates. After this time expires, the table entry with its ELIN is disregarded and no longer used (for PSAP callback). Therefore, table entries of only the most recently terminated
   E9-1-1 callers are considered in the ELIN table.

   The maximum entries in the ELIN table depend on the AudioCodes ELIN Gateway deployed in the Lync Server 2010 environment:
   - **Mediant 1000 Series and Mediant 2000:** 100 entries
   - **Mediant 3000:** 300 entries

4. The ELIN Gateway uses the ELIN number as the E9-1-1 calling number and sends it in the ISDN Setup message (as an ANI / Calling Party Number) to the PSTN.

An example of a SIP INVITE message received from an E9-1-1 caller is shown below. The SIP *Content-Type* header indicating the PIDF-LO, and the NAM field listing the ELINs are shown in **bold** typeface.

```
INVITE sip:911;phone-context=Redmond@192.168.1.12;user=phone
SIP/2.0
From:
"voip_911_user1"<sip:voip_911_user1@contoso.com>;epid=1D19090AED;t
ag=d04d65d924
To: <sip:911;phone-context=Redmond@192.168.1.12;user=phone>
CSeq: 8 INVITE
Call-ID: e6828be1-1cdd-4fb0-bdda-cda7faf46df4
VIA: SIP/2.0/TLS 192.168.0.244:57918;branch=z9hG4bK528b7ad7
```

```
CONTACT:
<sip:voip_911_user1@contoso.com;opaque=user:epid:R4bCDaUj51a06PUbk
raS0QAA;gruu>;text;audio;video;image
PRIORITY: emergency
CONTENT-TYPE: multipart/mixed; boundary= ------
=_NextPart_000_4A6D_01CAB3D6.7519F890
geolocation: <cid:voip_911_user1@contoso.com>;inserted-
by="sip:voip_911_user1@contoso .com"
Message-Body:
------=_NextPart_000_4A6D_01CAB3D6.7519F890
Content-Type: application/sdp ; charset=utf-8
v=0
o=- 0 0 IN IP4 Client
s=session
c=IN IP4 Client
t=0 0
m=audio 30684 RTP/AVP 114 111 112 115 116 4 3 8 0 106 97
c=IN IP4 172.29.105.23
a=rtcp:60423
a=label:Audio
a=rtpmap:3 GSM/8000/1
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=ptime:20

------=_NextPart_000_4A6D_01CAB3D6.7519F890
Content-Type: application/pidf+xml
Content-ID: <voip_911_user1@contoso.com>
<?xml version="1.0" encoding="utf-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
xmlns:bp="urn:ietf:params:xml:ns:pidf:geopriv10:basicPolicy"
xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
xmlns:ms="urn:schema:Rtc.LIS.msftE911PidfExtn.2008"
entity="sip:voip_911_user1@contoso.com"><tuple
id="0"><status><gp:geopriv><gp:location-
info><ca:civicAddress><ca:country>US</ca:country><ca:A1>WA</ca:A1>
<ca:A3>Redmond</ca:A3><ca:RD>163rd</ca:RD><ca:STS>Ave</ca:STS><ca:
POD>NE</ca:POD><ca:HNO>3910</ca:HNO><ca:LOC>40/4451</ca:LOC>
<ca:NAM>1111-222-333; 1234567890 </ca:NAM>
<ca:PC>98052</ca:PC></ca:civicAddress></gp:location-
info><gp:usage-rules><bp:retransmission-
allowed>true</bp:retransmission-allowed></gp:usage-
rules></gp:geopriv><ms:msftE911PidfExtn><ms:ConferenceUri>sip:+142
55550199@contoso.com;user=phone</ms:ConferenceUri><ms:ConferenceMo
de>twoway</ms:ConferenceMode><LocationPolicyTagID
xmlns="urn:schema:Rtc.Lis.LocationPolicyTagID.2008">user-
tagid</LocationPolicyTagID
></ms:msftE911PidfExtn></status><timestamp>1991-09-
22T13:37:31.03</timestamp></tuple></presence>
------=_NextPart_000_4A6D_01CAB3D6.7519F890--
```

19.6.11.3.2 Pre-empting Existing Calls for E9-1-1 Calls

If the ELIN Gateway receives an E9-1-1 call from the IP network and there are unavailable channels (for example, all busy), the ELIN Gateway immediately terminates one of the non-E9-1-1 calls (arbitrary) and accepts the E9-1-1 call on the freed channel.

The preemption is done only on a channel pertaining to the same Trunk Group for which the E9-1-1 call was initially destined. For example, if an E9-1-1 call is destined for Trunk Group #2 and all the channels belonging to this group are busy, the ELIN Gateway terminates one of the calls in this group to free a channel for accepting the E9-1-1 call.

This feature is initiated only if the received SIP INVITE message contains a *Priority* header set to "emergency", as shown below:

```
PRIORITY: emergency
```

19.6.11.3.3 PSAP Callback to Lync 2010 Clients for Dropped E9-1-1 Calls

As the E9-1-1 service automatically provides all the contact information of the E9-1-1 caller to the PSAP, the PSAP operator can call back the E9-1-1 caller. This is especially useful in cases where the caller disconnects prematurely. However, as the Enterprise sends ELINs to the PSAP for E9-1-1 calls, a callback can only reach the original E9-1-1 caller using the ELIN Gateway to translate the ELIN number back into the E9-1-1 caller's extension number.

In the ELIN table of the ELIN Gateway, the temporarily stored *From* header value of the SIP INVITE message originally received from the E9-1-1 caller is used for PSAP callback. When the PSAP makes a callback to the E9-1-1 caller, the ELIN Gateway translates the called number (i.e., ELIN) received from the PSAP to the corresponding E9-1-1 caller's extension number as matched in the ELIN table.

The handling of PSAP callbacks by the ELIN Gateway is as follows:

1. When the ELIN Gateway receives any call from the PSTN, it searches the ELIN table for an ELIN that corresponds to the received Called Party Number in the incoming PSTN call.
2. If a match is found in the ELIN table, it routes the call to the Mediation Sever by sending a SIP INVITE, where the values of the *To* and *Request-URI* are taken from the value of the original *From* header that is stored in the ELIN table (in the **Call From** column).
3. The ELIN Gateway updates the Time in the ELIN table. (The Count is not affected).

The PSAP callback can be done only within a user-defined timeout (see Section 19.6.11.4.2 on page 342) started from after the original E9-1-1 call established with the PSAP is terminated. After this time expires, the table entry with its ELIN is disregarded and no longer used (for PSAP callback). Therefore, table entries of only the most recently terminated
E9-1-1 callers are considered in the ELIN table. If the PSAP callback is done after this timeout expires, the ELIN Gateway is unable to route the call to the E9-1-1 caller and instead, either sends it as a regular call or most likely, rejects it if there are no matching routing rules. However, if another E9-1-1 caller has subsequently been processed with the same ELIN number, then the PSAP callback is routed to this new E9-1-1 caller.

In scenarios where the same ELIN number is being used by multiple E9-1-1 callers, upon receipt of a PSAP callback, the ELIN Gateway sends the call to the most recent E9-1-1 caller. For example, if the ELIN number "4257275678" is being used by three E9-1-1 callers, as shown in the table below, then when a PSAP callback is received, the ELIN Gateway sends it to the E9-1-1 caller with phone number "4258359555".

ELIN	Time	Call From
4257275678	11:00	4258359333

ELIN	Time	Call From
4257275678	11:01	4258359444
4257275678	11:03	**4258359555**

19.6.11.3.4 Selecting ELIN for Multiple Calls within Same ERL

The ELIN Gateway supports the receipt of up to five ELIN numbers in the XML message of each incoming SIP INVITE message. As discussed in the preceding sections, the ELIN Gateway sends the ELIN number as the E9-1-1 calling number to the PSTN-based emergency provider. If the XML message contains more than one ELIN number, the ELIN Gateway chooses the ELIN according to the following logic:

- If the first ELIN in the list is not being used by other active calls, it chooses this ELIN.
- If the first ELIN in the list is being used by another active call, the ELIN Gateway skips to the next ELIN in the list, and so on until it finds an ELIN that is not being used and sends this ELIN.
- If all the ELINs in the list are in use by active calls, the ELIN Gateway selects the ELIN number as follows:
  1. The ELIN with the lowest count (i.e., lowest number of active calls currently using this ELIN).
  2. If the count between ELINs is identical, the ELIN Gateway selects the ELIN with the greatest amount of time passed since the original E9-1-1 call using this ELIN was terminated with the PSAP. For example, if E9-1-1 caller using ELIN 4257275678 was terminated at **11:01** and E9-1-1 caller using ELIN 4257275670 was terminated at **11:03**, then the ELIN Gateway selects ELIN 4257275678.

  In this scenario, multiple E9-1-1 calls will be sent with the same ELIN.

19.6.11.4 Configuring AudioCodes ELIN Gateway

This chapter describes E9-1-1 configuration of the AudioCodes ELIN Gateway deployed in the Microsoft Lync Server 2010 environment.

19.6.11.4.1 Enabling the E9-1-1 Feature

By default, the E9-1-1 feature in the ELIN Gateway for Lync Server 2010 is disabled. To enable it, the following *ini* file parameter setting must be done:

```
E911Gateway = 1
```

19.6.11.4.2 Configuring the E9-1-1 Callback Timeout

The PSAP can use the ELIN to call back the E9-1-1 caller within a user-defined time interval (in minutes) from when the initial call established with the PSAP has been terminated. By default, an ELIN can be used for PSAP callback within 30 minutes after the call is terminated. You can change this interval, by using the following *ini* file parameter:

```
E911CallbackTimeout = <time value>   ; where <time value> can be
                                       any value from 0 through 60
```

19.6.11.4.3 Configuring the SIP Release Cause Code for Failed E9-1-1 Calls

When a Lync 2010 client makes an emergency call, the call is routed through the Microsoft Mediation Server to the ELIN Gateway, which sends it on to the PSTN. In some instances, the call may not be established due to either the PSTN (for example, the destination is busy or not found) or ELIN Gateway (for example, lack of resources or an internal error). In such a scenario, the Mediation Server requires that the ELIN Gateway "reject" the call with the SIP release cause code 503 "Service Unavailable", instead of the designated release call. Such a release cause code enables the Mediation Server to issue a failover to another entity (for example, another ELIN Gateway), instead of retrying the call or returning the release call to the user.

To support this requirement, the ELIN Gateway can be configured to send the 503 "Service Unavailable" release cause code instead of SIP 4xx if an emergency call cannot be established. To enable this support, the following *ini* file parameter setting must be done:

```
EmergencySpecialReleaseCause = 1
```

To disable this feature, set the EmergencySpecialReleaseCause parameter to 0 (default).

19.6.11.4.4 Viewing the ELIN Table

You can view the ELIN table of the ELIN Gateway:

- Using the following CLI command:

```
# show voip gw e911
ELIN                Time      Count Index Call From
-------------------------------------------------------------
4257275678          22:11:52  0     2     4258359333
4257275999          22:11:57  0     3     4258359444
4257275615          22:12:03  0     0     4258359555
4257275616          22:11:45  0     1     4258359777
------------ Current Time: 22:12:40
```

- Using Syslog, by invoking the following Web command shell:

```
SIP / GateWay / E911Dump
```

19.6.12 Multilevel Precedence and Preemption

The device's Multilevel Precedence and Preemption (MLPP) service can be enabled using the CallPriorityMode parameter. MLPP is a call priority scheme, which does the following:

- Assigns a precedence level (priority level of call) to specific phone calls or messages.
- Allows higher priority calls (*precedence call*) and messages to preempt lower priority calls and messages (i.e., terminates existing lower priority calls) that are recognized within a user-defined domain (*MLPP domain ID*). The domain specifies the collection of devices and resources that are associated with an MLPP subscriber. When an MLPP subscriber that belongs to a particular domain places a precedence call to another MLPP subscriber that belongs to the same domain, MLPP service can preempt the existing call that the called MLPP subscriber is on for a higher-precedence call. MLPP service availability does not go across different domains

MLPP is typically used in the military where for example, high-ranking personnel can preempt active calls during network stress scenarios, such as a national emergency or degraded network situations.

The Resource Priority value in the Resource-Priority SIP header can be any on of those listed in the table below. A default MLPP call Precedence Level (configured by the SIPDefaultCallPriority parameter) is used if the incoming SIP INVITE or PRI Setup message contains an invalid priority or Precedence Level value respectively. For each MLPP call priority level, the Multiple Differentiated Services Code Points (DSCP) can be set to a value from 0 to 63.

Table 19-19: MLPP Call Priority Levels (Precedence) and DSCP Configuration Parameters

MLPP Precedence Level	Precedence Level in Resource-Priority SIP Header	DSCP Configuration Parameter
0 (lowest)	routine	MLPPRoutineRTPDSCP
2	priority	MLPPPriorityRTPDSCP
4	immediate	MLPPImmediateRTPDSCP
6	flash	MLPPFlashRTPDSCP
8	flash-override	MLPPFlashOverRTPDSCP
9 (highest)	flash-override-override	MLPPFlashOverOverRTPDSCP

- **Precedence Ring Tone:** You can assign a ring tone (in the CPT file) that is played when a Precedence call is received from the IP side. This is configured by the parameter PrecedenceRingingType. In addition, you can define (using the PreemptionToneDuration parameter) the duration for which the device plays a preemption tone to the Tel and IP sides if a call is preempted.
- Emergency Telecommunications Services calls (e.g., E911): ETS calls can be configured to be regarded as having a higher priority than any MLPP call (default), using the E911MLPPBehavior parameter.
- **MLPP Preemption Events in SIP Reason Header:** The device sends the SIP Reason header (as defined in RFC 4411) to indicate the reason a preemption event occurred and the type of preemption event. The device sends a SIP BYE or CANCEL request, or 480, 486, 488 responses (as appropriate) with a Reason header whose Reason-params can includes one of the following preemption cause classes:
  - Reason: preemption ;cause=1 ;text="UA Preemption"
  - Reason: preemption ;cause=2 ;text="Reserved Resources Preempted"
  - Reason: preemption ;cause=3 ;text="Generic Preemption"
  - Reason: preemption ;cause=4 ;text="Non-IP Preemption"

- Reason: preemption; cause=5; text="Network Preemption"

Cause=4: The Reason cause code "Non-IP Preemption" indicates that the session preemption has occurred in a non-IP portion of the infrastructure. The device sends this code in the following scenarios:

- The device performs a network preemption of a busy call (when a high priority call is received), the device sends a SIP BYE or CANCEL request with this Reason cause code.
- The device performs a preemption of a B-channel for a Tel-to-IP outbound call request from the softswitch for which it has not received an answer response (e.g., Connect), and the following sequence of events occurs:
  - **a.** The device sends a Q.931 DISCONNECT over the ISDN MLPP PRI to the partner switch to preempt the remote end instrument.
  - **b.** The device sends a 488 (Not Acceptable Here) response with this Reason cause code.

Cause=5: The Reason cause code "Network Preemption" indicates preempted events in the network. Within the Defense Switched Network (DSN) network, the following SIP request messages and response codes for specific call scenarios have been identified for signaling this preemption cause:

- SIP:BYE - If an active call is being preempted by another call
- CANCEL - If an outgoing call is being preempted by another call
- 480 (Temporarily Unavailable), 486 (User Busy), 488 (Not Acceptable Here) - Due to incoming calls being preempted by another call.

The device receives SIP requests with preemption reason cause=5 in the following cases:

- The softswitch performs a network preemption of an active call - the following sequence of events occurs:
  - **a.** The softswitch sends the device a SIP BYE request with this Reason cause code.
  - **b.** The device initiates the release procedures for the B-channel associated with the call request and maps the preemption cause to PRI Cause = #8 'Preemption'. This value indicates that the call is being preempted. For PRI, it also indicates that the B-channel is not reserved for reuse.
  - **c.** The device sends a SIP 200 OK in response to the received BYE, before the SIP end instrument can proceed with the higher precedence call.
- The softswitch performs a network preemption of an outbound call request for the device that has not received a SIP 2xx response - the following sequence of events occur:
  - **a.** The softswitch sends the device a SIP 488 (Not Acceptable Here) response code with this Reason cause code. The device initiates the release procedures for the B-channel associated with the call request and maps the preemption cause to PRI Cause = #8 'Preemption'.
  - **b.** The device deactivates any user signaling (e.g., ringback tone) and when the call is terminated, it sends a SIP ACK message to the softswitch

Notes:

- If required, you can exclude the "resource-priority" tag from the SIP Require header in INVITE messages for Tel-to-IP calls when MLPP priority call handling is used. This is configured using the RPRequired parameter.
- For a complete list of the MLPP parameters, see 'MLPP Parameters' on page 890.

19.6.13 Denial of Collect Calls

You can configure the device to reject (disconnect) incoming Tel-to-IP collect calls and to signal this denial to the PSTN. This capability is required, for example, in the Brazilian telecommunication system to deny collect calls. When this feature is enabled upon rejecting the incoming call, the device sends a sequence of signals to the PSTN. This consists of an off-hook, an on-hook after one second, and then an off-hook after two seconds. In other words, this is in effect, a double-answer sequence.

This feature is enabled for all calls, using the EnableFXODoubleAnswer parameter.

Notes:

- This feature is applicable only to FXO interfaces.
- To support this feature, ensure that automatic dialing has not been configured for the FXO ports.
- Ensure that the PSTN side is configured to identify this double-answer signal.

19.6.14 Configuring ISDN Supplementary Services

The ISDN Supp Services Table page allows you to configure supplementary services for Integrated Services Digital Network (ISDN) Basic Rate Interface (BRI) phones connected to the device. This feature enables the device to route IP-to-Tel calls (including voice and fax) to specific BRI ports (channels).

This table allows you to define BRI phone extension numbers per BRI port pertaining to a specific BRI module. Therefore, this offers support for point-to-multipoint configuration of several phone numbers per BRI channel. Up to eight phone numbers can be defined per BRI trunk. In addition, for each BRI endpoint, the following optional configurations can be defined:

- User ID and password - for registering the BRI endpoint to a third-party softswitch for authentication and/or billing. For viewing BRI registration status, see 'Viewing Registration Status' on page 724.
- Caller ID name - for displaying the BRI endpoint's caller ID to a dialed destination, if enabled (i.e., "Presentation" is not restricted)
- Caller ID presentation or restriction
- Enable/disable sending caller ID to BRI endpoints

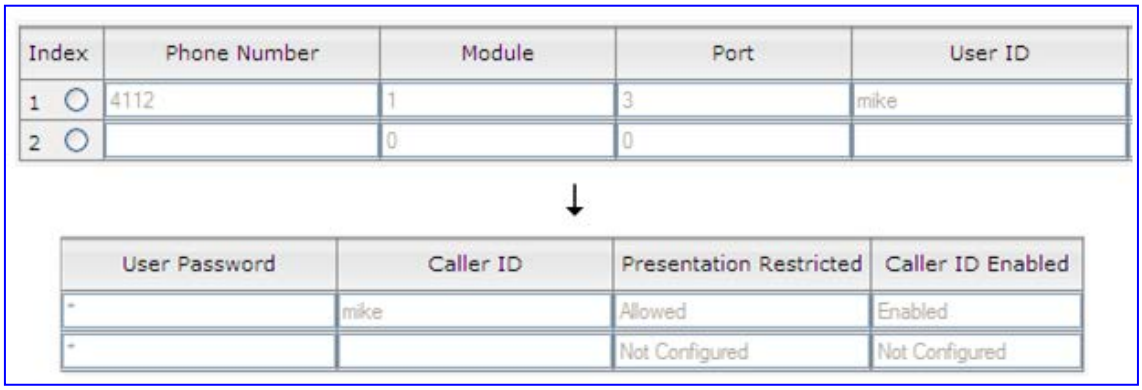
Notes:

- To use this table for routing of IP-to-Tel calls to specific BRI channels, the Channel Select Mode in the Trunk Group Settings must be set to 'Select Trunk by ISDN Supplementary Services Table' (see 'Configuring Trunk Group Settings' on page 273).
- You can also configure this table using the ISDNSuppServ *ini* file table parameter (see 'Configuration Parameters Reference' on page 752).
- To allow the end-user to hear a dial tone when picking up the BRI phone, it is recommended to set the Progress Indicator in the Setup Ack bit (0x10000=65536). Therefore, the recommended value is 0x10000 + 0 x1000 = 65536 + 4096 = 69632 (i.e., set the ISDNInCallsBehavior parameter to 69632).

➢ **To configure BRI supplementary services:**

1. Open the ISDN Supp Services Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Digital Gateway** submenu > **ISDN Supp Services**).

Figure 19-31: ISDN Supp Services Table Page



2. Configure the parameters as described in the table below.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.
5. To register the BRI endpoints, click the **Register** button. To unregister the BRI endpoints, click **Unregister**. The registration method for each BRI endpoint is according to the setting of the 'Registration Mode' parameter in the Trunk Group Settings page.

Table 19-20: ISDN Supp Services Table Parameters

Parameter	Description
Phone Number	The telephone extension number for the BRI endpoint.
Module	The BRI module number to which the BRI extension pertains.
Port	The port number (on the BRI module) to which the BRI extension is connected.
User ID	User ID for registering the BRI endpoint to a third-party softswitch for authentication and/or billing.
User Password	User password for registering the BRI endpoint to a third-party softswitch for authentication and/or billing. **Note:** For security, the password is displayed as an asterisk (*).
Caller ID	Caller ID name of the BRI extension (sent to the IP side). The valid value is a string of up to 18 characters.
Presentation Restricted	Determines whether the BRI extension sends its Caller ID information to the IP when a call is made. ▪ **[0]** Allowed = The device sends the string defined in the 'Caller ID' field when this BRI extension makes a Tel-to-IP call. ▪ **[1]** Restricted = The string defined in the 'Caller ID' field is not sent.
Caller ID Enabled	Enables the receipt of Caller ID. ▪ **[0]** Disabled = The device does not send Caller ID information to the BRI extension. ▪ **[1]** Enabled = The device sends Caller ID information to the BRI extension

19.6.15 Configuring Voice Mail Parameters

The Voice Mail Settings page allows you to configure the voice mail parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 752.

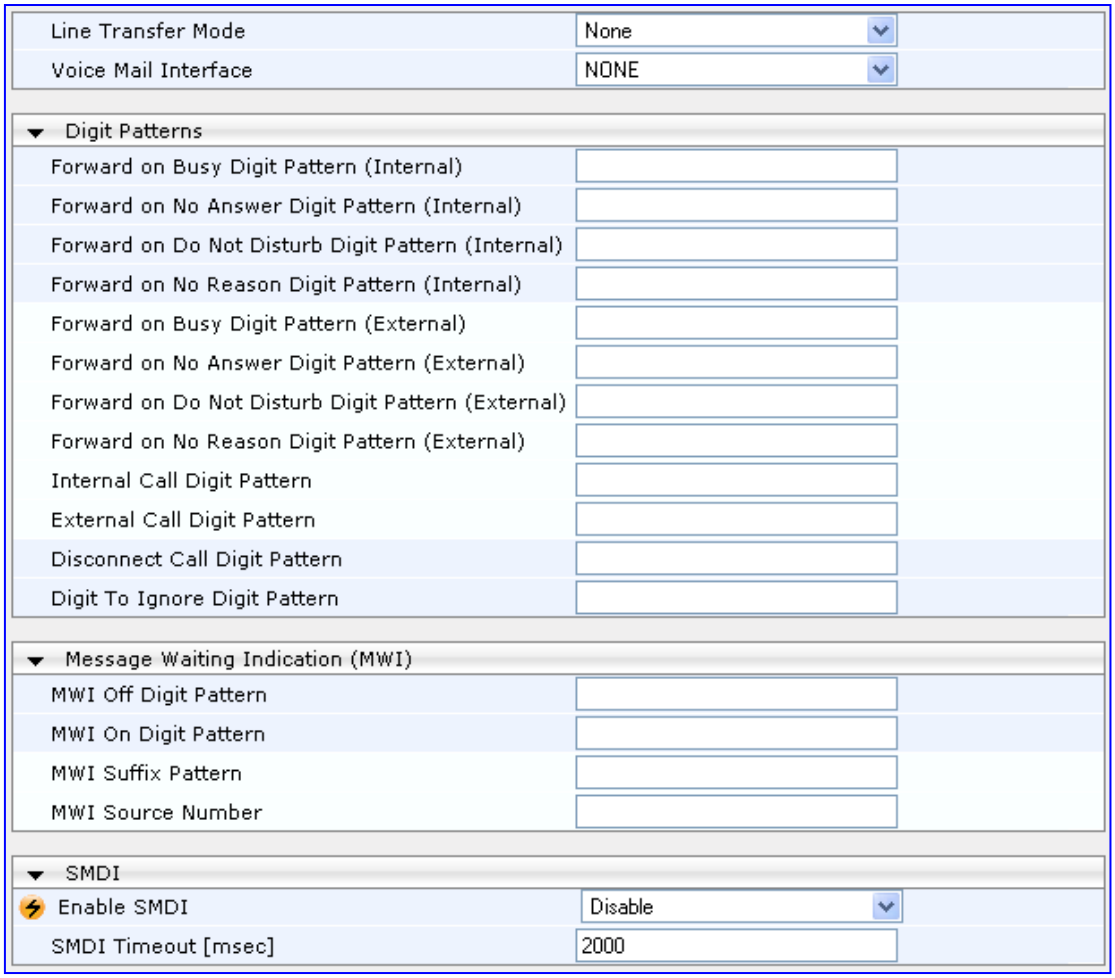
Notes:

- The Voice Mail Settings page is available only for FXO and CAS interfaces.
- For more information on configuring voice mail, refer to the *CPE Configuration Guide for Voice Mail User's Manual*.

➢ **To configure the Voice Mail parameters:**

1. Open the Voice Mail Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Advanced Applications** submenu > **Voice Mail Settings**).

Figure 19-32: Voice Mail Settings Page



2. Configure the parameters as required.
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.6.16 Advice of Charge Services for Euro ISDN

Advice of charge (AOC) is a pre-billing function that tasks the rating engine with calculating the cost of using a service and relaying that information back to the customer thus, allowing users to obtain charging information for all calls during the call (AOC-D) or at the end of the call (AOC-E), or both.

The AOC-D and AOC-E messages are part of the Facility Information Element (IE) message:

- AOC-D message—ISDN Advice of Charge information sent during a call. The message is sent periodically to subscribers of AOC during-call services.
- AOC-E message—ISDN Advice of Charge information sent at the end of a call.

The device supports the sending of AoC messages for Tel-to-IP calls, providing billing applications with the number of charged units. This feature can typically be implemented in the hotel industry, where external calls made by guests can be billed accurately. In such a setup, the device is connected on one side to a PBX through an E1 line (Euro ISDN), and on the other side to a SIP trunk provided by an ITSP. When a call is made by a guest, the device first sends an AOC-D Facility message to the PBX indicating the connection charge unit, and then sends subsequent AOC-D messages every user-defined interval to indicate the charge unit during the call. When the call ends, the device sends an AoC-E Facility message to the PBX indicating the total number of charged units.

➢ **To configure AoC:**

1. Ensure that the PSTN protocol for the E1 trunk line is EuroISDN and set to network side.
2. Ensure that the date and time of the device is correct. For accuracy, it is recommended to use an NTP server to obtain the date and time.
3. Enable the AoC service, using the EnableAOC parameter.
4. Configure charge codes in the Charge Code table (ChargeCode) - see 'Configuring Charge Codes' on page 353. Note that in the Chrage Code table, the table fields are as follows:
   - 'End Time' - time at which this charge code ends
   - 'Pulse Interval' - time between every sent AOC-D Facility message
   - 'Pulses On Answer' - number of charging units in first generated AOC-D Facility message
5. Assign the charge code index to the desired routing rules in the Outbound IP Routing table (see 'Configuring Outbound IP Routing Table' on page 291).

19.7 Analog Gateway

This section describes configuration of analog settings.

Note: The Analog Gateway submenu appears only if the device is installed with an FXS or FXO module.

19.7.1 Configuring Keypad Features

The Keypad Features page enables you to activate and deactivate the following features directly from the connected telephone's keypad:

- Call Forward - see 'Configuring Call Forward' on page 358
- Caller ID Restriction - see 'Configuring Caller Display Information' on page 357
- Hotline - see 'Configuring Automatic Dialing' on page 356
- Call Transfer
- Call Waiting - see 'Configuring Call Waiting' on page 361
- Rejection of Anonymous Calls

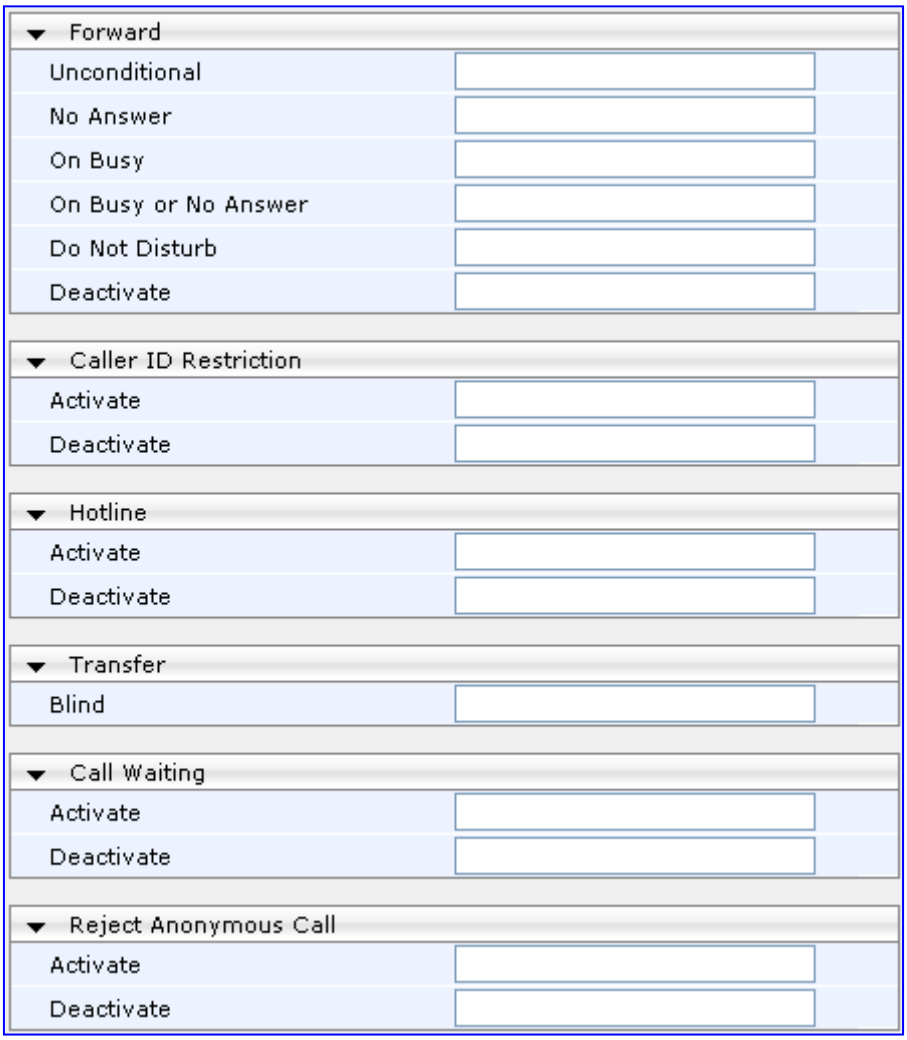
Notes:

- The Keypad Features page is available only for FXS interfaces.
- The method used by the device to collect dialed numbers is identical to the method used during a regular call (i.e., max digits, interdigit timeout, digit map, etc.).
- The activation of each feature remains in effect until it is deactivated (i.e., not deactivated after a call).

➢ To configure the keypad features

1. Open the Keypad Features page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Keypad Features**).

Figure 19-33: Keypad Features Page



2. Configure the keypad features as required. For a description of these parameters, see 'Configuration Parameters Reference' on page 752.
3. Click Submit to apply your changes.
4. To save the changes to the flash memory, see 'Saving Configuration' on page 680.

19.7.2 Configuring Metering Tones

The FXS interfaces can generate 12/16 KHz metering pulses toward the Tel side (e.g., for connection to a pay phone or private meter). Tariff pulse rate is determined according to the device's Charge Codes table. This capability enables users to define different tariffs according to the source/destination numbers and the time-of-day. The tariff rate includes the time interval between the generated pulses and the number of pulses generated on answer.

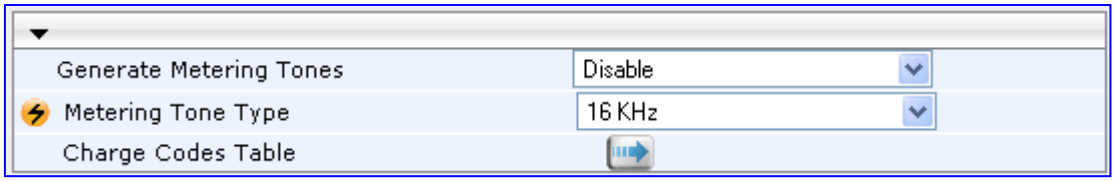
Notes:

- The Metering Tones page is available only for FXS interfaces.
- Charge Code rules can be assigned to routing rules in the Outbound IP Routing Table (see 'Configuring Outbound IP Routing Table' on page 291). When a new call is established, the Outbound IP Routing Table is searched for the destination IP address. Once a route is located, the Charge Code (configured for that route) is used to associate the route with an entry in the Charge Codes table.

➢ **To configure Metering tones:**

1. Open the Metering Tones page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Metering Tones**).

Figure 19-34: Metering Tones Page



2. Configure the Metering tones parameters as required. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 752.
3. Click **Submit** to apply your changes.
4. To save the changes to the flash memory, see 'Saving Configuration' on page 680.

If you set the 'Generate Metering Tones' parameter to **Internal Table**, access the Charge Codes Table page by clicking the **Charge Codes Table** button. For more information on configuring the Charge Codes table, see 'Configuring Charge Codes Table' on page 353.

19.7.3 Configuring Charge Codes

The Charge Codes Table page is used to configure the metering tones (and their time interval) that the FXS interfaces generate to the Tel side. To associate a charge code to an outgoing Tel-to-IP call, use the Outbound IP Routing Table'.

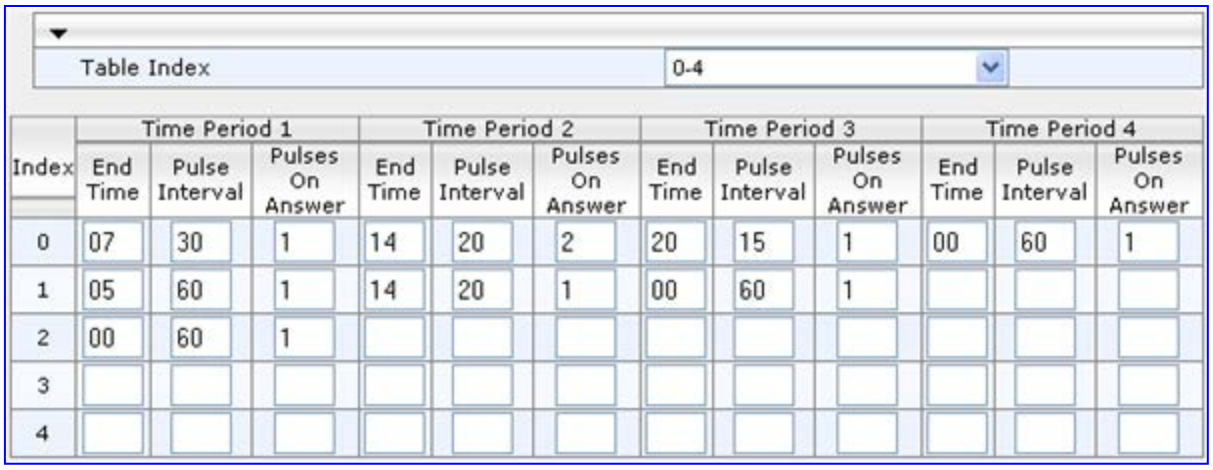
Notes:

- The Charge Codes Table page is available only for FXS interfaces.
- You can also configure the Charge Codes table using the *ini* file table parameter ChargeCode.

➢ **To configure the Charge Codes:**

1. Open the Charge Codes Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Charge Codes**). Alternatively, you can access this page from the Metering Tones page (see 'Configuring Metering Tones' on page 352).

Figure 19-35: Charge Codes Table Page



2. Define up to 25 different charge codes (each charge code is defined per row). Each charge code can include up to four different time periods in a day (24 hours). Each time period is composed of the following:
   - The end of the time period (in a 24 rounded-hour's format).
   - The time interval between pulses (in tenths of a second).
   - The number of pulses sent on answer.

   The first time period always starts at midnight (00). It is mandatory that the last time period of each rule ends at midnight (00). This prevents undefined time frames in a day. The device selects the time period by comparing the device 's current time to the end time of each time period of the selected Charge Code. The device generates the Number of Pulses on Answer once the call is connected and from that point on, it generates a pulse each Pulse Interval. If a call starts at a certain time period and crosses to the next, the information of the next time period is used.
3. Click **Submit** to apply your changes.
4. To save the changes to the flash memory, see 'Saving Configuration' on page 680.

19.7.4 Configuring FXO Settings

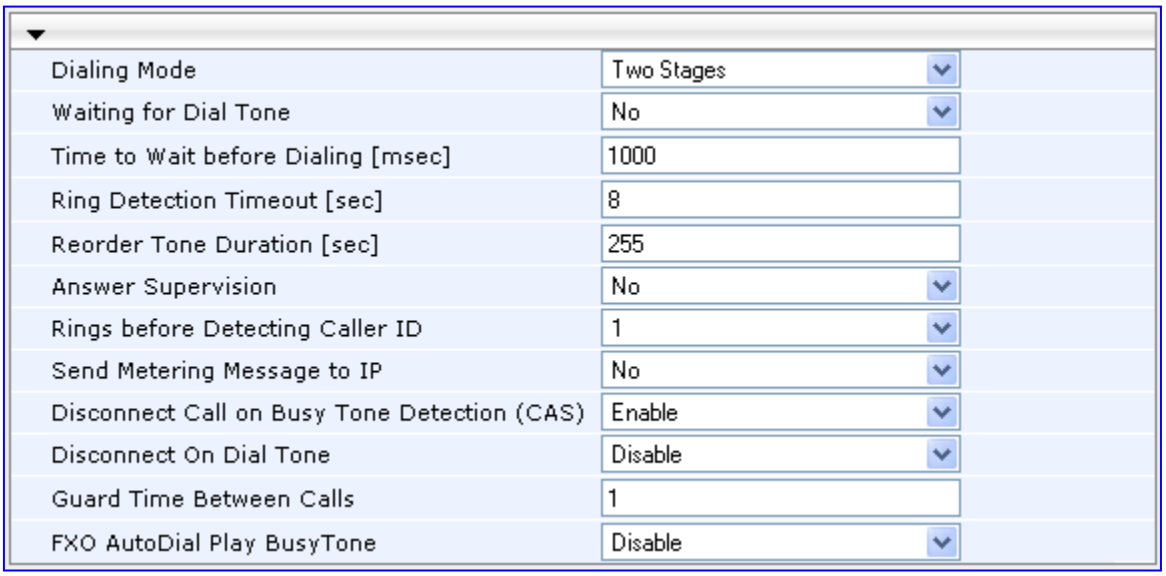
The FXO Settings page allows you to configure the device's specific FXO parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 752.

Note: The FXO Settings page is available only for FXO interfaces.

➢ **To configure the FXO parameters:**

1. Open the FXO Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **FXO Settings**).

Figure 19-36: FXO Settings Page



2. Configure the parameters as required.
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.7.5 Configuring Authentication

The Authentication page defines a user name and password for authenticating each device port. Authentication is typically used for FXS interfaces, but can also be used for FXO interfaces.

Notes:

- For configuring whether authentication is performed per port or for the entire device, use the parameter AuthenticationMode. If authentication is for the entire device, the configuration on this page is ignored.
- If either the user name or password fields are omitted, the port's phone number and global password (using the Password parameter) are used instead.
- After you click **Submit**, the password is displayed as an asterisk (*).
- You can also configure Authentication using the *ini* file table parameter Authentication (see 'Configuration Parameters Reference' on page 752).

➢ **To configure the Authentication Table:**

1. Set the parameter 'Authentication Mode' (AuthenticationMode) to **Per Endpoint.**
2. Open the Authentication page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Authentication**).

Figure 19-37: Authentication Page

Gateway Port	User Name	Password
Module 1 Port 1 FXS	user1	*******
Module 1 Port 2 FXS	user2	*******
Module 2 Port 1 FXO		
Module 2 Port 2 FXO		
Module 2 Port 3 FXO		
Module 2 Port 4 FXO		

3. In the 'User Name' and 'Password' fields corresponding to a port, enter the user name and password respectively.
4. Click Submit to apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.7.6 Configuring Automatic Dialing

The Automatic Dialing page allows you to define a telephone number that is automatically dialed when an FXS or FXO port is used (e.g., off-hooked).

Notes:

- After a ring signal is detected on an 'Enabled' FXO port, the device initiates a call to the destination number without seizing the line. The line is seized only after the call is answered.
- After a ring signal is detected on a 'Disabled' or 'Hotline' FXO port, the device seizes the line.
- You can also configure automatic dialing using the *ini* file table parameter TargetOfChannel.
- You can configure the device to play a Busy/Reorder tone to the Tel side upon receiving a SIP 4xx, 5xx, or 6xx response from the IP side (i.e., Tel-to-IP call failure), using the *ini* file parameter FXOAutoDialPlayBusyTone (see 'Configuration Parameters Reference' on page 752).

➢ **To configure Automatic Dialing:**

1. Open the Automatic Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Automatic Dialing**).

Figure 19-38: Automatic Dialing Page

Gateway Port	Destination Phone Number	Auto Dial Status
Module 1 Port 1 FXS	101	Enable
Module 1 Port 2 FXS	911	Hotline
Module 2 Port 1 FXO	302	Enable
Module 2 Port 2 FXO		Enable
Module 2 Port 3 FXO		Enable
Module 2 Port 4 FXO		Enable

2. In the 'Destination Phone Number' field corresponding to a port, enter the telephone number that you want automatically dialed.
3. From the 'Auto Dial Status' drop-down list, select one of the following:
   - **Disable [0]**: The automatic dialing feature for the specific port is disabled (i.e., the number in the 'Destination Phone Number' field is ignored).
   - **Enable [1]**: The number in the 'Destination Phone Number' field is automatically dialed if the phone is off-hooked (for FXS interfaces) or a ring signal (from PBX/PSTN switch) is detected (FXO interfaces). The FXO line is seized only after the SIP call is answered.
   - **Hotline [2]**:
     - **FXS interfaces:** When a phone is off-hooked and no digit is dialed for a user-defined time (configured using the parameter HotLineToneDuration), the number in the 'Destination Phone Number' field is automatically dialed.

- **FXO interfaces:** If a ring signal is detected, the device seizes the FXO line, plays a dial tone, and then waits for DTMF digits. If no digits are detected for a user-defined time (configured using the parameter HotLineToneDuration), the number in the 'Destination Phone Number' field is automatically dialed by sending a SIP INVITE message with this number.

4. Click Submit to apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.7.7 Configuring Caller Display Information

The Caller Display Information page allows you to define a caller identification string (Caller ID) for FXS and FXO ports and enable the device to send the Caller ID information to IP when a call is made. The called party can use this information for caller identification. The information configured on this page is sent in an INVITE message in the From header. For information on Caller ID restriction according to destination/source prefixes, see 'Configuring Number Manipulation Tables' on page 277.

➢ **To configure the Caller Display Information:**

1. Open the Caller Display Information page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Caller Display Information**).

Figure 19-39: Caller Display Information Page

Gateway Port	Caller ID/Name	Presentation
Module 1 Port 1 FXS	Private	Restricted
Module 1 Port 2 FXS	Susan C.	Restricted
Module 2 Port 1 FXO	Lee P.	Allowed
Module 2 Port 2 FXO	Ronaldo	Allowed
Module 2 Port 3 FXO		Allowed
Module 2 Port 4 FXO		Allowed

2. In the 'Caller ID/Name' field corresponding to the desired port, enter the Caller ID string (up to 18 characters).
3. From the 'Presentation' drop-down list, select one of the following:
   - **Allowed [0]** - sends the string defined in the 'Caller ID/Name' field when a Tel-to-IP call is made using the corresponding device port.
   - **Restricted [1]** - the string defined in the 'Caller ID/Name' field is not sent.
4. Click Submit to apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Notes:

- When FXS ports receive 'Private' or 'Anonymous' strings in the From header, they don't send the calling name or number to the Caller ID display.
- If Caller ID name is detected on an FXO line (EnableCallerID = 1), it is used instead of the Caller ID name defined on this page.
- When the 'Presentation' field is set to 'Restricted', the Caller ID is sent to the remote side using only the P-Asserted-Identity and P-Preferred-Identity headers (AssertedIdMode).
- The value of the 'Presentation' field can be overridden by configuring the 'Presentation' field in the Source Number Manipulation table (see 'Configuring Number Manipulation Tables' on page 277).
- You can also configure the Caller Display Information table using the *ini* file table parameter CallerDisplayInfo.

19.7.8 Configuring Call Forward

The Call Forwarding Table page allows you to forward (redirect) IP-to-Tel calls (using SIP 302 response) originally destined to specific device ports, to other device ports or to an IP destination.

Notes:

- Ensure that the Call Forward feature is enabled (default) for the settings on this page to take effect. To enable Call Forward, use the parameter EnableForward ('Configuring Supplementary Services' on page 310).
- You can also configure the Call Forward table using the *ini* file table parameter FwdInfo.

➢ **To configure Call Forward per port:**

1. Open the Call Forward Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Call Forward**).

Figure 19-40: Call Forward Table Page

Gateway Port	Forward Type	Forward to Phone Number	Time for No Reply Forward
Module 1 Port 1 FXS	On busy	201	30
Module 1 Port 2 FXS	Unconditional	202@10.2.1.1	30
Module 2 Port 1 FXO	No Answer	203	30
Module 2 Port 2 FXO	Deactivate		30
Module 2 Port 3 FXO	Deactivate		30
Module 2 Port 4 FXO	Deactivate		30

2. Configure the Call Forward parameters for each port according to the table below.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 19-21: Call Forward Table

Parameter	Description
Forward Type	Determines the scenario for forwarding a call. ▪ **[0]** Deactivate = Don't forward incoming calls (default). ▪ **[1]** On Busy = Forward incoming calls when the port is busy. ▪ **[2]** Unconditional = Always forward incoming calls. ▪ **[3]** No Answer = Forward incoming calls that are not answered within the time specified in the 'Time for No Reply Forward' field. ▪ **[4]** On Busy or No Answer = Forward incoming calls when the port is busy or when calls are not answered within the time specified in the 'Time for No Reply Forward' field. ▪ **[5]** Do Not Disturb = Immediately reject incoming calls.
Forward to Phone Number	The telephone number or URI (<number>@<IP address>) to where the call is forwarded. **Note:** If this field only contains a telephone number and a Proxy isn't used, the 'forward to' phone number must be specified in the Outbound IP Routing Table (see 'Configuring Outbound IP Routing Table' on page 291).
Time for No Reply Forward	If you have set the 'Forward Type' for this port to 'No Answer', enter the number of seconds the device waits before forwarding the call to the phone number specified.

19.7.9 Configuring Caller ID Permissions

The Caller ID Permissions page allows you to enable or disable (per port) the Caller ID generation (for FXS interfaces) and detection (for FXO interfaces). If a port isn't configured, its Caller ID generation / detection is determined according to the global parameter EnableCallerID described in 'Configuring Supplementary Services' on page 310.

Note: You can also configure the Caller ID Permissions table using the *ini* file table parameter EnableCallerID.

➢ **To configure Caller ID Permissions per port:**

1. Open the Caller ID Permissions page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Caller ID Permissions**).

Figure 19-41: Caller ID Permissions Page

Gateway Port	Caller ID
Module 1 Port 1 FXS	Enable
Module 1 Port 2 FXS	Disable
Module 2 Port 1 FXO	
Module 2 Port 2 FXO	
Module 2 Port 3 FXO	
Module 2 Port 4 FXO	

2. From the 'Caller ID' drop-down list, select one of the following:
   - **Enable:** Enables Caller ID generation (FXS) or detection (FXO) for the specific port.
   - **Disable:** Caller ID generation (FXS) or detection (FXO) for the specific port is disabled.
   - **Not defined:** Caller ID generation (FXS) or detection (FXO) for the specific port is determined according to the parameter 'Enable Caller ID' (described in 'Configuring Supplementary Services' on page 310).
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.7.10 Configuring Call Waiting

The Call Waiting page allows you to enable or disable call waiting per device FXS port**.**

Notes:

- This page is applicable only to FXS interfaces.
- Instead of using this page, you can enable or disable call waiting for all the device's ports, using the global call waiting parameter 'Enable Call Waiting' (see 'Configuring Supplementary Services' on page 310).
- You can also configure the Call Waiting table using the *ini* file table parameter CallWaitingPerPort (see 'Configuration Parameters Reference' on page 752).
- For additional call waiting configuration, see the following parameters: FirstCallWaitingToneID (in the CPT file), TimeBeforeWaitingIndication, WaitingBeepDuration, TimeBetweenWaitingIndications, and NumberOfWaitingIndications.

➢ **To configure Call Waiting:**

1. Open the Call Waiting page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Call Waiting**).

Figure 19-42: Call Waiting Page

Gateway Port	Call Waiting Configuration
Module 1 Port 1 FXS	Enable
Module 1 Port 2 FXS	Enable
Module 2 Port 1 FXO	
Module 2 Port 2 FXO	
Module 2 Port 3 FXO	
Module 2 Port 4 FXO	

2. From the 'Call Waiting Configuration' drop-down list corresponding to the port you want to configure for call waiting, select one of the following options:
   - **Enable:** Enables call waiting for the specific port. When the device receives a call on a busy endpoint (port), it responds with a 182 response (not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected by the device, the device switches to the waiting call. The device that initiated the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received.
   - **Disable:** No call waiting for the specific port.
   - **Empty:** Call waiting is determined according to the global parameter 'Enable Call Waiting' (described in 'Configuring Supplementary Services' on page 310).
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

19.7.11 Configuring FXS Distinctive Ringing and Call Waiting Tones per Source/Destination Number

You can configure a Distinctive Ringing tone and Call Waiting tone per calling (source) and/or called (destination) number (or prefix) for IP-to-Tel calls. This feature can be configured per FXS endpoint or for a range of FXS endpoints. Therefore, different tones can be played per FXS endpoint/s depending on the source and/or destination number of the received call. In addition, you can configure multiple entries with different source and/or destination prefixes and tones for the same FXS port.

Typically, the played Ringing and/or Call Waiting tone is indicated in the SIP Alert-info header field of the received INVITE message. If this header is not present in the received INVITE, then this feature is used and the tone played is according to the settings in this table.

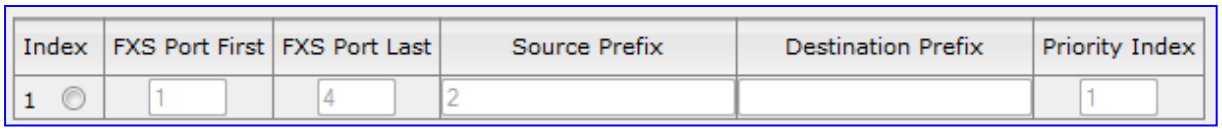
Notes:

- This page is applicable only to FXS interfaces.
- You can also configure the Tone Index table using the *ini* file table parameter ToneIndex.

➢ **To configure distinctive ringing and call waiting per FXS port:**

1. Open the Tone Index Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Tone Index**).

Figure 19-43: Tone Index Table Page



The figure above shows a configuration example for using Distinctive Ringing and Call Waiting tones of Index #9 in the CPT file for FXS endpoints 1 to 4 when a call is received from a source number with prefix 2.

2. In the 'Add' field, enter a table index number and then click **Add**.
3. Configure the table according to the table below.
4. Click Submit to apply your changes.

Table 19-22: Tone index Table Parameter Description

Parameter	Description
Index	Defines the table index entry. Up to 50 entries can be defined.
FXS Port First	Defines the starting range of FXS ports, where 1 is the first port.
FXS Port Last	Defines the end range of FXS ports.
Source Prefix	Defines the prefix of the calling number.
Destination Prefix	Defines the prefix of the called number.
Priority Index	Defines the index of the Distinctive Ringing and Call Waiting tones (default is 0). The Call Waiting tone index equals to the Priority Index plus the value of the FirstCallWaitingToneID parameter. For example, if you want to use the Call Waiting tone in the CPT file at Index #9, you need to enter "1" as the Priority Index value and set the FirstCallWaitingToneID parameter to "8". The summation of these values is 9, i.e., index #9.

19.7.12 FXS/FXO Coefficient Types

The FXS Coefficient and FXO Coefficient types used by the device can be one of the following:

- US line type of 600 ohm AC impedance and 40 V RMS ringing voltage for REN = 2
- European standard (TBR21)

These types can be selected using the *ini* file parameters FXSCountryCoefficients (for FXS) and CountryCoefficients (for FXO), or using the Web interface (see 'Configuring Analog Settings' on page 178).

These Coefficient types are used to increase return loss and trans-hybrid loss performance for two telephony line type interfaces (US or European). This adaptation is performed by modifying the telephony interface characteristics. This means, for example, that changing impedance matching or hybrid balance doesn't require hardware modifications, so that a single device is able to meet requirements for different markets. The digital design of the filters and gain stages also ensures high reliability, no drifts (over temperature or time) and simple variations between different line types.

The FXS Coefficient types provide best termination and transmission quality adaptation for two FXS line type interfaces. This parameter affects the following AC and DC interface parameters:

- DC (battery) feed characteristics
- AC impedance matching
- Transmit gain
- Receive gain
- Hybrid balance
- Frequency response in transmit and receive direction
- Hook thresholds
- Ringing generation and detection parameters

19.7.13 FXO Operating Modes

This section provides a description of the device's FXO operating modes:

- For IP-to-Tel calls (see 'FXO Operations for IP-to-Tel Calls' on page 363)
- For Tel-to-IP calls (see 'FXO Operations for Tel-to-IP Calls' on page 366)
- Call termination on FXO devices (see 'Call Termination on FXO Devices' on page 368)

19.7.13.1 FXO Operations for IP-to-Tel Calls

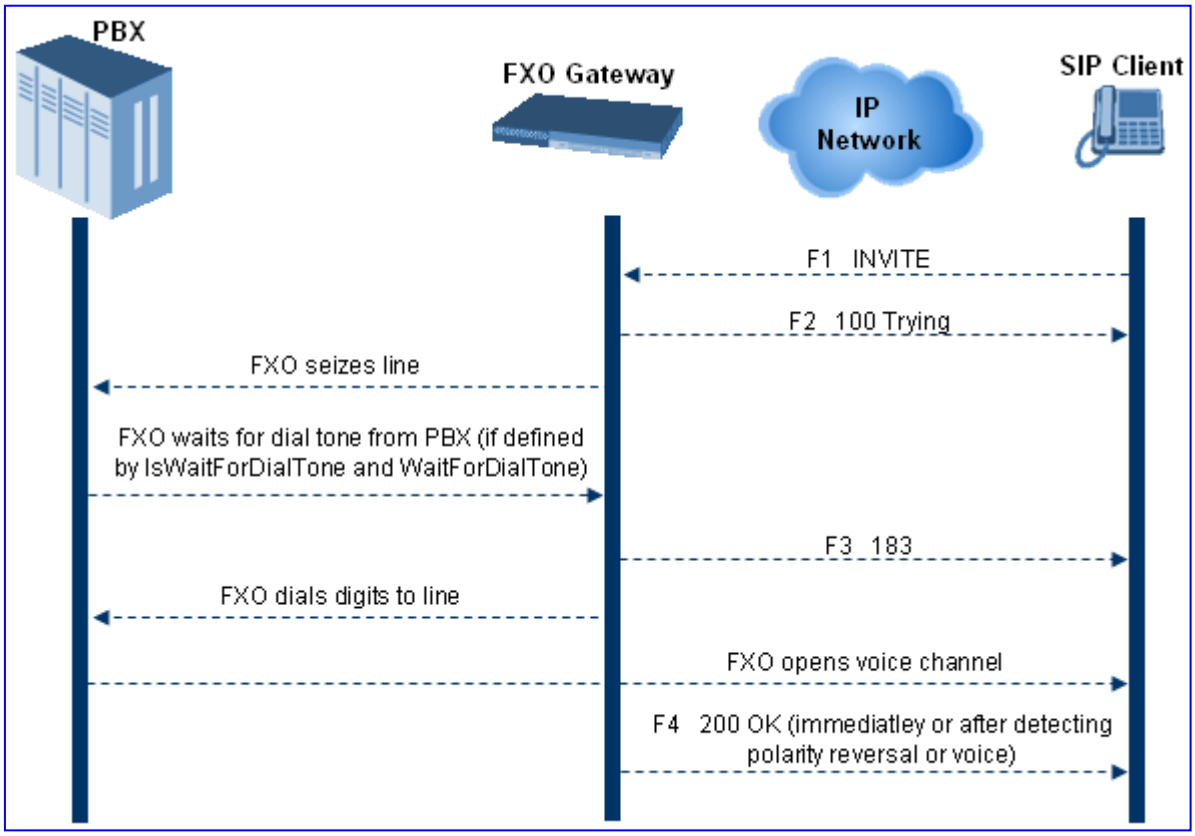
The FXO device provides the following operating modes for IP-to-Tel calls:

- One-stage dialing (see 'One-Stage Dialing' on page 364)
  - Waiting for dial tone (see 'Two-Stage Dialing' on page 365)
  - Time to wait before dialing
  - Answer supervision
- Two-stage dialing (see 'Two-Stage Dialing' on page 365)
- Dialing time: DID wink (see 'DID Wink' on page 365)

19.7.13.1.1 One-Stage Dialing

One-stage dialing is when the FXO device receives an IP-to-Tel call, off-hooks the PBX line connected to the telephone, and then immediately dials the destination telephone number. In other words, the IP caller doesn't dial the PSTN number upon hearing a dial tone.

Figure 19-44: CAS State Machine Page



One-stage dialing incorporates the following FXO functionality:

- **Waiting for Dial Tone:** Enables the device to dial the digits to the Tel side only after detecting a dial tone from the PBX line. The *ini* file parameter IsWaitForDialTone is used to configure this operation.
- **Time to Wait Before Dialing:** Defines the time (in msec) between seizing the FXO line and starting to dial the digits. The *ini* file parameter WaitForDialTime is used to configure this operation.

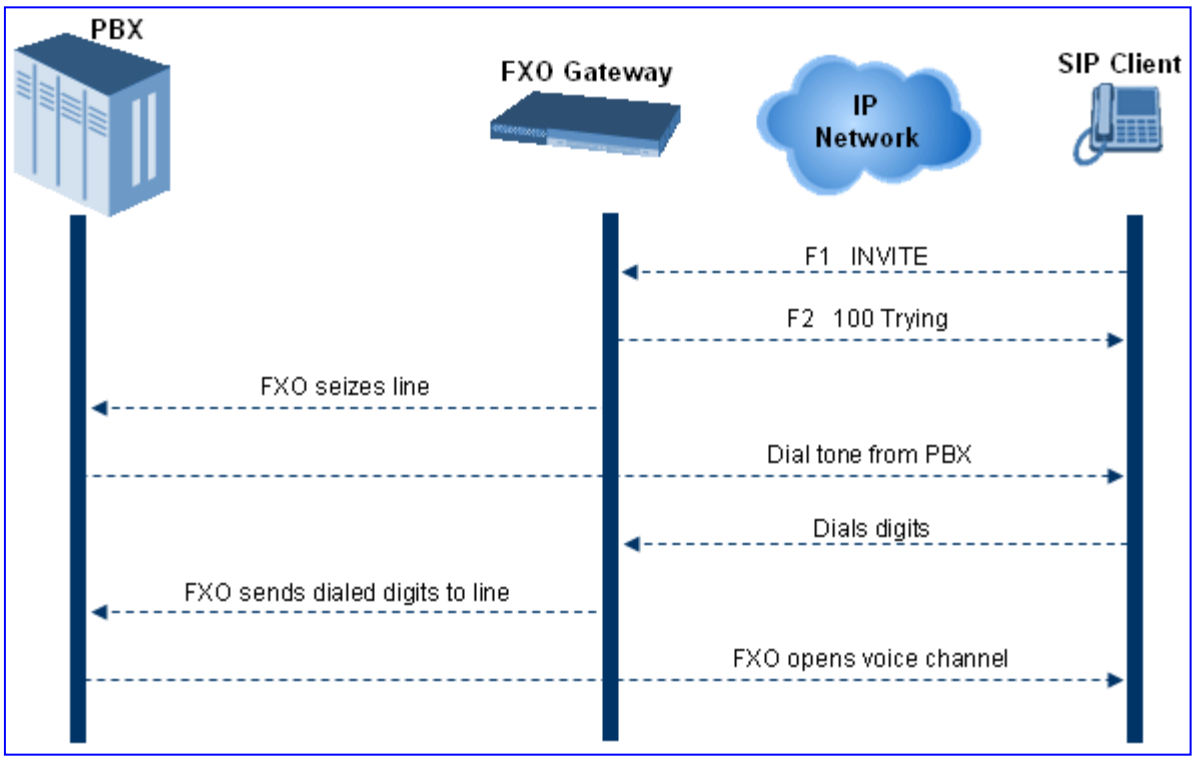
Note: The *ini* file parameter IsWaitForDialTone must be disabled for this mode.

- **Answer Supervision:** The Answer Supervision feature enables the FXO device to determine when a call is connected, by using one of the following methods:
  - **Polarity Reversal:** the device sends a 200 OK in response to an INVITE only when it detects a polarity reversal.
  - **Voice Detection:** the device sends a 200 OK in response to an INVITE only when it detects the start of speech (or ringback tone) from the Tel side. (Note that the IPM detectors must be enabled).

19.7.13.1.2 Two-Stage Dialing

Two-stage dialing is when the IP caller is required to dial twice. The caller initially dials to the FXO device and only after receiving a dial tone from the PBX (via the FXO device), dials the destination telephone number.

Figure 19-45: Call Flow for Two-Stage Dialing



Two-stage dialing implements the Dialing Time feature. Dialing Time allows you to define the time that each digit can be separately dialed. By default, the overall dialing time per digit is 200 msec. The longer the telephone number, the greater the dialing time.

The relevant parameters for configuring Dialing Time include the following:

- DTMFDigitLength (100 msec): time for generating DTMF tones to the PSTN (PBX) side
- DTMFInterDigitInterval (100 msec): time between generated DTMF digits to PSTN (PBX) side

19.7.13.1.3 DID Wink

The device's FXO ports support Direct Inward Dialing (DID). DID is a service offered by telephone companies that enables callers to dial directly to an extension on a PBX without the assistance of an operator or automated call attendant. This service makes use of DID trunks, which forward only the last three to five digits of a phone number to the PBX. If, for example, a company has a PBX with extensions 555-1000 to 555-1999, and a caller dials 555-1234, the local central office (CO) would forward, for example, only 234 to the PBX. The PBX would then ring extension 234.

DID wink enables the originating end to seize the line by going off-hook. It waits for acknowledgement from the other end before sending digits. This serves as an integrity check that identifies a malfunctioning trunk and allows the network to send a re-order tone to the calling party.

The "start dial" signal is a wink from the PBX to the FXO device. The FXO then sends the last four to five DTMF digits of the called number. The PBX uses these digits to complete the routing directly to an internal station (telephone or equivalent)

- DID Wink can be used for connection to EIA/TIA-464B DID Loop Start lines
- Both FXO (detection) and FXS (generation) are supported

19.7.13.2 FXO Operations for Tel-to-IP Calls

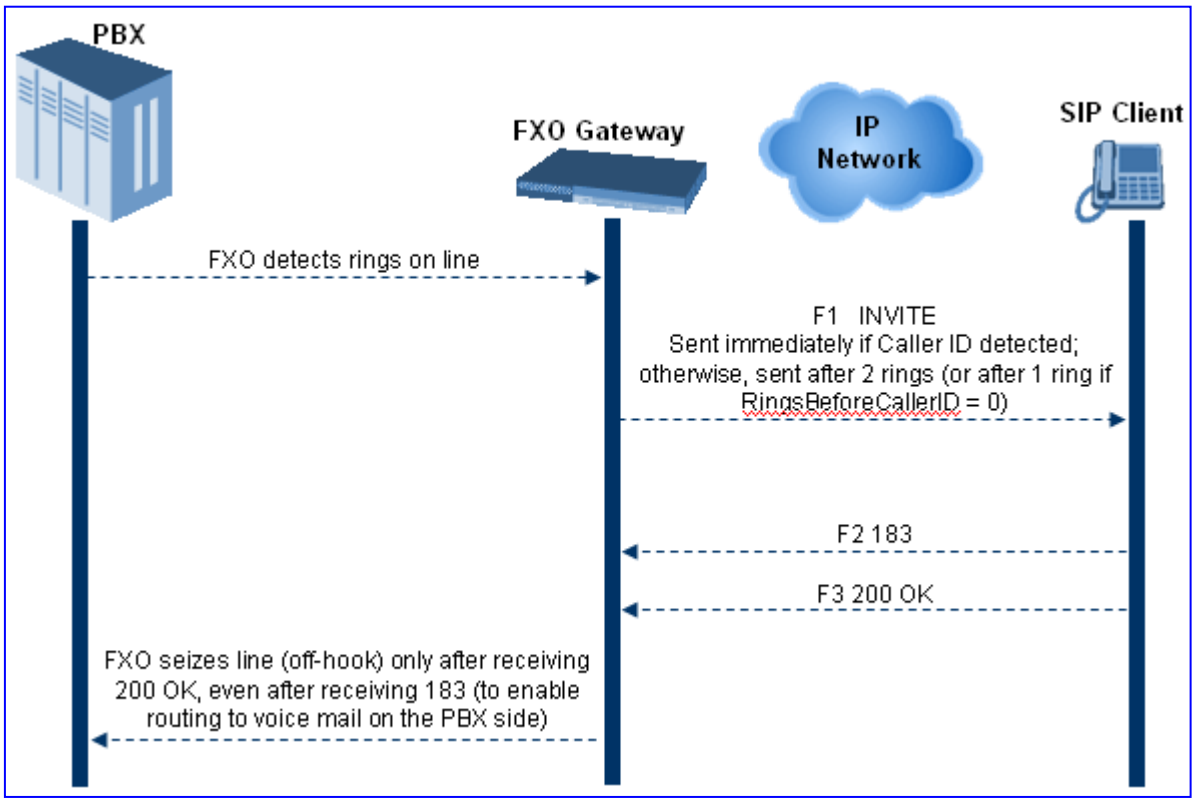
The FXO device provides the following FXO operating modes for Tel-to-IP calls:

- Automatic Dialing (see 'Automatic Dialing' on page 366)
- Collecting Digits Mode (see 'Collecting Digits Mode' on page 367)
- FXO Supplementary Services (see 'FXO Supplementary Services' on page 367)
  - Hold/Transfer Toward the Tel side
  - Hold/Transfer Toward the IP side
  - Blind Transfer to the Tel side

19.7.13.2.1 Automatic Dialing

Automatic dialing is defined using the Web interface's Automatic Dialing (TargetOfChannel ini file parameter) page, described in see 'Configuring Automatic Dialing' on page 356.

The SIP call flow diagram below illustrates Automatic Dialing.

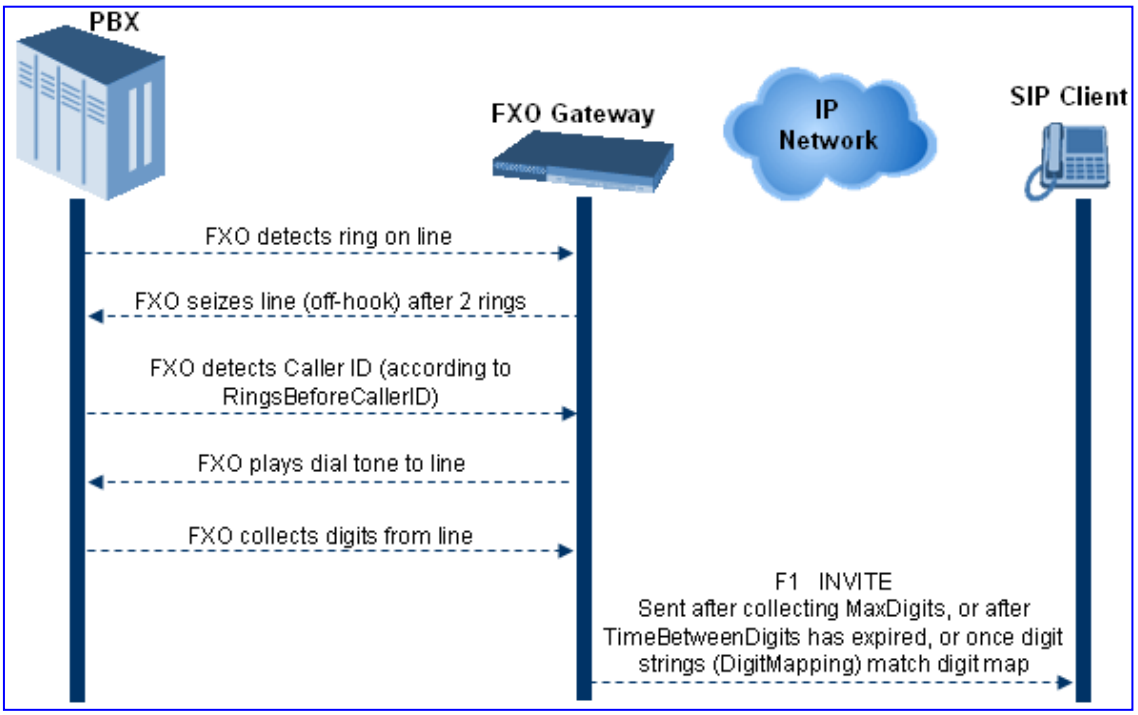


19.7.13.2.2 Collecting Digits Mode

When automatic dialing is not defined, the device collects the digits.

The SIP call flow diagram below illustrates the Collecting Digits Mode.

Figure 19-46: Call Flow for Collecting Digits Mode



19.7.13.2.3 FXO Supplementary Services

The FXO supplementary services include the following:

- **Hold / Transfer toward the Tel side:** The *ini* file parameter LineTransferMode must be set to 0 (default). If the FXO receives a hook-flash from the IP side (using out-of-band or RFC 2833), the device sends the hook-flash to the Tel side by performing one of the following:
  - Performing a hook flash (i.e., on-hook and off-hook)
  - Sending a hook-flash code (defined by the *ini* file parameter HookFlashCode)

  The PBX may generate a dial tone that is sent to the IP, and the IP side may dial digits of a new destination.
- **Blind Transfer to the Tel side:** A blind transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. The *ini* file parameter LineTransferMode must be set to 1.

  The blind transfer call process is as follows:
  - FXO receives a REFER request from the IP side
  - FXO sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then drops the line (on-hook). Note that the time between flash to dial is according to the WaitForDialTime parameter.
  - PBX performs the transfer internally
- **Hold / Transfer toward the IP side:** The FXO device doesn't initiate hold / transfer as a response to input from the Tel side. If the FXO receives a REFER request (with or without replaces), it generates a new INVITE according to the Refer-To header.

19.7.13.3 Call Termination on FXO Devices

This section describes the device's call termination capabilities for its FXO interfaces:

- Calls terminated by a PBX (see 'Call Termination by PBX' on page 368)
- Calls terminated before call establishment (see 'Call Termination before Call Establishment' on page 369)
- Ring detection timeout (see 'Ring Detection Timeout' on page 369)

19.7.13.3.1 Calls Termination by PBX

The FXO device supports various methods for identifying when a call has been terminated by the PBX.

The PBX doesn't disconnect calls, but instead signals to the device that the call has been disconnected using one of the following methods:

- **Detection of polarity reversal/current disconnect:** The call is immediately disconnected after polarity reversal or current disconnect is detected on the Tel side (assuming the PBX/CO generates this signal). This is the recommended method.

  Relevant parameters: EnableReversalPolarity, EnableCurrentDisconnect, CurrentDisconnectDuration, CurrentDisconnectDefaultThreshold, and TimeToSampleAnalogLineVoltage.

- **Detection of Reorder, Busy, Dial, and Special Information Tone (SIT) tones:** The call is immediately disconnected after a Reorder, Busy, Dial, or SIT tone is detected on the Tel side (assuming the PBX / CO generates this tone). This method requires the correct tone frequencies and cadence to be defined in the Call Progress Tones file. If these frequencies are not known, define them in the CPT file (the tone produced by the PBX / CO must be recorded and its frequencies analyzed -- refer to Adding a Reorder Tone to the CPT File in the Reference Manual). This method is slightly less reliable than the previous one. You can use the CPTWizard (described in the *Reference Manual*) to analyze Call Progress Tones generated by any PBX or telephone network.

  Relevant parameters: DisconnectOnBusyTone and DisconnectOnDialTone.

- **Detection of silence:** The call is disconnected after silence is detected on both call directions for a specific (configurable) amount of time. The call isn't disconnected immediately; therefore, this method should only be used as a backup option.

  Relevant parameters: EnableSilenceDisconnect and FarEndDisconnectSilencePeriod.

- **Special DTMF code:** A digit pattern that when received from the Tel side, indicates to the device to disconnect the call.

  Relevant *ini* file parameter: TelDisconnectCode.

- **Interruption of RTP stream:** Relevant parameters: BrokenConnectionEventTimeout and DisconnectOnBrokenConnection.

Note: This method operates correctly only if silence suppression is not used.

- **Protocol-based termination of the call from the IP side**

Note: The implemented disconnect method must be supported by the CO or PBX.

19.7.13.3.2 Call Termination before Call Establishment

The device supports the following call termination methods before a call is established:

- **Call termination upon receipt of SIP error response (in Automatic Dialing mode):** By default, when the FXO device operates in Automatic Dialing mode, there is no method to inform the PBX if a Tel-to-IP call has failed (SIP error response - 4xx, 5xx or 6xx - is received). The reason is that the FXO device does not seize the line until a SIP 200 OK response is received. Use the FXOAutoDialPlayBusyTone parameter to allow the device to play a Busy/Reorder tone to the PSTN line if a SIP error response is received. The FXO device seizes the line (off-hook) for the duration defined by the TimeForReorderTone parameter. After playing the tone, the line is released (on-hook).
- **Call termination after caller (PBX) on-hooks phone (Ring Detection Timeout feature):** This method operates in one of the following manners:
  - **Automatic Dialing is enabled:** if the remote IP party doesn't answer the call and the ringing signal (from the PBX) stops for a user-defined time (configured by the parameter FXOBetweenRingTime), the FXO device releases the IP call.
  - **No automatic dialing and Caller ID is enabled:** the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP.

19.7.13.3.3 Ring Detection Timeout

The operation of Ring Detection Timeout depends on the following:

- **Automatic dialing is disabled and Caller ID is enabled:** if the second ring signal is not received for a user-defined time (using the parameter FXOBetweenRingTime), the FXO device doesn't initiate a call to the IP.
- **Automatic dialing is enabled:** if the remote party doesn't answer the call and the ringing signal stops for a user-defined time (using the parameter FXOBetweenRingTime), the FXO device releases the IP call.

Ring Detection Timeout supports full ring cycle of ring on and ring off (from ring start to ring start).

19.7.14 Remote PBX Extension between FXO and FXS Devices

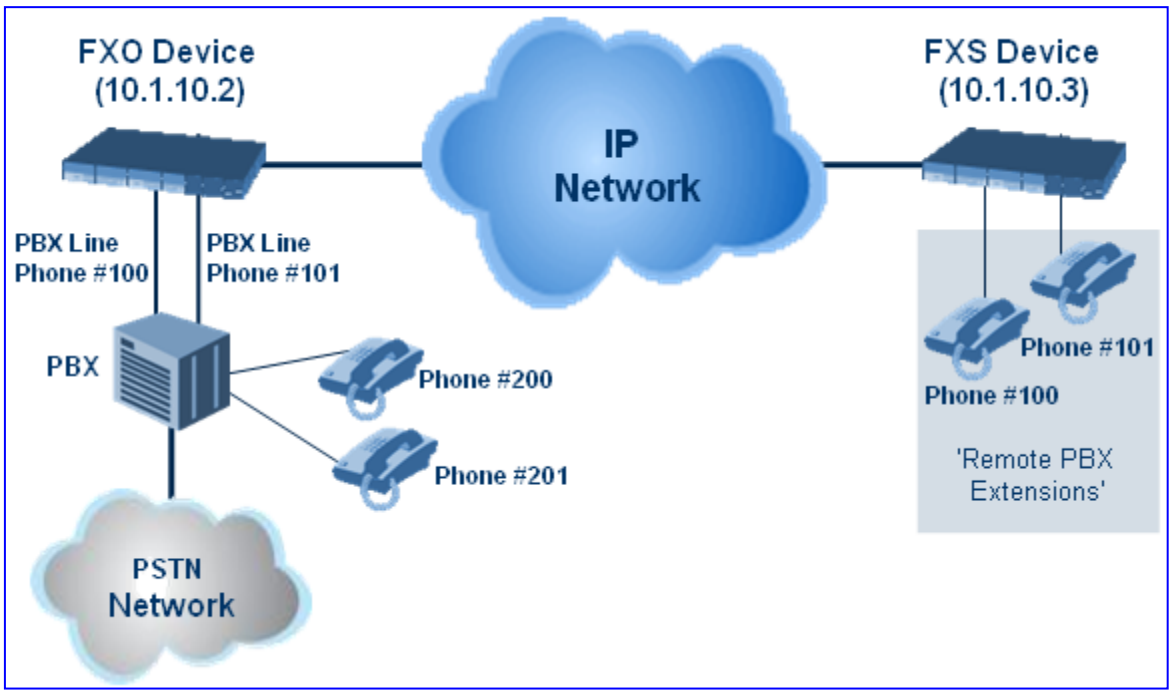
Remote PBX extension offers a company the capability of extending the "power" of its local PBX by allowing remote phones (remote offices) to connect to the company's PBX over the IP network (instead of via PSTN). This is as if the remote office is located in the head office (where the PBX is installed). PBX extensions are connected through FXO ports to the IP network, instead of being connected to individual telephone stations. At the remote office, FXS units connect analog phones to the same IP network. To produce full transparency, each FXO port is mapped to an FXS port (i.e., one-to-one mapping). This allows individual extensions to be extended to remote locations. To call a remote office worker, a PBX user or a PSTN caller simply dials the PBX extension that is mapped to the remote FXS port.

This section provides an example on how to implement a remote telephone extension through the IP network, using FXO andFXS interfaces. In this configuration, the FXO device routes calls received from the PBX to the 'Remote PBX Extension' connected to the FXS device. The routing is transparent as if the telephone connected to the FXS device is directly connected to the PBX.

The following is required:

- FXO interfaces with ports connected directly to the PBX lines (shown in the figure below)
- FXS interfaces for the 'remote PBX extension'

- Analog phones (POTS)
- PBX (one or more PBX loop start lines)
- LAN network



19.7.14.1 Dialing from Remote Extension (Phone at FXS)

The procedure below describes how to dial from the 'remote PBX extension' (i.e., phone connected to the FXS interface).

➢ **To make a call from the FXS interface:**

1. Off-hook the phone and wait for the dial tone from the PBX. This is as if the phone is connected directly to the PBX. The FXS and FXO interfaces establish a voice path connection from the phone to the PBX immediately after the phone is off-hooked.
2. Dial the destination number (e.g., phone number 201). The DTMF digits are sent over IP directly to the PBX. All the audible tones are generated from the PBX (such as ringback, busy, or fast busy tones). One-to-one mapping occurs between the FXS ports and PBX lines.
3. The call disconnects when the phone connected to the FXS goes on-hook.

19.7.14.2 Dialing from PBX Line or PSTN

The procedure below describes how to dial from a PBX line (i.e., from a telephone directly connected to the PBX) or from the PSTN to the 'remote PBX extension' (i.e., telephone connected to the FXS interface).

➢ **To dial from a telephone directly connected to the PBX or from the PSTN:**

- Dial the PBX subscriber number (e.g., phone number 101) in the same way as if the user's phone was connected directly to the PBX. As soon as the PBX rings the FXO device, the ring signal is 'sent' to the phone connected to the FXS device. Once the phone connected to the FXS device is off-hooked, the FXO device seizes the PBX line and the voice path is established between the phone and PBX.

There is one-to-one mapping between PBX lines and FXS device ports. Each PBX line is routed to the same phone (connected to the FXS device). The call disconnects when the phone connected to the FXS device is on-hooked.

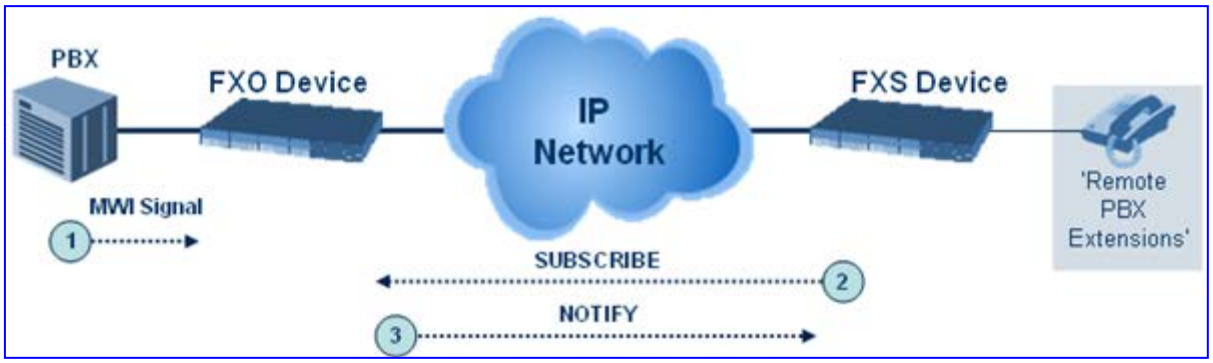
19.7.14.3 Message Waiting Indication for Remote Extensions

The device supports the relaying of Message Waiting Indications (MWI) for remote extensions (and voice mail applications). Instead of subscribing to an MWI server to receive notifications of pending messages, the FXO device receives subscriptions from the remote FXS device and notifies the appropriate extension when messages (and the number of messages) are pending.

The FXO device detects an MWI message from the Tel (PBX) side using any one of the following methods:

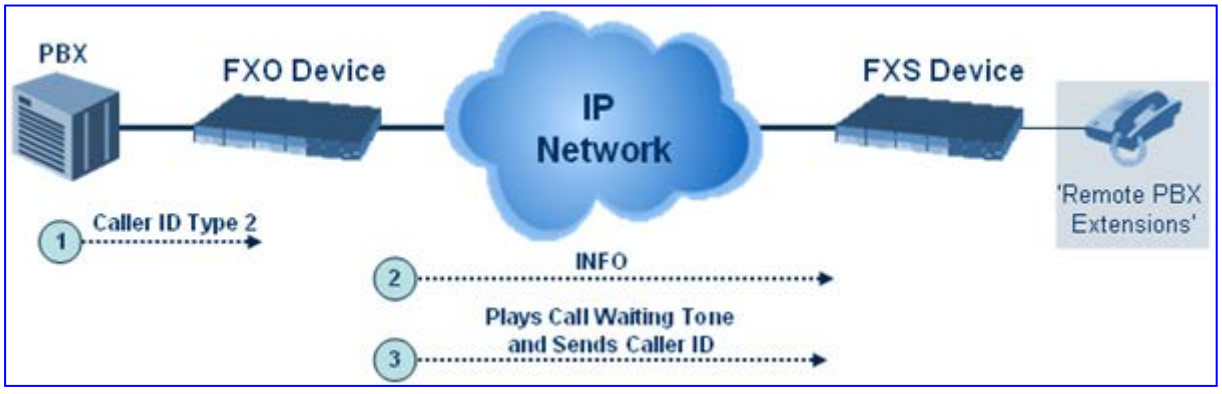
- 100 VDC (sent by the PBX to activate the phone's lamp)
- Stutter dial tone from the PBX
- MWI display signal (according to the parameter CallerIDType)

Upon detection of an MWI message, the FXO device sends a SIP NOTIFY message to the IP side. When receiving this NOTIFY message, the remote FXS device generates an MWI signal toward its Tel side.



19.7.14.4 Call Waiting for Remote Extensions

When the FXO device detects a Call Waiting indication (FSK data of the Caller Id - CallerIDType2) from the PBX, it sends a proprietary INFO message, which includes the caller identification to the FXS device. Once the FXS device receives this INFO message, it plays a call waiting tone and sends the caller ID to the relevant port for display. The remote extension connected to the FXS device can toggle between calls using the Hook Flash button.



19.7.14.5 FXS Gateway Configuration

The procedure below describes how to configure the FXS interface (at the 'remote PBX extension').

➢ **To configure the FXS interface:**

1. In the Trunk Group Table page (see , assign the phone numbers 100 to 104 to the device's endpoints.

Figure 19-47: Assigning Phone Numbers to FXS Endpoints

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	100	0

2. In the 'Automatic Dialing page (see 'Configuring Automatic Dialing' on page 356), enter the phone numbers of the FXO device in the 'Destination Phone Number' fields. When a phone connected to Port #1 off-hooks, the FXS device automatically dials the number '200'.

Figure 19-48: Automatic Dialing for FXS Ports

Gateway Port	Destination Phone Number	Auto Dial Status
Module 3 Port 1 FXS	200	Enable
Module 3 Port 2 FXS	201	Enable
Module 3 Port 3 FXS	202	Enable
Module 3 Port 4 FXS	203	Enable

3. In the Outbound IP Routing Table page (see 'Configuring Outbound IP Routing Table' on page 291), enter 20 for the destination phone prefix, and 10.1.10.2 for the IP address of the FXO device.

Figure 19-49: FXS Tel-to-IP Routing Configuration

	Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address
1		20	*		10.1.10.2

Note: For the transfer to function in remote PBX extensions, Hold must be disabled at the FXS device (i.e., Enable Hold = 0) and hook-flash must be transferred from the FXS to the FXO (HookFlashOption = 4).

19.7.14.6 FXO Gateway Configuration

The procedure below describes how to configure the FXO interface (to which the PBX is directly connected).

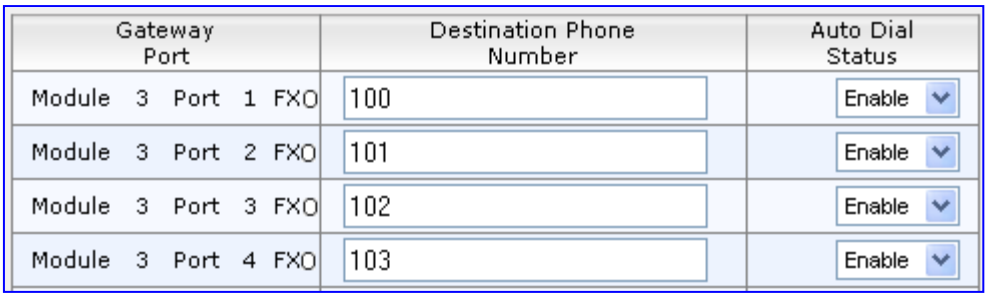
➢ **To configure the FXO interface:**

1. In the Trunk Group Table page (see , assign the phone numbers 200 to 204 to the device's FXO endpoints.

Figure 19-50: Assigning Phone Numbers to FXO Ports

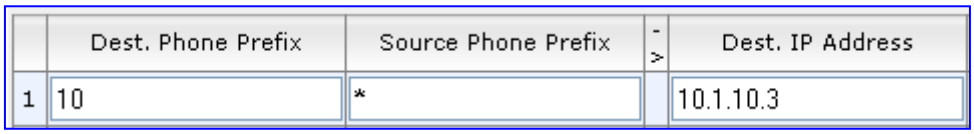
2. In the 'Automatic Dialing page, enter the phone numbers of the FXS device in the 'Destination Phone Number' fields. When a ringing signal is detected at Port #1, the FXO device automatically dials the number '100'.

Figure 19-51: FXO Automatic Dialing Configuration



3. In the Outbound IP Routing Table page, enter 10 in the 'Destination Phone Prefix' field, and the IP address of the FXS device (10.1.10.3) in the field 'IP Address'.

Figure 19-52: FXO Tel-to-IP Routing Configuration



4. In the 'FXO Settings page (see 'Configuring FXO Parameters' on page 354), set the parameter 'Dialing Mode' to 'Two Stages' (IsTwoStageDial = 1).

19.8 Dialing Plan Features

This section discusses various dialing plan features supported by the device:

- Dialing plan notations (see 'Dialing Plan Notation for Routing and Manipulation' on page 374)
- Digit mapping (see 'Digit Mapping' on page 375)
- External Dial Plan file containing dial plans (see 'External Dial Plan File' on page 377)
- Dial plan prefix tags for enhanced IP-to-Tel routing (see Dial Plan Prefix Tags for IP-to-Tel Routing on page 379)

19.8.1 Dialing Plan Notation for Routing and Manipulation

The device supports flexible dialing plan notations for denoting the prefix and/or suffix source and/or destination phone numbers and SIP URI user names in the routing (see 'Configuring Outbound IP Routing Table' on page 291 and 'Configuring Inbound IP Routing Table' on page 299) and manipulation tables (see 'Configuring Number Manipulation Tables' on page 277).

Table 19-23: Dialing Plan Notations for Prefixes and Suffixes

Notation	Description
x (letter "x")	Denotes any single digit.
# (pound symbol)	When used at the end of a prefix, it denotes the end of a number. For example, **54324xx#** represents a 7-digit number that starts with the digits 54324. When used anywhere in the suffix, it is part of the number. For example, **(3#45)** can represent the number string, 123#45.
***** (asterisk symbol)	When used in the prefix, it denotes any number. When used in the suffix, it is part of the number. For example, **(3*45)** can represent the number string, 123*45.
Range of Digits **Notes:** ▪ Dial plans denoting a prefix that is a range must be enclosed in square brackets, e.g., **[4-8]** or **23xx[456]**. ▪ Dial plans denoting a prefix that is not a range is not enclosed, e.g., **12345#**. ▪ Dial plans denoting a suffix must be enclosed in parenthesis, e.g., **(4)** and **(4-8)**. ▪ Dial plans denoting a suffix that include multiple ranges, the range must be enclosed in square brackets, e.g., **(23xx[4,5,6])**. ▪ An example for entering a combined prefix and suffix dial plan - assume you want to match a rule whose destination phone prefix is 4 to 8, and suffix is 234, 235, or 236. The entered value would be the following: **[4-8](23[4,5,6])**.	
[n-m] or **(n-m)**	Represents a range of numbers. For example: ▪ To denote numbers from 5551200 to 5551300: ✓ Prefix: **[5551200-5551300]#** ✓ Suffix: **(5551200-5551300)** ▪ To denote numbers from 123100 to 123200: ✓ Prefix: **123[100-200]** ✓ Suffix: **(123[100-200])** ▪ To denote prefix and suffix numbers together: ✓ 03(100): for any number that starts with 03 and ends with 100.

Notation	Description
	✓ **[100-**199](100,101,105): for a number that starts with 100 to 199 and ends with 100, 101 or 105. ✓ 03(abc): for any number that starts with 03 and ends with abc. ✓ 03(5xx): for any number that starts with 03 and ends with 5xx. ✓ 03(400,401,405): for any number that starts with 03 and ends with 400 or 401 or 405. **Notes:** ▪ The value *n* must be less than the value *m*. ▪ Only numerical ranges are supported (not alphabetical letters). ▪ For suffix ranges, the starting (*n*) and ending (*m*) numbers in the range must have the same number of digits. For example, (23-34) is correct, but (3-12) is not.
[n,m,...] or **(n,m,...)**	Represents multiple numbers. For example, to denote a one-digit number starting with 2, 3, 4, 5, or 6: ▪ Prefix: **[2,3,4,5,6]#** ▪ Suffix: **(2,3,4,5,6)** ▪ Prefix with Suffix: **[2,3,4,5,6](8,7,6)** - prefix is denoted in square brackets; suffix in parenthesis For **prefix only**, the notations *d**[n,**m]e* and *d**[n-**m]e* can also be used: ▪ To denote a five-digit number that starts with 11, 22, or 33: **[11,22,33]xxx#** ▪ To denote a six-digit number that starts with 111 or 222: **[111,222]xxx#** **Note:** Up to three digits can be used to denote each number.
[n1-m1,n2-m2,a,b,c,n3-m3] or **(n1-m1,n2-m2,a,b,c,n3-m3)**	Represents a mixed notation of single numbers and multiple ranges. For example, to denote numbers 123 to 130, 455, 766, and 780 to 790: ▪ Prefix: **[123-130,455,766,780-790]** ▪ Suffix: **(123-130,455,766,780-790)** **Note:** The ranges and the single numbers used in the dial plan must have the same number of digits. For example, each number range and single number in the dialing plan example above consists of three digits.

19.8.2 Digit Mapping

Digit map pattern rules are used for Tel-to-IP ISDN overlap dialing (by setting the ISDNRxOverlap parameter to 1) to reduce the dialing period (for digital interface). For more information on digit maps for ISDN overlapping, see ISDN Overlap Dialing on page 266. The device collects digits until a match is found in the user-defined digit pattern (e.g., for closed numbering schemes). The device stops collecting digits and starts sending the digits (collected number) when any one of the following scenarios occur:

- Maximum number of digits is received. You can define (using the MaxDigits parameter) the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side by the device. When the number of collected digits reaches the maximum (or a digit map pattern is matched), the device uses these digits for the called destination number.
- Inter-digit timeout expires (e.g., for open numbering schemes). This is defined using the TimeBetweenDigits parameter. This is the time that the device waits between each received digit. When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number.
- The phone's pound (#) key is pressed.

- Digit string (i.e., dialed number) matches one of the patterns defined in the digit map.

Digit map (pattern) rules are defined using the DigitMapping parameter. The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar ("|"). The maximum length of the entire digit pattern is 152 characters. The available notations are described in the table below:

Table 19-24: Digit Map Pattern Notations

Notation	Description
[n-m]	Range of numbers (not letters).
.	(single dot) Repeat digits until next notation (e.g., T).
x	Any single digit.
T	Dial timeout (configured by the TimeBetweenDigits parameter).
S	Short timer (configured by the TimeBetweenDigits parameter; default is two seconds) that can be used when a specific rule is defined after a more general rule. For example, if the digit map is 99|998, then the digit collection is terminated after the first two 9 digits are received. Therefore, the second rule of 998 can never be matched. But when the digit map is 99s|998, then after dialing the first two 9 digits, the device waits another two seconds within which the caller can enter the digit 8.

Below is an example of a digit map pattern containing eight rules:

```
DigitMapping = 11xS|00[1-
7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x|xx.T
```

In the example, the rule "00**[1-**7]xxx" denotes dialed numbers that begin with 00, and then any digit from 1 through 7, followed by three digits (of any number). Once the device receives these digits, it does not wait for additional digits, but starts sending the collected digits (dialed number) immediately.

Notes:

- If you want the device to accept/dial any number, ensure that the digit map contains the rule "xx.T"; otherwise, dialed numbers not defined in the digit map are rejected.
- If you are using an external Dial Plan file for dialing plans (see 'External Dial Plan File' on page 377), the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map (configured by the DigitMapping parameter).
- It may be useful to configure both Dial Plan file and Digit Maps. For example, the Digit Map can be used for complex digit patterns (which are not supported by the Dial Plan) and the Dial Plan can be used for long lists of relatively simple digit patterns. In addition, as timeout between digits is not supported by the Dial Plan, the Digit Map can be used to define digit patterns (MaxDigits parameter) that are shorter than those defined in the Dial Plan, or left at default. For example, “xx.T” Digit Map instructs the device to use the Dial Plan and if no matching digit pattern, it waits for two more digits and then after a timeout (TimeBetweenDigits parameter), it sends the collected digits. Therefore, this ensures that calls are not rejected as a result of their digit pattern not been completed in the Dial Plan.

19.8.3 External Dial Plan File

The device allows you to select a specific Dial Plan (index) defined in an external Dial Plan file. This file is loaded to the device as a .dat file (binary file), converted from an *ini* file using the DConvert utility. This file can include up to eight Dial Plans (Dial Plan indices), with a total of up to 8,000 dialing rules (lines). The required Dial Plan is selected using the DialPlanIndex parameter. This parameter can use values 0 through 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The Dial Plan index can be configured globally or per Tel Profile.

The format of the Dial Plan index file is as follows:

- A name in square brackets ("**[...]") on** a separate line indicates the beginning of a new Dial Plan index.
- Every line under the Dial Plan index defines a dialing prefix and the number of digits expected to follow that prefix. The prefix is separated by a comma (",") from the number of additional digits.
- The prefix can include numerical ranges in the format **[x-**y], as well as multiple numerical ranges **[n-**m]**[x-**y] (no comma between them).
- The prefix can include asterisks ("*") and number signs ("#").
- The number of additional digits can include a numerical range in the format x-y.
- Empty lines and lines beginning with a semicolon (";") are ignored.

An example of a Dial Plan file with indices (in *ini*-file format before conversion to binary .dat) is shown below:

```
[ PLAN1 ]
; Area codes 02, 03, - phone numbers include 7 digits.
02,7
03,7
; Cellular/VoIP area codes 052, 054 - phone numbers include 8
digits.
052,8
054,8
; International prefixes 00, 012, 014 - number following
prefix includes 7 to 14 digits.
00,7-14
012,7-14
014,7-14
; Emergency number 911 (no additional digits expected).
911,0
[ PLAN2 ]
; Supplementary services such as Call Camping and Last Calls
(no additional digits expected), by dialing *41, *42, or *43.
*4[1-3],0
```

Notes:

- If you are using an external Dial Plan file for dialing plans (see 'External Dial Plan File' on page 377), the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map (configured by the DigitMapping parameter).
- It may be useful to configure both Dial Plan file and Digit Maps. For example, the Digit Map can be used for complex digit patterns (which are not supported by the Dial Plan) and the Dial Plan can be used for long lists of relatively simple digit patterns. In addition, as timeout between digits is not supported by the Dial Plan, the Digit Map can be used to define digit patterns (MaxDigits parameter) that are shorter than those defined in the Dial Plan, or left at default. For example, “xx.T” Digit Map instructs the device to use the Dial Plan and if no matching digit pattern, it waits for two more digits and then after a timeout (TimeBetweenDigits parameter), it sends the collected digits. Therefore, this ensures that calls are not rejected as a result of their digit pattern not been completed in the Dial Plan.
- For E1 CAS MFC-R2 variants (which don't support terminating digit for the called party number, usually I-15), the external Dial Plan file and the DigitMapping parameter are ignored. Instead, you can define a Dial Plan template per trunk using the parameter CasTrunkDialPlanName_x.

19.8.3.1 Modifying ISDN-to-IP Calling Party Number

The device can use the Dial Plan file to change the Calling Party Number value (source number) of the incoming ISDN call when sending to IP. For this feature, the Dial Plan file supports the following syntax:
<ISDN Calling Party Number>,0,<new calling number>

- The first number contains the calling party number (or its prefix) received in the ISDN call SETUP message. The source number can also be a range, using the syntax **[x-**y] in the Dial Plan file. This number is used as the display name in the From header of the outgoing INVITE.
- The second number must always be set to "0".
- The third number is a string of up to 12 characters containing the mapped number that is used as the URI user part in the From and Contact headers of the outgoing INVITE.

The Dial Plan index used in the Dial Plan file for this feature is defined by the Tel2IPSourceNumberMappingDialPlanIndex parameter.

An example of such a configuration in the Dial Plan file is shown below:

```
[ PLAN1 ]
; specific received number changed to 04343434181.
0567811181,0,04343434181
; number range that changes to 04343434181.
056788118[2-4],0,04343434181
```

If we take the first Dial Plan rule in the example above (i.e., "0567811181,0,04343434181"), the received Calling Number Party of 0567811181 is changed to 04343434181 and sent to the IP with a SIP INVITE as follows:

```
Via: SIP/2.0/UDP 211.192.160.214:5060;branch=z9hG4bK3157667347
From: <sip:04343434181@kt.co.kr:5060>;tag=de0004b1
To: sip:01066557573@kt.co.kr:5060
Call-ID: 585e60ec@211.192.160.214
CSeq: 1 INVITE
Contact:<sip:04343434181@211.192.160.214:5060;transport=udp>
```

The initial Dial Plan text file must be converted to *.dat file format using the DConvert utility. This is done by clicking the DConvert's **Process Dial Plan File** button. For a detailed description of the DConvert utility, refer to the Product Reference Manual. You can load this *.dat file to the device using the Web interface (see 'Loading Auxiliary Files' on page 681), BootP & TFTP utility, or using the Auto-update mechanism from an external HTTP server.

Notes:

- Tel-to-IP routing is performed on the original source number if the parameter 'Tel to IP Routing Mode' is set to 'Route calls before manipulation'.
- Tel-to-IP routing is performed on the modified source number as defined in the Dial Plan file, if the parameter 'Tel To IP Routing Mode' is set to 'Route calls after manipulation'.
- Source number Tel-to-IP manipulation is performed on the modified source number as defined in the Dial Plan file.

19.8.4 Dial Plan Prefix Tags for IP-to-Tel Routing

The device supports the use of string labels (or "tags") in the external Dial Plan file for tagging incoming IP-to-Tel calls. The special “tag” is added as a prefix to the called party number, and then the Inbound IP Routing Table uses this “tag” instead of the original prefix. Manipulation is then performed after routing in the Manipulation table, which strips the “tag” characters before sending the call to the endpoint.

This feature resolves the limitation of entries in the Inbound IP Routing Table (IP-to-Tel call routing) for scenarios in which many different routing rules are required. For example, a city may have many different area codes, some for local calls and others for long distance calls (e.g. 425-202-xxxx for local calls, but 425-200-xxxx for long distance calls).

For using tags, the Dial Plan file is defined as follows:

- Number of dial plan (text)
- Dial string prefix (ranges can be defined in brackets)
- User-defined routing tag (text)

Note: Dial Plan Prefix Tags are not applicable to FXS and FXO interfaces.

The example configuration below assumes a scenario where multiple prefixes exist for local and long distance calls:

➢ **To use Dial Plan file routing tags:**

1. Load an *ini* file to the device that selects the Dial Plan index (e.g., 1) for routing tags, as shown below:

```
IP2TelTaggingDestDialPlanIndex = 1
```

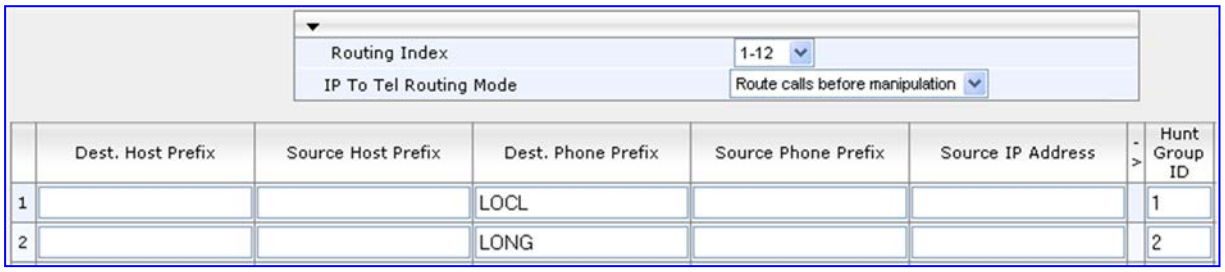
2. Define the external Dial Plan file with two routing tags (as shown below):
   - "LOCL" - for local calls
   - "LONG" - for long distance calls

```
[ PLAN1 ]
42520[3-5],0,LOCL
425207,0,LOCL
42529,0,LOCL
425200,0,LONG
425100,0,LONG
```

   Therefore, if an incoming IP call to destination prefix 425203 (for example) is received, the device adds the prefix tag "LOCL" (as specified in the Dial Plan file), resulting in the number "LOCL425203".

3. Assign the different tag prefixes to different Trunk Groups in the Inbound IP Routing Table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **IP to Trunk Group Routing**):
   - The Dest. Phone Prefix' field is set to the value "LOCL" and this rule is assigned to a local Trunk Group (e.g. Trunk Group ID 1).
   - The Dest. Phone Prefix' field is set to the value "LONG" and this rule is assigned to a long distance Trunk Group (e.g. Trunk Group ID 2).

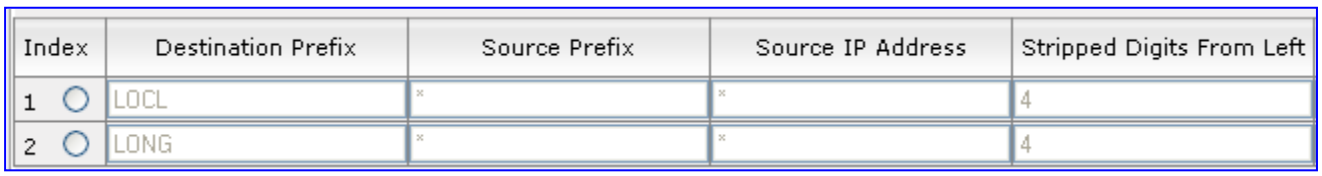
Figure 19-53: Configuring Dial Plan File Label for IP-to-Tel Routing



The above routing rules are configured to be performed before manipulation (described in the step below).

4. Configure manipulation in the Destination Phone Number Manipulation Table for IP to Tel Calls table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu > **Dest Number IP->Tel**) for removing the first four characters of the called party number "tag" (in our example, "LOCL" and "LONG"):
   - The Destination Prefix' field is set to the value "LOCL" and the 'Stripped Digits From Left' field is set to '4'.
   - The Destination Prefix' field is set to the value "LONG" and the 'Stripped Digits From Left' field is set to '4'.

Figure 19-54: Configuring Manipulation for Removing Label

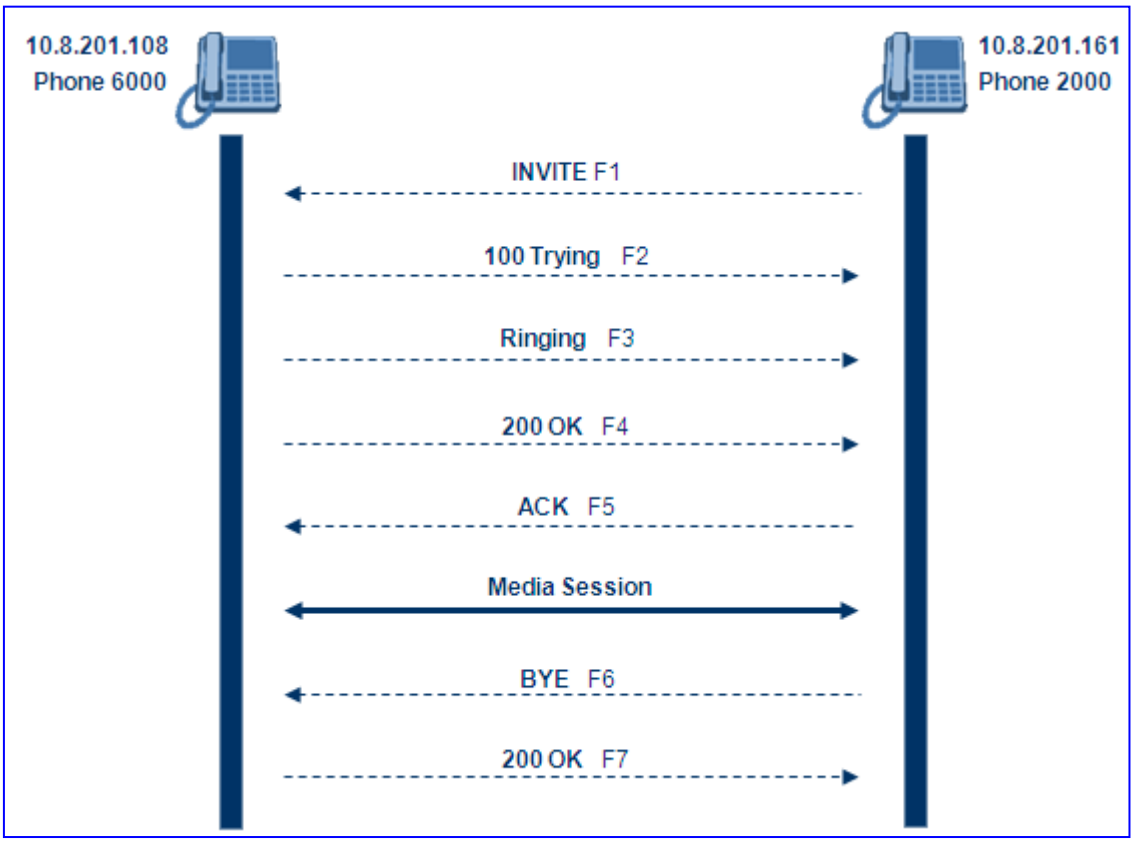


19.9 SIP Call Routing Examples

19.9.1 SIP Call Flow Example

The SIP call flow (shown in the following figure), describes SIP messages exchanged between two devices during a basic call. In this call flow example, device (10.8.201.158) with phone number '6000' dials device (10.8.201.161) with phone number '2000'.

Figure 19-55: SIP Call Flow



- **F1 INVITE (10.8.201.108 >> 10.8.201.161):**

```
INVITE sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
CSeq: 18153 INVITE
Contact: <sip:8000@10.8.201.108;user=phone>
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000
MSBG/v.6.40.010.006
Supported: 100rel,em
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,
NOTIFY,PRACK,REFER,INFO
Content-Type: application/sdp
Content-Length: 208
v=0
o=AudiocodesGW 18132 74003 IN IP4 10.8.201.108
s=Phone-Call
```

```
c=IN IP4 10.8.201.108
t=0 0
m=audio 4000 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
```

- **F2 TRYING (10.8.201.161 >> 10.8.201.108):**

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000 MSBG/v.6.40.010.006
CSeq: 18153 INVITE
Content-Length: 0
```

- **F3 RINGING 180 (10.8.201.161 >> 10.8.201.108):**

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000 MSBG/v.6.40.010.006
CSeq: 18153 INVITE
Supported: 100rel,em
Content-Length: 0
```

Note: Phone '2000' answers the call and then sends a 200 OK message to device 10.8.201.108.

- **F4 200 OK (10.8.201.161 >> 10.8.201.108):**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
CSeq: 18153 INVITE
Contact: <sip:2000@10.8.201.161;user=phone>
Server: Audiocodes-Sip-Gateway/Mediant 1000 MSBG/v.6.40.010.006
Supported: 100rel,em
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,
NOTIFY,PRACK,REFER,INFO
Content-Type: application/sdp
Content-Length: 206
v=0
o=AudiocodesGW 30221 87035 IN IP4 10.8.201.161
s=Phone-Call
c=IN IP4 10.8.201.10
t=0 0
```

```
m=audio 7210 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=ptime:20
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
```

- **F5 ACK (10.8.201.108 >> 10.8.201.10):**

```
ACK sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacZYpJWxZ
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000
MSBG/v.6.40.010.006
CSeq: 18153 ACK
Supported: 100rel,em
Content-Length: 0
```

Note: Phone '6000' goes on-hook and device 10.8.201.108 sends a BYE to device 10.8.201.161. A voice path is established.

- **F6 BYE (10.8.201.108 >> 10.8.201.10):**

```
BYE sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacRKCVBud
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000
MSBG/v.6.40.010.006
CSeq: 18154 BYE
Supported: 100rel,em
Content-Length: 0
```

- **F7 OK 200 (10.8.201.10 >> 10.8.201.108):**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacRKCVBud
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000 MSBG/v.6.40.010.006
CSeq: 18154 BYE
Supported: 100rel,em
Content-Length: 0
```

19.9.2 SIP Message Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261 standard. A proxy server might require authentication before forwarding an INVITE message. A Registrar/Proxy server may also require authentication for client registration. A proxy replies to an unauthenticated INVITE with a 407 Proxy Authorization Required

response, containing a Proxy-Authenticate header with the form of the challenge. After sending an ACK for the 407, the user agent can then re-send the INVITE with a Proxy-Authorization header containing the credentials.

User agents, Redirect or Registrar servers typically use 401 Unauthorized response to challenge authentication containing a WWW-Authenticate header, and expect the re-INVITE to contain an Authorization header.

The following example describes the Digest Authentication procedure, including computation of user agent credentials:

1. The REGISTER request is sent to a Registrar/Proxy server for registration:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c17940
To: <sip: 122@10.1.1.200>
Call-ID: 634293194@10.1.1.200
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000
MSBG/v.6.40.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
```

2. Upon receipt of this request, the Registrar/Proxy returns a 401 Unauthorized response:

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.2.1.200
From: <sip:122@10.2.2.222 >;tag=1c17940
To: <sip:122@10.2.2.222 >
Call-ID: 634293194@10.1.1.200
Cseq: 1 REGISTER
Date: Mon, 30 Jul 2001 15:33:54 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
WWW-Authenticate: Digest realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
stale=FALSE,
algorithm=MD5
```

3. According to the sub-header present in the WWW-Authenticate header, the correct REGISTER request is created.
4. Since the algorithm is MD5:
   - The username is equal to the endpoint phone number 122.
   - The realm return by the proxy is audiocodes.com.
   - The password from the *ini* file is AudioCodes.
   - The equation to be evaluated is (according to RFC this part is called A1) **'122:audiocodes.com:AudioCodes'.**
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is 'a8f17d4b41ab8dab6c95d3c14e34a9e1'.
5. Next, the par called A2 needs to be evaluated:
   - The method type is 'REGISTER'.
   - Using SIP protocol 'sip'.
   - Proxy IP from *ini* file is '10.2.2.222'.
   - The equation to be evaluated is **'REGISTER:sip:10.2.2.222'.**

- The MD5 algorithm is run on this equation and stored for future usage.
- The result is 'a9a031cfddcb10d91c8e7b4926086f7e'.

6. Final stage:

- The A1 result: The nonce from the proxy response is '11432d6bce58ddf02e3b5e1c77c010d2'.
- The A2 result: The equation to be evaluated is **'A1:11432d6bce58ddf02e3b5e1c77c010d2:A2'.**
- The MD5 algorithm is run on this equation. The outcome of the calculation is the response needed by the device to register with the Proxy.
- The response is 'b9c45d0234a5abf5ddf5c704029b38cf'.

At this time, a new REGISTER request is issued with the following response:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Server: Audiocodes-Sip-Gateway/Mediant 1000 MSBG/v.6.40.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
Authorization: Digest, username: 122,
realm="audiocodes.com”,
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
uri=”10.2.2.222”,
response=“b9c45d0234a5abf5ddf5c704029b38cf”
```

7. Upon receiving this request and if accepted by the Proxy, the proxy returns a 200 OK response closing the REGISTER transaction:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Cseq: 1 REGISTER
Date: Thu, 26 Jul 2001 09:34:42 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
Contact: <sip:122@10.1.1.200>; expires="Thu, 26 Jul 2011 10:34:42
GMT"; action=proxy; q=1.00
Contact: <122@10.1.1.200:>; expires="Tue, 19 Jan 2038 03:14:07
GMT"; action=proxy; q=0.00
Expires: Thu, 26 Jul 2001 10:34:42 GMT
```

19.9.3 Establishing a Call between Two Devices

This section provides an example on configuring two AudioCodes' devices with FXS interfaces for establishing call communication. This setup enables the establishment of calls between telephones connected to the same device, and between the two devices.

This example assumes the following:

- IP address of the first device is 10.2.37.10 and its endpoint numbers are 101 to 104.
- IP address of the second device is 10.2.37.20 and its endpoint numbers are 201 to 204.
- SIP Proxy is not used. Internal call routing is performed using the device's Outbound IP Routing Table.

➢ **To configure the two devices for call communication:**

1. For the *first* device (10.2.37.10), in the Trunk Group Table page (see), assign the phone numbers 101 to 104 to the device's endpoints.

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	101	0

2. For the *second* device (10.2.37.20), in the Trunk Group Table page, assign the phone numbers 201 to 204 to the device's endpoints.

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	201	0

3. Configure the following for *both* devices:

   In the Outbound IP Routing Table page (see 'Configuring Outbound IP Routing Table' on page 291), add the following routing rules:

   a. In the first row, enter 10 for the destination phone prefix and enter 10.2.37.10 for the destination IP address (i.e., IP address of the first device).
   b. In the second row, enter 20 for the destination phone prefix and 10.2.37.20 for the destination IP address (i.e., IP address of the second device).

   These settings enable the routing (from both devices) of outgoing Tel-to-IP calls that start with 10 to the first device and calls that start with 20 to the second device.

	Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Dest. IP Group ID
1		10	*		10.2.37.10	
2		20	*		10.2.37.20	

4. Make a call. Pick up the phone connected to port #1 of the first device and dial 102 (to the phone connected to port #2 of the same device). Listen for progress tones at the calling phone and for the ringing tone at the called phone. Answer the called phone, speak into the calling phone, and check the voice quality. Dial 201 from the phone connected to port #1 of the first device; the phone connected to port #1 of the second device rings. Answer the call and check the voice quality.

19.9.4 Trunk-to-Trunk Routing Example

This example describes two devices, each interfacing with the PSTN through four E1 spans. Device **A** is configured to route all incoming Tel-to-IP calls to Device **B**. Device **B** generates calls to the PSTN on the same E1 trunk on which the call was originally received (in Device **A**).

- Device **A** IP address: 192.168.3.50
- Device **B** IP address: 192.168.3.51

The *ini* file parameters configuration for devices **A** and **B** are as follows:

1. At both devices, define four Trunk Groups, each with 30 B-channels:
   - TrunkGroup_1 = 0/1-31,1000
   - TrunkGroup_2 = 1/1-31,2000
   - TrunkGroup_3 = 2/1-31,3000
   - TrunkGroup_4 = 3/1-31,4000
2. At Device **A**, add the originating Trunk Group ID as a prefix to the destination number for Tel-to-IP calls:

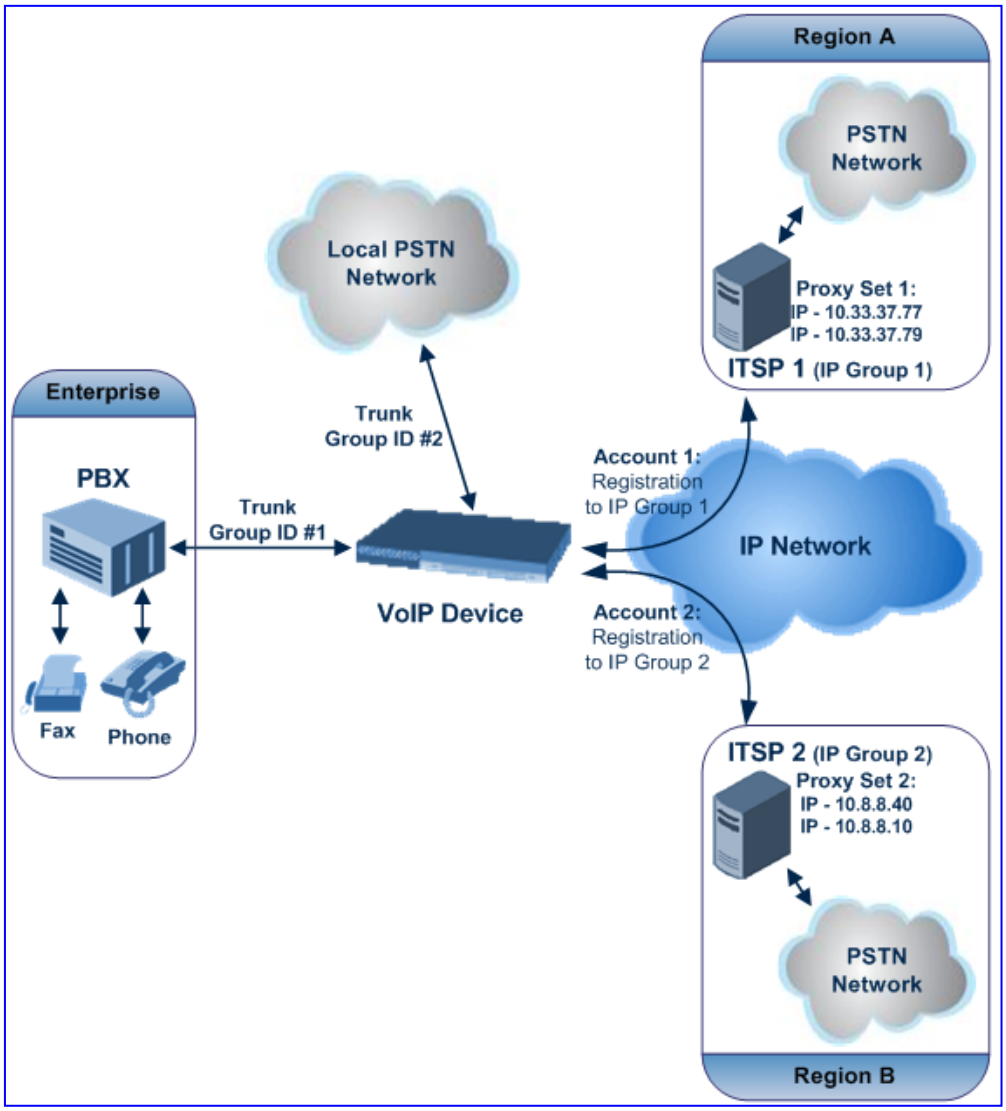
   AddTrunkGroupAsPrefix = 1
3. At Device **A**, route all incoming PSTN calls starting with prefixes 1, 2, 3, and 4, to the IP address of Device **B**:
   - Prefix = 1, 192.168.3.51
   - Prefix = 2, 192.168.3.51
   - Prefix = 3, 192.168.3.51
   - Prefix = 4, 192.168.3.51

   Note: You can also define Prefix = *,192.168.3.51, instead of the four lines above.
4. At Device **B**, route IP-to-PSTN calls to Trunk Group ID according to the first digit of the called number:
   - PSTNPrefix = 1,1
   - PSTNPrefix = 2,2
   - PSTNPrefix = 3,4
   - PSTNPrefix = 4,4
5. At Device **B**, remove the first digit from each IP-to-PSTN number before it is used in an outgoing call: NumberMapIP2Tel = *,1.

19.9.5 SIP Trunking between Enterprise and ITSPs

By implementing the device's enhanced and flexible routing capabilities, you can design complex routing schemes. This section provides an example of an elaborate routing scheme for SIP trunking between an Enterprise's PBX and two Internet Telephony Service Providers (ITSP), using the device.

Scenario: In this example, the Enterprise wishes to connect its TDM PBX to two different ITSPs, by implementing the device in its network environment. It's main objective is for the device to route Tel-to-IP calls to these ITSPs according to a dial plan. The device is to register (on behalf of the PBX) to each ITSP, which implements two servers for redundancy and load balancing. The Register messages must use different URI's in the From, To, and Contact headers per ITSP. In addition, all calls dialed from the Enterprise PBX with prefix '02' is sent to the local PSTN. The figure below illustrates this example setup:



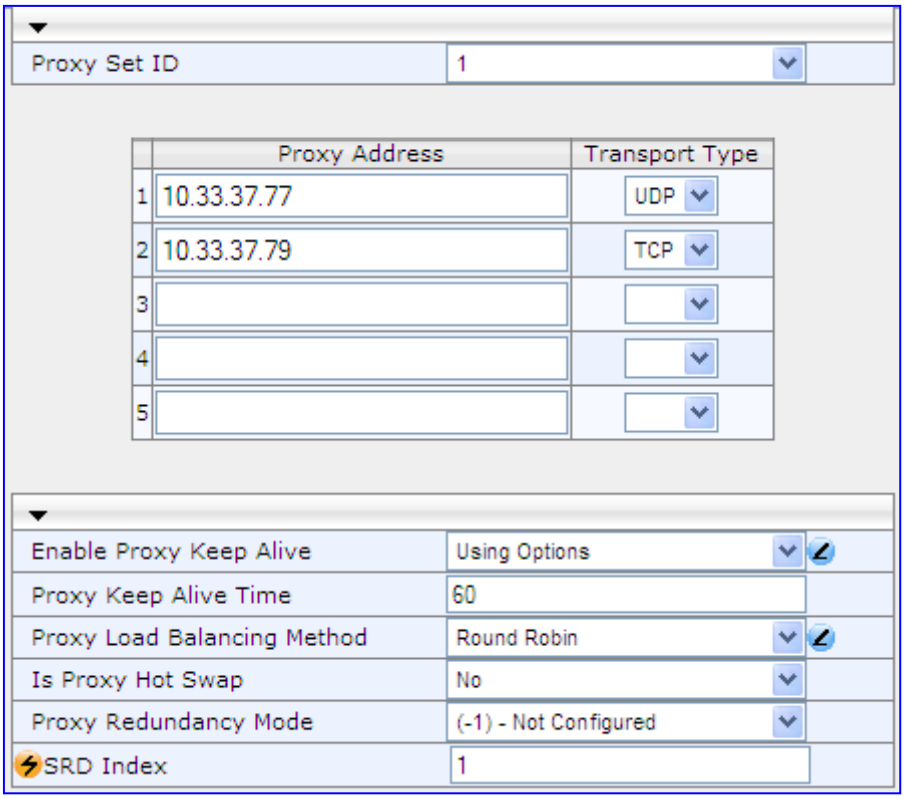
➢ **To configure call routing between an Enterprise and two ITSPs:**

1. Enable the device to register to a Proxy/Registrar server using the parameter IsRegisterNeeded.
2. In the Proxy Sets Table page (see 'Configuring Proxy Sets Table' on page 219), configure two Proxy Sets and for each, enable Proxy Keep-Alive (using SIP OPTIONS) and 'round robin' load-balancing method:

- Proxy Set #**1** includes two IP addresses of the first ITSP (**ITSP 1**) - 10.33.37.77 and 10.33.37.79 - and using UDP.
- Proxy Set #**2** includes two IP addresses of the second ITSP (**ITSP 2**) - 10.8.8.40 and 10.8.8.10 - and using TCP.

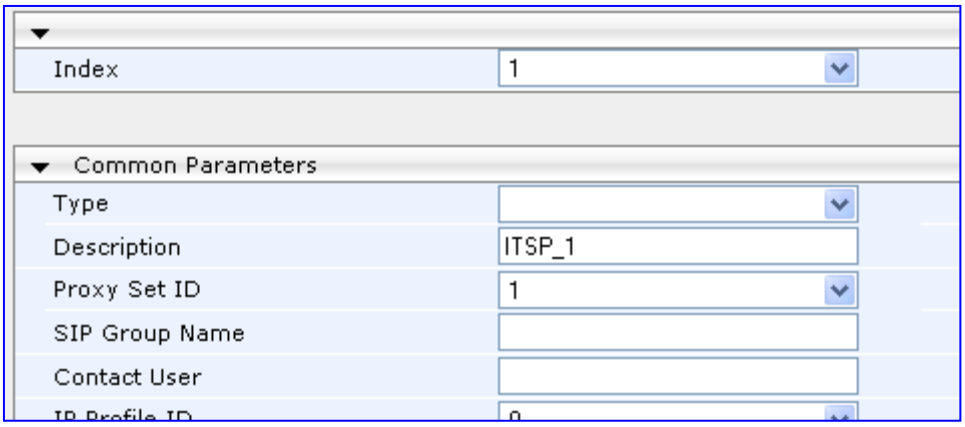
The figure below displays the configuration of Proxy Set ID #1. Perform similar configuration for Proxy Set ID #2, but using different IP addresses.

Figure 19-56: Configuring Proxy Set ID #1 in the Proxy Sets Table Page



3. In the IP Group Table page (see 'Configuring IP Groups' on page 212), configure the two IP Groups #1 and #2. Assign Proxy Sets #1 and #2 to IP Groups #1 and #2 respectively.

Figure 19-57: Configuring IP Groups #1 and #2 in the IP Group Table Page



4. In the Trunk Group Table page, enable the Trunks connected between the Enterprise's PBX and the device (Trunk Group ID #1), and between the local PSTN and the device (Trunk Group ID #2).

Figure 19-58: Assigning Trunks to Trunk Group ID #1

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID	Tel Profile ID
1	Module 1 PRI	1	1	1-30	1100	1	0
2	Module 1 PRI	2	2	1-30	2200	2	0

5. In the Trunk Group Settings page, configure 'Per Account' registration for Trunk Group ID #1 (without serving IP Group)

Figure 19-59: Configuring Trunk Group #1 for Registration per Account in Trunk Group Settings Page

	Trunk Group ID	Channel Select Mode	Registration Mode	Serving IP Group ID	Gateway Name	Contact User
1	1	Cyclic Ascending	Per Account			username

6. In the Account Table page, configure the two Accounts for PBX trunk registration to ITSPs using the same Trunk Group (i.e., ID #1), but different serving IP Groups #1 and #2. For each account, define user name, password, and hostname, and ContactUser. The Register messages use different URI's (Hostname and ContactUser) in the From, To, and Contact headers per ITSP. Enable registration for both accounts.

Figure 19-60: Configuring Accounts for PBX Registration to ITSPs in Account Table Page

Index	ServedTrunkGroup	ServingIPGroup	Username	Password	HostName	Register	ContactUser
1	1	1	user1	1234	ITSP1	1	ITSP1user
2	1	2	user2	5555	ITSP2	1	ITSP2user

7. In the Inbound IP Routing Table page, configure IP-to-Tel routing for calls from ITSPs to Trunk Group ID #1 (see 1 below) and from the device to the local PSTN (see 2 below).

Figure 19-61: Configuring ITSP-to-Trunk Group #1 Routing in IP to Trunk Group Table Page

	Dest. Host Prefix	Source Host Prefix	Dest. Phone Prefix	Source Phone Prefix	Source IP Address	->	Trunk Group ID	IP Profile ID	Source IPGroup ID
1			*	*			1		
2			02	*			2		

8. In the Outbound IP Routing Table page, configure Tel-to-IP routing rules for calls to ITSPs (see first entry below) and to local PSTN (see second and third entries below).

Figure 19-62: Configuring Tel-to-IP Routing to ITSPs in Tel to IP Routing Table Page

Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Port	Transport Type	Dest. IPGroup ID
1	0[3,4,5[	*				Not Configured	1
1	0[6,7,8]	*				Not Configured	2
1	02	*		10.13.4.13		Not Configured	

19.10 IP-to-IP Routing Application

The device's supports IP-to-IP VoIP call routing (or SIP Trunking). The IP-to-IP call routing application enables enterprises to seamlessly connect their IP-based PBX (IP-PBX) to SIP trunks, typically provided by an Internet Telephony Service Provider (ITSP). By implementing the device, enterprises can then communicate with PSTN networks (local and overseas) through ITSP's, which interface directly with the PSTN. Therefore, the IP-to-IP application enables enterprises to replace the bundles of physical PSTN wires with SIP trunks provided by ITSP's and use VoIP to communicate within and outside the enterprise network using its standard Internet connection. At the same time, the device can also provide an interface with the traditional PSTN network, enabling PSTN fallback in case of IP connection failure with the ITSP's.

In addition, the device supports multiple SIP Trunking. This can be useful in scenarios where if a connection to one ITSP fails, the call can immediately be transferred to another ITSP. In addition, by allowing multiple SIP trunks where each trunk is designated a specific ITSP, the device can route calls to an ITSP based on call destination (e.g., country code).

Therefore, in addition to providing VoIP communication within an enterprise's LAN, the device allows the enterprise to communicate outside of the corporate LAN using SIP Trunking. This includes remote (roaming) IP-PBX users, for example, employees using their laptops to communicate with one another from anywhere in the world such as at airports.

The IP-to-IP application can be implemented by enterprises in the following example scenarios:

- VoIP between an enterprise's headquarters and remote branch offices
- VoIP between an enterprise and the PSTN via an ITSP

The IP-to-IP call routing capability is feature-rich, allowing interoperability with different ITSP's or service providers:

- Easy and smooth integration with multiple ITSP SIP trunks.
- Supports SIP registration and authentication with ITSP servers (on behalf of the enterprise's IP telephony system) even if the enterprise's IP telephony system does no support registration and authentication.
- Supports SIP-over-UDP, SIP-over-TCP, and SIP-over-TLS transport protocols, one of which is generally required by the ITSP.
- Provides alternative routing to different destinations (to another ITSP or the PSTN) when the connection with an ITSP network is down.
- Provides fallback to the legacy PSTN telephone network upon Internet connection failure.
- Provides Transcoding from G.711 to G.729 coder with the ITSP for bandwidth reduction.
- Supports SRTP, providing voice traffic security toward the ITSP.
- IP-to-IP routing can be used in combination with the regular Gateway application. For example, an incoming IP call can be sent to an E1/T1 span or it can be forwarded to an IP destination.

Therefore, the device provides the ideal interface between enterprises' IP-PBX's and ITSP SIP trunks.

In the IP-to-IP application, SIP Methods\Responses are handled and terminated at each leg independently:

- Initiating Dialog INVITE: terminated at one leg and initiated on the other leg, 180\182\183\200\4xx uses the same logic and same limitations, in some cases the result may be a different response code.
- OPTIONS: terminated at each leg independently.

- INFO: only specific INFO's (such as DTMF) are handled; other types are omitted.
- UPDATE: terminated at each leg independently and may cause only changes in the RTP flow - Hold\Retrieve are the only exceptions that traverse the two legs.
- ReINVITE: terminated at each leg independently and may cause only changes in the RTP flow - Hold\Retrieve are the only exceptions that traverse the two legs.
- PRACK: terminated at each leg independently.
- REFER (within a dialog): terminated at each leg independently.
- 3xx Responses: terminated at each leg independently.
- 401\407 Responses to initial INVITE: in case the B2B session is associated with an Account, the responses is terminated at the receiving leg; in other cases, the responses are passed transparently.
- REGISTER: handled only in cases associated with a USER IP Group - Contact\To\From specific parameters are omitted.

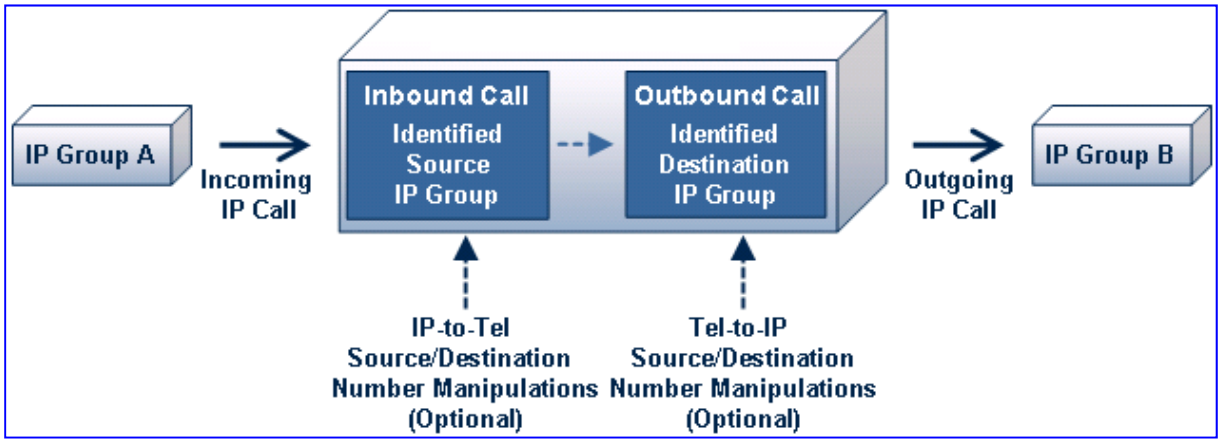
19.10.1 Theory of Operation

The device's IP-to-IP SIP session is performed by implementing Back-to-Back User Agent (B2BUA). The device acts as a user agent for both ends (*legs*) of the SIP call (from call establishment to termination). The session negotiation is performed independently for each call leg, using global parameters such as coders or using IP Profiles associated with each call leg to assign different configuration behaviors for these two IP-to-IP call legs.

The device also supports NAT traversal for SIP clients behind NAT, where the device is defined with a global IP address.

The figure below provides a simplified illustration of the device's handling of IP-to-IP call routing:

Figure 19-63: Basic Schema of the Device's IP-to-IP Call Handling



The basic IP-to-IP call handling process can be summarized as follows:

1. Incoming IP calls are identified as belonging to a specific logical entity in the network (referred to as a *Source IP Group*), according to Inbound IP Routing rules.
2. The Source IP Group is associated with a specific IP Group (*Destination IP Group*), and then sent to the appropriate destination address (defined by a *Proxy Set*) associated with this Destination IP Group.

3. Number manipulation can be performed at both legs (inbound and outbound).

The following subsections discuss the main terms associated with the IP-to-IP call routing application.

19.10.1.1 Proxy Sets

A Proxy Set is a group of up to five Proxy servers (for Proxy load balancing and redundancy), defined by IP address or fully qualified domain name (FQDN). The Proxy Set is assigned to IP Groups (of type SERVER only), representing the address of the IP Group to where the device sends the INVITE message (**destination** of the call). Typically, for IP-to-IP call routing, two Proxy Sets are defined for call destination – one for each leg (i.e., each IP Group) of the call (i.e., both directions).

19.10.1.2 IP Groups

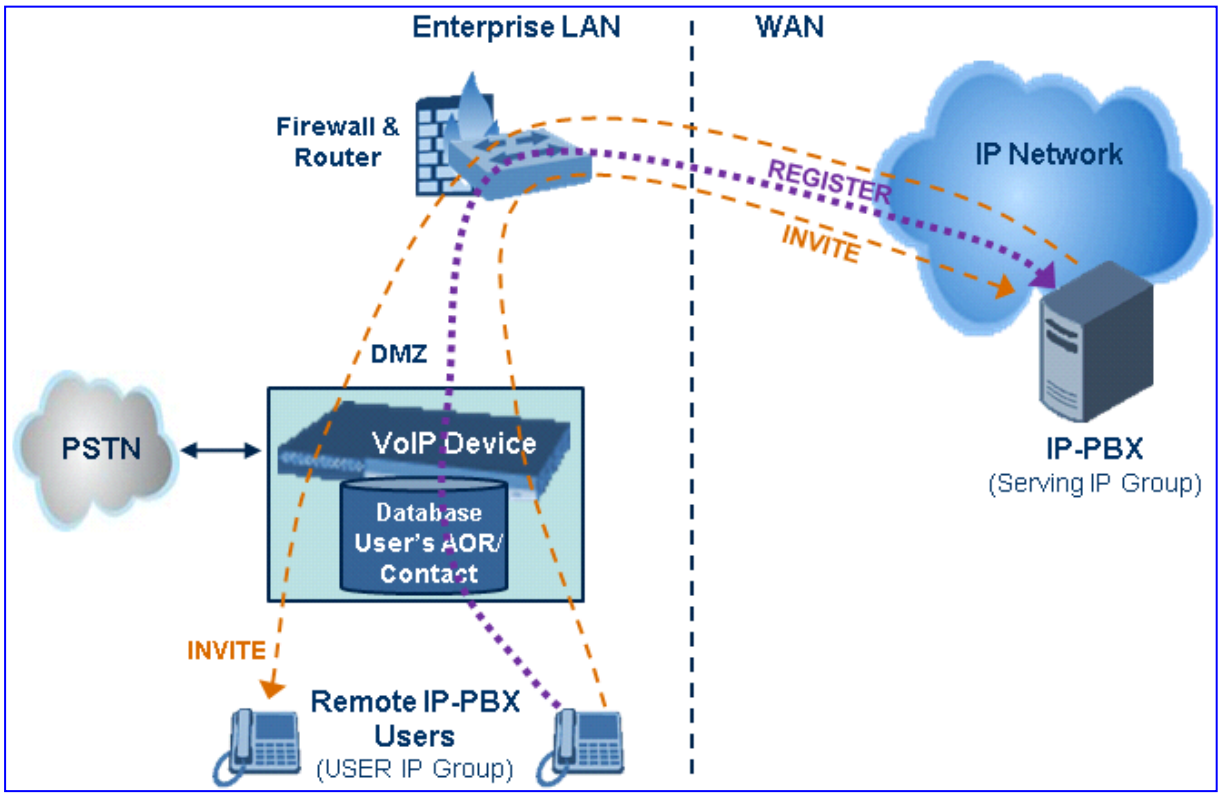
An IP Group represents a logical SIP entity in the device's network environment such as an ITSP SIP trunk, ITSP Proxy/Registrar server, IP-PBX, or remote IP-PBX users. The address of the IP Group is typically defined by the Proxy Set that is assigned to it.

The opposite legs of the call are each presented by an IP Group: one being a *Serving* IP Group; the other the *Served* IP Group. The Serving IP Group depicts the IP Group (e.g., ITSP) that provides service ("serves") to the Served IP Group (e.g., IP-PBX). This is the IP Group to where the device sends INVITE messages received from the Served IP Group as well as REGISTER messages for registering on behalf of the Served IP Group.

In addition, IP Groups can be *SERVER* or *USER* type. In SERVER IP Groups (e.g., ITSP or IP-PBX), the destination address (defined by the Proxy Set) is known. In contrast, USER IP Groups represents groups of users whose location is dynamically obtained by the device when REGISTER requests and responses traverse (or are terminated) by the device. Generally, these are remote IP-PBX users (e.g., IP phones and soft phones).

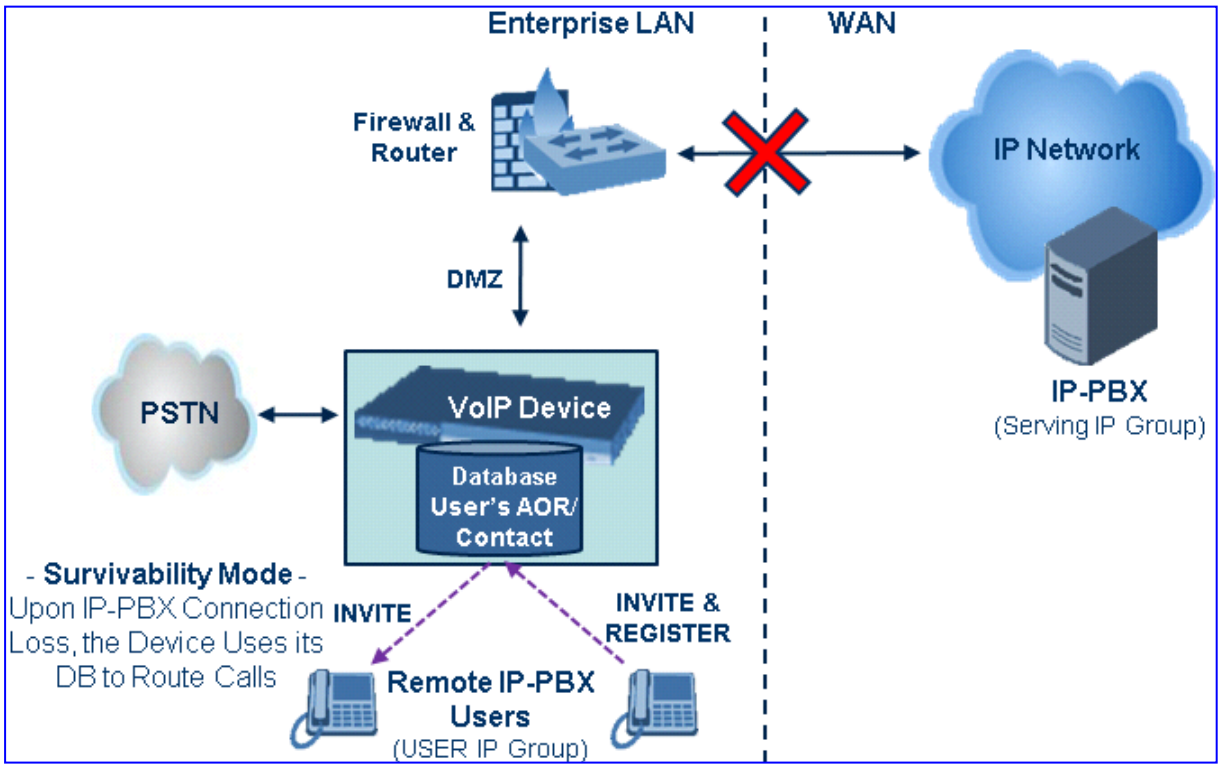
For registrations of USER IP Groups, the device updates its internal database with the AOR and Contacts of the users (refer to the figure below) Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group (e.g., IP-PBX). The device forwards these responses directly to the remote SIP users. For a call to a registered remote user, the device searches its dynamic database (by using the Request URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained and a SIP request is then sent to this user.

Figure 19-64: IP-to-IP Routing/Registration/Authentication of Remote IP-PBX Users (Example)



The device also supports the IP-to-IP call routing Survivability mode feature (refer to the figure below) for USER IP Groups. The device records (in its database) REGISTER messages sent by the clients of the USER IP Group. If communication with the Serving IP Group (e.g., IP-PBX) fails, the USER IP Group enters into Survivability mode in which the device uses its database for routing calls between the clients of the USER IP Group. The RTP packets between the clients traverse through the device. When the Serving IP Group is available again, the device returns to normal mode, sending INVITE and REGISTER messages to the Serving IP Group.

Figure 19-65: IP-to-IP Routing for IP-PBX Remote Users in Survivability Mode (Example)



19.10.1.3 Inbound and Outbound IP Routing Rules

The device's IP-to-IP call routing is performed using the following two routing rule stages:

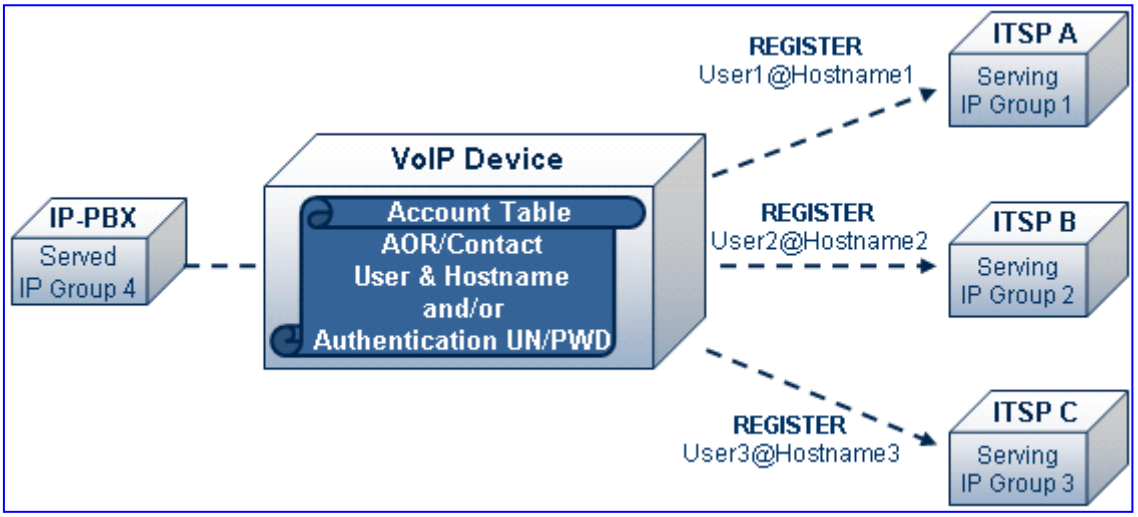
1. **Inbound IP Routing Mapping Rule:** Identifies the received call as an IP-to-IP call based on various characteristics such as the call's source IP address, and assigns it to an IP Group.
2. **Outbound IP Routing Mapping Rule:** Determines the destination (i.e., IP address) to where the incoming call (classified to a specific IP Group by the Inbound IP Routing rules) is finally routed. The destination address is typically depicted by another IP Group (destination IP Group) and therefore, the call is sent to the IP address that is defined in the Proxy Set associated with this IP Group. If the destination is a USER IP Group, the device searches for a match between the request URI (of the received INVITE) to an AOR registration record in the device's internal database. If a match is found, the INVITE is sent to the IP address of the registered contact.

19.10.1.4 Accounts

Accounts are used by the device to register to a Serving IP Group (e.g., an ITSP) on behalf of a Served IP Group (e.g., IP-PBX). This is necessary for ITSP's that require registration to provide services. Accounts are also used for defining user name/password for digest authentication (with or without registration) if required by the ITSP. Multiple Accounts per

Served IP Group can be configured for registration to more than one Serving IP Group (e.g., an IP-PBX that requires registering to multiple ITSP's).

Figure 19-66: Registration with Multiple ITSP's on Behalf of IP-PBX



19.10.2 IP-to-IP Routing Configuration Example

This section provides step-by-step procedures for configuring IP-to-IP call routing. These procedures are based on the setup example described below. In this example, the device serves as the communication interface between the enterprise's IP-PBX (located on the LAN) and the following network entities:

- ITSP SIP trunks (located on the WAN)
- Remote IP-PBX users (located on the WAN)
- Local PSTN network

Calls from the Enterprise are routed according to destination.

This example assumes the following:

- The device has the public IP address 212.25.125.136 and is connected to the enterprise's firewall/NAT demilitarized zone (DMZ) network, providing the interface between the IP-PBX, and two ITSP's and the local PSTN.
- The enterprise has an IP-PBX located behind a Firewall/NAT:
  - IP-PBX IP address: 10.15.4.211
  - Transport protocol: UDP
  - Voice coder: G.711
  - IP-PBX users: 4-digit length extension number and served by two ITSPs.
  - The enterprise also includes remote IP-PBX users that communicate with the IP-PBX via the device. All dialed calls from the IP-PBX consisting of four digits starting with digit "4" are routed to the remote IP-PBX users.
- Using SIP trunks, the IP-PBX connects (via the device) to two different ITSP's:
  - **ITSP-A:**
    - Implements Proxy servers with fully qualified domain names (FQDN): "Proxy1.ITSP-A" and "Proxy2.ITSP-B", using TLS.
    - Allocates a range of PSTN numbers beginning with +1919, which is assigned to a range of IP-PBX users.
    - Voice coder: G.723.
  - **ITSP-B:**
    - Implements Proxy servers with IP addresses 216.182.224.202 and

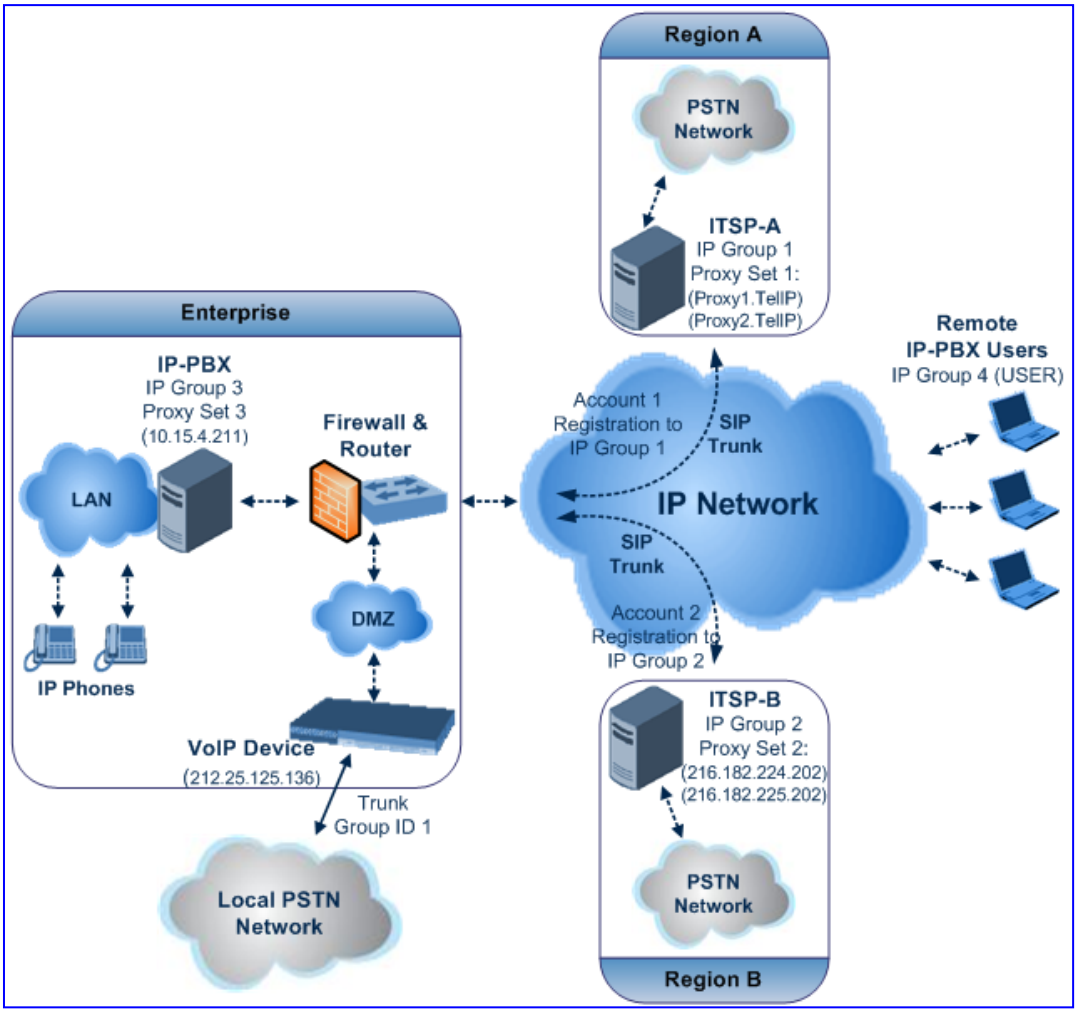
216.182.225.202, using TCP.

- Allocates a range of PSTN numbers beginning with 0200, which is assigned to a range of IP-PBX users.
- Voice coder: G.723.

- Registration and authentication is required by both ITSP's, which is performed by the device on behalf of the IP-PBX. The SIP REGISTER messages use different URI's (host name and contact user) in the From, To, and Contact headers per ITSP as well as username and password authentication.
- Outgoing calls from IP-PBX users are routed according to destination:
  - If the calls are dialed with the prefix "+81", they are routed to ITSP-A (Region A).
  - If the calls are dialed with the prefix "9", they are routed to the local PSTN network.
  - For all other destinations, the calls are routed to ITSP-B.
- The device is also connected to the PSTN through a traditional T1 ISDN trunk for local incoming and outgoing calls. Calls dialed from the enterprise's IP-PBX with prefix '9' are sent to the local PSTN. In addition, in case of Internet interruption and loss of connection with the ITSP trunks, all calls are rerouted to the PSTN.

The figure below provides an illustration of this example scenario:

Figure 19-67: SIP Trunking Setup Scenario Example



The steps for configuring the device according to the scenario above can be summarized as follows:

- Enable the IP-to-IP feature (see 'Step 1: Enable the IP-to-IP Capabilities' on page 398).
- Configure the number of media channels (see 'Step 2: Configure the Number of Media Channels' on page 399).
- Configure a Trunk Group for interfacing with the local PSTN (see 'Step 3: Define a Trunk Group for the Local PSTN' on page 399).
- Configure Proxy Sets (see 'Step 4: Configure the Proxy Sets' on page 400).
- Configure IP Groups (see 'Step 5: Configure the IP Groups' on page 402).
- Configure Registration Accounts (see 'Step 6: Configure the Account Table' on page 404).
- Configure IP Profiles (see 'Step 7: Configure IP Profiles for Voice Coders' on page 405).
- Configure inbound IP routing rules (see 'Step 8: Configure Inbound IP Routing' on page 407).
- Configure outbound IP routing rules (see 'Step 9: Configure Outbound IP Routing' on page 408).
- Configure destination phone number manipulation (see 'Step 10: Configure Destination Phone Number Manipulation' on page 410).

19.10.2.1 Step 1: Enable the IP-to-IP Capabilities

This step describes how to enable the device's IP-to-IP application.

➢ **To enable IP-to-IP capabilities:**

1. Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** submenu > **Applications Enabling**).
2. From the 'Enable IP2IP Application' drop-down list, select **Enable**, as shown below:

Note: For the IP-to-IP feature, the device must also be installed with the appropriate Software Upgrade Feature Key.

19.10.2.2 Step 2: Configure the Number of Media Channels

The number of media channels represents the number of digital signaling processors (DSP) channels that the device allocates to IP-to-IP calls. The remaining DSP channels can be used for PSTN calls. Two IP media channels are used per IP-to-IP call.

➢ **To configure the number of media channels:**

1. Open the IP Media Settings page (**Configuration** tab > **VoIP** menu > **IP Media** submenu > **IP Media Settings**).

Figure 19-68: Defining Required Media Channels

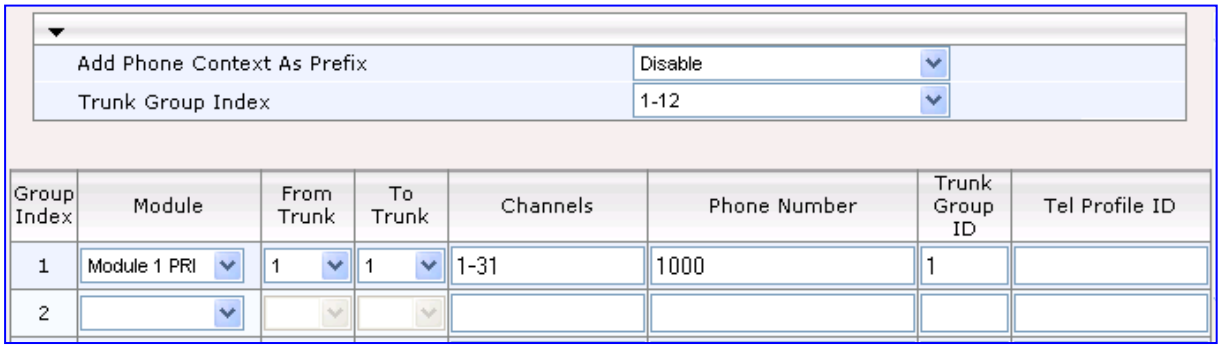
2. In the 'Number of Media Channels' field, enter the required number of media channels (in the example above, "120" to enable up to 60 IP-to-IP calls).
3. Click **Submit**.
4. Save the settings to flash memory ("burn") and reset the device (see 'Saving Configuration' on page 680).

19.10.2.3 Step 3: Define a Trunk Group for the Local PSTN

For incoming and outgoing local PSTN calls with the IP-PBX, you need to define the Trunk Group ID (#1) for the T1 ISDN trunk connecting between the device and the local PSTN. This Trunk Group is also used for alternative routing to the legacy PSTN network in case of a loss of connection with the ITSP's.

➢ **To configure the Trunk Group for local PSTN:**

1. Open the Trunk Group Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Trunk Group** > **Trunk Group**).
2. Configure Trunk Group ID #1 (as shown in the figure below):
   - From the 'From Trunk' and 'To Trunk' drop-down lists, select **1** to indicate Trunk 1 for this Trunk Group.
   - In the 'Channels' field, enter the Trunk channels or ports assigned to the Trunk Group (e.g. 1-31 for E1 and 1-24 for T1).
   - In the 'Phone Number' field, enter any phone number (logical) for this Trunk (e.g. 1000).
   - In the 'Trunk Group ID' field, enter "1" as the ID for this Trunk Group.



3. Configure the Trunk in the Trunk Settings page (**Configuration** tab > **VoIP** menu > **PSTN** submenu > **Trunk Settings**).

19.10.2.4 Step 4: Configure the Proxy Sets

This step describes how to configure the following Proxy Sets:

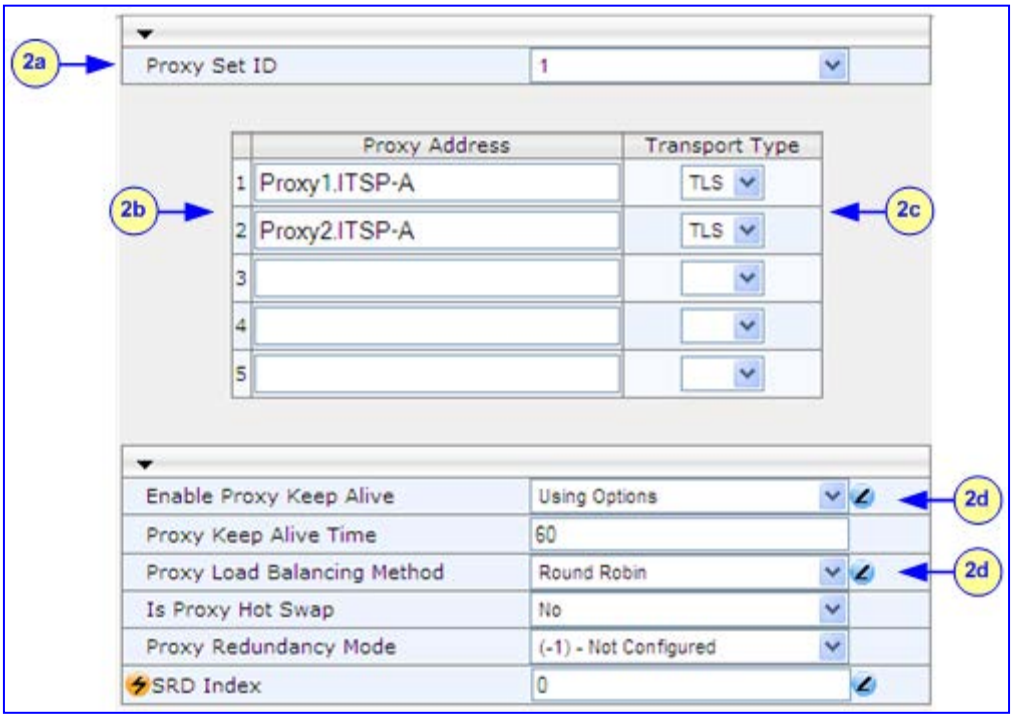
- Proxy Set ID #1 defined with two FQDN's for ITSP-A
- Proxy Set ID #2 defined with two IP addresses for ITSP-B
- Proxy Set ID #3 defined with an IP address for the IP-PBX

The Proxy Sets represent the actual destination (IP address or FQDN) to which the call is routed. These Proxy Sets are later assigned to IP Groups (see 'Step 5: Configure the IP Groups' on page 402).

➢ **To configure the Proxy Sets:**

1. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **Proxy Sets Table**).
2. Configure Proxy Set ID **#1** for ITSP-A:
   a. From the 'Proxy Set ID' drop-down list, select **1**.
   b. In the 'Proxy Address' column, enter the FQDN of ITSP-A SIP trunk Proxy servers (e.g., Proxy1.ITSP-A and Proxy2. ITSP-A).
   c. From the 'Transport Type' drop-down list corresponding to the Proxy addresses entered above, select **TLS**.
   d. In the 'Enable Proxy Keep Alive' drop-down list, select **Using Options**, and then in the 'Proxy Load Balancing Method' drop-down list, select **Round Robin**.

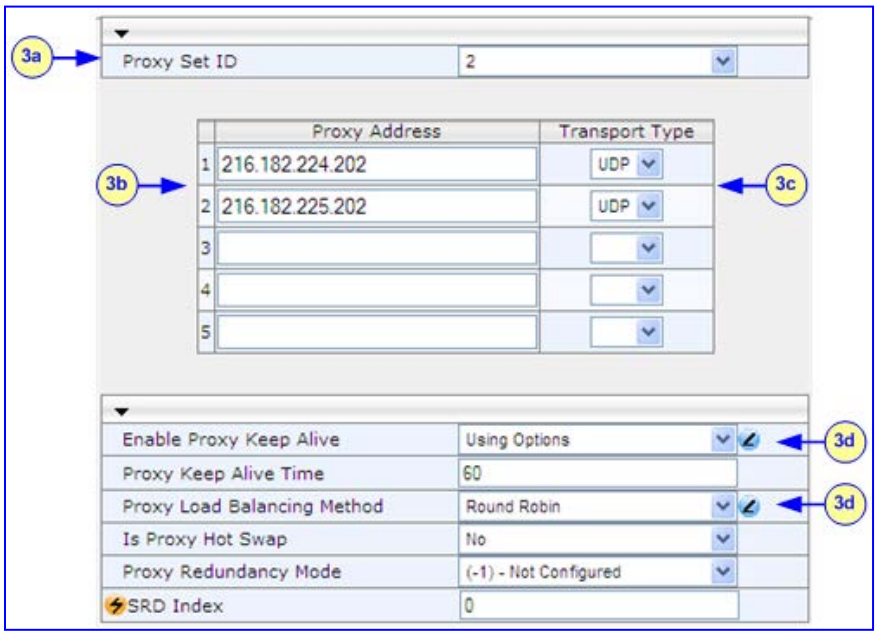
Figure 19-69: Proxy Set ID #1 for ITSP-A



3. Configure Proxy Set ID **#2** for ITSP-B:
   a. From the 'Proxy Set ID' drop-down list, select **2**.
   b. In the 'Proxy Address' column, enter the IP addresses of the ITSP-B SIP trunk (e.g., 216.182.224.202 and 216.182.225.202).
   c. From the 'Transport Type' drop-down list corresponding to the IP address entered above, select **UDP**.

d. In the 'Enable Proxy Keep Alive' drop-down list, select "Using Options", and then in the 'Proxy Load Balancing Method' drop-down list, select **Round Robin**.

Figure 19-70: Proxy Set ID #2 for ITSP-B



4. Configure Proxy Set ID #3 for the IP-PBX:

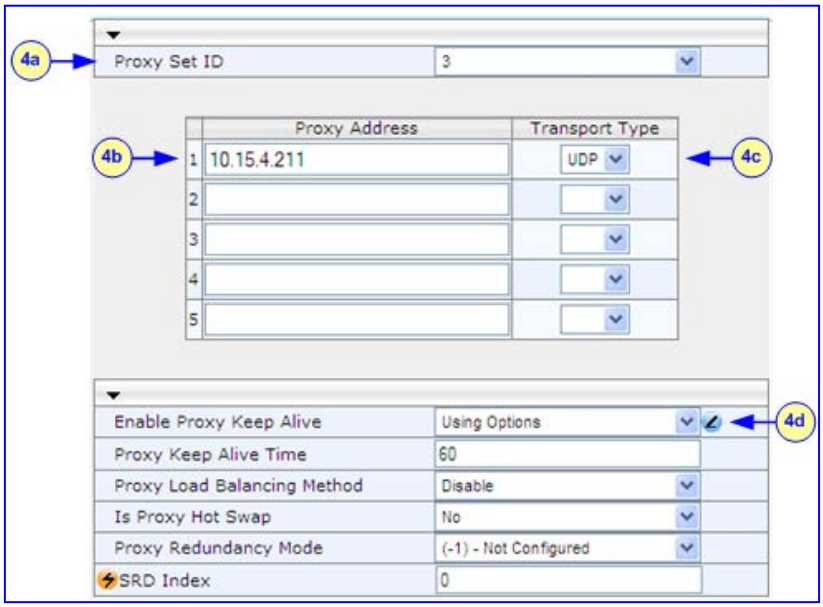
a. From the 'Proxy Set ID' drop-down list, select **3**.

b. In the 'Proxy Address' column, enter the IP address of the IP-PBX (e.g., 10.15.4.211).

c. From the 'Transport Type' drop-down list corresponding to the IP address entered above, select **UDP**".

d. In the 'Enable Proxy Keep Alive' drop-down list, select **Using Options** – this is used in Survivability mode for remote IP-PBX users.

Figure 19-71: Proxy Set ID #3 for the IP-PBX



19.10.2.5 Step 5: Configure the IP Groups

This step describes how to create the IP Groups for the following entities in the network:

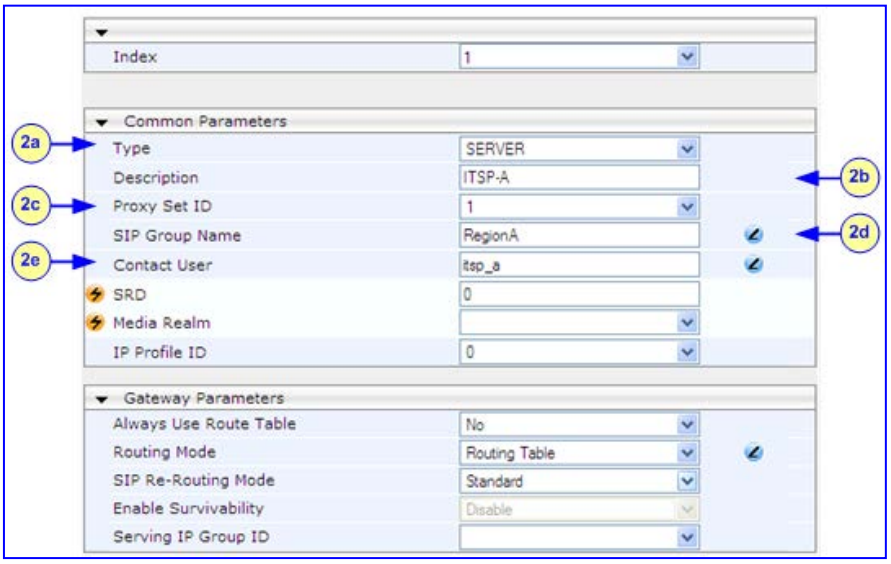
- ITSP-A SIP trunk
- ITSP-B SIP trunk
- IP-PBX
- IP-PBX remote users

These IP Groups are later used by the device for routing calls.

➢ **To configure the IP Groups:**

1. Open the IP Group Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **IP Group Table**).
2. Define IP Group #**1** for ITSP-A:
   a. From the 'Type' drop-down list, select **SERVER**.
   b. In the 'Description' field, type an arbitrary name for the IP Group (e.g., ITSP A).
   c. From the 'Proxy Set ID' drop-down lists, select **1** (represents the IP addresses, configured in **,** for communicating with this IP Group).
   d. In the 'SIP Group Name' field, enter the host name sent in the SIP Request From\To headers for this IP Group, as required by ITSP-A (e.g., RegionA).
   e. Contact User = name that is sent in the SIP Request's Contact header for this IP Group (e.g., ITSP-A).

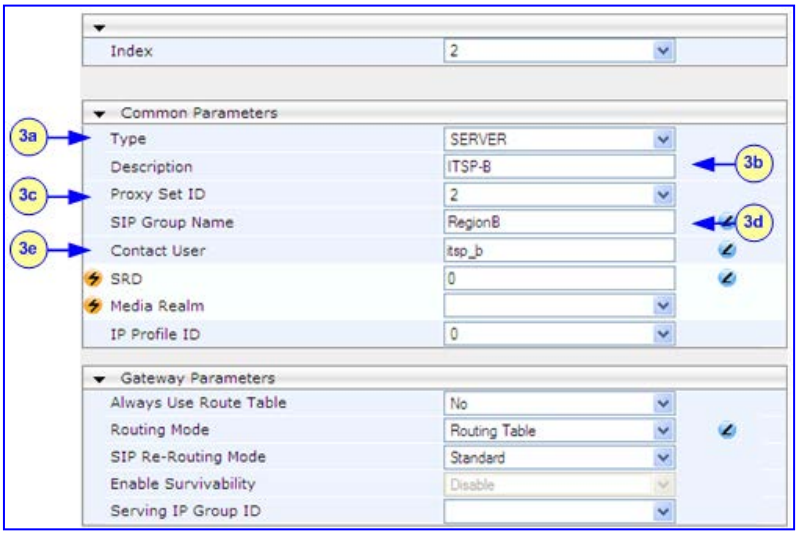
Figure 19-72: Defining IP Group 1



3. Define IP Group #**2** for ITSP-B:
   a. From the 'Type' drop-down list, select **SERVER**.
   b. In the 'Description' field, type an arbitrary name for the IP Group (e.g., ITSP B).
   c. From the 'Proxy Set ID' drop-down lists, select **2** (represents the IP addresses, configured in **,** for communicating with this IP Group).
   d. In the 'SIP Group Name' field, enter the host name sent in SIP Request From\To headers for this IP Group, as required by ITSP-B (e.g., RegionB).

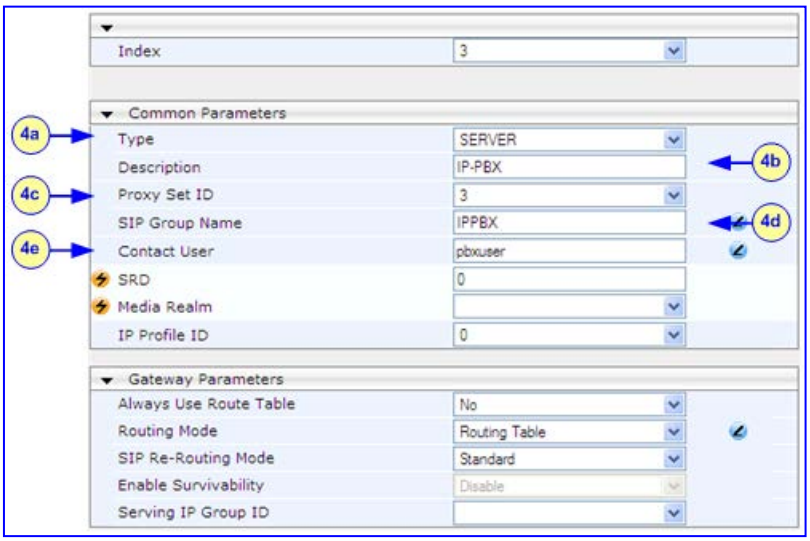
e. Contact User = name that is sent in the SIP Request Contact header for this IP Group (e.g., ITSP-B).

Figure 19-73: Defining IP Group 2



4. Define IP Group #**3** for the IP-PBX:
   a. From the 'Type' drop-down list, select **SERVER**.
   b. In the 'Description' field, type an arbitrary name for the IP Group (e.g., IP-PBX).
   c. From the 'Proxy Set ID' drop-down lists, select **3** (represents the IP address, configured in **,** for communicating with this IP Group).
   d. In the 'SIP Group Name' field, enter the host name that is sent in SIP Request From\To headers for this IP Group (e.g., IPPBX).
   e. Contact User = name that is sent in the SIP Request Contact header for this IP Group (e.g., PBXUSER).

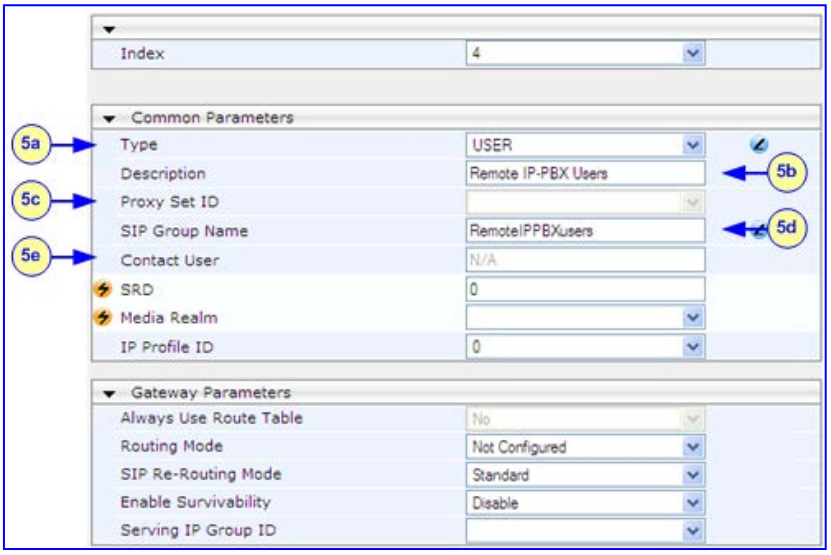
Figure 19-74: Defining IP Group 3



5. Define IP Group #**4** for the remote IP-PBX users:
   a. From the 'Type' drop-down list, select **USER**.
   b. In the 'Description' field, type an arbitrary name for the IP Group (e.g., IP-PBX).
   c. In the 'SIP Group Name' field, enter the host name that is used internal in the device's database for this IP Group (e.g., RemoteIPPBXusers).

d. From the 'Serving IP Group ID' drop-down list, select **3** (i.e. the IP Group for the IP-PBX).

Figure 19-75: Defining IP Group 4



Note: No Serving IP Groups are defined for ITSP-A and ITSP-B. Instead, the Outbound IP Routing table (see 'Step 9: Configure Outbound IP Routing' on page 408) is used to configure outbound call routing for calls originating from these ITSP IP Groups.

19.10.2.6 Step 6: Configure the Account Table

The Account table is used by the device to register to an ITSP on behalf of the IP-PBX. As described previously, the ITSP's requires registration and authentication to provide service. For the example, the Served IP Group is the IP-PBX (IP Group ID #3) and the Serving IP Groups are the two ITSP's (IP Group ID's #1 and #2).

➢ **To configure the Account table:**

1. Open the Account Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Account Table**).

Figure 19-76: Defining Accounts for Registration

2. Configure Account ID **#1** for IP-PBX authentication and registration with ITSP-A:
   - In the 'Served IP Group' field, enter "3" to indicate that authentication is performed on behalf of IP Group #3 (i.e., the IP-PBX).
   - In the 'Serving IP Group' field, enter "1" to indicate that registration/authentication is with IP Group #1 (i.e., ITSP-A).

- In the 'Username', enter the SIP username for authentication supplied by ITSP-A (e.g., itsp_a).
- In the 'Password' field, enter the SIP password for authentication supplied by ITSP-A (e.g., 12345).
- In the 'Register' field, enter "1" to enable registration with ITSP-A.

3. Configure Account ID **#2** for IP-PBX registration) with ITSP-B Registrar server:
   - In the 'Served IP Group' field, enter "3" to indicate that registration is performed on behalf of IP Group #3 (i.e., the IP-PBX).
   - In the 'Serving IP Group' field, enter "2" to indicate that registration is with IP Group #3 (e.g., ITSP-B).
   - In the 'Username', enter the SIP username for the registration/authentication supplied by ITSP-B (e.g., itsp_b).
   - In the 'Password' field, enter the SIP password for registration/authentication supplied by ITSP-B (e.g., 11111).
   - In the 'Register' field, enter "1" to enable registration with ITSP-B.

19.10.2.7 Step 7: Configure IP Profiles for Voice Coders

Since different voice coders are used by the IP-PBX (G.711) and the ITSP's (G.723), you need to define two IP Profiles:

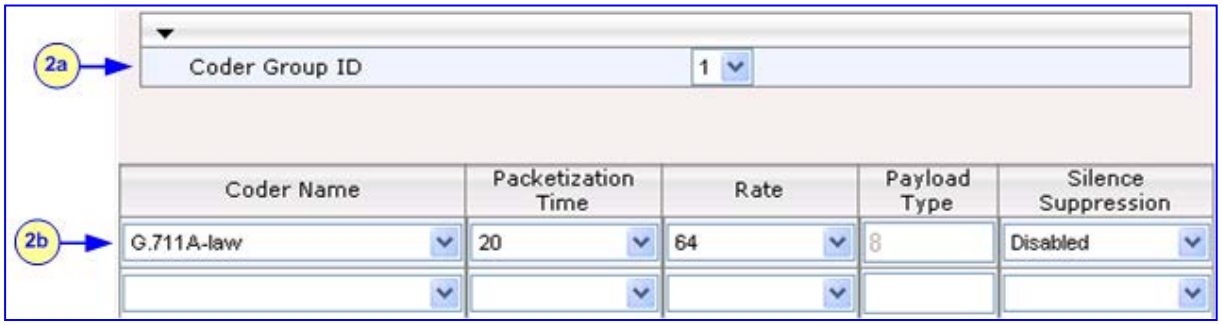
- Profile ID #1 - configured with G.711 for the IP-PBX
- Profile ID #2 - configured with G.723 for the ITSP's

These profiles are later used in the Inbound IP Routing table and Outbound IP Routing table.

➢ **To configure IP Profiles for voice coders:**

1. Open the Coder Group Settings page (**Configuration** tab > **VoIP** menu > **Coders And Profiles** submenu > **Coders Group Settings**)
2. Configure Coder Group ID **#1** for the IP-PBX (as shown in the figure below):
   a. From the 'Coder Group ID' drop-down list, select **1**.
   b. From the 'Coder Name' drop-down list, select **G.711A-law**.
   c. Click **Submit**.

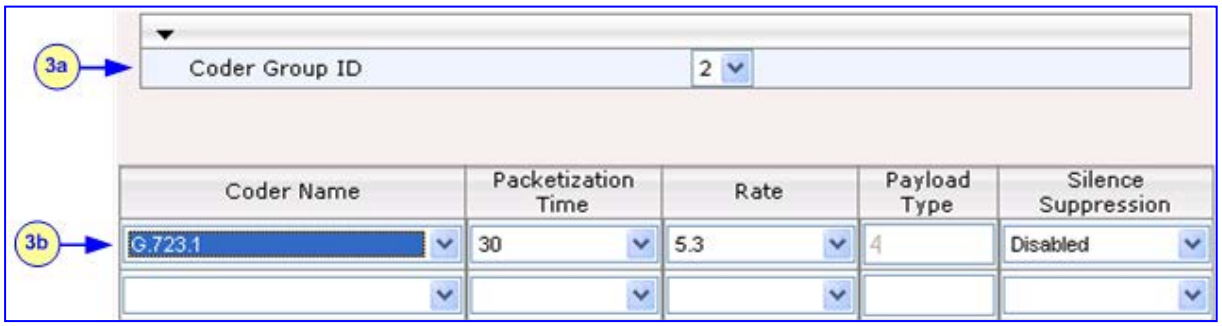
Figure 19-77: Defining Coder Group ID 1



3. Configure Coder Group ID #2 for the ITSP's (as shown in the figure below):
   a. From the 'Coder Group ID' drop-down list, select **2**.
   b. From the 'Coder Name' drop-down list, select **G.723.1**.

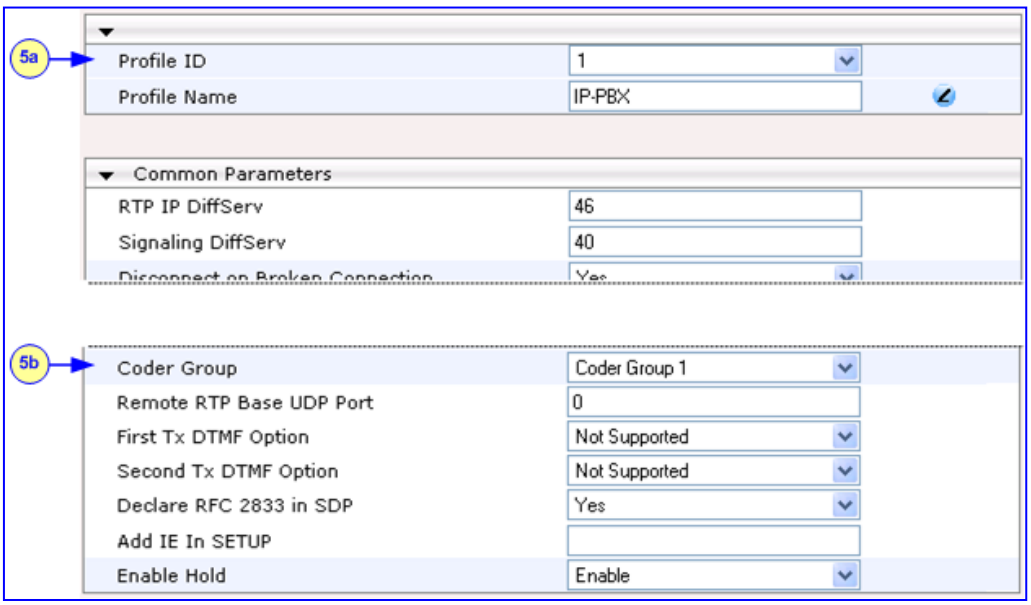
c. Click **Submit**.

Figure 19-78: Defining Coder Group ID 2



4. Open the IP Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders And Profiles** submenu > **IP Profile Settings**).
5. Configure Profile ID #1 for the IP-PBX (as shown below):
   a. From the 'Profile ID' drop-down list, select **1**.
   b. From the 'Coder Group' drop-down list, select **Coder Group 1**.
   c. Click **Submit**.

Figure 19-79: Defining IP Profile ID 1



6. Configure Profile ID #2 for the ITSP's:
   a. From the 'Profile ID' drop-down list, select **2**.
   b. From the 'Coder Group' drop-down list, select **Coder Group 2**.
   c. Click **Submit**.

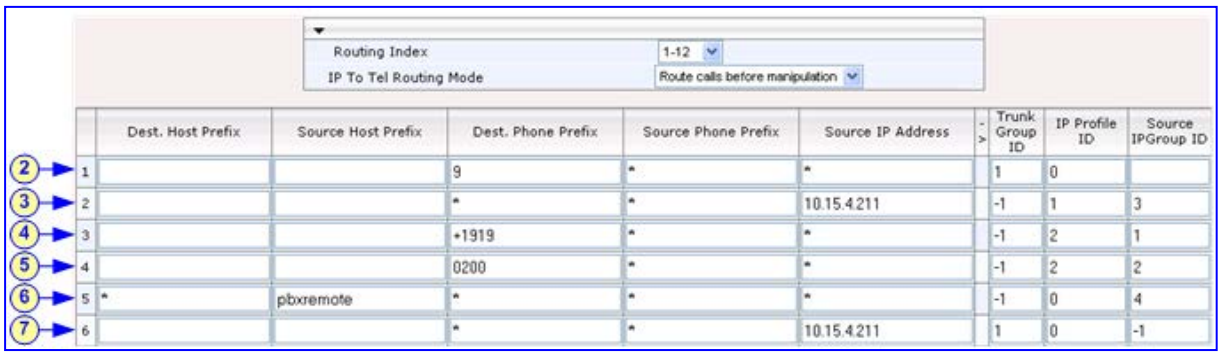
19.10.2.8 Step 8: Configure Inbound IP Routing

This step defines how to configure the device for routing inbound (i.e., received) IP-to-IP calls. The table in which this is configured uses the IP Groups that you defined in 'Step 5: Configure the IP Groups' on page 402.

➢ **To configure inbound IP routing:**

1. Open the Inbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **IP to Trunk Group Routing**).

Figure 19-80: Defining Inbound IP Routing Rules



2. **Index #1:** routes calls with prefix 9 (i.e., local calls) dialed from IP-PBX users to the local PSTN:
   - 'Dest Phone Prefix': enter "9" for the dialing prefix for local calls.
   - 'Trunk Group ID': enter "1" to indicate that these calls are routed to the Trunk (belonging to Trunk Group #1) connected between the device and the local PSTN network.
3. **Index #2:** identifies IP calls received from the IP-PBX as IP-to-IP calls and assigns them to the IP Group ID configured for the IP-PBX:
   - 'Dest Phone Prefix': enter the asterisk (*) symbol to indicate all destinations.
   - 'Source IP Address': enter the IP address of the IP-PBX (i.e., 10.15.4.211).
   - 'Trunk Group ID': enter "-1" to indicate that these calls are IP-to-IP calls.
   - 'IP Profile ID': enter "1" to assign these calls to Profile ID #1 to use G.711.
   - 'Source IP Group ID': enter "3" to assign these calls to the IP Group pertaining to the IP-PBX.
4. **Index #3:** identifies IP calls received from ITSP-A as IP-to-IP calls and assigns them to the IP Group ID configured for ITSP-A:
   - 'Dest Phone Prefix': ITSP-A assigns the Enterprise a range of numbers that start with +1919. Enter this prefix to indicate calls received from this ITSP.
   - 'Trunk Group ID': enter "-1" to indicate that these calls are IP-to-IP calls.
   - 'IP Profile ID': enter "2" to assign these calls to Profile ID #2 to use G.723.
   - 'Source IP Group ID': enter "1" to assign these calls to IP Group pertaining to ITSP-A.
5. **Index #4:** identifies IP calls received from ITSP-B as IP-to-IP calls and assigns them to the IP Group ID configured for ITSP-B:
   - 'Dest Phone Prefix': ITSP-B assigns the Enterprise a range of numbers that start with 0200. Enter this prefix to indicate calls coming from this ITSP.
   - 'Trunk Group ID': enter "-1" to indicate that these calls are IP-to-IP calls.
   - 'IP Profile ID': enter "2" to assign these calls to Profile ID #2 to use G.723.

- 'Source IP Group ID': enter "2" to assign these calls to IP Group pertaining to ITSP-B.

6. **Index #5:** identifies all IP calls received from IP-PBX remote users:
   - 'Source Host Prefix': enter "PBXuser". This is the host name that appears in the From header of the Request URI received from remote IP-PBX users.
   - 'Trunk Group ID': enter "-1" to indicate that these calls are IP-to-IP calls.
   - 'Source IP Group ID': enter "4" to assign these calls to the IP Group pertaining to the remote IP-PBX users.
7. **Index #6:** is used for alternative routing. This configuration identifies all IP calls received from the IP-PBX and which can't reach the ITSP's servers (e.g. loss of connection with ITSP's) and routes them to the local PSTN network:
   - 'Dest Phone Prefix': enter the asterisk (*) symbol to indicate all destinations.
   - 'Source IP Address': enter the IP address of the IP-PBX (i.e., 10.15.4.211).
   - 'Trunk Group ID': enter "1" to route these calls to the Trunk Group ID configured for the Trunk connected to the device and interfacing with the local PSTN.
   - 'Source IP Group ID': enter "-1" to indicate that these calls are not assigned to any source IP Group.

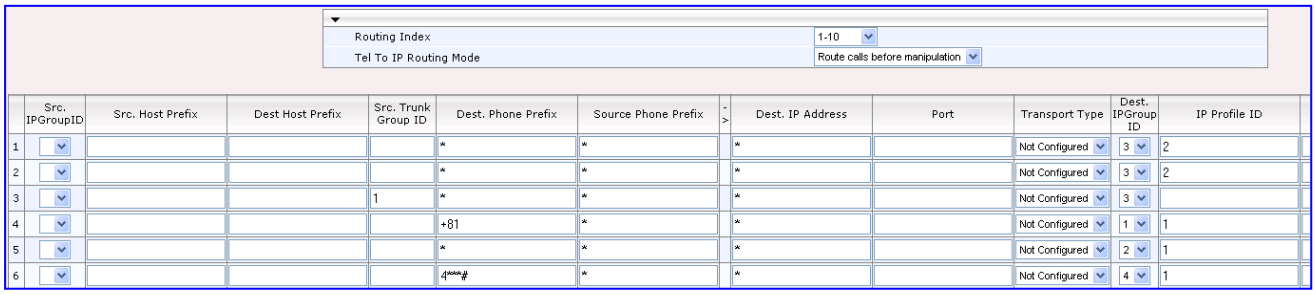
19.10.2.9 Step 9: Configure Outbound IP Routing

This step defines how to configure the device for routing outbound (i.e., sent) IP-to-IP calls. In our example scenario, calls from both ITSP's must be routed to the IP-PBX, while outgoing calls from IP-PBX users must be routed according to destination. If the calls are destined to the Japanese market, then they are routed to ITSP-B; for all other destinations, the calls are routed to ITSP-A. This configuration uses the IP Groups defined in 'Step 5: Configure the IP Groups' on page 402 and IP Profiles defined in 'Step 7: Configure IP Profiles for Voice Coders' on page 405.

➢ **To configure outbound IP routing rules:**

1. Open the Outbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **Tel to IP Routing**).

Figure 19-81: Defining Outbound IP Routing Rules



2. **Index #1:** routes IP calls received from ITSP-A to the IP-PBX:
   - 'Source IP Group ID': select **1** to indicate received (inbound) calls identified as belonging to the IP Group configured for ITSP-A.
   - 'Dest Phone Prefix' and 'Source Phone Prefix' : enter the asterisk (*) symbol to indicate all destinations and callers respectively.
   - 'Dest IP Group ID': select **3** to indicate the destination IP Group to where these calls are sent, i.e., to the IP-PBX.
   - 'IP Profile ID': enter "2" to indicate the IP Profile configured for G.723.
3. **Index #2:** routes IP calls received from ITSP-B to the IP-PBX:
   - 'Source IP Group ID': select **2** to indicate received (inbound) calls identified as belonging to the IP Group configured for ITSP-B.

- 'Dest Phone Prefix' and 'Source Phone Prefix': enter the asterisk (*) symbol to indicate all destinations and callers respectively.
- 'Dest IP Group ID': select **3** to indicate the destination IP Group to where these calls are sent, i.e., to the IP-PBX.
- 'IP Profile ID': enter "2" to indicate the IP Profile configured for G.723.

4. **Index #3:** routes calls received from the local PSTN network to the IP-PBX:
   - 'Source Trunk Group ID': enter "1" to indicate calls received on the trunk connecting the device to the local PSTN network.
   - 'Dest IP Group ID': select **3** to indicate the destination IP Group to where the calls must be sent, i.e., to the IP-PBX.
5. **Index #4:** routes IP calls received from the IP-PBX to ITSP-A:
   - 'Source IP Group ID': select **3** to indicate received (inbound) calls identified as belonging to the IP Group configured for the IP-PBX.
   - 'Dest Phone Prefix': enter "+81" to indicate calls to Japan (i.e., with prefix +81).
   - 'Source Phone Prefix': enter the asterisk (*) symbol to indicate all sources.
   - 'Dest IP Group ID': select **1** to indicate the destination IP Group to where the calls must be sent, i.e., to ITSP-A.
   - 'IP Profile ID': enter "1" to indicate the IP Profile configured for G.711.
6. **Index #5:** routes IP calls received from the IP-PBX to ITSP-B:
   - 'Source IP Group ID': select **3** to indicate received (inbound) calls identified as belonging to the IP Group configured for the IP-PBX.
   - 'Dest Phone Prefix' and 'Source Phone Prefix': enter the asterisk (*) symbol to indicate all destinations (besides Japan) and all sources respectively.
   - 'Dest IP Group ID': select **2** to indicate the destination IP Group to where the calls must be sent, i.e., to ITSP-A.
   - 'IP Profile ID': enter "1" to indicate the IP Profile configured for G.711.
7. **Index #6:** routes dialed calls (four digits starting with digit 4) from IP-PBX to remote IP-PBX users. The device searches its database for the remote users registered number, and then sends an INVITE to the remote user's IP address (listed in the database):
   - 'Source IP Group ID': select **3** to indicate received (inbound) calls identified as belonging to the IP Group configured for the IP-PBX.
   - 'Dest Phone Prefix': enter the "4xxx#" to indicate all calls dialed from IP-PBX that include four digits and start with the digit 4.
   - 'Dest IP Group ID': select **4** to indicate the destination IP Group to where the calls must be sent, i.e., to remote IP-PBX users.
   - 'IP Profile ID': enter "1" to indicate the IP Profile configured for G.711.

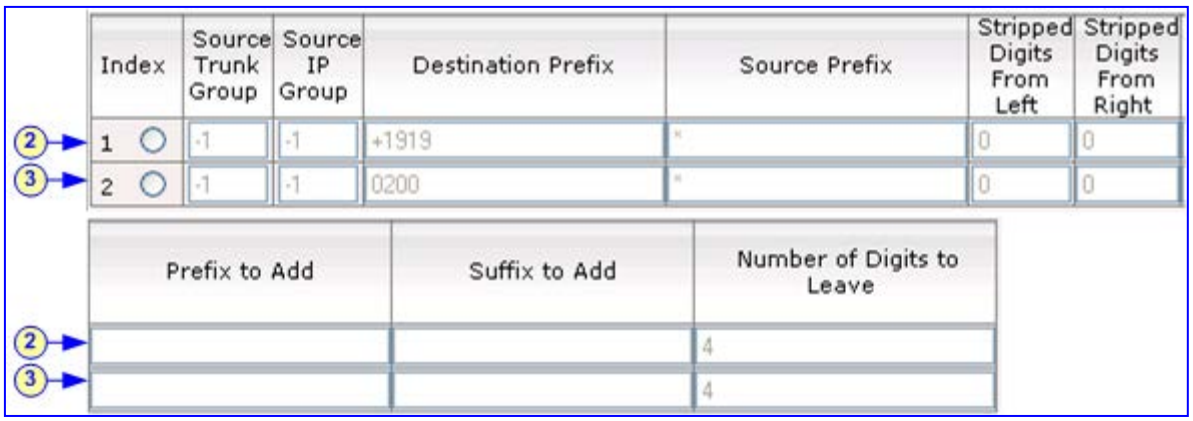
19.10.2.10 Step 10: Configure Destination Phone Number Manipulation

This step defines how to manipulate the destination phone number. The IP-PBX users in our example scenario use a 4-digit extension number. The incoming calls from the ITSP's have different prefixes and different lengths. This manipulation leaves only the four digits of the user's destination number coming from the ITSP's.

➢ **To configure destination phone number manipulation:**

1. Open the Destination Phone Number Manipulation Table for IP -> Tel calls page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu > **Dest Number Tel->IP**).

Figure 19-82: Defining Destination Phone Number Manipulation Rules



2. **Index #1:** defines destination number manipulation of IP calls received from ITSP-A. The phone number of calls received with prefix +1919 (i.e., from ITSP-A) are removed except for the last four digits:
   - 'Destination Prefix': enter the prefix "+1919".
   - 'Source Prefix': enter the asterisk (*) symbol to indicate all sources.
   - 'Number of Digits to Leave': enter "4" to leave only the last four digits.
3. **Index #2:** defines destination number manipulation of IP calls received from ITSP-B. The phone number of calls received with prefix 0200 (i.e., from ITSP-B) are removed except for the last four digits**:**
   - 'Destination Prefix': enter the prefix "0200".
   - 'Source Prefix': enter the asterisk (*) symbol to indicate all sources.
   - 'Number of Digits to Leave': enter "4" to leave only the last four digits.

20 Session Border Controller

This section provides a detailed description of the device's SBC application.

Note: For guidelines on how to deploy your E-SBC device based on network topology and for SBC deployment examples, please refer to the *Typical SBC Deployments Guide* document.

20.1 SBC Overview

This section provides a detailed description of the device's SBC application.

This section includes the following subsections:

- Overview of the SBC application (see 'Overview' on page 411)
- SIP networking definitions (see 'SIP Network Definitions' on page 413)
- SIP dialog-initiation process (see 'SIP Dialog Initiation Process' on page 413)
- User registration and the device's database (see 'User Registration and Internal Database' on page 421)
- Media handling (see 'SBC Media Handling' on page 424)
- SBC Dialog Admission Control (see 'SIP Dialog Admission Control' on page 434)
- Handling SIP 3xx Redirect Responses (see 'Handling SIP 3xx Redirect Responses' on page 435)
- SIP Diversion and History-Info headers interworking (see 'Interworking SIP Diversion and History-Info Headers' on page 436)

20.1.1 Overview

The SBC application provides the following main features:

- NAT traversal (see 'NAT Traversal' on page 412)
- VoIP firewall and security for signaling and media (see 'VoIP Firewall' on page 412)
- Topology hiding (see 'Topology Hiding' on page 412)
- SIP normalization (see 'SIP Normalization' on page 412)
- Survivability (see 'Survivability' on page 413)
- Routing (see 'SIP Network Definitions' on page 413 and 'SIP Dialog Initiation Process' on page 413):
  - IP-to-IP routing translations of SIP, UDP, TCP, TLS (when extensive transcoding is not required)
  - Load balancing and redundancy of SIP servers
  - Routing according to Request-URI\Specific IP address\Proxy\FQDN
  - Alternative routing
  - Routing between different Layer-3 networks (e.g., LAN and WAN)
- Load balancing\redundancy of SIP servers
- Internet Telephony Service Providers (ITSP) accounts
- SIP URI user and host name manipulations (see 'SIP Dialog Initiation Process' on page 413)
- Coder Transcoding (see 'SBC Media Handling' on page 424)

20.1.1.1 NAT Traversal

The device supports NAT traversal, allowing, for example, communication with ITSPs with globally unique IP addresses, for LAN-to-WAN VoIP signaling (and bearer), using two independent legs. In addition, it also enables communication for "far-end" users located behind a NAT on the WAN. The device supports this by:

- Continually registering far-end users in its dynamic database
- Maintaining remote NAT binding state by frequent registrations, thereby, off-loading far-end registrations from the LAN IP PBX
- Using Symmetric RTP (RFC 4961) to overcome bearer NAT traversal

20.1.1.2 VoIP Firewall

The device provides a firewall for VoIP:

- SIP signaling:
  - Deep and stateful inspection of all SIP signaling packets
  - SIP dialog initiations may be rejected based on values of incoming SIP INVITE message and other Layer-3 characteristics
  - Packets not belonging to an authorized SIP dialog are discarded
- RTP:
  - Opening pinholes (ports) in the device's firewall based on Offer-Answer SDP negotiations
  - Deep packet inspection of all RTP packets
  - Late rouge detection - if a SIP session was gracefully terminated and someone tries to "ride on it" with rouge traffic from the already terminated RTP and SIP context, the VoIP Firewall prevents this from occurring
  - Disconnects call (after user-defined time) if RTP connection is broken
  - Black/White lists for both Layer-3 firewall and SIP classification

20.1.1.3 Topology Hiding

The device intrinsically supports topology hiding, limiting the amount of topology information displayed to external parties. For example, IP addresses of ITSPs' equipment (e.g. proxies, gateways, and application servers) can be hidden from outside parties.

The device's topology hiding is provided by implementing back-to-back user agent (B2BUA) leg routing:

- Strips all incoming SIP Via header fields and creates a new Via value for the outgoing message
- Each leg has its own Route/Record Route set
- Modifies SIP To, From, and Request-URI host names (must be configured using the Message Manipulations table - see 'Configuring Message Manipulations' on page 464)
- Generates a new SIP Call-ID header value (different between legs)
- Changes the SIP Contact header to the device's own address
- Layer-3 topology hiding by modifying source IP address in the SIP IP header

20.1.1.4 SIP Normalization

The device supports SIP normalization, whereby the SBC application can overcome interoperability problems between SIP user agents. This is achieved by the following:

- Manipulation of SIP URI user and host parts

- Connection to ITSP SIP trunks on behalf of an IP-PBX - the device can register and utilize user and password to authenticate for the IP-PBX

20.1.1.5 Survivability

The device's SBC application provides two survivability features:

- Routing calls to alternative routes such as the PSTN
- Routing calls between user agents in the local network using a dynamic database (built according to registrations of SIP user agents)

20.1.2 SIP Network Definitions

The device's SBC application can implement multiple SIP signaling and RTP (media) interfaces. For more information, see 'Mutiple SIP Signaling/Media Interfaces Environment' on page 225.

20.1.3 SIP Dialog Initiation Process

The device's SIP dialog initiation process concerns all incoming SIP dialog initiation requests. This includes SIP methods such as INVITE, SUBSCRIBE, OPTIONS, REFER, INFO, UNSOLICITED NOTIFY, MESSAGE, and REGISTER.

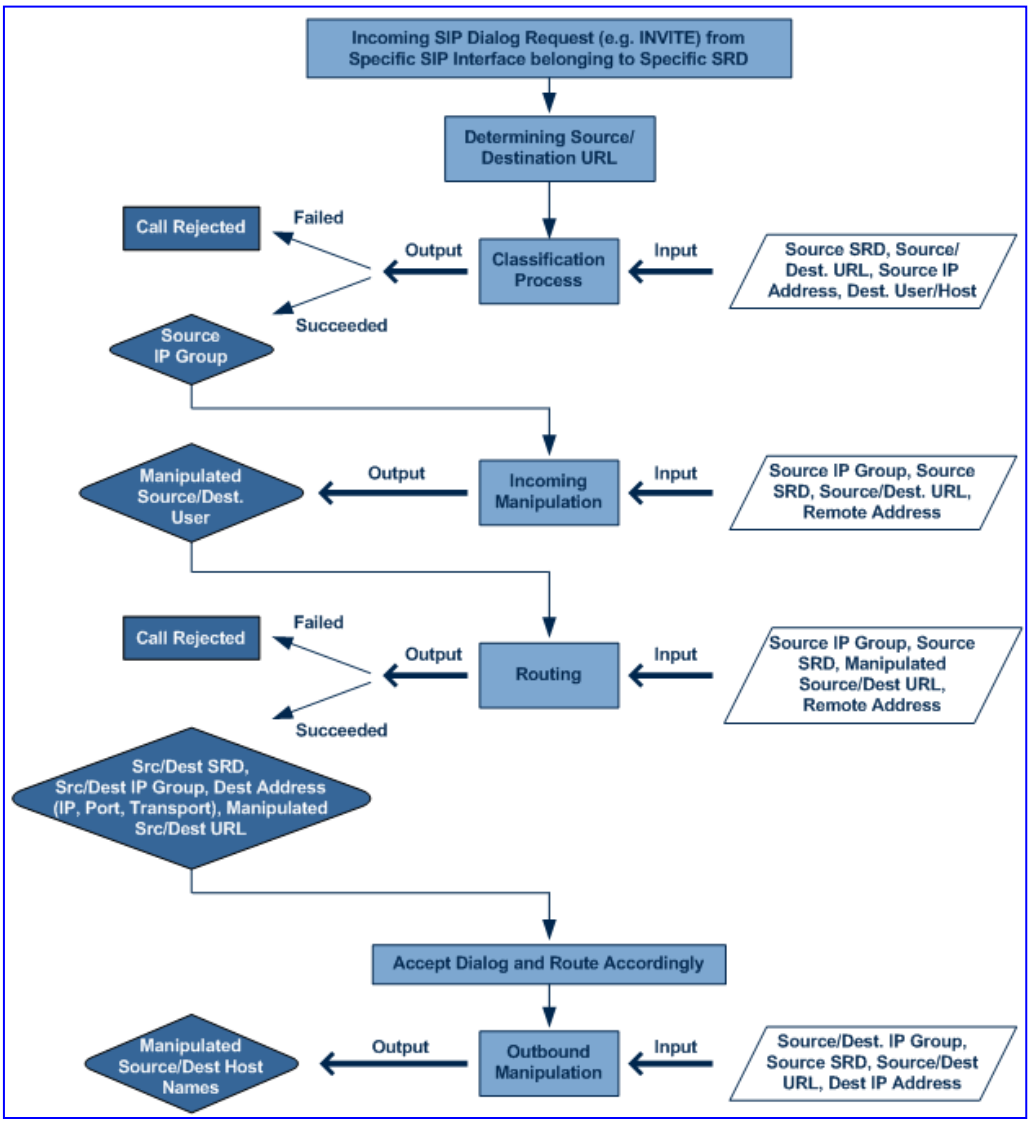
The SIP dialog initiation process consists of the following stages:

- Determining Source and Destination URL (see 'Determining Source and Destination URL' on page 414)
- Classifying Source IP Group (see 'Source IP Group Classification' on page 415)
- IP-to-IP Routing (see 'SBC IP-to-IP Routing' on page 416)
- Manipulating IP-to-IP Inbound and Outbound SIP dialogs (see 'IP-to-IP Inbound and Outbound Manipulation' on page 417)

For a description of the Registration process, see 'User Registration and Internal Database' on page 421.

The flowchart below illustrates this process:

Figure 20-1: Routing Process



20.1.3.1 Determining Source and Destination URL

The SIP protocol has more than one URL in a dialog establishing request that might represent the source and destination URL. When handling an incoming request, the device determines which SIP headers are used for source and destination URLs. Once these URLs are determined, the input user and host are taken from them.

- **INVITE dialogs:**
  - Source URL: if exists, obtained from the P-Asserted\Preferred-Identity header; otherwise, from the From header
  - Destination URL: obtained from the Request-URI
- **REGISTER dialogs:**
  - Source URL: obtained from the To header
  - Destination URL: obtained from the Request-URI

20.1.3.2 Source IP Group Classification

The device supports the configuration of rules for classifying incoming SIP dialog initiating request. The classification identifies the incoming SIP dialog request as belonging to a specific IP Group (from where the SIP dialog request originated).

Classification begins with the device's Registration database, where it searches for a match by checking if the request arrived from a registered user:

- Compares received Contact to the Contact of the registered user
- Compares P-Asserted/From URL to the registered AOR

If the database search is unsuccessful, the classification process proceeds with locating a Proxy Set (associated with the SIP dialog request's IP address, or IP address, port, and transport type if the ClassificationInput parameter is enabled in the Proxy Set) and then finding a match with a corresponding IP Group in the IP Group table. Each IP Group can be classified according to its Proxy Set (if in the IP Group table the parameter ClassifyByProxySet is enabled). If enabled, the device classifies Requests arriving from the IP Group's Proxy Set as coming from this IP Group. The classification is done according to the Proxy IP list (in case of host names, then according to the dynamically resolved IP address list). Note that this classification is not relevant in cases where multiple IP Groups use the same Proxy Set.

If classification based on Proxy Set is unsuccessful, the device proceeds to the Classification table, which searches for a source IP Group based on the following matching rules: Source IP Address, Source Username Prefix, Source Host Prefix, Destination Username Prefix, Destination Host Prefix, and Source SRD.

If the above classification process fails to determine the source IP Group to which the incoming packet belongs, the call can either be rejected, or allowed and processed (by assigning it to the default IP Group of the default SRD). This last classification is determined by the parameter AllowUnclassifiedCalls.

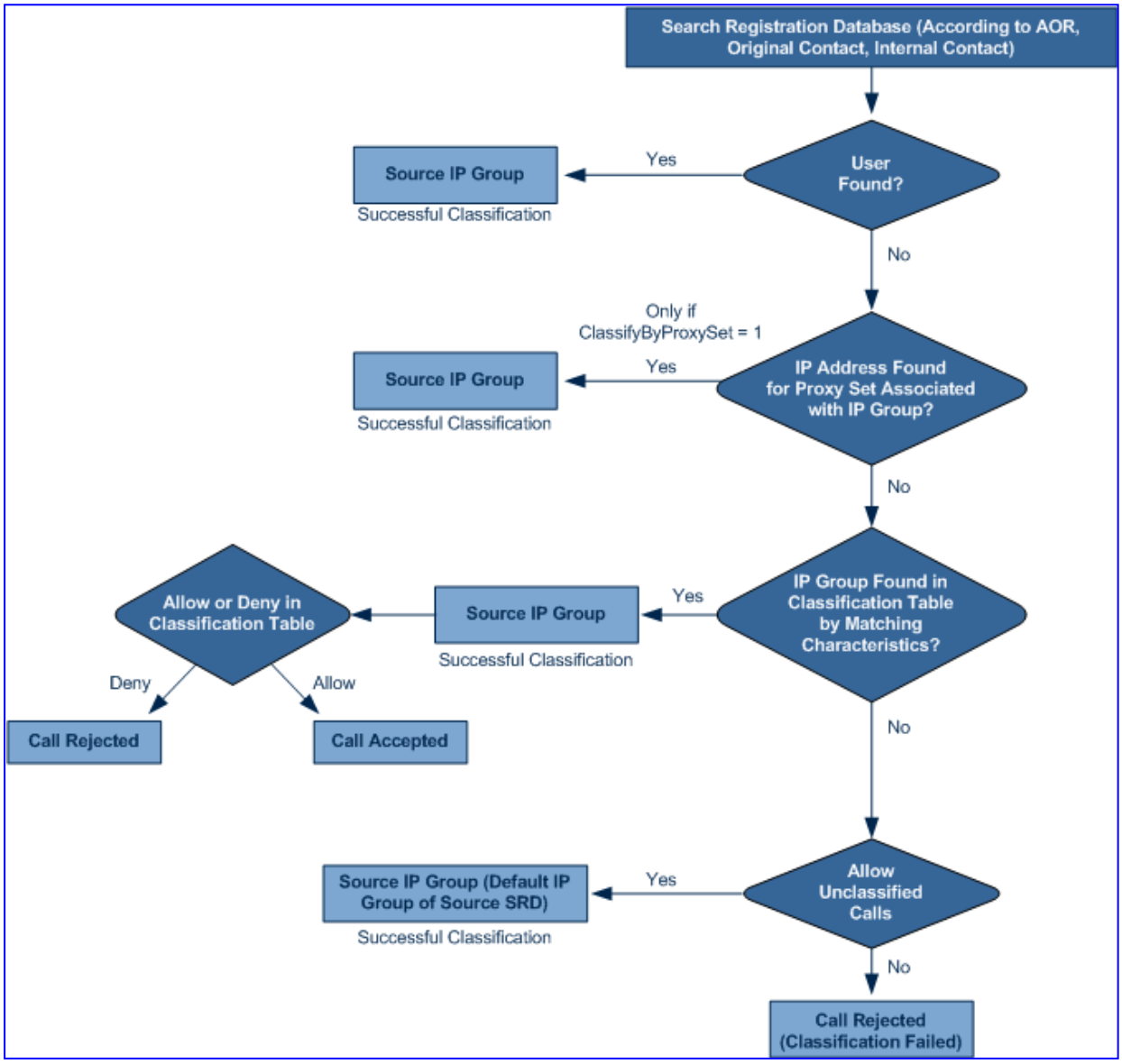
This IP Group is afterwards used for the following purposes:

- Input for the manipulation and routing processes
- Defining SIP behavior and IP Profile, Media Realm and matching account

Note: Incoming REGISTER messages are recorded in the device's database and sent to a destination only if they are associated with a source IP Group that is of USER type.

The flowchart below illustrates the classification process:

Figure 20-2: Classification Process (Identifying IP Group or Rejecting Call)



20.1.3.3 SBC IP-to-IP Routing

The device's SBC application employs a comprehensive and flexible routing scheme:

- Routing rules according to Layer-3/4 and SIP characteristics
- Routing to different destination types:
  - Request-URI (of incoming SIP dialog initiating requests)
  - Specific destination IP address (based on IP address, host name, port, transport type, and/or SRD). Routing to a host name can be resolved using NAPTR/SRV/A-Record.
  - Specific FQDN (NAPTR/SRV/A-Record Resolutions)
  - Registered User Contact listed in the device's database (only for USER-type IP Groups)

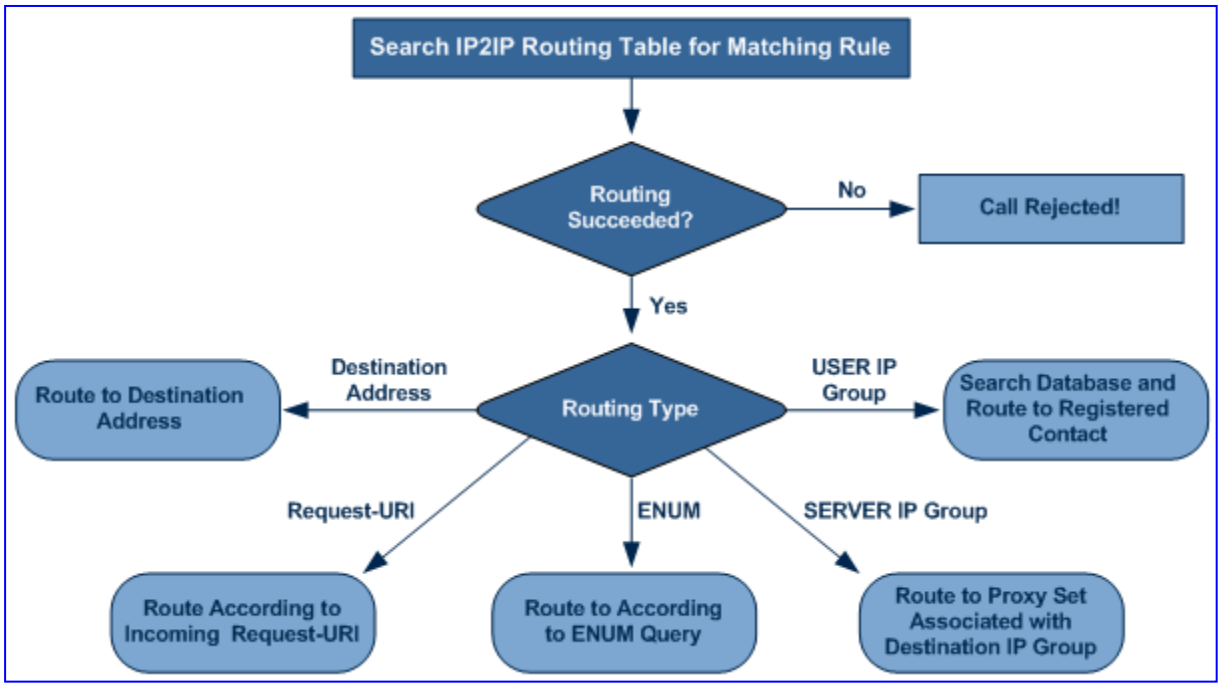
- Destination IP Group (address defined by Proxy Set associated with the IP Group) with the ability of load balancing and redundancy
- ENUM query

- Alternative Routing
- Routing between two different Layer-3 networks
- Transport protocol translator (UDP to TCP to TLS)
- Source and destination user name manipulation (pre/post routing)

The device's IP-to-IP routing rules are configured in the IP-to-IP Routing table. This table provides enhanced IP-to-IP call routing capabilities for routing received SIP messages such as INVITE messages to a destination IP address. The routing rule must match one of the following input characteristics: Source IP Group, Source Phone Prefix, and/or Source Host Prefix.

For all destination types listed above except destination IP Group, the IP Group can optionally be itself, configured to provide destination SRD and/or IP Profile. If neither destination SRD nor destination IP Group are defined, the destination SRD is the source SRD and the destination IP Group is its default IP Group.

Figure 20-3: IP-to-IP Routing Types

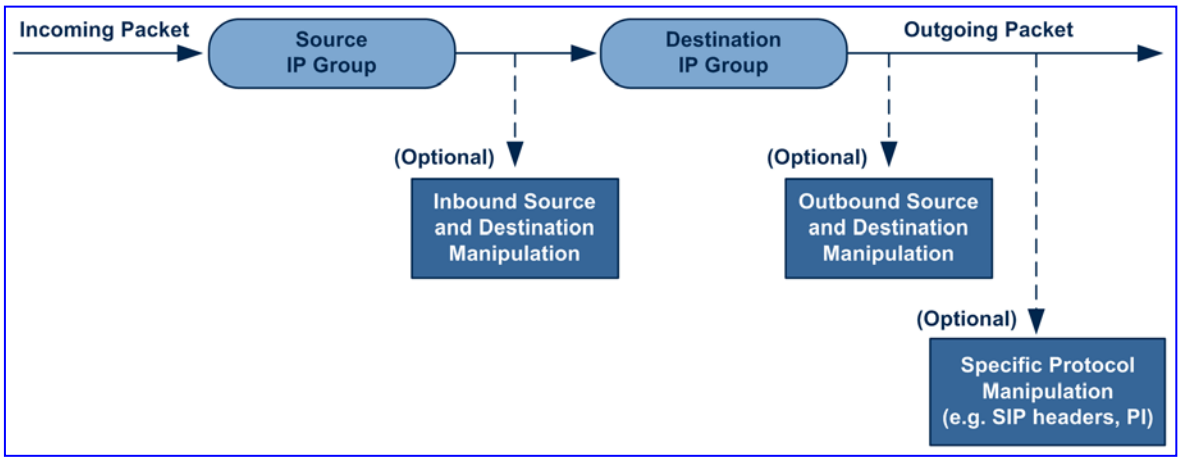


20.1.3.4 IP-to-IP Inbound and Outbound Manipulation

The device supports SIP URI user part (source and destination) manipulations for inbound and outbound routing. These manipulations can be applied to a source IP group, source and destination host and user prefixes, and/or user-defined SIP request (e.g., INVITE, OPTIONS, SUBSCRIBE, and/or REGISTER). Since outbound manipulations are performed after routing, the outbound manipulation rule matching can also be done by destination IP Group.

Manipulated destination user and host are performed on the following SIP headers: Request-URI, To, and Remote-Party-ID (if exists). Manipulated source user and host are performed on the following SIP headers: From, P-Asserted (if exists), P-Preferred (if exists), and Remote-Party-ID (if exists).

Figure 20-4: SIP URI Manipulation in IP-to-IP Routing



In addition, you can restrict source user identity in outgoing SIP dialogs in the Outbound Manipulation table (using the column PrivacyRestrictionMode):

- **[0]** Transparent (default): no device intervention in anything SIP data related to privacy
- **[1]** Don't change privacy: the user identity remains the same as in the incoming INVITE. If a restricted number exists, the restricted presentation is normalized as follows:
  - From URL header - anonymous@anonymous.invalid.
  - If a P-Asserted-Identity header exists (either in the incoming INVITE or added by the device), a Privacy header is added with the value "id".
- **[2]** Restrict: the user identity is restricted (the restricted presentation is as mentioned above).
- **[3]** Remove Restriction: the device attempts to reveal the user identity by setting user values to the From header and removing the privacy "id" value if the Privacy header exists.
- If the From header user is anonymous, the value is taken from the P-Preferred-Identity, P-Asserted-Identity, or Remote-Party-ID header (if exists).

The device identifies an incoming user as restricted if one of the following exists:

- From header user is anonymous.
- P-Asserted-Identity and Privacy headers contain the value "id".

All restriction logic is performed after the user number has been manipulated.

The manipulations are configured using the IPOutboundManipulation and IPInboundManipulation parameters.

Host name (source and destination) manipulations are simply host name substitutions with the names defined for the source and destination IP Groups respectively (if any, in the IP Group table).

Below is an example of a call flow and consequent SIP URI manipulations:

Figure 20-5: SIP INVITE (Manipulations) from LAN to WAN

Incoming INVITE (from LAN)

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
Via: SIP/2.0/UDP 10.2.2.6;branch=z9hGLLLLLan
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=OILAN;paramer1=arik
To: <sip:1000@10.2.2.3;user=phone>
Call-ID: USELLLAN@10.2.2.3
CSeq: 1 INVITE
Contact: <sip:7000@10.2.2.3>
Supported: em,100rel,timer,replaces
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK
User-Agent: Sip Message Generator V1.0.0.5
Content-Type: application/sdp
Content-Length: 155

v=0
o=SMG 791285 795617 IN IP4 10.2.2.6
s=Phone-Call
c=IN IP4 10.2.2.6
t=0 0
m=audio 6000 RTP/AVP 8
a=rtpmap:8 pcma/8000
a=sendrecv
a=ptime:20
```

Outgoing INVITE (to WAN)

```
INVITE sip: 9721000@ITSP;user=phone;x=y;z=a SIP/2.0
Via: SIP/2.0/UDP 212.179.1.12;branch=z9hGWwan
From: <sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;paramer1=arik
To: <sip: 9721000@ITSP;user=phone>
Call-ID: USEVWWAN@212.179.1.12
CSeq: 38 INVITE
Contact: <sip:7000@212.179.1.12>
Supported: em,100rel,timer,replaces
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER
User-Agent: Sip Message Generator V1.0.0.5
Content-Type: application/sdp
Content-Length: 155

v=0
o=SMG 5 9 IN IP4 212.179.1.11
s=Phone-Call
c=IN IP4 212.179.1.11
t=0 0
m=audio 8000 RTP/AVP 8
a=rtpmap:8 pcma/8000
a=sendrecv
a=ptime:20
```

The SIP message manipulations in the example above (contributing to typical topology hiding) are as follows:

SIP Manipulation	From	To
Inbound Source SIP URI User Name	7000	97000 (blue)
Source IP Group Name (SIP URI Host Name)	10.2.2.6	IP_PBX (blue)
Inbound Destination SIP URI User Name	1000	9721000 (red)
Destination IP Group Name (SIP URI Host Name)	10.2.2.3	ITSP (red)

20.1.3.5 SIP Header Manipulation

The device provides enhanced SIP header manipulation, including insertion, removal, and/or modification of SIP headers and parameters. This manipulation is configured in the Message Manipulations table (MessageManipulations parameter). This feature enables the normalization of SIP messaging fields between communicating network segments. For example, it allows service providers to design their own policies on the SIP messaging fields that must be present before a SIP call enters their network. Similarly, enterprises and small businesses may have policies for the information that can enter or leave their networks for policy or security reasons from a service provider. The manipulations can also be implemented to resolve incompatibilities between SIP devices inside the enterprise network.

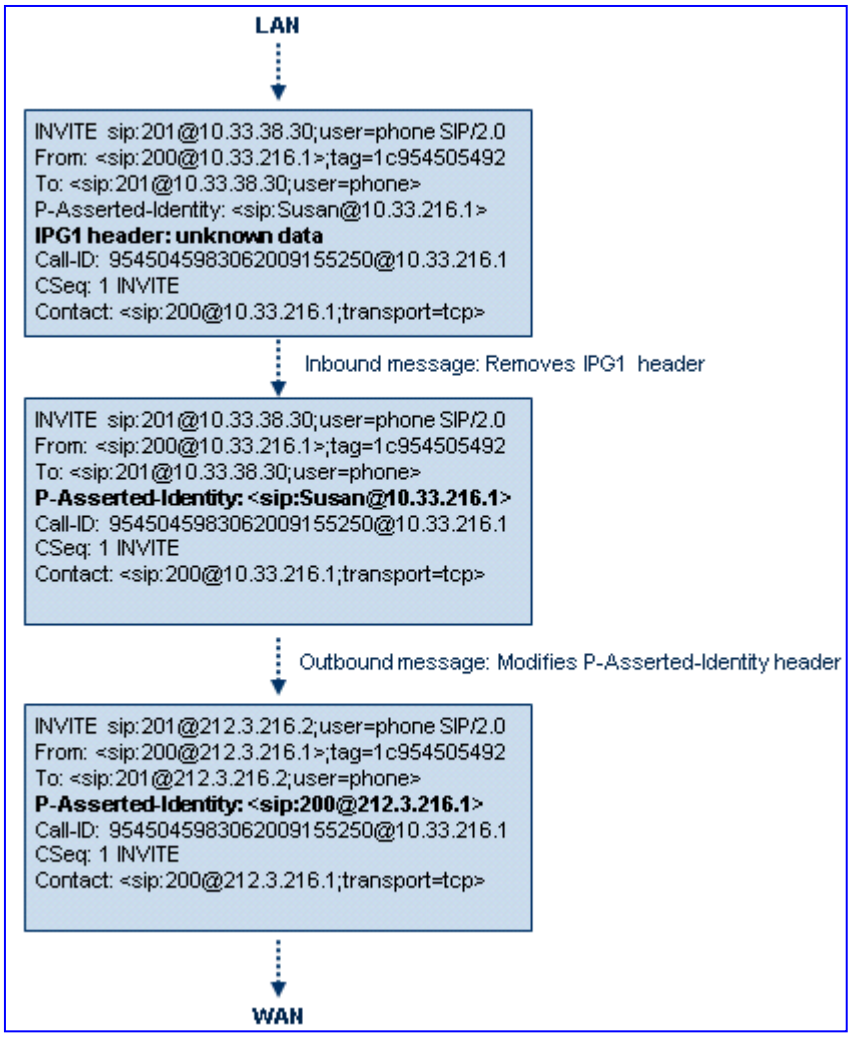
SIP Messaging manipulation supports the following:

- Addition of new headers.
- Removal of headers ("Black list").
- Modification of header components - value, header value (e.g., URI value of the P-Asserted-Identity header can be copied to the From header), call's parameter values.
- Deletion of SIP body (e.g., if a message body isn't supported at the destination network this body is removed).
- Translating one SIP response code to another.
- Topology hiding (generally present in SIP headers such as Via, Record Route, Route

and Service-Route).

- Configurable identity hiding (information related to identity of subscribers for example, P-Asserted-Identity, Referred-By, Identity and Identity-Info).
- Apply conditions per rule - the condition can be on parts of the message or call's parameters.
- Multiple manipulation rules on the same SIP message.

Figure 20-6: SIP Header Manipulation Example



The manipulation is performed on SIP messages according to the Classification table (source/destination of username/host prefixes and source IP address). The manipulation can be performed on message type (Method, Request/Response, and Response type). Message manipulations are performed only after the classification, inbound manipulations and routing are successfully preformed (i.e., manipulations are performed only in the outgoing leg).

SIP Message manipulation rules can be assigned to an IP Group in the IP Group table (IPGroup parameter) and determined whether they must be performed for inbound or outbound messages.

Notes:

- Unknown SIP parts can only be added or removed.
- SIP manipulations do not allow you to remove or add mandatory SIP headers. Only the modify option is available for mandatory headers and is performed only on requests that initiate new dialogs. Mandatory SIP headers include To, From, Via, CSeq, Call-Id, and Max-Forwards. Mandatory SDP headers include v, o, s, t ,c, and m.
- When multiple manipulations rules apply to the same header, the second rule applies to the result string of the first rule.
- Manipulating any value in the message body causes a change to the Content-length header automatically.
- SDP body manipulations are currently not supported.

20.1.4 User Registration and Internal Database

To allow registrations to traverse the SBC, the device must be configured with at least one IP Group of type USER. USER IP Groups represent a group of user agents that share the following characteristics:

- Perform registrations and share the same serving proxy\registrar
- Possess identical SIP and media behavior
- Reside on the same Layer-3 network and are associated with the same SRD

Typically, the device is configured as the user agent's outbound proxy and the device is configured (using the IP2IP Routing table) to route requests received from this IP Group to the serving proxy and vice versa. Survivability can be achieved using the alternative routing feature.

20.1.4.1 Initial Registration Request Processing

Registration requests have different processing policies than other SIP methods:

1. Determining source and destination URL's:
   - The source URL is obtained from the To header
   - The destination URL is obtained from the Request URI
2. Classification: The REGISTER classification process is the same as the general classification process (described in previous sections). The source IP Group must be of type USER. If classification fails or the source IP Group is not of type USER, the registration is rejected.
3. Routing: The REGISTER routing is performed using the IP2IP Routing table:
   - The destination type can be an IP Group, specific IP address, Request-URI, or ENUM query (can also use DNS queries).
   - If the destination IP Group is of type USER, then the registration is not be forwarded. Instead, the device accepts (replies with 200 OK response) or rejects (Reply with 4xx) the request according to the user group policy.
4. Internal registration database: If the source IP Group is of type User and registration succeeds (replied with 200 OK by the IP-PBX), then the device adds a record to its database that identified the specific contact of this specific user (AOR). This record is used later to route requests to this specific user (either in normal or in survivability modes).
5. Alternative Routing: Alternative routing can be configured in the IP2IP Routing table

for REGISTER requests.

6. Inbound Manipulation: The SBC record in the device's database includes the Contact header. Every REGISTER request is added to the database before manipulation, allowing correct user identification in the SBC Classification process for the next received request.
7. Session Admission Control: Applies various limitations on incoming and outgoing REGISTER requests. For example, limiting REGISTER requests from a certain IP Group/SRD. Note that this limitation is only for concurrent register dialogs and not concurrent registrations in the internal database.
8. The device can retain the original value of the SIP Expires header received from the user or proxy, in the outgoing REGISTER message. This feature also applies when the device is in "survivability" state (i.e., REGISTER requests cannot be forwarded to the proxy and is terminated by the device). This is configured by the SBCUserRegistrationTime, SBCProxyRegistrationTime, and SBCSurvivabilityRegistrationTime parameters.
9. By default, the Contact of the outgoing REGISTER is populated with a unique Contact generated by the device and associated with this specific registration. Alternatively, the original user can be retained in the Contact and used in the outgoing REGISTER request (using the SBCKeepContactUserinRegister parameter).

20.1.4.2 Internal Database

The device manages a dynamic database that is updated according to registration requests that traverse the SBC. Each database entry represents a binding between an AOR and one or more contact. Database bindings are added upon successful registration responses. For specific registrations, the AOR is obtained from the SIP To header and the contact is taken from the SIP Contact header.

Database bindings are removed in the following cases:

- Successful de-registration responses (REGISTER with Expires header that equals zero)
- Registration failure responses
- Timeout of the Expires header value (in scenarios where the user agent did not send a refresh registration request)

The device's database can include up to 600 registered SBC users.

The database has the following limitations:

- Maximum of five contacts per AOR
- The same contact cannot belong to more than one AOR
- Contacts with identical URIs and different ports and transport types are not supported (same key is created)
- Multiple contacts in a single REGISTER is not supported
- One database is shared between all USER-type IP Groups

20.1.4.3 Routing using Internal Database

Typically, routing using the database is applicable to all method types other than registrations. To route to a registered user (using the internal dynamic database), the following steps must be taken:

1. An IP2IP Routing rule with the desired input parameters (matching characteristics) and the destination type as IP Group (operation rule).
2. The destination IP Group must be of type USER.
3. To find a match for these specific rules, the device attempts to locate a match between the incoming Request-URI and (according to the description order):

a. Unique contact - the Contact generated by the SBC and sent in the initial registration request to the serving proxy
b. Registered AOR - the AOR of the incoming REGISTER request
c. Registered contact - the Contact of the incoming REGISTER request

If registrations are destined to the database (using the above rules), the device does not attempt to find a database match, but instead replies with 200 OK (used for Survivability). Once a match is found, the request is routed either to the contact received in the initial registration or (if the device identifies that the user agent is behind a NAT) to the source IP address of the initial registration.

20.1.4.4 Registration Refreshes

Registration refreshes are incoming REGISTER requests that are associated with a specific registered user. The association is performed by searching the internal registration database. These refreshes are routed to the serving proxy only if the serving proxy Expires time is about to expire; otherwise, the device responds with a 200 OK without routing the REGISTER. Each such refreshes also refresh the internal timer time set on the device for this specific registration.

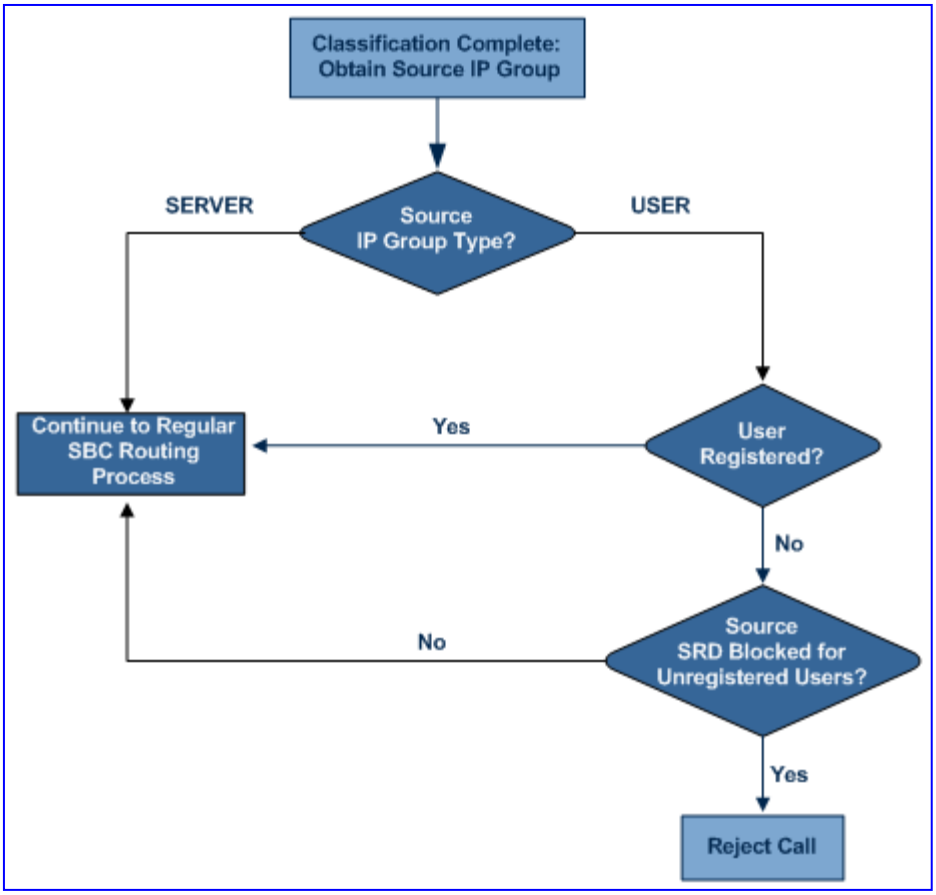
20.1.4.5 Registration Restriction Control

The device provides flexibility in controlling user's registration:

- **Limiting Number of Registrations per Source SRD and/or IP Group:** You can limit the number of users that can register with the device. This limitation can be applied per source IP Group and/or SRD. By default, no limitation exists for registered users. This is configured using the parameters SRD or IPGroup.

- **Blocking Incoming Calls from Unregistered Users:** You can block incoming calls (INVITE requests) from unregistered users (pertaining to USER-type IP Groups). By default, calls from unregistered users are not blocked. This is configured using the parameter SRD. The flowchart below depicts the process for blocking unregistered users. When the call is rejected, the device sends a SIP 500 "Server Internal Error" response to the remote end.

Figure 20-7: Blocking Incoming Calls from Unregistered Users



20.1.5 SBC Media Handling

Media behavior includes anything related to the establishment, management and termination of media sessions within the SIP protocol. Media sessions are created using the SIP "offer"/"answer" mechanism. If successful, the result is a bi-directional media (RTP) flow (e.g. audio, fax, modem, DTMF). Each offer/answer may create more than one media session of different types (e.g. audio and fax). In a SIP dialog, multiple offer/answer transactions may occur, each may change the media sessions characteristics (e.g. IP address, port, coders, media types, and RTP mode). The media capabilities exchanged in an offer/answer transactions include the following:

- Media types (Audio, Secure Audio, Video, Fax, Text...)
- IP addresses and ports of the media flow
- Media flow mode (send receive, receive only, send only, inactive)
- Media coders (coders and their characteristics used in each media flow)
- Other (standard or proprietary) media and session characteristics

Even though the device usually does not change the negotiated media capabilities (mainly performed by the remote user agents), it does examine the media exchange to control negotiated media types (if necessary) and to know how to open the RTP media channels (IP addresses, coder type, payload type etc.).

The device is aware and sometimes active in the offer\answer process due to the following:

- NAT traversal: the device changes the SDP address to be its own address, thereby, resolving NAT problems.
- Firewall and security:
  - RTP pin holes - only RTP packets related to a successful offer\answer negotiation traverse the device: When the device initializes, there are no RTP pin holes opened, this means that each RTP\RTCP packets destined to the device are discarded. Once an offer\answer transaction ends successfully, an RTP pin hole is opened and RTP\RTCP flows between the two remote user agents. Once a pin hole is opened, the payload type and RTP header version is validated for each packet. RTP pin holes close if one of the associated SIP dialogs is closed (may also be due to broken connection).
  - Late rogue detection - once a dialog is disconnected, the related pin holes also disconnect.
  - Deep Packet inspection of the RTP that flows through the opened pin holes.
- Adding of media functionality to SIP user agents:
  - Transcoding (for a description on the transcoding modes, see 'Transcoding Modes' on page 430)
  - Broken connection

According to the above functionalities, the call can be configured to operate in one of the following modes:

- **Media Anchoring without Transcoding (Transparent):** RTP traverses the device with minimal RTP packet changes (no DSP resources needed). This is typically used to solve NAT, firewall, and security issues. In this mode, all the "audio" coders in the received offer are included in the SBC outgoing offer. The Coder Table configuration has no effect on the coders in the outgoing offer. For more information, see 'Media Anchoring without Transcoding (Transparent)' on page 425.
- **Media Anchoring with Transcoding:** RTP traverses the device and each leg uses a different coder or coder parameters (DSP resources are required). For more information, see 'Media Anchoring with Transcoding' on page 426.
- **No Media Anchoring:** The RTP packet flow does not traverse the device. Instead, the two SIP UA's establish a direct RTP/SRTP flow between one another (see 'No Media Anchoring' on page 428).

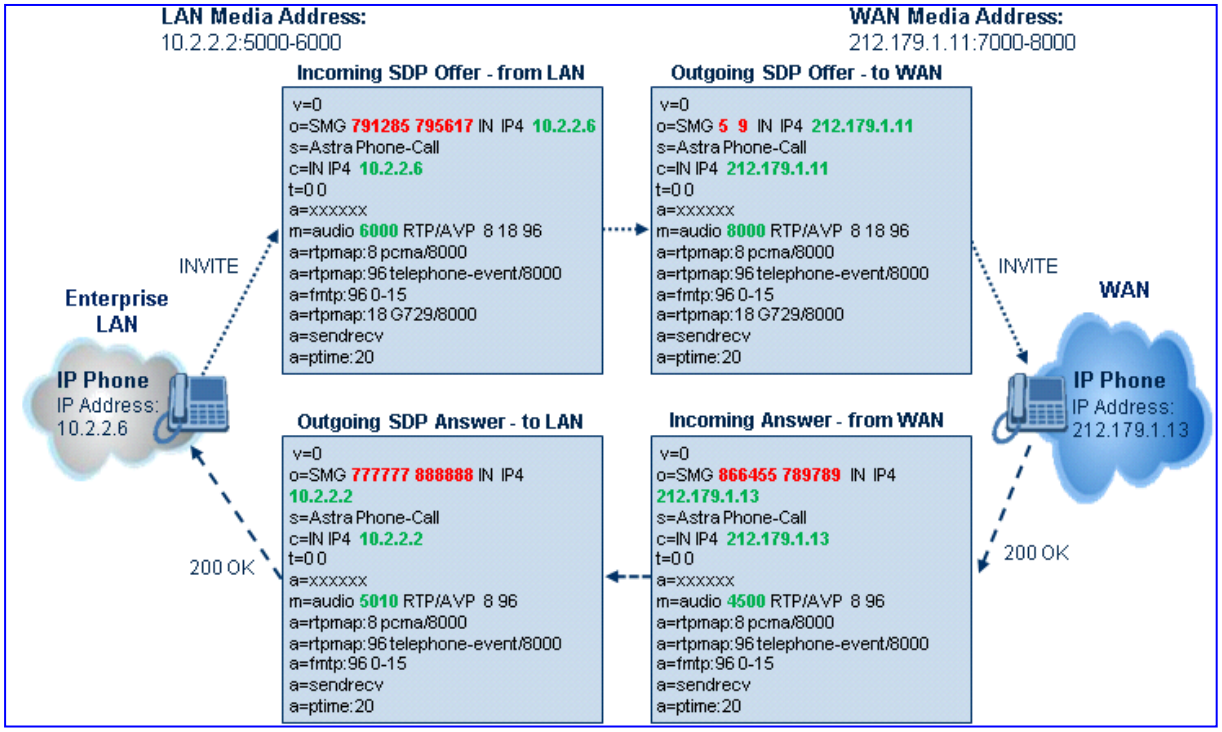
20.1.5.1 Media Anchoring without Transcoding (Transparent)

To direct the RTP to flow through the device (for NAT traversal, firewall and security), all IP address fields in the SDP are modified:

- Origin: IP address, session and version id
- Session connection attribute ('c=' field)
- Media connection attribute ('c=' field)
- Media port number
- RTCP media attribute IP address and port (if the parameter EnableRTCPAttribute is set to 1)

Each SBC leg allocates and uses the device's local ports (e.g., for RTP\RTCP\fax). The local ports are allocated from a Media Realm associated with each leg. The legs are associated with a Media Realm as follows: If the leg's IP Group is configured with a Media Realm, then this is the associated Media Realm; otherwise, the leg's SRD Media Realm is the associated one. The figure below illustrates an example of SDP handling for a call between IP Phone 10.2.2.6 (LAN) and a remote IP Phone 212.179.1.13 ().

Figure 20-8: SDP Offer/Answer Example

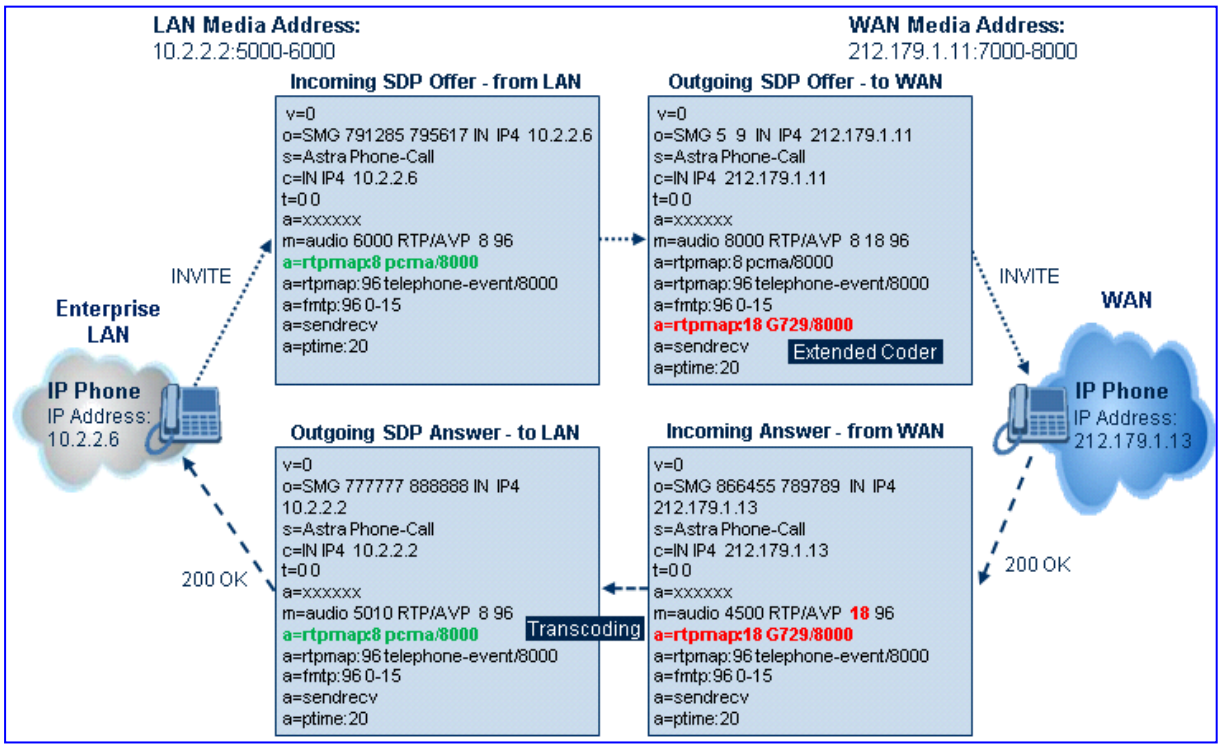


20.1.5.2 Media Anchoring with Transcoding

The device performs transcoding when there are no common coders between the two user agents (i.e., the SDP answer from one user agent doesn't include any coder included in the offer previously sent by the other user agent). For transcoding, the device can be configured to add media capabilities to user agents pertaining to a specific IP Group, and then perform transcoding in cases where the selected coder in the answer SDP is not one that appears in the original offer. The capabilities that can be added are one or more of the device's supported coders and are configured by using the parameter SBCExtensionCodersGroupID (points to a coders list) in the IP Profile table (which is assigned to the IP Group). Therefore, to allow user agents of different IP Groups to communicate with each other (regardless of their capabilities), an extended coders table with at least one coder that is supported by each IP Groups' user agents needs to be assigned to each IP Group. Therefore, each offer destined to specific IP Groups include this coder.

In the scenario depicted in the figure below, the IP phone on the LAN side initiates a call to the IP phone on the WAN. The initial SDP offer (from the LAN leg) includes codec G.711 as its supported codec. Since this is sent to a Destination IP Group that is configured with an extended coder list, on the WAN leg the device adds another supported codec G.729 to the SDP, which is now offered to the WAN IP phone. The WAN IP phone chooses the extended codec (G.729) in its SDP answer to the device's WAN leg. Since this codec was not included in the original incoming offer, the device performs transcoding (between G.729 and G.711) between its two legs, allowing the streaming of media to occur.

Figure 20-9: Transcoding using Extended Coders (Example)



For an SDP offer to provide an extended coder list to a remote user agent, the following prerequisites must be fulfilled:

- An extended coders list has been configured for the user agent's IP Group (i.e., Destination IP Group)
- The incoming offer contains at least one supported coder (otherwise, transcoding can't be performed)
- Both legs have available DSP's
- T.38 doesn't appear in the offer

If the above prerequisites are not met, the SDP offer is sent without the extended coders list. The coders from the extended list are added after the ones from the original offer (decreases transcoding probability). Coders common between the extended coders list and those in the original SDP offer are not added. Transcoding may be performed even in scenarios when the same coder has been chosen - this occurs if the coders use different coder parameters (e.g. rate and packetization time).

The device also supports early media, whereby the first offer\answer transaction is finalized and the media flow starts before the SIP call is connected (before the INVITE 200 OK response). The offer and answer options can be included in the following SIP messages:

- Offer in first INVITE, answer on 180, and no or same answer in the 200 OK
- Offer in first INVITE, answer on 180, and a different answer in the 200 OK (not standard)
- INVITE without SDP, offer in 180, and answer in PRACK

- PRACK and UPDATE transactions can also be used for initiating subsequent offer\answer transactions before the INVITE 200 OK response.
- In a SIP dialog life time, media characteristics after originally determined by the first offer\answer transaction can be changed by using subsequent offer\answer transactions. These transactions may be carried either in UPDATE or ReINVITE SIP transactions. The media handling is similar to the original offer/answer handling. If the offer is rejected by the remote party, then no media changes occur (e.g. INVITE without SDP, then 200 OK and ACK, offer\answer within an offer/answer, and Hold ReINVITE with IP address of 0.0.0.0 - IP address is unchanged).

20.1.5.3 No Media Anchoring

The No Media Anchoring feature enables the use of SBC signaling capabilities without handling the RTP/SRTP (media) flow between remote SIP user agents (UA). The RTP packet flow does not traverse the device, instead, the two SIP UA's establish a direct RTP/SRTP flow (i.e., direct call) between one another. Signaling continues to traverse the device with minimal intermediation and involvement to enable certain SBC abilities such as routing.

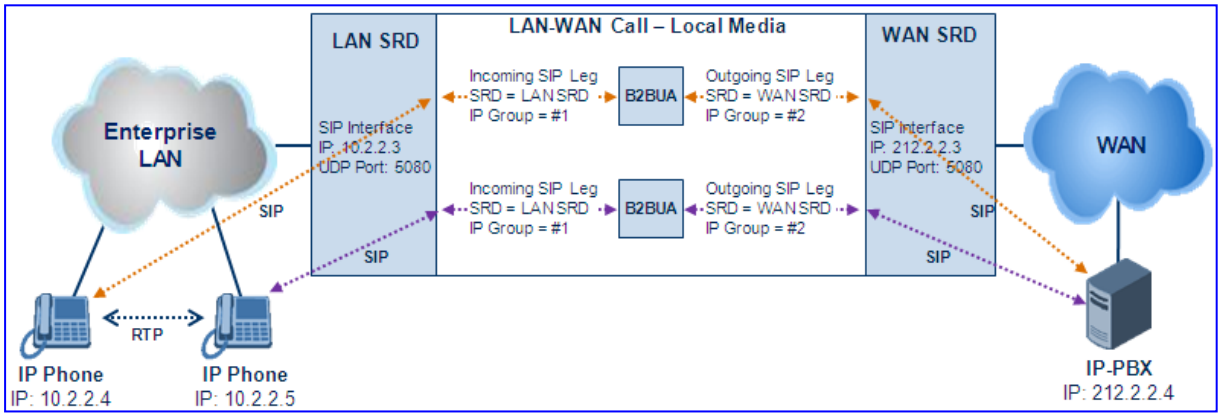
In contrast to the regular SBC implementation, the No Media Anchoring feature:

- Does not perform any manipulation on SDP data (offer/answer transaction) such as ports, IP address, coders.
- Opening voice channels and allocation of IP media ports are not required.

The No Media Anchoring feature is typically implemented in the following scenarios:

- SBC device is located within the LAN.
- Calls between two SIP UA's in the same LAN and signals are sent to a SIP proxy server that is located in the WAN (as illustrated in the figure below).

Figure 20-10: SBC SIP Signaling without RTP Media Flow



The benefits of implementing the No Media Anchoring feature include the following:

- Saves network bandwidth
- Reduces CPU usage (no RTP/SRTP handling)
- Avoids interference in SDP negotiation and header manipulation on RTP/SRTP

The No Media Anchoring process is as follows:

1. Identifies a No Media Anchoring call - according to configuration and the call's properties (such as source, destination, IP Group, and SRD).
2. Handles the identified No Media Anchoring call.

The No Media Anchoring feature is enabled using the SBCDirectMedia parameter. You can also enable No Media Anchoring per SRD (using the IntraSRDMediaAnchoring parameter), whereby calls between two UA's that pertain to the same SRD (source and destination) are handled as No Media Anchoring (direct media) calls.

Notes:

- No Media Anchoring can be used when the SBC does not do NAT traversal (for media) where all the users are in the same domain.
- No Media Anchoring calls cannot operate simultaneously with the following SBC features:
  - Force transcoding
  - Extension Coders
  - Extension of RFC 2833/Out-of-band DTMF/In-band DTMF
  - Extension of SRTP/RTP

  All restriction features (Allowed Coders, restrict SRTP/SRT, restrict RFC 2833) can operate simultaneously. Once No Media Anchoring is enabled, the features listed above are disabled.
- The Coder Restriction feature operates simultaneously with No Media Anchoring calls. Restricted coders are removed from the SDP offer message.
- When two UA's pertain to the same SRD, the parameter IntraSRDMediaAnchoring is set to 1, and one of the UA's is defined as a foreign user (example, "follow me service") located in the WAN, while the other UA is located in the LAN: calls between these two UA's can't be established until IntraSRDMediaAnchoring is set to 0, as the device doesn't interfere in the SIP signaling. In other words, parameters such as IP addresses are not manipulated for calls between LAN and WAN (although required).
- When the parameter SBCDirectMedia is disabled, No Media Anchoring calls between two UA's belonging to separate SRD's cannot be configured. No Media Anchoring calls between two UA's belonging to the same SRD is configurable only (in this case).

20.1.5.4 Interworking DTMF Methods

The device supports interworking between various DTMF methods such as RFC 2833, In-Band DTMF's, and SIP INFO (Cisco\Nortel\Korea). By default, the device allows the remote user agents to negotiate (in case of RFC 2833) and passes DTMF without intervention. However, if two user agents (UA) support different DTMF methods, the device can interwork these different DTMF methods at each leg.

This DTMF interworking feature is enabled using IP Profiles (*ini* file parameter IPProfile):

- SBCRFC2833Behavior - affects the RFC 2833 SDP offer\answer negotiation:
  - **[0]** (default): the device does not intervene in the RFC 2833 negotiation.
  - **[1]:** each outgoing offer\answer includes RFC 2833 in the offered SDP (the device adds RFC 2833 only if the incoming offer does not include RFC 2833).
  - **[2]:** the device removes RFC 2833 from the incoming offer.
- SBCAlternativeDTMFMethod – the device's first priority for DTMF method at each leg is RFC 2833. Therefore, if a specific leg negotiates RFC 2833 successfully, then the chosen DTMF method for this leg is RFC 2833. For legs where RFC 2833 is not negotiated successfully, the device uses this parameter to determine the DTMF method for the leg.
  - **[0]** (default): the device does not attempt to interwork any special DTMF method
  - **[1]:** In Band
  - **[2]:** INFO, Cisco
  - **[3]:** INFO, Nortel

- **[4]:** INFO, Korea

The chosen DTMF method determines (for each leg) which DTMF method is used for sending DTMF's. If the device interworks between different DTMF methods and one of the methods is In-band\RFC 2833, detection and generation of DTMF methods requires DSP allocation.

20.1.5.5 Transcoding Modes

The device supports the configuration of the voice transcoding mode (media negotiation) between the two SBC legs. The device can be configured to perform transcoding only when necessary. Typically, the SBC passes RTP packets transparently (RTP-to-RTP) between the two user agents. If the device is configured to always perform transcoding, then transcoding is performed on the outgoing SBC leg and the device's SBC application interworks the media by implementing PSTN transcoding (since both legs have different media capabilities).

In the SBC application, forced transcoding of voice in an SBC session allows the device to receive capabilities that are not negotiated between the SBC legs. For example, if one would like to force Gain Control on the SBC session to use voice transcoding, even though both sides of the session have negotiated without SBC intervention (for example, coder extension).

Note: To implement transcoding, you must configure the number of required DSP channels for transcoding (for example, `MediaChannels = 120`). Each transcoding session uses two DSP resources. It is recommended to use the MPM module(s) when implementing transcoding.

The transcoding mode can be configured using the parameters TranscodingMode and IPProfile.

20.1.5.6 Restricting Coders

The SBC Allowed Coders (coders restriction) feature determines the coders that can be used for a specific SBC leg. This provides greater control over bandwidth by enforcing the use of specific coders (*allowed coders groups*) while preventing the use of other coders. This is done by defining a group of allowed coders for the SBC leg, as described below:

1. Configure a Coders Group for allowed coders, using the CodersGroup parameter.
2. Select this Coders Group using the SBCAllowedCodersGroupID parameter of the IP Profile table.
3. Enable this feature by setting the SBCAllowedCodersMode parameter of the IP Profile table to **Restriction** or **Restriction and Preference**.

Coders that are not listed in the Allowed Coders Group are removed from the SDP offer. Therefore, only coders common between the SDP offer and Allowed Coders Group are used.

The device can also add coders (referred to as e*xtension coders*) to the SDP offer in the outgoing SBC leg. This is done as follows:

1. Configure a Coders Group for extension coders, using the CodersGroup parameter.
2. Select this Coders Group using the SBCExtensionCodersGroupID parameter of the IP Profile table.
3. Enable this feature by setting the SBCAllowedCodersMode parameter of the IP Profile table to **Restriction** or **Restriction and Preference**.

In addition to restricting the use of coders, the device can prioritize the coders listed in the SDP offer. This feature is referred to as *Coder Preference*. This is done on both SBC legs:

- Incoming SDP offer: The coders list is re-arranged according to their order in the

Allowed Coders Group table. The coders listed higher up in the table take preference over ones listed lower down in the table. This feature is enabled by setting the SBCAllowedCodersMode parameter of the IP Profile table to **Preference** or **Restriction and Preference**.

- Outgoing SDP offer: The coders are arranged in the SDP offer according to the above if only allowed coders are used. However, if extension coders are also used, then the coders list is arranged according to the SBCPreferencesMode parameter of the IP Profile table. This parameter can be configured to add the extension coders at the end of the list (default) and arrange the allowed coders according to their order in the Coders Group table, or arrange allowed and extension coders according to their position in the Coders Group table.

Notes:

- For a list of supported coders, refer to the *ini* file parameter table CodersGroup.
- If Allowed Coder Groups are configured, unknown coders are blocked by the device.
- Allowed Coder Groups are applicable only to audio media.
- For configuring Allowed Coder Groups and Extension Coder Groups, use the AllowedCodersGroup parameter.

The Allowed Coders process is as follows:

a. The device receives an incoming SIP message with SDP (offer) and checks the offered coders.

b. The source (first) leg may have Allowed Coders (i.e. list of coders that can be used - enforced).

c. The device checks for common coders between the SDP offered coders and the Allowed Coders Group list.

For example, assume the following:

- The SDP coder offer includes the following coders: G.729, G.711, and G.723.
- The source (first) leg includes the following Allowed Coders: G.711 and G.729.

The device selects the common coders, i.e., G.711 and G.729 (with changed preferred coder priority - highest for G.711). In other words, it removes the coders that are not in the Allowed Coders list and the order of priority is first according to the Allowed Coders list.

Now assume that the destination (second) leg also includes Allowed Coders and/or extensions. Therefore, the device performs the Allowed Coders procedure (common coders) between the updated coder list and the destination leg's Allowed Coders list (Coders Extension procedure is performed before Allowed Coders). Adding to the example, assume the following:

- For the first leg, the device selects the common coders G.711 and G.729 (explained in the example above).
- Assume that the second leg includes the Extended coder G.726.
- Assume that the second leg includes the following Allowed Coders: G.723, G.726, and G.729.

As a result, the device selects the common coders, i.e., G.729 and G.726 (coder priority did not change Extension coder order).

If the Allowed Coders policy on SDP returns an empty coders list, the device (source leg) rejects the call (SIP 488 or ACK and BYE). If both Coders Extension and Allowed Coders policies on SDP (in this order) returns an empty coders list, the second leg rejects the call (SIP 488, or ACK and BYE).

Below is an example, assuming that Allowed Coders list (ordered) includes G711A-law (PCMA), G729, and G711U-law (PCMU), and Extension Coder is G729.

1. SDP offer - original offer:

```
m=audio 6050 RTP/AVP 0 8 4 96
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:4 G723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

2. SDP offer - after manipulation:

```
m=audio 6010 RTP/AVP 8 0 96 18
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
```

In the SDP, the "m=audio 6010 RTP/AVP 8 0 96 18" line shows that the coder priority has changed - G.711A-law ("8") and then G.711U-law ("0") - and that the Extension coder G.729 ("18") has been added. The G.723 coder ("4) in the original offer was removed as it was not defined in the Allowed Coders list (i.e., a restricted coder).

20.1.5.7 SRTP-RTP Transcoding

The device supports transcoding between SRTP and RTP. The device can also enforce SBC legs to use SRTP\RTP, using the IP Profile parameter SBCMediaSecurityBehaviour:

- As is (default): no special handling for RTP\SRTP is done.
- SRTP: SBC legs negotiate only SRTP media lines, and RTP media lines are removed from the incoming SDP offer\answer.
- RTP: SBC legs negotiate only RTP media lines, and SRTP media lines are removed from the incoming offer\answer.
- Both: each offer\answer is extended (if not already) to two media lines - one RTP and the other SRTP.

If two SBC legs (after offer\answer negotiation) use different security types (i.e., one RTP and the other SRTP), then the device performs RTP-SRTP transcoding.

To transcode between RTP and SRTP, the following prerequisites must be met:

- At least one supported SDP "crypto" attribute and parameters
- EnableMediaSecurity must be set to 1

If one of the above transcoding prerequisites is not met:

- Any value other than “As is” is discarded.
- If the incoming offer is SRTP, force transcoding, coder transcoding, and DTMF extensions are not applied.

Transcoding between RTP and SRTP requires one DSP allocation. SRTP to SRTP does not require DSP allocation.

20.1.5.8 Multiple RTP Media Streams per Call Session

The device's SBC application supports multiple RTP media streams per SBC call session. Up to five different media types can be included in a session:

- Audio (m=audio)
- Video (m=video)
- Text (m=text)
- Fax (m=image)

Therefore, the device can provide transcoding of various attributes in the SDP offer/answer (e.g., codec, port, and packetization time) per media type. If the device is unable to perform transcoding (for example, does not support the codec), it relays the SBC dialog transparently.

20.1.6 Fax Negotiation and Transcoding

The device can allow fax transmissions to traverse transparently (i.e., without transcoding) or it can handle the fax as follows:

- Allow interoperability between different fax machines, supporting fax transcoding if required.
- Restrict usage of specific fax coders to save bandwidth, enhance performance, or comply with supported coders. These coders include G.711 (A-Law or Mu-Law), VBD (G.711 A-Law or G.711 Mu-Law), and T38.

Fax configuration is done in the IP Profile and Coder Group Settings tables. The IP Profile table determines the supported fax coders and the negotiation method used between the incoming and outgoing fax legs, using the following fax-related parameters:

- SBCFaxBehavior: defines the offer negotiation method - pass fax transparently, negotiate fax according to fax settings in IP Profile, or enforce remote UA to first establish a voice channel before fax negotiation.
- SBCFaxCodersGroupID: defines the supported fax coders (from the Coders Group Settings table).
- SBCFaxOfferMode: determines the fax coders sent in the outgoing SDP offer.
- SBCFaxAnswerMode: determines the fax coders sent in the outgoing SDP answer.

Note: The voice-related coder configuration (Allowed and Extended coders) is independent of the fax-related coder configuration, with the exception of the G.711 coder. If the G.711 coder is restricted by the Allowed Coders Group table, it is not used for fax processing even if it is listed in the Coders Group Settings table for faxes. However, support for G.711 coders for voice is not dependent upon which fax coders are listed in the Coders Group Settings table.

20.1.7 SIP Dialog Admission Control

The device allows you to limit the number of concurrent calls (SIP dialogs). These call limits can be applied per SRD and/or IP Group, and per user (identified by its registered contact). This is especially important for applications where VoIP and Data traffic contend on the WAN throughput, which may be limited by itself. For example, DSL WAN access interface is very limited in the uplink. Therefore, by controlling the number of calls allowed, bandwidth can be reserved for Data applications. This feature can be useful for implementing Service Level Agreements (SLA) policies.

The SIP dialog limits can be defined per SIP request type and direction (inbound or outbound). These relate to requests that initiate SIP dialogs and not the subsequent requests that can be of different type and direction. The SIP dialog-initiating request types can include SIP INVITEs, REGISTER, and/or SUBSCRIBE, or it can be configured to include all dialogs. Requests that supersede the defined limit are rejected with a SIP 486 "Busy Here" response.

SIP-dialog rate control can also be configured using the “token bucket” mechanism. The token bucket is a control mechanism that dictates the rate of SIP-dialog setups based on the presence of tokens in the bucket – a logical container that holds aggregate SIP dialogs to be accepted or transmitted. Tokens in the bucket are removed ("cashed in") for the ability to setup a dialog. Therefore, a flow can set up dialogs up to its peak burst rate if there are adequate tokens in the bucket and if the burst threshold is configured appropriately:

- Every SIP dialog setup request must attempt to take a token from the bucket.
- If there are no tokens, the request is dropped.
- New tokens are added to the bucket at a user-defined rate (token rate).
- If the bucket contains the maximum number of tokens, tokens to be added at that moment are dropped.

A token bucket is configured using the following new parameters:

- Rate = Rate at which tokens are added to the bucket (i.e., token rate). One token is added to the bucket every 1000/Rate milliseconds. The rate of dialog setups per second or unlimited if set to 0 (default).
- Max Burst = Maximum tokens that can fill the bucket. At any given time, the bucket cannot contain more than this amount of tokens. The maximum burst size for the dialog setup rate, unlimited if set to 0 (default).

Dropped requests are replied with the 486 “Busy Here” SIP response. Dropped requests are not counted in the bucket.

The SIP dialog limits are defined in the Admission Control table (SBCAdmissionControl).

20.1.8 Limiting SBC Call Duration

You can define a maximum allowed duration (in minutes) for SBC calls. If an established call reaches this user-defined limit, the device terminates the call. This feature ensures calls are properly terminated, allowing available resources for new calls. This feature is configured using the MaxCallDuration parameter.

20.1.9 SIP Authentication Server for SBC Users

The device can function as an authentication server for SIP SBC message requests, based on HTTP authentication DIGEST with MD5. Alternatively, such requests can be authenticated by an external, third-party server.

When functioning as an authentication server (set by the IP Group table parameter, AuthenticationMode), the device authenticates users belonging to a USER-type IP Group. When the device receives an INVITE or REGISTER request from a client (e.g., SIP phone) for SIP message authorization, the device processes the authorization as follows:

1. The device verifies the type of incoming SIP method (e.g., INVITE) that must be challenged for authorization. This is configured using the IP Group table parameter, MethodList.
2. If the message is received without an Authorization header, the device "challenges" the client by sending a 401 or 407 SIP response. The client then resends the request with an Authorization header (containing the user name and password).
3. The device validates the SIP message according to the settings of the parameters, AuthNonceDuration, AuthChallengeMethod and AuthQOP.
   - If validation fails, the message is rejected and the device sends a 403 "Forbidden" response.
   - If validation succeeds, the device verifies identification of the SBC user. This is done by checking that the user name and password received from the user is the same username and password that appears in the device's database. The SBC users in the database are obtained from the User Information file. If the SIP SBC user is not successfully authenticated after three attempts, the device sends a 403 "Forbidden" response.
4. If the user is successfully identified, the SIP message request is processed.

20.1.10 Handling SIP 3xx Redirect Responses

By default, the device's handling of SIP 3xx responses is to send the Contact header unchanged. However, some network setups require that the new INVITE message sent as a result of the 3xx traverse the device. This is enabled by the parameter SBC3xxBehavior.

Reasons for enforcing resultant INVITEs to traverse the SBC may vary:

- The user that receives the 3xx can't route to the 3xx contact (i.e., the user is on the LAN and the new contact is on the WAN). In such a scenario, the device helps the user reach the WAN contact and overcome NAT problems.
- Enforce certain SBC policies (e.g., call admission control, header manipulation, and transcoding) on the resultant INVITE.

The device enforces this by modifying each Contact in the 3xx response as follows:

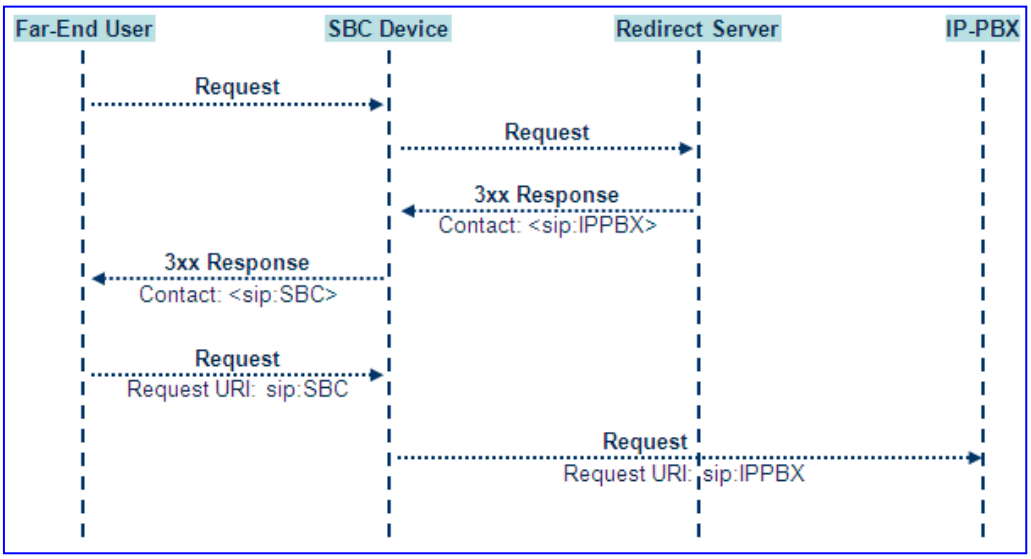
- Changes the host part to the device's IP address – this change causes the remote user agent to send the INVITE to the device.
- Adds a special prefix ("T~&R_") to the Contact user part – to identify the new INVITE as a 3xx resultant INVITE.

The SBC handling for the 3xx resultant INVITE is as follows:

1. The incoming INVITE is identified as a 3xx resultant INVITE according to the special prefix.
2. The device automatically replaces the SBC host part (in the Request-URI) with the host from the 3xx Contact.
3. The prefix ("T~&R_") remains in the user part for the classification, manipulation, and routing mechanisms.
4. The classification, manipulation, and routing processes are done exactly like any other INVITE handling. The special prefix can be used for specific routing rules for 3xx resultant INVITEs.

5. The prefix is removed before the resultant INVITE is sent to the destination.

Figure 20-11: SIP 3xx Response Handling



The process of this feature is described using an example:

1. The device receives the Redirect server's SIP 3xx response (e.g., Contact: <sip:User@IPPBX:5060;transport=tcp;param=a>;q=0.5).
2. The device replaces the Contact header value with the special prefix and database key value as user part, and with the device's URL as host part (e.g., Contact: <sip:Prefix_Key_User@SBC:5070;transport=udp>;q=0.5).
3. The device sends this manipulated SIP 3xx response to the Far-End User (FEU).
4. The FEU sends a new request with the Request-URI set to the value of the received 3xx response's Contact header (e.g., RequestURI: sip:Prefix_Key_User@SBC:5070;transport=udp).
5. Upon receipt of the new request from the FEU, the device replaces the Request-URI with the new destination address (e.g., RequestURI: sip:Prefix_User@IPPBX:5070;transport=tcp;param=a).
6. The device removes the user prefix from the Request-URI, and then sends this Request-URI to the new destination (e.g., RequestURI: sip:User@IPPBX:5070;transport=tcp;param=a).

20.1.11 Interworking SIP Diversion and History-Info Headers

This device can be configured to interwork between the SIP Diversion and History-Info headers. This is important, for example, to networks that support the Diversion header but not the History-Info header, or vice versa. Therefore, mapping between these headers is crucial for preserving the information in the SIP dialog regarding how and why (e.g., call redirection) the call arrived at a certain SIP UA.

This feature is configured in the IP Profile table (IPProfile parameter) using the following new parameters:

- SBCDiversionMode - defines the device's handling of the Diversion header
- SBCHistoryInfoMode - defines the device's handling of the History-Info header

The handling of the SIP Diversion and History-Info headers is described in the table below:

Table 20-1: Handling of SIP Diversion and History-Info Headers

Parameter Value	SIP Header Present in Received SIP Message		
	Diversion	**History-Info**	**Diversion and History-Info**
HistoryInfoMode = Add **DiversionMode = Remove**	Diversion converted to History-Info. Diversion removed.	Not present	Diversion removed.
HistoryInfoMode = Remove **DiversionMode = Add**	Not present.	History-Info converted to Diversion. History-Info removed.	History-Info added to Diversion. History-Info removed.
HistoryInfoMode = Disable **DiversionMode = Add**	Diversion converted to History-Info.	Not present.	Diversion added to History-Info.
HistoryInfoMode = Disable **DiversionMode = Add**	Not present.	History-Info converted to Diversion.	History-Info added to Diversion.
HistoryInfoMode = Add **DiversionMode = Add**	Diversion converted to History-Info.	History-Info converted to Diversion.	Headers are synced and sent.
HistoryInfoMode = Remove **DiversionMode = Remove**	Diversion removed.	History-Info removed.	Both removed.

20.1.12 Call Survivability

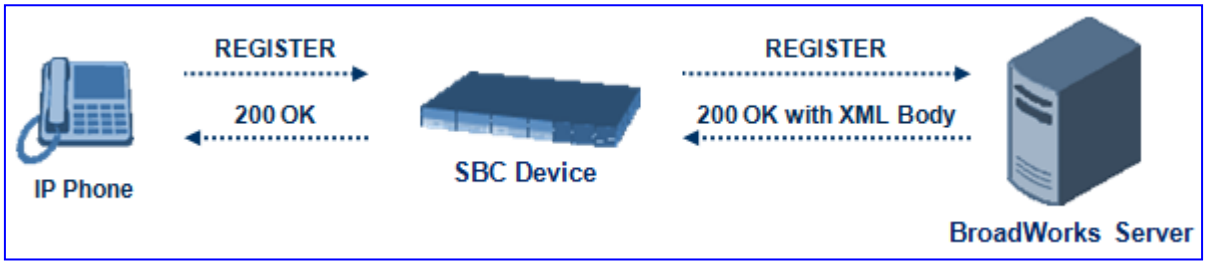
This section describes various call survivability features supported by the SBC device.

20.1.12.1 Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server for Survivability

This feature enables SBC user registration for interoperability with BroadSoft BroadWorks server to provide call survivability in case of connectivity failure with the BroadWorks server, for example, due to a WAN failure. This feature enables local users to dial a local extension (or any other configured alias) that identifies another local user, in survivability mode. This feature is enabled using the SBCExtensionsProvisioningMode parameter.

In normal operation, when subscribers (such as IP phones) register to the BroadWorks server through the device, the device includes the SIP Allow-Events header in the sent REGISTER message. In response, the BroadWorks server sends the device a SIP 200 OK containing an XML body with subscriber information such as extension number, phone number, and URIs (aliases). The device forwards the 200 OK to the subscriber (without the XML body).

Figure 20-12: Interoperability with BroadWorks Registration Process



The device saves the users in its registration database with their phone numbers and extensions, enabling future routing to these destinations during survivability mode. When in survivability mode, the device routes the call to the Contact associated with the dialed phone number or extension number in the registration database.

Below is an example of an XML body received from the BroadWorks server:

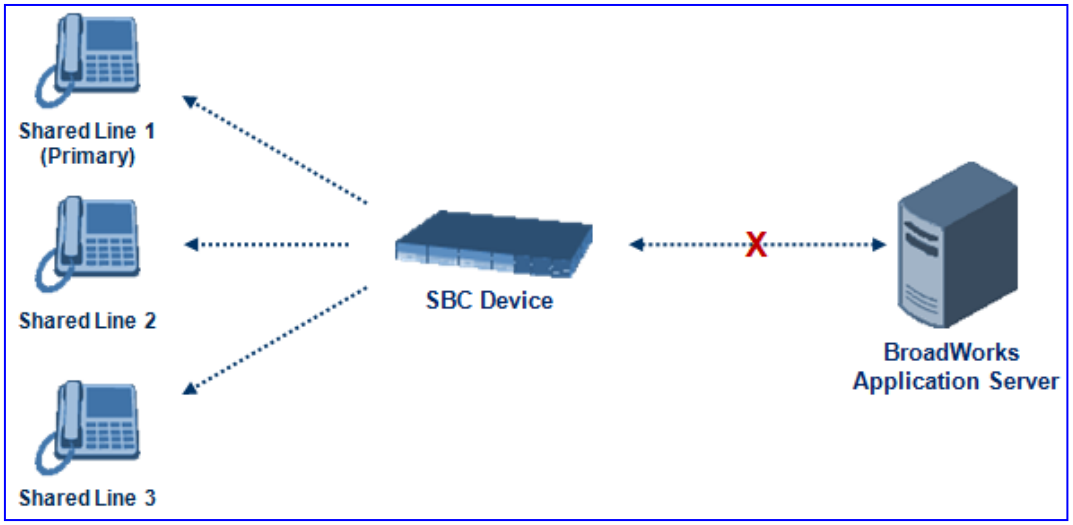
```
<?xml version="1.0" encoding="utf-8"?>
  <BroadsoftDocument version="1.0" content="subscriberData">
    <phoneNumbers>
      <phoneNumber>2403645317</phoneNumber>
      <phoneNumber>4482541321</phoneNumber>
    </phoneNumbers>
    <aliases>
      <alias>sip:bob@broadsoft.com</alias>
      <alias>sip:rhughes@broadsoft.com</alias>
    </aliases>
    <extensions>
      <extension>5317</extension>
      <extension>1321</extension>
    </extensions>
  </BroadSoftDocument>
```

20.1.12.2 BroadSoft's Shared Phone Line Call Appearance for SBC Survivability

The device can provide redundancy for BroadSoft's Shared Call Appearance feature. When the BroadSoft application server switch (AS) fails or does not respond, or when the network connection between the device and the BroadSoft AS is down, the device manages the Shared Call Appearance feature for the SIP clients.

This feature is supported by configuring a primary extension and associating it with secondary extensions (i.e., *shared lines*) so that incoming calls to the primary extension also ring at the secondary extensions. The call is established with the first extension to answer the call, and consequently, the ringing at the other extensions stop. For example, assume primary extension number 600 is shared with secondary extensions 601 and 602. In the case of an incoming call to 600, all three phone extensions ring simultaneously (using the device's call forking feature as described in 'SIP Forking Initiated by SIP Proxy Server' on page 443). Note that incoming calls specific to extensions 601 or 602 ring only at these specific extensions.

Figure 20-13: Call Survivability for BroadSoft's Shared Line Appearance



To configure this capability, you need to configure a shared-line, inbound manipulation rule for registration requests to change the destination number of the secondary extension numbers (e.g. 601 and 602) to the primary extension (e.g., 600). In addition, call forking must also be enabled. The procedure below describes the main configuration required.

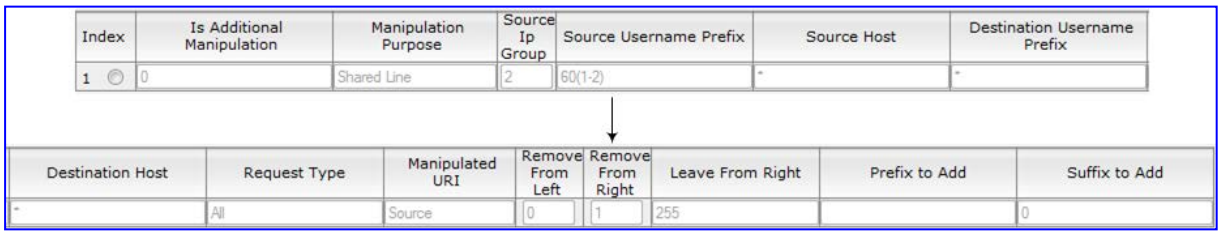
Notes:

- The device enables outgoing calls from all equipment that share the same line simultaneously (usually only one simultaneous call is allowed per a specific shared line).
- The LED indicator of a shared line may display the wrong current state.

➢ **To configure the Shared Line feature:**

1. Create a SERVER-type IP Group for the BroadWorks server.
2. Create a USER-type IP Group for the IP phone users and set the EnableSBCClientForking to enable so that the device forks incoming calls to all contacts under the same AOR that are registered in the device's registration database.
3. Create a call routing rule in the IP2IP Routing table for routing calls between the above configured IP Groups.
4. Create a manipulation rule in the IP to IP Inbound Manipulation table for the secondary extensions (e.g., 601 and 602) so that they also register in the device's database under the primary extension (e.g., 600) contact:
   - Set the 'Manipulation Purpose' field to **Shared Line**.
   - Set the 'Source IP Group' field to the IP Group ID that you created for the users (e.g., 2).
   - Set the 'Source Username Prefix' field to represent the secondary extensions (e.g., 601 and 602).
   - Set the 'Manipulated URI' field to **Source** to manipulate the source URI.
   - Set the 'Remove From Right' field to "1" to remove the last digit of the extensions (e.g., 601 is changed to 60).
   - Set the 'Suffix to Add' field to "0" to add 0 to the end of the manipulated number (e.g., 60 is changed to 600).

Figure 20-14: Manipulation Rule for Shared Line



20.1.12.3 Call Survivability for Call Centers

The device supports call survivability for call centers. When a communication failure (e.g., in the network) occurs with the remote voice application server responsible for handling the call center application (such as IVR), the device routes the incoming calls received from the customer (i.e., from the TDM gateway) to the call center agents.

In normal operation, the device registers the agents in its users registration database. Calls received from the TDM gateway are forwarded by the device to the application server, which processes the calls and sends them to specific call center agents, through the device. Upon a failure with the application server, the device routes the calls from the TDM Gateway to the agents. The device routes the call to the first available user it founds. If the call is not answered by the user, the device routes it to the next available user. The SBC can handle a sequence of up to five users, after which the session is timed out and the call is dropped.

Figure 20-15: Normal Operation in Call Center Application

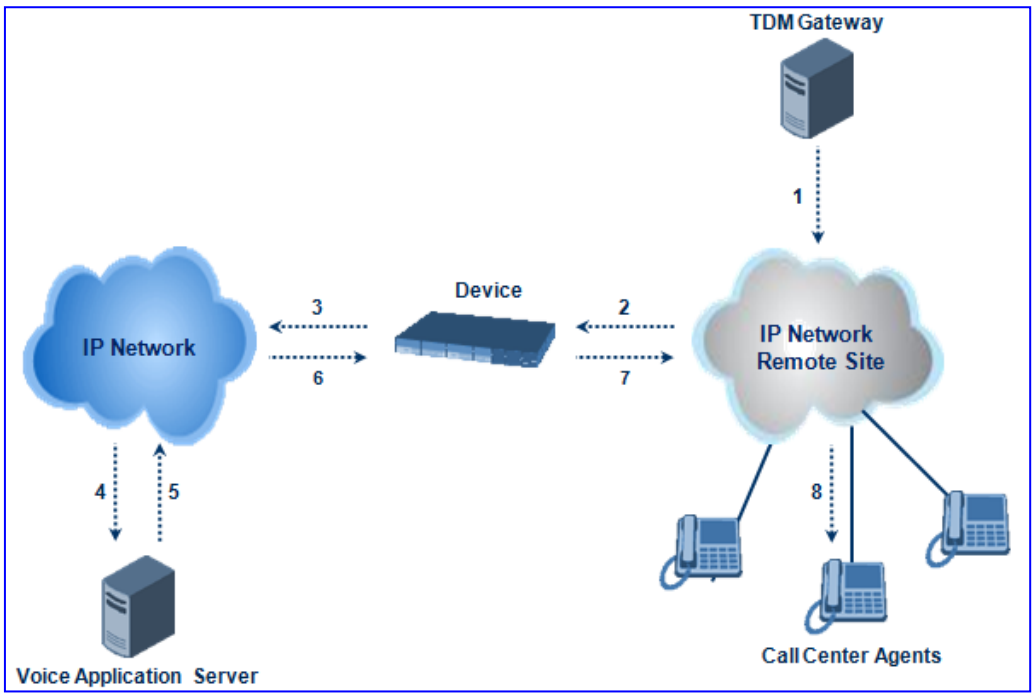
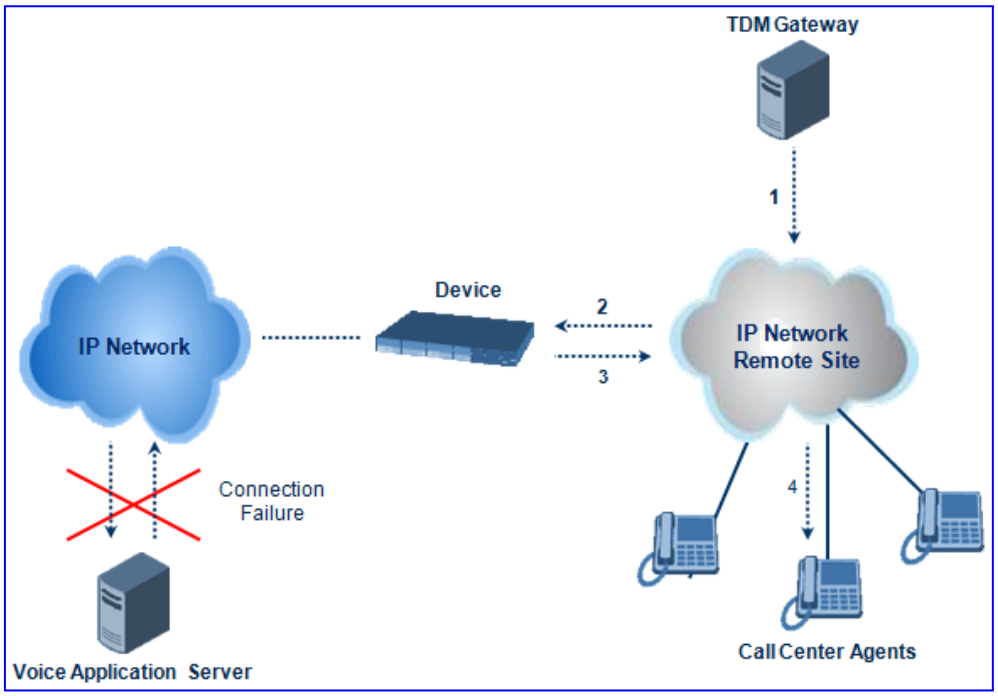


Figure 20-16: Call Survivability for Call Center



➢ **To configure call survivability for a call center application:**

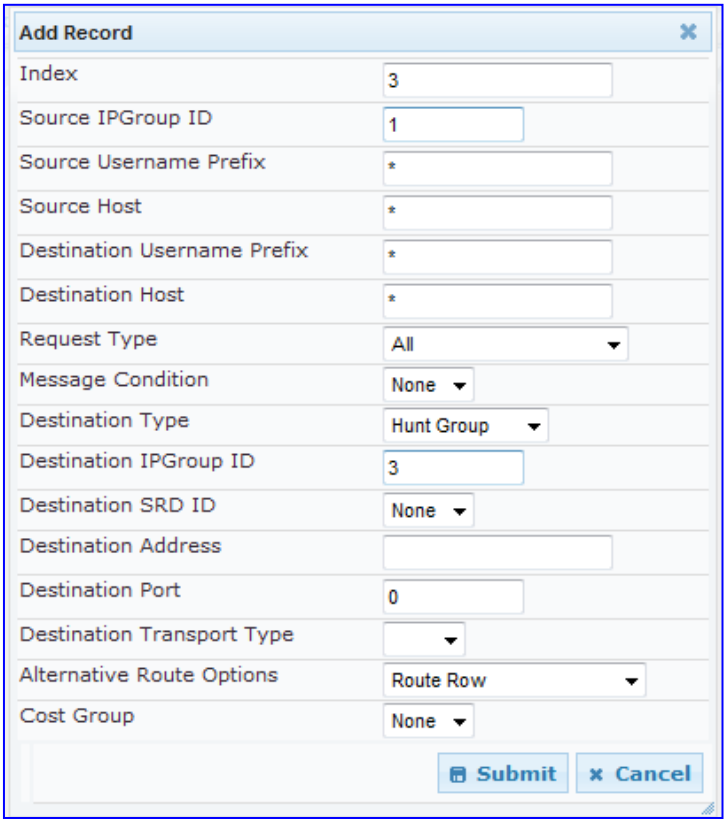
1. Configure IP Groups in the IP Group table (see 'Configuring IP Groups' on page 212) for the following entities:
   - TDM Gateway (SERVER-type IP Group). This entity forwards the customer calls, through the device, to the Application server.
   - Application server (SERVER-type IP Group). This entity processes the call and sends the call, through the device, to the specific call center agent located on a different network (remote).

- Call center agents (USER-type IP Group). You can configure multiple IP Groups to represent different groups of call center agents, for example, agents and managers.

2. In the Classification table (see 'Configuring Classification Table' on page 452), configure rules to classify incoming calls received from the entities listed in Step 1 to IP Groups.
3. In the SBC IP2IP Routing table (see 'Configuring SBC IP-to-IP Routing' on page 457), configure the following IP-to-IP routing rules:
   - For normal operation:
     - Routing from TDM Gateway to Application server.
     - Routing from Application server to call center agents.
   - For call survivability mode: Routing from TDM Gateway to call center agents. This configuration is unique due to the following settings:
     - The 'Source IP Group ID' field is set to the IP Group of the TDM Gateway.
     - The 'Destination Type' field is set to **Hunt Group**, which is specifically used for call center survivability.
     - The 'Destination IP Group ID' field is set to the IP Group of the call center agents.

   The figure below displays a routing rule example, assuming IP Group "1" represents the TDM Gateway and IP Group "3" represents the call center agents:

Figure 20-17: Routing Rule for Call Survivability of Call Center



20.1.12.4 Survivability Mode Display on Aastra IP Phones

If the SBC device is deployed in an Enterprise network with Aastra IP phones and connectivity with the WAN fails, the device provides call survivability by enabling communication between IP phone users within the LAN enterprise. In such a scenario, the device can be configured to notify the IP phones that it is currently operating in Survivability

mode. When this occurs, the Aastra IP phones display the message, "StandAlone Mode" on their LCD screens. This feature is enabled by setting the SBCEnableAASTRASurvivabilityNotice parameter to 1.

When this feature is enabled and the SBC device is in Survivability mode, it responds to SIP REGISTER messages from the IP phones with a SIP 200 OK containing the following XML body:

```
Content-Type: application/xml
<?xml version="1.0" encoding="utf-8"?>
<LMIDocument version="1.0">
<LocalModeStatus>
   <LocalModeActive>true</LocalModeActive>
   <LocalModeDisplay>StandAlone Mode</LocalModeDisplay>
</LocalModeStatus>
</LMIDocument>
```

20.1.13 Call Forking

20.1.13.1 Initiating SIP Call Forking

The SBC device enables call forking, whereby an incoming call is forked to multiple SBC users (destinations). In such a scenario, upon an incoming call, all the extensions of a user ring simultaneously and the first extension to pick up the call receives the call and all other extensions stop ringing. Call forking is supported by the device's ability to register in its database multiple SIP client user phone contacts (mobile and fixed-line extensions) to the same Address of Record (AOR).

This feature can be implemented in the following example scenarios:

- An enterprise Help Desk, where incoming customer calls are simultaneously sent to multiple customer service agent extensions.
- An employee's phone devices, where the incoming call is simultaneously sent to multiple devices (e.g., to the employee's office phone and mobile SIP phone).
- An enterprise reception desk, where an incoming call is simultaneously sent to multiple receptionists.

The Call Forking feature is configured by creating a USER-type IP Group with the EnableSBCClientForking parameter set to enabled (in the IP Group table).

20.1.13.2 SIP Forking Initiated by SIP Proxy Server

The device can handle SIP forking responses received from a proxy server in response to an INVITE sent by the device from a UA. In other words, received responses with a different SIP To header 'tag' parameter for the request forwarded by the device. This occurs in scenarios, for example, where a proxy server forks the INVITE request to several UAs, and hence, the SBC device may receive several replies for a single request.

Forked SIP responses may result in a single SDP offer with two or more SDP answers during call setup. The SBC handles this scenario by "hiding" the forked responses from the INVITE-initiating UA. This is achieved by marking the UA that responded first to the INVITE as the active UA, and only requests/responses from that UA are subsequently forwarded. All other requests/responses from other UAs are handled by the SBC (SDP offers from these users are answered with an 'inactive' media).

If the active UA is the first one to send the final response (e.g., 200 OK), the call is established and all other final responses are acknowledged and a BYE is sent if needed. If another UA sends the first final response, then it is possible that the SDP answer that was forwarded to the INVITE-initiating UA is not relevant, and media synchronization is needed between the two UAs. Media synchronization is done by sending a re-INVITE request

immediately after the call is established. The re-INVITE is sent without an offer to the INVITE-initiating UA. This causes the UA to send an offer which is forwarded to the UA that confirmed the call. The media synchronization process is enabled by the EnableSBCMediaSync parameter.

20.1.14 Alternative Routing on Detection of Failed SIP Response

The device can detect failure of a sent SIP response (e.g., TCP timeout, and UDP ICMP). In such a scenario, the device re-sends the response to an alternative destination. This support is in addition to alternative routing if the device detects failed SIP requests.

For example, assume the device sends a SIP 200 OK in response to a received INVITE request. If the device does not receive a SIP ACK in response to this, it sends a new 200 OK to the next alternative destination (e.g., to the next given IP address resolved from a DNS from the Contact or Record-Route header in the request related to the response).

20.2 SBC Configuration

This section describes the configuration of the SBC application.

Note: For the SBC application, the following requirements must be met:

- The SBC application is enabled (see 'Enabling Applications' on page 205).
- The 'SBC' Software Upgrade Key is installed on the device (see 'Loading Software Upgrade Key' on page 696).

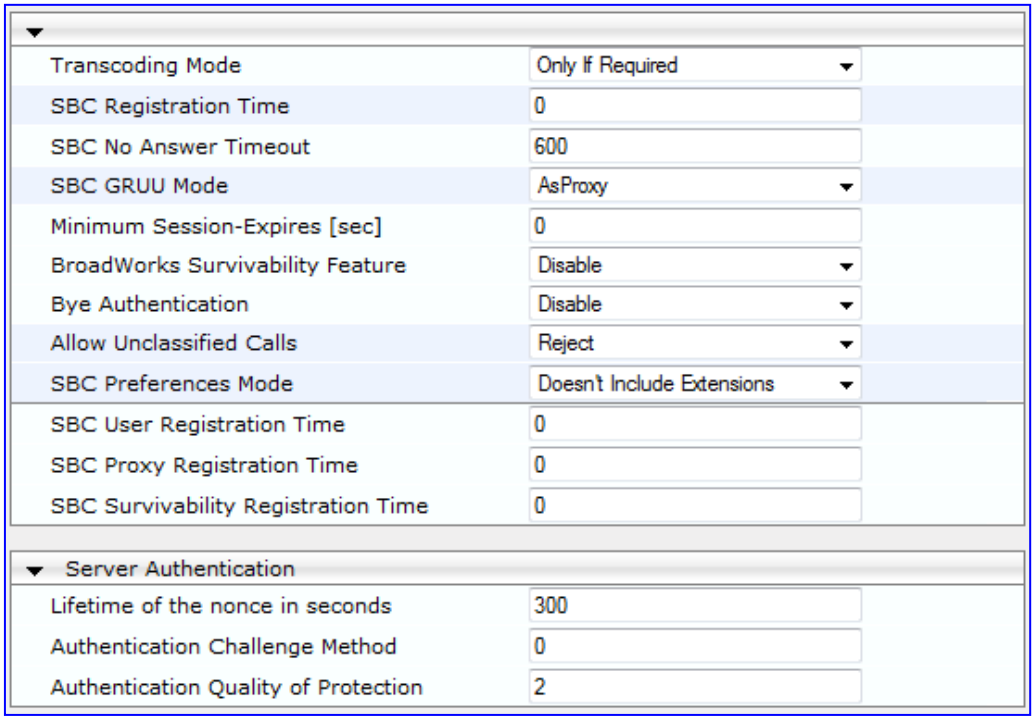
20.2.1 Configuring General Settings

The General Settings page allows you to configure general SBC parameters. For a description of these parameters, see 'SBC Parameters' on page 979.

➢ **To configure general parameters:**

1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **General Settings**).

Figure 20-18: General Settings Page



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

20.2.2 Configuring Admission Control

The Admission Control page allows you to define up to 100 rules for limiting the number of concurrent calls (SIP dialogs). These call limits can be applied per SRD, IP Group, SIP request type (e.g., INVITEs), SIP dialog direction (e.g., inbound), and/or per user (identified by its registered contact). This is especially important for applications where VoIP and Data traffic contend on the WAN throughput, which may be limited by itself. For example, DSL WAN access interface is very limited in the uplink. Therefore, by controlling the number of calls allowed, bandwidth can be reserved for specific Data applications. This feature can be useful for implementing Service Level Agreements (SLA) policies.

The SIP dialog limits can be defined per SIP request type and direction. These relate to requests that initiate SIP dialogs and not the subsequent requests that can be of different type and direction. The SIP dialog-initiating request types can include SIP INVITEs, REGISTER, and/or SUBSCRIBE, or it can be configured to include the total number of all dialogs.

This feature also provides support for SIP-dialog rate control, using the “token bucket” mechanism. The token bucket is a control mechanism that dictates the rate of SIP-dialog setups based on the presence of tokens in the bucket – a logical container that holds aggregate SIP dialogs to be accepted or transmitted. Tokens in the bucket are removed ("cashed in") for the ability to setup a dialog. Therefore, a flow can setup dialogs up to its peak burst rate if there are adequate tokens in the bucket and if the burst threshold is configured appropriately.

Requests that reach the user-defined call limit (maximum concurrent calls and/or call rate) are sent to an alternative route, if configured in the IP2IP Routing table for the SRD or IP Group. If no alternative routing rule is located, the device rejects the SIP request with a SIP 486 "Busy Here" response.

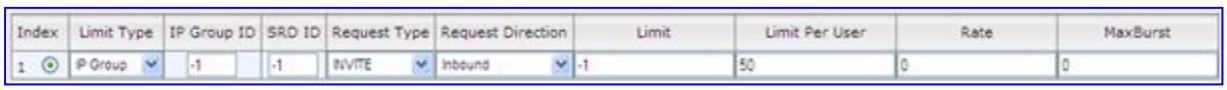
Notes:

- The enforcement of a configured limitation for the incoming leg is performed immediately after the Classification process. If the call/request is rejected at this stage, no routing is performed. The enforcement for the outgoing leg is performed within each alternative route iteration. This is accessed from two places: one during initial classification/routing, and another during alternative routing process.
- For configuring Admission Control using the *ini* file, refer to the parameter SBCAdmissionControl.

➢ **To configure Admission Control rules:**

1. Open the Admission Control page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Admission Control**).

Figure 20-19: Admission Control Page



2. Add an entry and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 20-2: Admission Control Parameters

Parameter	Description
Limit Type	Limitation rule defined per IP group or SRD. ▪ **[0]** IP Group (default) ▪ **[1]** SRD
IP Group ID	IP Group to which you want to apply the SIP dialog limit. To apply the rule to all IP Groups, set this parameter to -1 (default). **Note:** This parameter is applicable only if Limit Type is set to IP Group.
SRD ID	SRD to which you want to apply the SIP dialog limit. To apply the rule to all SRD's, set this parameter to -1 (default). **Note:** This parameter is applicable only if Limit Type is set to SRD.
Request Type	SIP dialog-initiating request type that initiates the SIP dialog to which you want to apply the SIP dialog limit (not the subsequent requests that can be of different type and direction). The SIP dialog-initiating request types can include: ▪ **[0]** All = include the total number of all dialogs (default) ▪ **[1]** INVITE ▪ **[2]** SUBSCRIBE ▪ **[3]** Other
Request Direction	The direction of the SIP request to which the limitation is applied. ▪ **[0]** Both = Applied to inbound and outbound SIP dialogs (default) ▪ **[1]** Inbound = Applies only to inbound SIP dialogs ▪ **[2]** Outbound = Applies only to outbound SIP dialogs
Limit	Maximum number of concurrent SIP dialogs per IP Group or SRD. You can also use the following special values: ▪ **[0]** 0 = Disallow/block all these dialogs ▪ **[-1]** -1 = No limit (default)
Limit Per User	Maximum number of concurrent SIP dialogs per user belonging to the configured IP Group or SRD. You can also use the following special values: ▪ **[0]** 0 = Disallow/block all these dialogs ▪ **[-1]** -1 = No limit (default)
Rate	Rate at which tokens are added to the bucket per second (i.e., token rate) or unlimited if set to 0 (default). One token is added to the bucket every 1000 divided by the value of this parameter (in milliseconds). **Note:** The token bucket feature is per IP Group, SRD, SIP request type, and SIP request direction.

Parameter	Description
MaxBurst	The maximum number of tokens (SIP dialogs) that the bucket can hold, where 0 is unlimited (default). The device only accepts a SIP dialog if a token exists in the bucket. Once the SIP dialog is accepted, a token is removed from the bucket. If a SIP dialog is received by the device and the token bucket is empty, then the device rejects the SIP dialog. Alternatively, if the bucket is full, for example, 100 tokens, and 101 SIP dialogs arrive (before another token is added to the bucket, i.e., faster than that defined in the Rate field), then the device accepts the first 100 SIP dialogs and rejects the last one. Dropped requests are replied with the 486 “Busy Here” SIP response. Dropped requests are not counted in the bucket. **Note:** The token bucket feature is per IP Group, SRD, SIP request type, and SIP request direction.

20.2.3 Configuring Allowed Coder Groups

The Allowed Coders Group page allows you to define up to five Allowed Coder Groups, each with up to 10 coders. Allowed Coder Groups determine the coders that can be used for a specific SBC leg. Therefore, the device's SBC application can enforce the use of specific coders while preventing the use of other coders. Coders excluded from the Allowed Coders Group are removed from the SDP offer. Only common coders between SDP offered coders and coders configured in the Allowed Coder Groups are used. The Allowed Coders Group table is also used to configure Extension coders. These are coders that you want to add to the SDP offer.

The order of appearance of coders in the Allowed Coder Group determines the coder priority (preference), whereby the first coder is given the highest priority. For more information on working with Allowed and Extension coders, and coder preference, see 'Restricting Coders' on page 430.

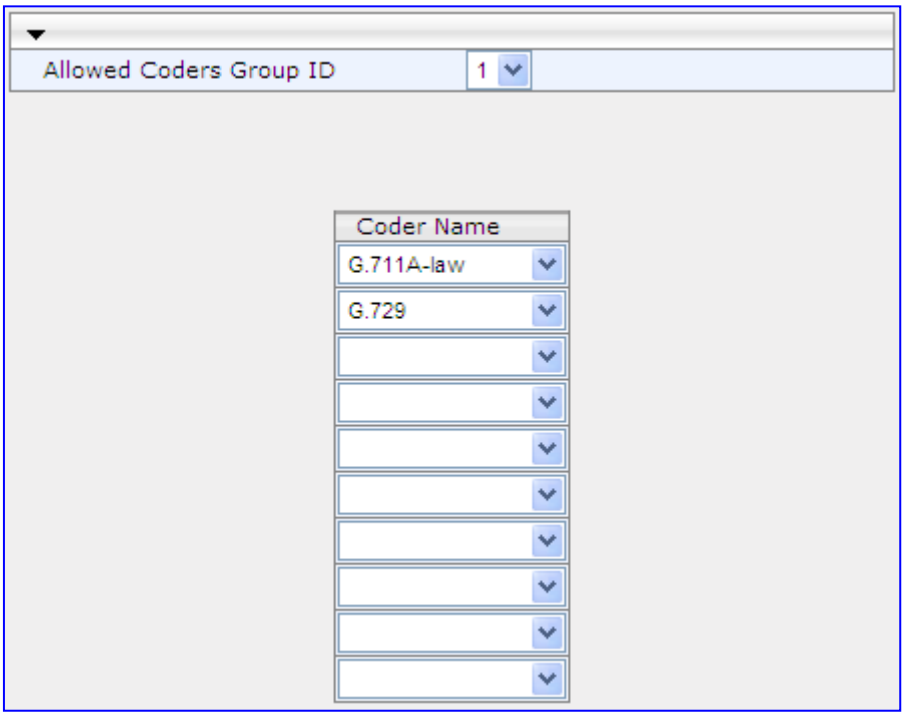
Notes:

- For a list of supported coders, refer to the *ini* file parameter table CodersGroup.
- Each coder can appear only once per Allowed Coder Group.
- If Allowed Coder Groups are configured, coders not included are blocked by the device.
- Allowed Coder Groups are applicable only to audio media.
- Allowed Coder Groups can be assigned to IP Profiles (see 'Configuring IP Profiles' on page 248).
- You can also configure Allowed Coder Groups using the *ini* file parameter table AllowedCodersGroup.

➢ **To configure Allowed Coder Groups:**

1. Open the Allowed Coders Group page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Allowed Coders Group**).

Figure 20-20: Allowed Coders Group Page



2. From the 'Allowed Coders Group ID' drop-down list, select an ID for the Allowed Coder Group.
3. In the Coder Name table, select coders for the Allowed Coder Group.
4. Click Subm**it to** apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

20.2.4 Configuring SIP Message Policy Rules

You can configure SIP message policies for blocking (blacklist) unwanted incoming SIP messages and allowing (whitelist) receipt of desired messages. This feature allows you to define legal and illegal characteristics of a SIP message. The message policy can apply globally (default) or per signaling domain (i.e., assigned to a SIP interface in the SIP Interface table - see 'Configuring SIP Interface Table' on page 210).

This feature is helpful against VoIP fuzzing (also known as robustness testing), which sends different types of packets to its "victims" for finding bugs and vulnerabilities. For example, the attacker might try sending a SIP message containing either an over-sized parameter or too many occurrences of a parameter.

SIP message security rules are configured in the new Message Policy table (MessagePolicy). Each policy can be defined with the following:

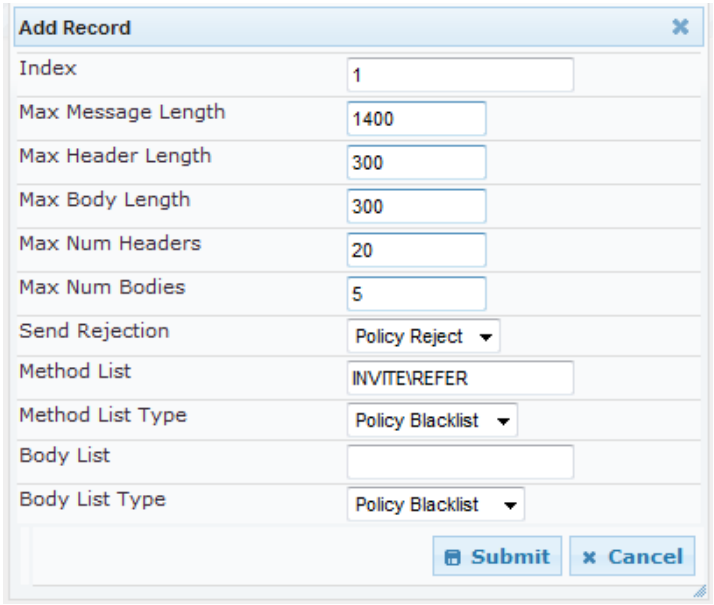
- Maximum message length
- Maximum SIP header length
- Maximum message body length
- Maximum number of headers

- Maximum number of bodies
- Option to send 400 "Bad Request" response if message request is rejected
- Blacklist and whitelist for defined SIP methods (e.g., INVITE)
- Blacklist and whitelist for defined SIP bodies

➢ **To configure SIP message policy rules:**

1. Open the Message Policy Table page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Message Policy Table**).
2. Click the **Add** button; the Add Record dialog box appears:

Figure 20-21: Adding SIP Message Policy Rule



The policy defined above limits messages to 32768 characters, headers to 256 characters, bodies to 512 characters, limits number of headers to 16, and only permits two bodies. Invalid requests are rejected. Only INVITE and BYE requests are permitted and there are no restrictions on bodies.

3. Configure the SIP message policy rule as required. See the table below for a description of each parameter.
4. Click Submit to apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 20-3: SIP Message Policy Parameters

Parameter	Description
Index **[MessagePolicy_Index]**	Defines the table index entry.
Max Message Length **[MessagePolicy_MaxMessageLength]**	Defines the maximum SIP message length. The valid value is up to 32768 characters.
Max Header Length **[MessagePolicy_MaxHeaderLength]**	Defines the maximum SIP header length. The valid value is up to 256 characters.
Max Body Length **[MessagePolicy_MaxBodyLength]**	Defines the maximum SIP message body length. This is the value of the Content-Length header. The valid value is up to 512 characters.

Parameter	Description
Max Num Headers **[MessagePolicy_MaxNumHeaders]**	Defines the maximum number of headers. The valid value is any number up to 16.
Max Num Bodies **[MessagePolicy_MaxNumBodies]**	Defines the maximum number of bodies. The valid value is any number up to 2.
Send Rejection **[MessagePolicy_SendRejection]**	Determines whether the device sends a 400 "Bad Request" response if a message request is rejected. ▪ Policy Reject = If the message is a request, then the device sends a response to reject the request. ▪ Policy Drop = The device ignores the message without sending any response.
Method List **[MessagePolicy_MethodList]**	Defines the SIP methods (e.g., INVITE\BYE) to which the rule applies. The syntax for entering the methods is as follows: ▪ Methods must be separated by a backslash (\). ▪ The entered value is not case sensitive.
Method List Type **[MessagePolicy_MethodListType]**	Determines the policy for the SIP methods. ▪ Policy Blacklist = The specified methods (in the 'Method List' field) are rejected by the policy. ▪ Policy Whitelist = The specified methods (in the 'Method List' field) are allowed by the policy.
Body List **[MessagePolicy_BodyList]**	Defines the SIP body (i.e., value of the Content-Type header) to which the rule applies.
Body List Type **[MessagePolicy_BodyListType]**	Determines the policy for the defined SIP body. ▪ Policy Blacklist =The specified SIP body (in the 'Body List' field) is rejected by the policy. ▪ Policy Whitelist = The specified SIP body (in the 'Body List' field) is allowed by the policy.

20.2.5 Routing SBC

This section describes the configuration of the routing entities for the SBC application. These include the following:

- Classification rules - see 'Configuring the Classification Table' on page 452
- Condition rules - see 'Configuring Condition Rules' on page 456
- IP-to-IP routing rules - see 'Configuring the IP-to-IP Routing' on page 457
- Alternative routing reasons - see 'Configuring Alternative Routing Reasons' on page 462

20.2.5.1 Configuring Classification Rules

The Classification table allows you to configure classification rules. Classification rules are used to classify incoming SIP dialog-initiating requests (e.g., SIP INVITE messages) to source IP Groups from where the SIP dialog request originated. Classification rules also enhance security by allowing you to create a SIP access list of whitelists and blacklists. Incoming SIP dialog requests matching the user-defined classification characteristics can be denied (i.e., blacklist) or allowed (i.e., whitelist). The identified source IP Group is later used in manipulation and routing processes.

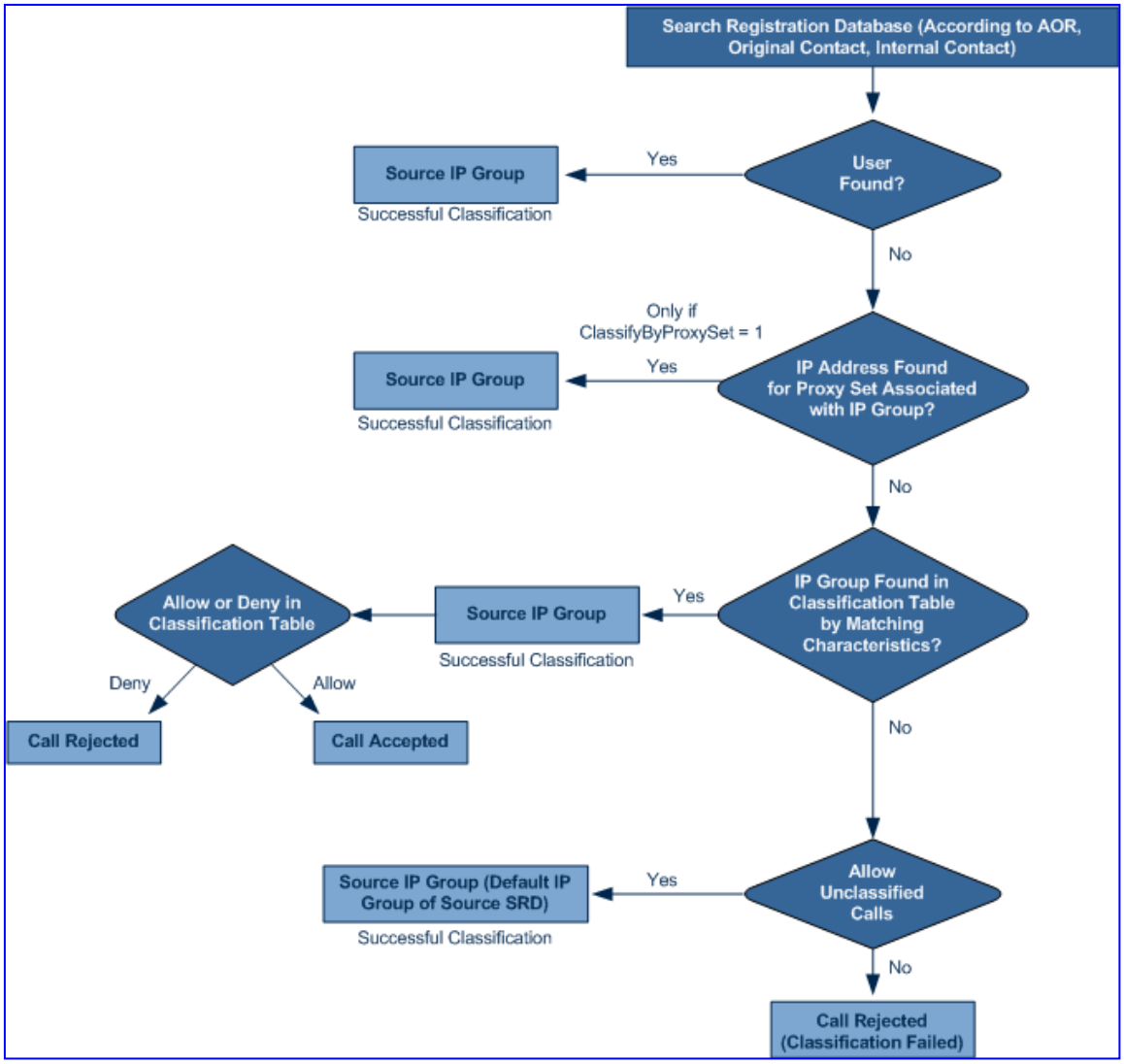
The Classification table is used to classify the incoming SIP dialog request only if classification based on the device's registration database and Proxy Set fails. The classification process is as follows:

1. Classification starts with the device's registration database, where it searches for a match by checking if the request arrived from a registered user in the database:
   - Compares Contact header of the received SIP dialog to the Contact of the registered user
   - Compares P-Asserted/From URL to the registered AOR
2. If the database search fails, the classification process proceeds with locating a Proxy Set associated with the IP address of the SIP dialog request, and then locating an IP Group associated with this Proxy Set (in the IP Group table). This classification stage is relevant only if enabled for the IP Group, using the ClassifyByProxySet parameter (see 'Configuring IP Groups' on page 212). If enabled, the device classifies requests arriving from the IP Group's Proxy Set as coming from this IP Group. The classification is done according to the Proxy IP list (in case of host names, then according to the dynamically resolved IP address list). Note that this classification is irrelevant in cases where multiple IP Groups use the same Proxy Set.
3. If classification based on Proxy Set fails, the device uses the Classification table to classify the SIP dialog to an IP Group. If it locates a classification rule whose characteristics (such as source IP address) match the incoming SIP dialog, then the SIP dialog is assigned to the associated IP Group. If the classification rule is defined as a whitelist, the SIP dialog is allowed and proceeds in the manipulation, routing and other processes. If the classification rule is defined as a blacklist, the SIP dialog is denied.
4. If the above classification process fails to determine the source IP Group to which the incoming packet belongs, the call is rejected or allowed, depending on the setting of the AllowUnclassifiedCalls parameter. If this parameter is enabled, the incoming SIP dialog is allowed and assigned to the default IP Group of the default SRD.

If the incoming SIP dialog cannot be classified according to the Classification table, the call is accepted or rejected based on the settings of the AllowUnclassifiedCalls parameter.

The flowchart below illustrates the classification process:

Figure 20-22: Classification Process (Identifying IP Group or Rejecting Call)



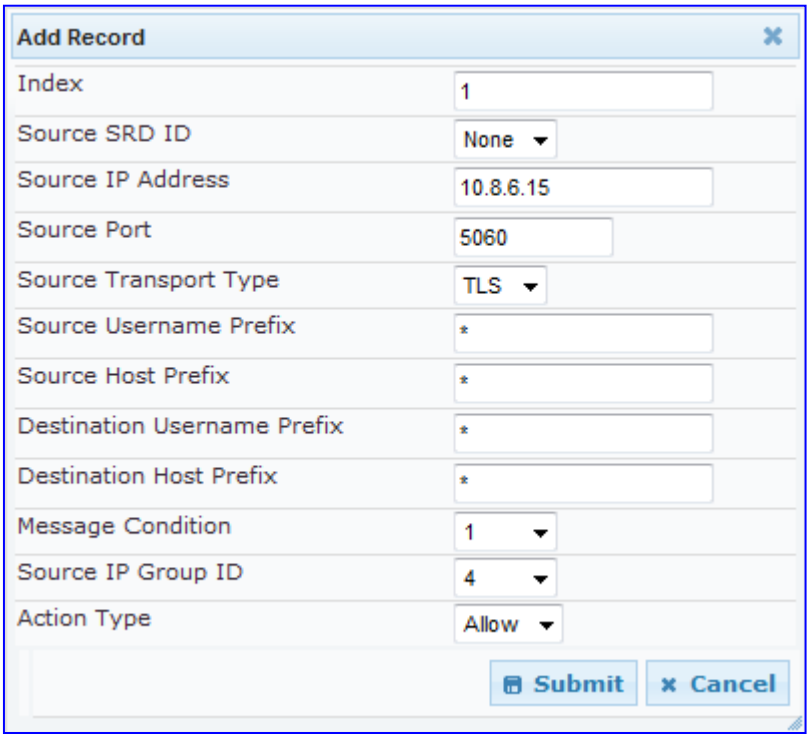
Notes:

- Incoming REGISTER messages are saved in the device's registration database and sent to a destination only if they are associated with a source IP Group that is of USER type.
- The Classification table can also be configured using the *ini* file table parameter Classification (see 'SBC Parameters' on page 979).

➢ **To configure classification rules:**

1. Open the Classification Table page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Routing SBC** submenu > **Classification Table**).
2. Click the **Add** button; the following appears:

Figure 20-23: Classification Table Page



The figure above shows an example classification rule that identifies an incoming SIP dialog to IP Group ID #4, if its source IP address is 10.8.6.15, source port is 5060, SIP transport is TLS, and matches the Message Condition rule 1. The rule also allows (whitelist) this SIP dialog.

3. Configure the classification rule as required. For a description of the parameters, see the table below.
4. Click Subm**it to** apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 20-4: Classification Table Parameters

Parameter	Description
Index	Defines the index number of the table row entry.
Matching Characteristics	
Source SRD ID **[Classification_SrcSRDID]**	Selects the SRD ID of the incoming SIP dialog. The default is -1 (i.e., no SRD is assigned). **Notes:** ▪ The SRDs are configured in the SRD table (see 'Configuring SRD Table' on page 207). ▪ The SRDs are also associated with a port number as defined by the SIP Interface used by the SRD (see 'Configuring SIP Interface Table' on page 210).
Source IP Address	Defines the source IP address (in dotted-decimal notation)

Parameter	Description
[Classification_SrcAddress]	of the incoming SIP dialog. **Notes:** ▪ If this parameter is not configured or is configured as an '*' (asterisk), then any source IP address is accepted. ▪ The IP address can include the "x" wildcard to represent single digits. For example: 10.8.8.xx represents all the addresses between 10.8.8.10 to 10.8.8.99. ▪ The IP address can include the asterisk (*) wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255.
Source Port **[Classification_SrcPort]**	Defines the source port number of the incoming SIP dialog.
Source Transport Type **[Classification_SrcTransportType]**	Defines the source transport type (UDP, TCP, or TLS) of the incoming SIP dialog.
Source Username Prefix **[Classification_SrcUsernamePrefix]**	Defines the prefix of the source URI user part of the incoming SIP dialog. This is typically located in the SIP From URI. **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 374.
Source Host Prefix **[Classification_SrcHost]**	Defines the prefix of the source URI host name in the From header of the incoming SIP dialog request. If this routing rule is not required, leave the field empty. The asterisk (*) symbol can be used to denote any source host prefix.
Destination Username Prefix **[Classification_DestUsernamePrefix]**	Defines the prefix of the destination Request-URI user part of the incoming SIP dialog. **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 374.
Destination Host Prefix **[Classification_DestHost]**	Defines the prefix of the destination Request-URI host name of the incoming SIP dialog request. If this routing rule is not required, leave the field empty. The asterisk (*) symbol can be used to denote any destination host prefix.
Message Condition **[Classification_MessageCondition]**	Assigns a Condition rule which can also be used to classify the incoming SIP dialog. **Note:** Condition rules are configured in the Condition Table (see 'Configuring Condition Rules for Classification' on page 456).
Operation Rule	
Source IP Group ID **[Classification_SrcIPGroupID]**	Assigns an IP Group to the incoming SIP dialog request if this SIP dialog matches the matching rule. The default is -1 (i.e., no IP Group is assigned). **Notes:** ▪ The IP Group must be associated with the selected SRD. ▪ The IP Group is used for SBC routing and manipulations. ▪ To define IP Groups, see 'Configuring IP Groups' on

Parameter	Description
	page 212.
Action Type **[Classification_ActionType]**	Defines a whitelist or blacklist for incoming SIP dialog requests that match the characteristics of the classification rule. ▪ **[0]** Deny = Blocks incoming SIP dialogs that match the characteristics of the Classification rule (blacklist). ▪ **[1]** Allow = Allows incoming SIP dialogs that match the characteristics of the Classification rule (whitelist), and assigns it to the associated IP Group. (default)

20.2.5.2 Configuring Condition Rules

Condition rules allow you to enhance the process of classifying an incoming SIP dialog to an IP Group by using SIP message rules. Condition rules are later assigned to classification rules in the Classification table (see 'Configuring Classification Table' on page 452). When a classification rule is associated with a condition rule, the classification is used only if the classification rule and its associated condition rule are matched.

Condition rules are SIP message conditions configured using the same syntax (match-condition) as in the Message Manipulations table (for example, "header.to.host contains company"). You can also define complex rules using the "AND" or "OR" Boolean operands.

➢ **To configure condition rules:**

1. Open the Condition Table page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Routing SBC** submenu > **Condition Table**).

Figure 20-24: Condition Table Page

Index	Condition	Description
1	param.ipg.src.type.==user	IP Group USER
2	header.via.exsts	Includes Via header
3	header.from.url.user==101	101 user part of From header

The figure above shows the following configurations:

- **Index 1:** Incoming SIP dialog that is classified as belonging to a USER-type IP Group.
- **Index 2:** Incoming SIP dialog with a SIP Via header.
- **Index 3:** Incoming SIP dialog with "101" as the user part in the SIP From header.

2. Click the Add button to add an entry, and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 20-5: Condition Table Parameters

Parameter	Description
Condition **[ConditionTable**_Condition]	Defines the condition rule of the SIP message. The valid value is a string. For a description of the string syntax, see 'Syntax' on page 1057. **Note:** Enclose user and host parts in single quotes.

Parameter	Description
Description **[ConditionTable**_Description]	Defines a brief description of the condition rule.

20.2.5.3 Configuring SBC IP-to-IP Routing

The IP2IP Routing Table page configures up to 120 SBC IP-to-IP routing rules. This table provides enhanced IP-to-IP call routing capabilities for routing received SIP dialog messages (e.g., INVITE) to a destination IP address. The SIP message is routed according to a routing rule whose configured input characteristics (e.g., Source IP Group) match the incoming SIP message. If the characteristics of an incoming call does not match the first rule, the call characteristics is then compared to those of the second rule, and so on until a matching rule is located. If no rule is matched, the call is rejected.

The IP-to-IP call destination can be one of the following:

- Registered user Contact listed in the device's database (only for USER-type IP Groups).
- Proxy Set associated with the destination IP Group (allows redundancy/load balancing).
- Specific destination address (can be based on IP address, host name, port, transport type, and/or SRD). Routing to a host name can be resolved using NAPTR/SRV/A-Record.
- Incoming Request-URI.
- ENUM query.

For all destination types listed above except destination IP Group, the IP Group can optionally be itself, configured to provide the destination SRD and/or IP Profile. If neither destination SRD nor destination IP Group is defined, the destination SRD is the source SRD and the destination IP Group is its default IP Group.

In addition to the alternative routing/load balancing provided by the Proxy Set associated with the destination IP Group, the table allows the configuration of alternative routes whereby if a route fails, the next adjacent (below) rule in the table that is configured as 'Alt Route Ignore/Consider Inputs' are used. The alternative routes rules can be set to enforce the input matching criteria or to ignore any matching criteria.

Alternative routing occurs upon one of the following conditions:

- A request sent by the device is responded with one of the following:
  - SIP response code (i.e., 4xx, 5xx, and 6xx SIP responses) configured in the SBC Alternative Routing Reasons table (see 'Configuring Alternative Routing Reasons' on page 462).
  - SIP 408 Timeout or no response (after timeout).
- The DNS resolution includes IP addresses that the device has yet to try (for the current call).

Messages are re-routed with the same SIP Call-ID and CSeq header fields (increased by 1).

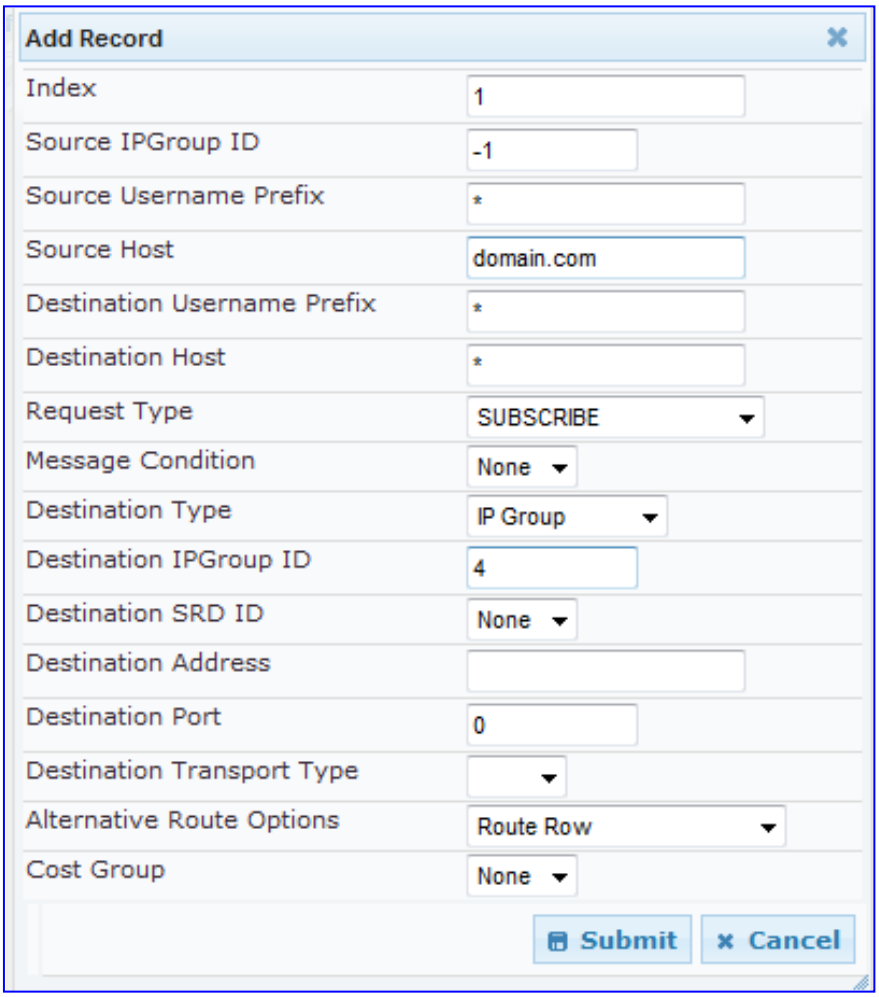
Notes:

- For a specific IP-to-IP routing rule to be effective, the incoming SIP dialog message must match the characteristics configured for that rule.
- The IP2IP Routing table can also be configured using the *ini* file table parameter IP2IPRouting (see 'SBC Parameters' on page 979).

➢ **To configure SBC IP-to-IP routing rules:**

1. Open the IP2IP Routing Table page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Routing SBC** submenu > **IP to IP Routing Table**).
2. Click the **Add** button; the Add Record dialog box appears:

Figure 20-25: SBC IP2IP Routing Table - Add Record Dialog Box



3. Add an entry and then configure it according to the table below.
4. Click the **Apply** button to save your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 20-6: IP2IP Routing Table Parameters

Parameter	Description
Matching Characteristics	
Source IP Group ID **[IP2IPRouting_SrcIPGroupID]**	Selects the IP Group from where the IP-to-IP call originated. Typically, the IP Group of an incoming SIP dialog is determined (or classified) using the 'Classification' table (see 'Configuring Classification Table' on page 452). If not used (i.e., any IP Group), simply leave the field empty. The default is -1.

Parameter	Description
Source Username Prefix **[IP2IPRouting_SrcUsernamePrefix]**	Defines the prefix of the user part of the incoming SIP dialog's source URI (usually the From URI). The default is "*". **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 374.
Source Host **[IP2IPRouting_SrcHost]**	Defines the host part of the incoming SIP dialog's source URI (usually the From URI). If this rule is not required, leave the field empty. To denote any host name, use the asterisk (*) symbol. The default is "*".
Destination Username Prefix **[IP2IPRouting_DestUsernamePrefix]**	Defines the prefix of the incoming SIP dialog's destination URI (usually the Request URI) user part. If this rule is not required, leave the field empty. To denote any prefix, use the asterisk (*) symbol. The default is "*". **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 374.
Destination Host **[IP2IPRouting_DestHost]**	Defines the host part of the incoming SIP dialog's destination URI (usually the Request-URI). If this rule is not required, leave the field empty. The asterisk (*) symbol can be used to denote any destination host. The default is "*".
Request Type **[IP2IPRouting_RequestType]**	Defines the SIP dialog request type of the incoming SIP dialog. ▪ **[0]** All (default) ▪ **[1]** INVITE ▪ **[2]** REGISTER ▪ **[3]** SUBSCRIBE ▪ **[4]** INVITE and REGISTER ▪ **[5]** INVITE and SUBSCRIBE ▪ **[6]** OPTIONS
Message Condition **[IP2IPRouting_MessageCondition]**	Selects a Message Condition rule. To configure Message Condition rules, see 'Configuring Condition Rules' on page 456.

Parameter	Description
Operation Routing Rule (when match occurs in characteristics)	
Destination Type **[IP2IPRouting_DestType]**	Determines the destination type to which the outgoing SIP dialog is sent. ▪ **[0]** IP Group (default) = The SIP dialog is sent to the IP Group's Proxy Set (SERVER-type IP Group) or registered contact from the database (if USER-type IP Group). ▪ **[1]** Dest Address = The SIP dialog is sent to the address configured in the following fields: 'Destination SRD ID', 'Destination Address', 'Destination Port', and 'Destination Transport Type'. ▪ **[2]** Request URI = The SIP dialog is sent to the address indicated in the incoming Request-URI. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence. ▪ **[3]** ENUM = An ENUM query is sent to include the destination address. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence. ▪ **[4]** Hunt Group = Used for call center survivability. For more information, see 'Call Survivability for Call Centers' on page 440.
Destination IP Group ID **[IP2IPRouting_DestIPGroupID]**	Defines the IP Group ID to where you want to route the call. The SIP dialog messages are sent to the IP address defined for the Proxy Set associated with this IP Group. If you select an IP Group, it is unnecessary to configure a destination IP address (in the 'Destination Address' field). However, if both parameters are configured, then the IP Group takes precedence. If the destination IP Group is of USER type, the device searches for a match between the Request-URI (of the received SIP dialog) to an AOR registration record in the device's database. The SIP dialog is then sent to the IP address of the registered contact. The default is -1. **Notes:** ▪ This parameter is only relevant if the parameter 'Destination Type' is set to 'IP Group'. However, regardless of the settings of the parameter 'Destination Type', the IP Group is still used - only for determining the IP Profile or outgoing SRD. If neither IP Group nor SRD are defined in this table, the destination SRD is determined according to the source SRD associated with the Source IP Group (configured in the IP Group table, see 'Configuring IP Groups' on page 212). If this table does not define an IP Group but only an SRD, then the first IP Group associated with this SRD (in the IP Group table) is used. ▪ If the selected destination IP Group ID is type SERVER, the request is routed according to the IP Group addresses.

Parameter	Description
	▪ If the selected destination IP Group ID is type USER, the request is routed according to the IP Group specific database (i.e., only to registered users of the selected database). ▪ If the selected destination IP Group ID is ANY USER (**[-2**]), the request is routed according to the general database (i.e., any matching registered user).
Destination SRD ID **[IP2IPRouting_DestSRDID]**	Defines the SRD ID. The default is None. **Note:** The destination IP Group must belong to the destination SRD, if both are configured in this table.
Destination Address **[IP2IPRouting_DestAddress]**	Defines the destination IP address (or domain name, e.g., domain.com) to where the call is sent. **Notes:** ▪ This parameter is applicable only if the parameter 'Destination Type' is set to 'Dest Address' **[1].** ▪ When using domain names, enter a DNS server IP address or alternatively, define these names in the 'Internal DNS Table' (see 'Configuring the Internal SRV Table' on page 138).
Destination Port **[IP2IPRouting_DestPort]**	Defines the destination port to where the call is sent.
Destination Transport Type **[IP2IPRouting_DestTransportType]**	Defines the transport layer type for sending the call: ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When this parameter is set to -1, the transport type is determined by the parameter SIPTransportType.
Alternative Route Options **[IP2IPRouting_AltRouteOptions]**	Determines whether this routing rule is the main routing rule or an alternative routing rule (to the rule defined directly above it in the table). ▪ **[0]** Route Row (default) = Main routing rule - the device first attempts to route the call to this route if the incoming SIP dialog's input characteristics matches this rule. ▪ **[1]** Alt Route Ignore Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route regardless of the incoming SIP dialog's input characteristics. ▪ **[2]** Alt Route Consider Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route only if the incoming SIP dialog matches this routing rule's input characteristics. **Notes:** ▪ The alternative routing entry (**[1]** or **[2]**) **must be defined in the next consecutive table entry index** to the Route Row entry (i.e., directly below it). For example, if Index 4 is configured as a Route Row, Index 5 must be configured as the alternative route. ▪ For IP-to-IP alternative routing, configure SBC alternative routing reasons upon receipt of 4xx, 5xx, and 6xx SIP

Parameter	Description
	responses (see 'Configuring Alternative Routing Reasons' on page 462). However, if no response, ICMP, or a SIP 408 response is received, the device attempts to use the alternative route even if no entries are configured in the 'SBC Alternative Routing Reasons' table. ▪ Multiple alternative route entries can be configured (e.g., Index 1 is the main route - Route Row - and indices 2 through 4 are configured as alternative routes).
Cost Group **[IP2IPRouting_CostGroup]**	Reserved for future use.

20.2.5.4 Configuring Alternative Routing Reasons

The SBC Alternative Routing Reasons page allows you to define up to five different call release (termination) reasons for call releases. If a call is released as a result of one of these reasons provided in SIP 4xx, 5xx, and 6xx response codes, the device attempts to locate an alternative route for the call. The call release reason type can be configured, for example, when there is no response to an INVITE message (after INVITE re-transmissions), where the device issues an internal 408 'No Response' implicit release reason.

Release reasons can also be configured to indicate that a route for an SRD or IP Group has reached its call admission control limit (i.e., maximum concurrent calls and/or call rate), as set in the Admission Control table (see 'Configuring Admission Control' on page 446). In such a scenario, an alternative route configured in the IP-to-IP Routing table can be used.

Alternative routes are configured in the IP2IP Routing table (see 'Configuring SBC IP-to-IP Routing Table' on page 457).

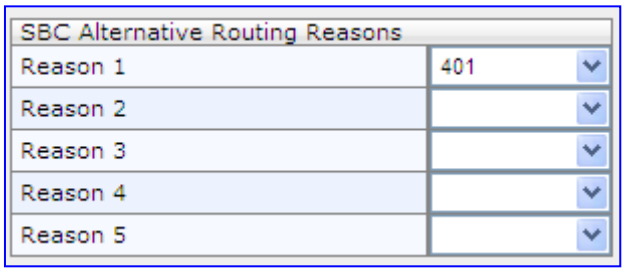
Notes:

- Alternative routing occurs even if this table is not configured upon scenarios where no response, ICMP, or a SIP 408 response is received.
- SIP requests pertaining to an SRD or IP Group that reach the call limit (maximum concurrent calls and/or call rate) as defined in the Call Admission table are sent to an alternative route if configured in the IP2IP Routing table for the SRD or IP Group. If no alternative routing rule is located, the device automatically rejects the SIP request with a SIP 486 "Busy Here" response.
- You can also configure alternative routing reasons using the *ini* file table parameter SBCAlternativeRoutingReasons.

➢ **To configure SIP reason codes for alternative IP routing:**

1. Open the SBC Alternative Routing Reasons page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Routing SBC** submenu > **Alternative Routing Reasons**).

Figure 20-26: Alternative Routing Reasons Page



2. Configure different call failure reasons that invoke alternative routing.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

20.2.6 Manipulations SBC

This section describes the configuration of the manipulation rules for the SBC application. These include the following:

- SIP message manipulation - see 'SIP Message Manipulations' on page 464
- IP-to-IP inbound manipulation - see 'Configuring IP-to-IP Inbound Manipulations' on page 467
- IP-to-IP outbound manipulation - see 'Configuring IP-to-IP Outbound Manipulations' on page 469

20.2.6.1 Configuring Message Manipulations

The Message Manipulations page allows you to define up to 200 SIP message manipulation rules. This manipulation includes insertion, removal, and/or modification of SIP headers. Multiple manipulation rules can be configured for the same SIP message.

Once you have defined SIP message manipulation rules (*Manipulation Set ID*), you can assign them to inbound and outbound SIP messages for IP Groups in the IP Group table.

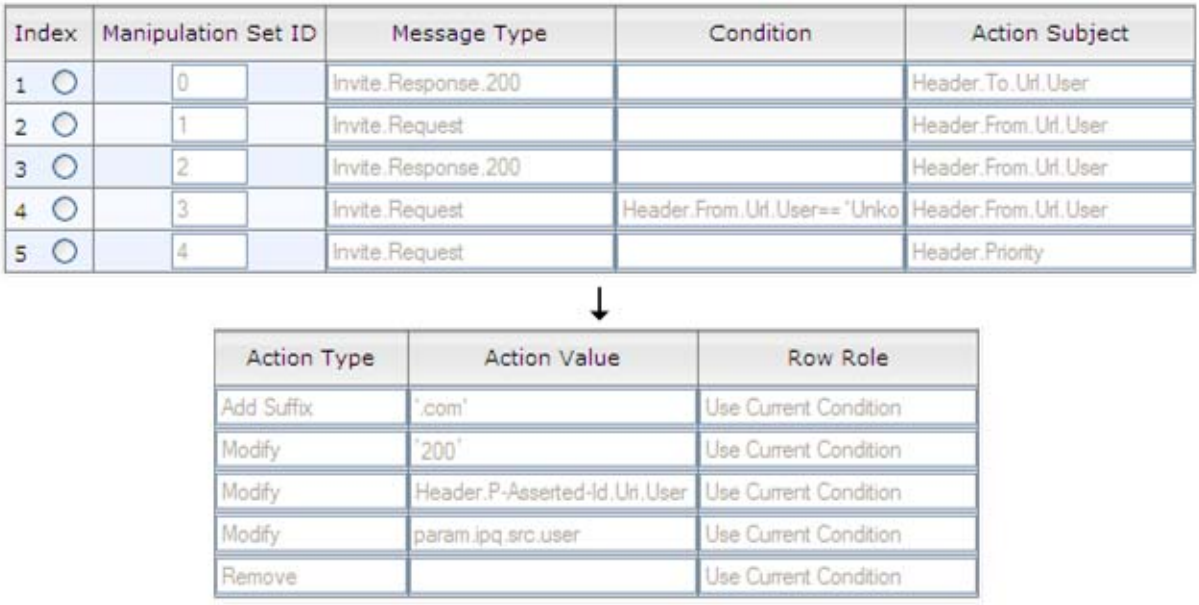
Notes:

- For more information on the syntax for configuring SIP message manipulation rules in the Message Manipulation table, see Appendix B, 'SIP Message Manipulation Syntax'.
- The values entered in the table are not case-sensitive.
- For configuring Message Manipulation using the *ini* file, see the parameter MessageManipulations.
- For the SBC application, SIP message manipulation is done on the inbound and outbound legs (i.e., only after classification, inbound/outbound manipulations, and routing).
- Each message can be manipulated twice - once for the source leg manipulation rules and once in the destination leg (source and destination IP Groups).
- Unknown SIP parts can only be added or removed.
- SIP manipulations do not allow you to remove or add mandatory SIP headers. They can only be modified and only on requests that initiate new dialogs. Mandatory SIP headers include To, From, Via, CSeq, Call-Id, and Max-Forwards.
- Manipulation of SDP body is currently not supported.

➢ **To configure SIP message manipulation rules:**

1. Open the Message Manipulations page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Manipulations SBC** submenu > **Message**).

Figure 20-27: Message Manipulations Page



Index	Manipulation Set ID	Message Type	Condition	Action Subject
1	0	Invite.Response.200		Header.To.Url.User
2	1	Invite.Request		Header.From.Url.User
3	2	Invite.Response.200		Header.From.Url.User
4	3	Invite.Request	Header.From.Url.User== 'Unko	Header.From.Url.User
5	4	Invite.Request		Header.Priority

Action Type	Action Value	Row Role
Add Suffix	'.com'	Use Current Condition
Modify	'200'	Use Current Condition
Modify	Header.P-Asserted-Id.Uri.User	Use Current Condition
Modify	param.ipq.src.user	Use Current Condition
Remove		Use Current Condition

The previous figure shows the following message manipulation rules:

- Index 1: adds the suffix ".com" to the host part of the To header.
- Index 2: changes the user part of the SIP From header to "200".
- Index 3: changes the user part of the From header to the user part of the P-Asserted-ID.
- Index 4: if the user part of the From header equals "unknown", then it is changed according to the srcIPGroup call's parameter.
- Index 5: removes the Priority header from an incoming INVITE message.

2. Add an entry and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 20-7: Message Manipulations Parameters

Parameter	Description
Manipulation Set ID **[ManSetID]**	Defines a Manipulation Set ID for the rule. You can define the same Manipulation Set ID for multiple rules and thereby, create a group of rules that you can assign to an IP entity. The Manipulation Set IDs are later used to assign the manipulation rules to an IP Group (see 'Configuring IP Groups' on page 212) for inbound and/or outbound messages.
Matching Characteristics	
Message Type **[MessageType]**	Defines the SIP message type that you want to manipulate. The valid value is a string denoting the SIP message. For example: ▪ Empty = rule applies to all messages

Parameter	Description
	▪ Invite = rule applies to all INVITE requests and responses ▪ Invite.Request = rule applies to INVITE requests ▪ Invite.Response = rule applies to INVITE responses ▪ subscribe.response.2xx = rule applies to SUBSCRIBE confirmation responses **Note:** Currently, SIP 100 Trying messages cannot be manipulated.
Condition **[Condition]**	Defines the condition that must exist for the rule to apply. The valid value is a string. For example: ▪ header.from.url.user== '100' (indicates that the user part of the From header must have the value "100") ▪ header.contact.param.expires > '3600' ▪ header.to.url.host contains 'domain' ▪ param.call.dst.user != '100' **Note:** Currently, SDP body message types are not supported.
Operation	
Action Subject **[ActionSubject]**	Defines the SIP header upon which the manipulation is performed.
Action Type **[ActionType]**	Defines the type of manipulation. ▪ **[0]** Add (default) = adds new header/param/body (header or parameter elements). ▪ **[1]** Remove = removes header/param/body (header or parameter elements). ▪ **[2]** Modify = sets element to the new value (all element types). ▪ **[3]** Add Prefix = adds value at the beginning of the string (string element only). ▪ **[4]** Add Suffix = adds value at the end of the string (string element only). ▪ **[5]** Remove Suffix = removes value from the end of the string (string element only). ▪ **[6]** Remove Prefix = removes value from the beginning of the string (string element only).
Action Value **[ActionValue]**	Defines a value (string) that you want to use in the manipulation. The syntax is as follows: string/<message-element>/<call-param> "+" string/<message-element>/<call-param> For example: ▪ 'itsp.com' ▪ header.from.url.user ▪ param.call.dst.user ▪ param.call.dst.host + '.com' ▪ param.call.src.user + '<' + header.from.url.user + '@' + header.p-asserted-id.url.host + '>' **Note:** Only single quotation marks must be used.
Row Role **[RowRole]**	Determines which condition must be used for the rule of this table row. ▪ **[0]** Use Current Condition = The condition entered in this row must be matched in order to perform the defined action (default). ▪ **[1]** Use Previous Condition = The condition of the rule configured

Parameter	Description
	directly above this row must be used in order to perform the defined action. This option allows you to configure multiple actions for the same condition. **Note:** When multiple manipulations rules apply to the same header, the next rule applies to the result string of the previous rule.

20.2.6.2 Configuring IP-to-IP Inbound Manipulations

The IP to IP Inbound Manipulation page allows you to configure up to 100 manipulation rules for manipulating the SIP URI user part (source and destination) of inbound SIP dialog requests. You can apply these manipulations to different SIP dialog message types (e.g., INVITE or REGISTER).

- Manipulated destination URI user part are done on the following SIP headers: Request-URI, To, and Remote-Party-ID (if exists)
- Manipulated source URI user part are done on the following SIP headers: From, P-Asserted (if exists), P-Preferred (if exists), and Remote-Party-ID (if exists)

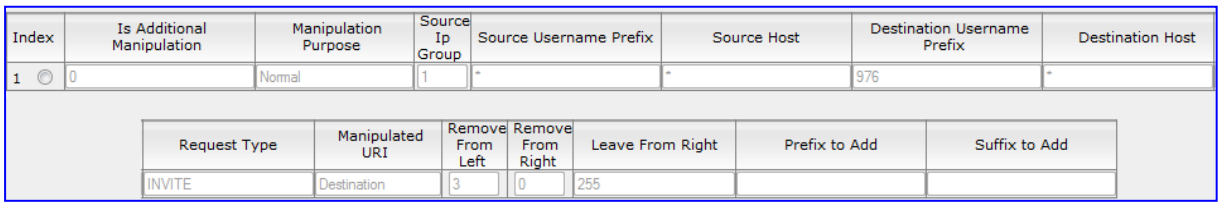
Notes:

- For a specific manipulation rule to be effective, the incoming SIP dialog must match the configured characteristics.
- SIP URI host name (source and destination) manipulations are configured in the IP Group table (see 'Configuring IP Groups' on page 212). These manipulations are simply host name substitutions with the names defined for the source and destination IP Groups respectively.
- The IP to IP Inbound Manipulation table can also be configured using the *ini* file table parameter IPInboundManipulation (see 'SBC Parameters' on page 979).

➢ **To configure IP-to-IP inbound manipulation rules:**

1. Open the IP to IP Inbound Manipulation page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Manipulations SBC** submenu > **IP to IP Inbound**).

Figure 20-28: IP to IP Inbound Manipulation Page



The figure above shows a manipulation configuration example that removes the destination URI user name prefix "976" in incoming INVITE messages received from IP Group #1 .

2. Add an entry and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 20-8: IP to IP Inbound Manipulation Parameters

Parameter	Description
Matching Characteristics	
Is Additional Manipulation **[IsAdditionalManipulation]**	Determines whether additional SIP URI user part manipulation is done for the table entry rule listed directly above it. ▪ **[0]** 0 = Regular manipulation rule (not done in addition to the rule above it). ▪ **[1]** 1 = If the above row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule. **Note:** Additional manipulation can only be performed on a different SIP URI (either source or destination) to the rule configured in the row above (defined by the parameter ManipulatedURI).
Manipulation Purpose **[ManipulationPurpose]**	Defines the purpose of the manipulation: ▪ **[0]** Normal = Inbound manipulations affect the routing input and source and/or destination number (default). ▪ **[1]** Routing input only = Inbound manipulations affect the routing input only, retaining the original source and destination number. ▪ **[2]** Shared Line = Used for the Shared-Line Appearance feature. This manipulation is for registration requests to change the destination number of the secondary extension numbers to the primary extension.
Source IP Group **[SrcIpGroup]**	Defines the IP Group from where the incoming INVITE is received. For any Source IP Group, enter the value -1.
Source Username Prefix **[SrcUsernamePrefix]**	Defines the prefix of the source SIP URI user name (usually in the From header). For any prefix, enter the asterisk "*" symbol (default). **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 374.
Source Host **[SrcHost]**	Defines the source SIP URI host name - full name (usually in the From header). For any host name, enter the asterisk "*" symbol (default).
Destination Username Prefix **[DestUsernamePrefix]**	Defines the prefix of the destination SIP URI user name (usually in the Request-URI). For any prefix, enter the asterisk "*" symbol (default). **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 374.
Destination Host **[DestHost]**	Defines the destination SIP URI host name - full name (usually in the Request URI). For any host name, enter the asterisk "*" symbol (default).
Request Type **[RequestType]**	Defines the SIP request type to which the manipulation rule is applied. ▪ **[0]** All = all SIP messages (default) ▪ **[1]** INVITE = all SIP messages except REGISTER and SUBSCRIBE ▪ **[2]** REGISTER = only SIP REGISTER messages ▪ **[3]** SUBSCRIBE = only SIP SUBSCRIBE messages ▪ **[4]** INVITE and REGISTER = all SIP messages except SUBSCRIBE ▪ **[5]** INVITE and SUBSCRIBE = all SIP messages except REGISTER

Parameter	Description
Manipulated URI **[ManipulatedURI]**	Determines whether the source or destination SIP URI user part is manipulated. ▪ **[0]** Source = Manipulation is done on the source SIP URI user part. (default) ▪ **[1]** Destination = Manipulation is done on the destination SIP URI user part.
Operation Manipulation Rule (when match occurs in characteristics)	
Remove From Left **[RemoveFromLeft]**	Defines the number of digits to remove from the left of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".
Remove From Right **[RemoveFromRight]**	Defines the number of digits to remove from the right of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "j".
Leave From Right **[LeaveFromRight]**	Defines the number of characters that you want retained from the right of the user name.
Prefix to Add **[Prefix2Add]**	Defines the number or string that you want added to the front of the user name. For example, if you enter 'user' and the user name is "john", the new user name is "userjohn".
Suffix to Add **[Suffix2Add]**	Defines the number or string that you want added to the end of the user name. For example, if you enter '01' and the user name is "john", the new user name is "john01".

20.2.6.3 Configuring IP-to-IP Outbound Manipulations

The IP to IP Outbound Manipulation page allows you to configure up to 100 manipulation rules for manipulating SIP URI user part (source and destination) of outbound SIP dialog requests. Manipulation rules in the table are located according to the source IP Group, and source and destination host and user prefixes and can be applied to a user-defined SIP request type (e.g., INVITE, OPTIONS, SUBSCRIBE, and /or REGISTER). However, since outbound manipulations are done only after routing, the outbound manipulation rule matching can also be done by destination IP Group.

Manipulated destination URI user part are performed on the following SIP headers: Request URI, To, and Remote-Party-ID (if exists). Manipulated source URI user part are performed on the following SIP headers: From, P-Asserted (if exists), P-Preferred (if exists), and Remote-Party-ID (if exists).

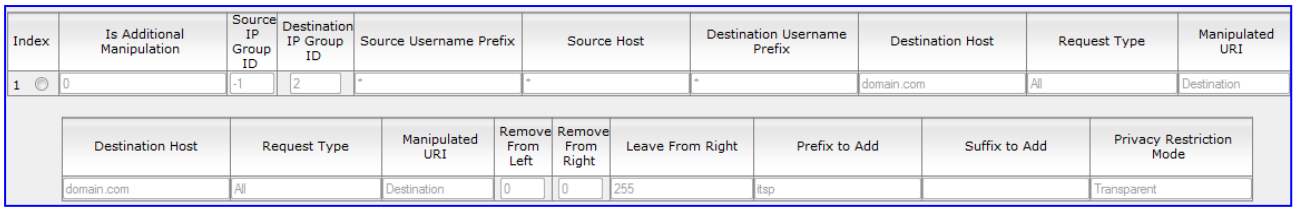
Notes:

- For a specific manipulation rule to be effective, the incoming SIP dialog must match the characteristics configured for that rule.
- SIP URI host name (source and destination) manipulations are configured in the IP Group table. These manipulations are simply host name substitutions with the names defined for the source and destination IP Groups respectively.
- The IP to IP Outbound Manipulation table can also be configured using the *ini* file table parameter IPOutboundManipulation (see 'SBC Parameters' on page 979).

➢ **To configure IP-to-IP outbound manipulation rules:**

1. Open the IP to IP Outbound Manipulation page (**Configuration** tab > **VoIP** menu > **SBC** submenu > **Manipulations SBC** submenu > **IP to IP Outbound**).

Figure 20-29: IP to IP Outbound Manipulation Page



2. Add an entry and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Table 20-9: IP to IP Outbound Manipulation Table Parameters

Parameter	Description
Matching Characteristics	
Is Additional Manipulation **[IsAdditionalManipulation]**	Determines whether additional SIP URI user part manipulation is done for the table entry rule listed directly above it. ▪ **[0]** 0 = Regular manipulation rule - not done in addition to the rule above it (default). ▪ **[1]** 1 = If the previous table row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule. **Note:** Additional manipulation can only be performed on a different SIP URI (either source or destination) to the rule configured in the row above (defined by the parameter ManipulatedURI).
Source IP Group ID **[SrcIPGroupID]**	Defines the IP Group from where the INVITE is received. For any Source IP Group, enter the value -1.
Destination IP Group ID **[DestIPGroupID]**	Defines the IP Group to where the INVITE is to be sent. For any Destination IP Group, enter the value -1.
Source Username Prefix **[SrcUsernamePrefix]**	Defines the prefix of the source SIP URI user name (usually in the From header). For any prefix, enter the asterisk "*" symbol (default). **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 374.
Source Host **[SrcHost]**	Defines the source SIP URI host name - full name (usually in the From header). For any host name, enter the asterisk "*" symbol (default).
Destination Username Prefix **[DestUsernamePrefix]**	Defines the prefix of the destination SIP URI user name (usually in the Request-URI). For any prefix, enter the asterisk "*" symbol (default). **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 374.
Destination Host **[DestHost]**	Defines the destination SIP URI host name - full name (usually in the Request URI). For any host name, enter the asterisk "*" symbol (default).

Parameter	Description
Request Type **[RequestType]**	Defines the SIP request type to which the manipulation rule is applied. ▪ **[0]** All = all SIP messages (default) ▪ **[1]** INVITE = all SIP messages except REGISTER and SUBSCRIBE ▪ **[2]** REGISTER = only SIP REGISTER messages ▪ **[3]** SUBSCRIBE = only SIP SUBSCRIBE messages ▪ **[4]** INVITE and REGISTER = all SIP messages except SUBSCRIBE ▪ **[5]** INVITE and SUBSCRIBE = all SIP messages except REGISTER
Manipulated URI **[IsAdditionalManipulation]**	Determines whether the source or destination SIP URI user part is manipulated. ▪ **[0]** Source = Manipulation is done on the source SIP URI user part (default). ▪ **[1]** Destination = Manipulation is done on the destination SIP URI user part.
Operation Manipulation Rule (when match occurs in characteristics)	
Remove From Left **[RemoveFromLeft]**	Defines the number of digits to remove from the left of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".
Remove From Right **[RemoveFromRight]**	Defines the number of digits to remove from the right of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "j".
Leave From Right **[LeaveFromRight]**	Defines the number of characters that you want retained from the right of the user name.
Prefix to Add **[Prefix2Add]**	Defines the number or string that you want added to the front of the user name. For example, if you enter 'user' and the user name is "john", the new user name is "userjohn".
Suffix to Add **[Suffix2Add]**	Defines the number or string that you want added to the end of the user name. For example, if you enter '01' and the user name is "john", the new user name is "john01".
Privacy Restriction Mode **[PrivacyRestrictionMode]**	Determines user privacy handling (i.e., restricting source user identity in outgoing SIP dialogs). ▪ **[0]** Transparent = No intervention in SIP privacy (default). ▪ **[1]** Don't change privacy = The user identity remains the same as in the incoming SIP dialog. If a restricted number exists, the restricted presentation is normalized as follows: ✓ From URL header: anonymous@anonymous.invalid. ✓ If a P-Asserted-Identity header exists (either in the incoming SIP dialog or added by the device), a Privacy header is added with the value "id". ▪ **[2]** Restrict = The user identity is restricted (the restricted presentation is as mentioned above). ▪ **[3]** Remove Restriction = The device attempts to reveal the user identity by setting user values in the From header and removing the privacy "id" value if the Privacy header exists. If the From header user is anonymous, the value is taken from the P-Preferred-Identity, P-Asserted-Identity, or Remote-Party-ID header (if

Parameter	Description
	exists). The device identifies an incoming user as restricted if one of the following exists: ▪ From header user is anonymous. ▪ P-Asserted-Identity and Privacy headers contain the value "id". **Note:** All restriction logic is performed after the user number has been manipulated.

21 Stand-Alone Survivability (SAS) Application

This section describes the Sand-Alone Survivability (SAS) application.

21.1 Overview

The device's Stand-Alone Survivability (SAS) feature ensures telephony communication continuity (survivability) for enterprises using hosted IP services (such as IP Centrex) or IP-PBX in cases of failure of these entities. In case of failure of the IP Centrex, IP-PBX servers (or even WAN connection and access Internet modem), the enterprise typically loses its internal telephony service at any branch, between its offices, and with the external environment. In addition, typically these failures lead to the inability to make emergency calls (e.g., 911 in North America). Despite these possible points of failure, the device's SAS feature ensures that the enterprise's telephony services (e.g., SIP IP phones or soft phones) are maintained, by routing calls to the PSTN (i.e., providing PSTN fallback).

Notes:

- The SAS application is available only if the device is installed with the SAS Software Upgrade Key.
- Throughput this section, the term *user agent* (UA) refers to the enterprise's LAN phone user (i.e., SIP telephony entities such as IP phones).
- Throughout this section, the term *proxy* or *proxy server* refers to the enterprise's centralized IP Centrex or IP-PBX.
- Throughout this section. the term *SAS* refers to the SAS application running on the device.

21.1.1 SAS Operating Modes

The device's SAS application can be implemented in one of the following main modes:

- **Outbound Proxy:** In this mode, SAS receives SIP REGISTER requests from the enterprise's UAs and forwards these requests to the external proxy (i.e., outbound proxy). When a connection with the external proxy fails, SAS enters SAS emergency state and serves as a proxy, by handling internal call routing for the enterprise's UAs - routing calls between UAs and if setup, routing calls between UAs and the PSTN. For more information, see 'SAS Outbound Mode' on page 474.
- **Redundant Proxy:** In this mode, the enterprise's UAs register with the external proxy and establish calls directly through the external proxy, without traversing SAS (or the device per se'). Only when connection with the proxy fails, do the UAs register with SAS, serving now as the UAs redundant proxy. SAS then handles the calls between UAs, and between the UAs and the PSTN (if setup). This mode is operational only during SAS in emergency state. This mode can be implemented, for example, for proxies that accept only SIP messages that are sent directly from the UAs. For more information, see 'SAS Redundant Mode' on page 475.

Note: It is recommended to implement the SAS outbound mode.

21.1.1.1 SAS Outbound Mode

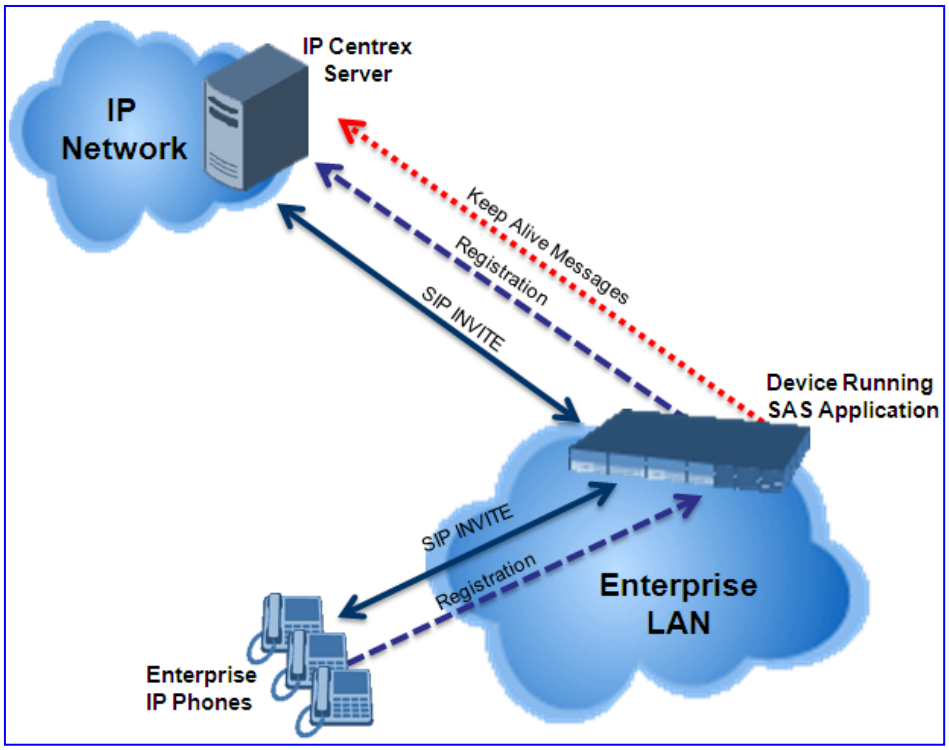
This section describes the SAS outbound mode, which includes the following states:

- Normal state (see 'Normal State' on page 474)
- Emergency state (see 'Emergency State' on page 474)

21.1.1.1.1 Normal State

In normal state, SAS receives REGISTER requests from the enterprise's UAs and forwards them to the external proxy (i.e., outbound proxy). Once the proxy replies with a SIP 200 OK, the device records the Contact and address of record (AOR) of the UAs in its internal SAS registration database. Therefore, in this mode, SAS maintains a database of all the registered UAs in the network. In addition, SAS continuously maintains a keep-alive mechanism toward the external proxy, using SIP OPTIONS messages. The figure below illustrates the operation of SAS outbound mode in normal state:

Figure 21-1: SAS Outbound Mode in Normal State (Example)



21.1.1.1.2 Emergency State

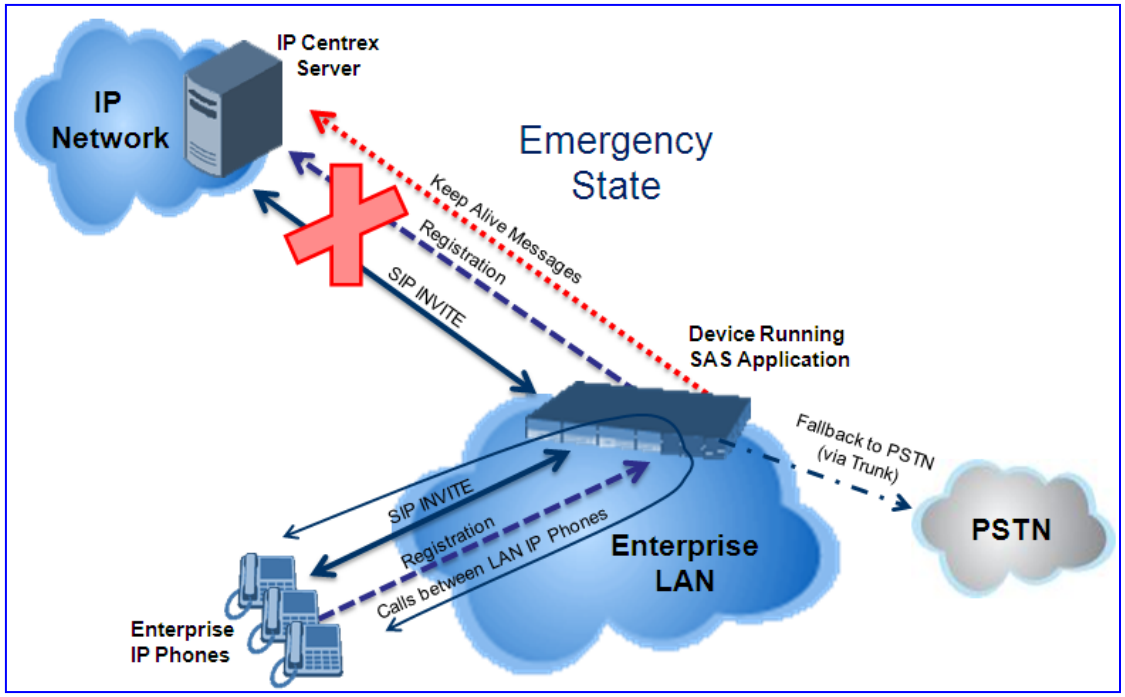
When a connection with the external proxy fails (detected by the device's keep-alive messages), the device enters SAS emergency state. The device serves as a proxy for the UAs, by handling internal call routing of the UAs (within the LAN enterprise).

When the device receives calls, it searches its SAS registration database to locate the destination address (according to AOR or Contact). If the destination address is not found, SAS forwards the call to the default gateway. Typically, the default gateway is defined as the device itself (on which SAS is running), and if the device has PSTN interfaces, the enterprise preserves its capability for outgoing calls (from UAs to the PSTN network).

The routing logic of SAS in emergency state is described in detail in 'SAS Routing in Emergency State' on page 479.

The figure below illustrates the operation of SAS outbound mode in emergency state:

Figure 21-2: SAS Outbound Mode in Emergency State (Example)



When emergency state is active, SAS continuously attempts to communicate with the external proxy, using keep-alive SIP OPTIONS. Once connection to the proxy returns, the device exits SAS emergency state and returns to SAS normal state, as explained in 'Exiting Emergency and Returning to Normal State' on page 477.

21.1.1.2 SAS Redundant Mode

In SAS redundant mode, the enterprise's UAs register with the external proxy and establish calls directly through it, without traversing SAS (or the device per se'). Only when connection with the proxy fails, do the UAs register with SAS, serving now as the UAs redundant proxy. SAS then handles the calls between UAs, and between the UAs and the PSTN (if setup).

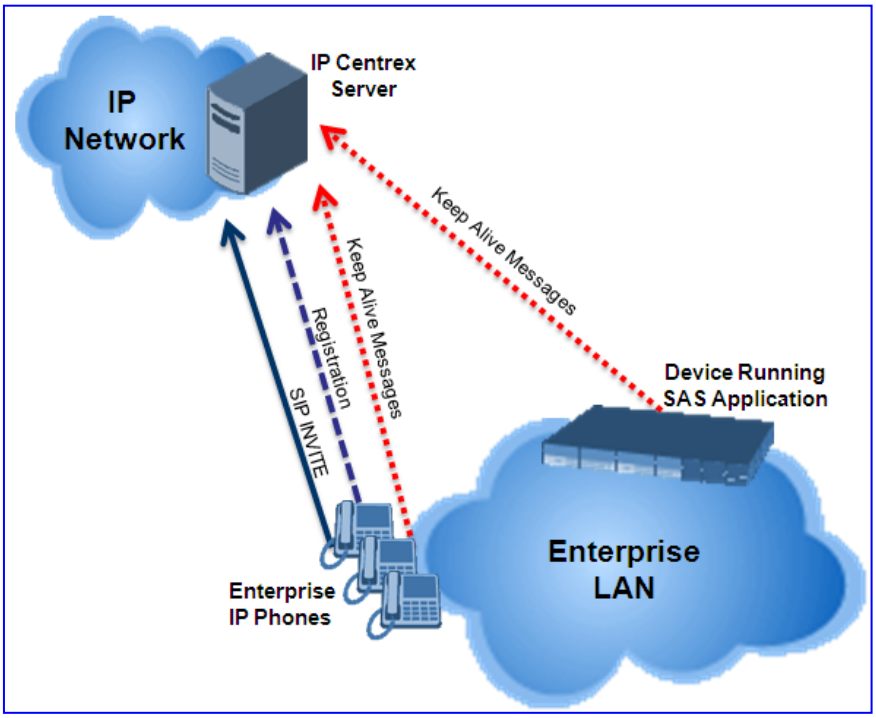
This mode is operational only during SAS in emergency state.

Note: In this SAS deployment, the UAs (e.g., IP phones) must support configuration for primary and secondary proxy servers (i.e., proxy redundancy), as well as homing. Homing allows the UAs to switch back to the primary server from the secondary proxy once the connection to the primary server returns (UAs check this using keep-alive messages to the primary server). If homing is not supported by the UAs, you can configure SAS to ignore messages received from UAs in normal state (the 'SAS Survivability Mode' parameter must be set to 'Always Emergency' / 2) and thereby, “force” the UAs to switch back to their primary proxy.

21.1.1.2.1 Normal State

In normal state, the UAs register and operate directly with the external proxy.

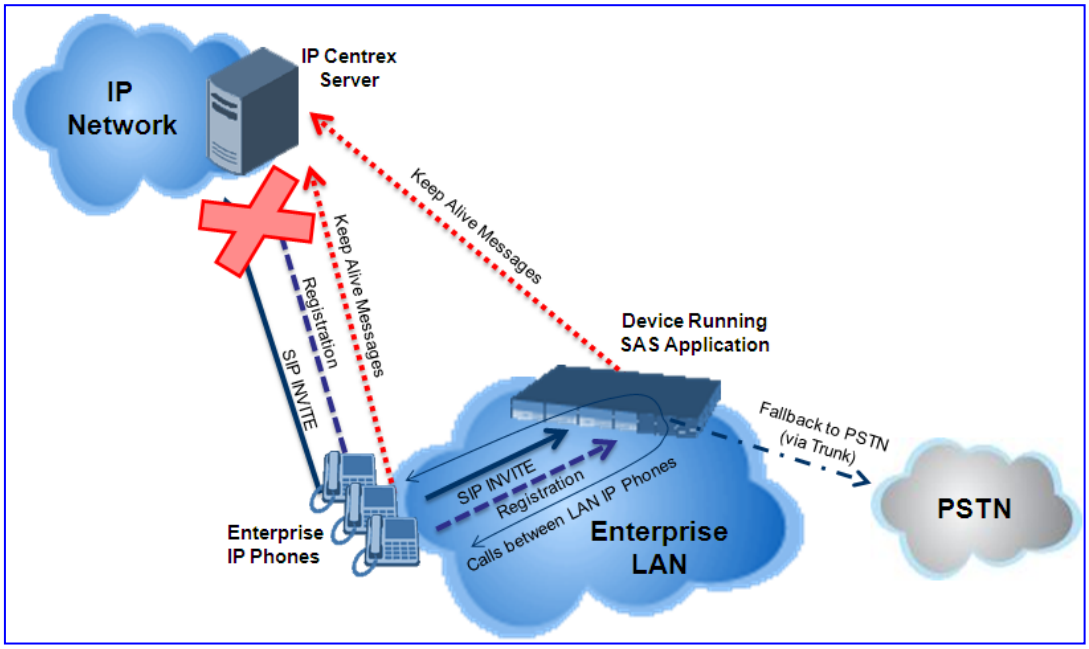
Figure 21-3: SAS Redundant Mode in Normal State (Example)



21.1.1.2.2 Emergency State

If the UAs detect that their primary (external) proxy does not respond, they immediately register to SAS and start routing calls to it.

Figure 21-4: SAS Redundant Mode in Emergency State (Example)



21.1.1.2.3 Exiting Emergency and Returning to Normal State

Once the connection with the primary proxy is re-established, the following occurs:

- **UAs:** switch back to operate with the primary proxy.
- **SAS:** ignores REGISTER requests from the UAs, forcing the UAs to switch back to the primary proxy.

  Note: This is applicable only if the 'SAS Survivability Mode' parameter is set to 'Always Emergency' (2).

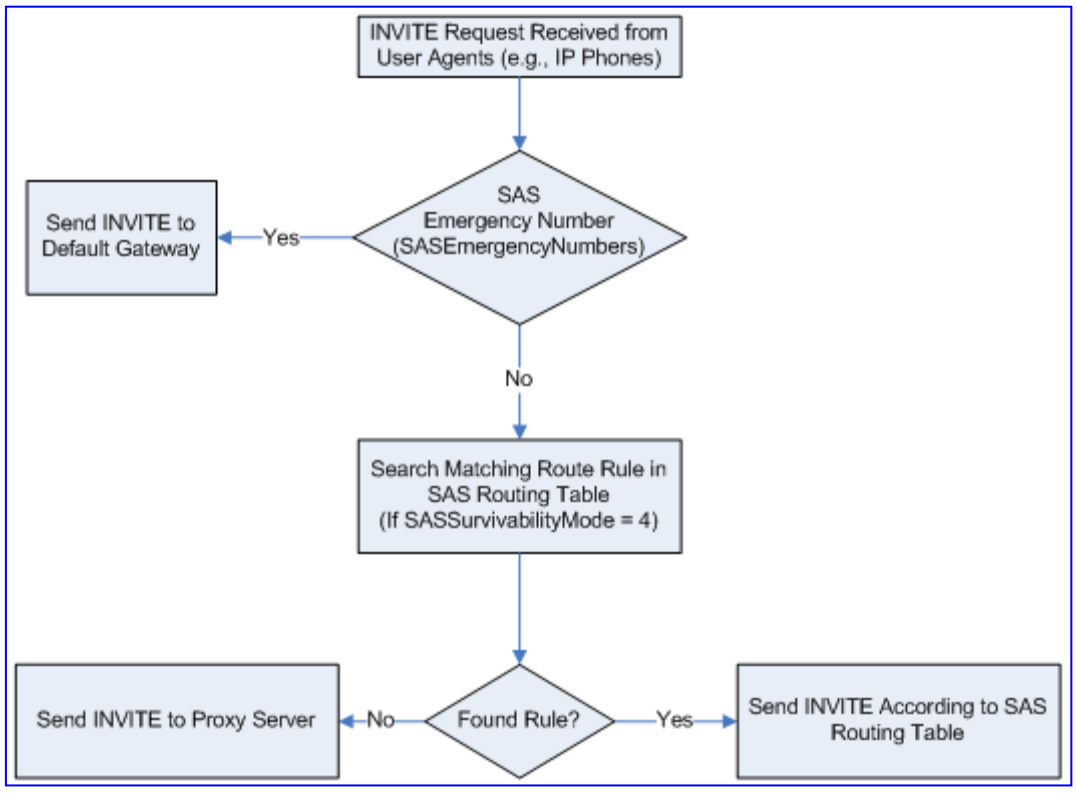
21.1.2 SAS Routing

This section provides flowcharts describing the routing logic for SAS in normal and emergency states.

21.1.2.1 SAS Routing in Normal State

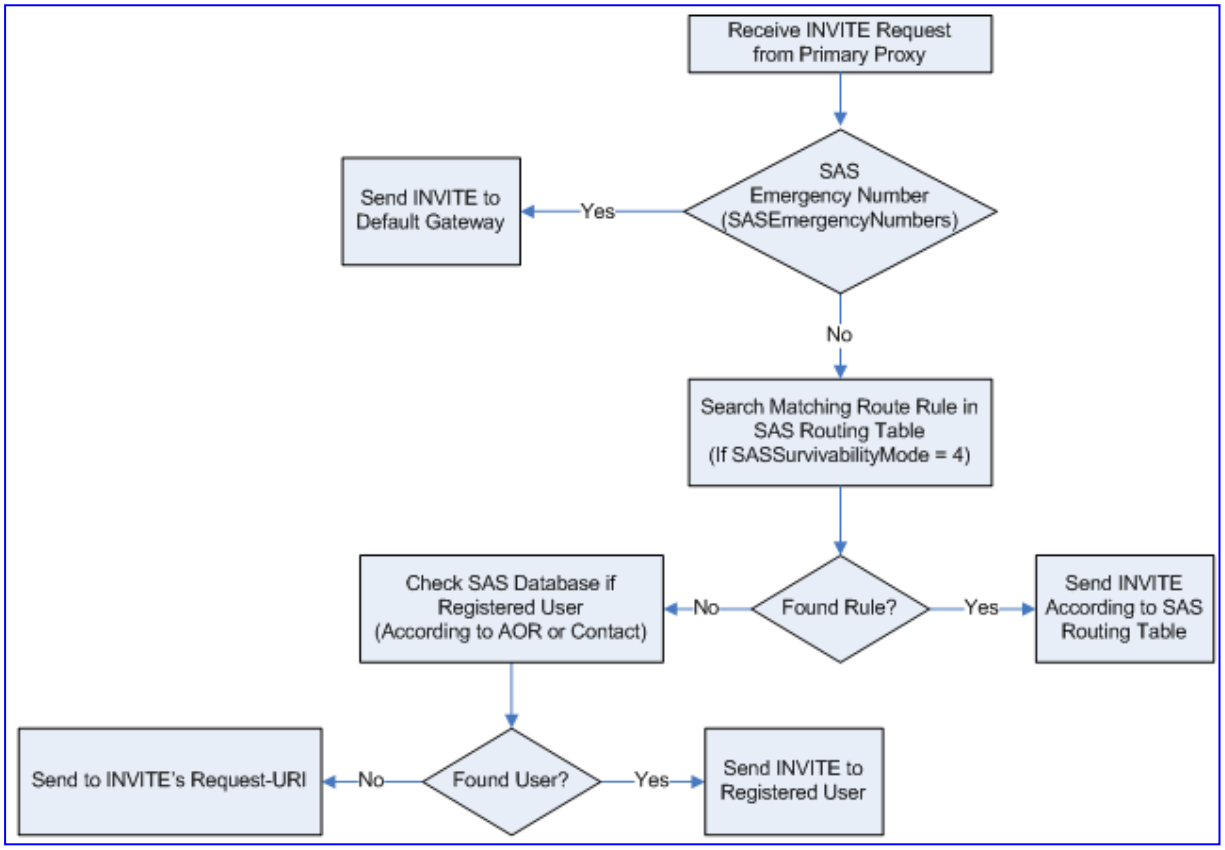
The flowchart below displays the routing logic for SAS in normal state for INVITE messages received from the UAs:

Figure 21-5: Flowchart of INVITE from UA's in SAS Normal State



The flowchart below displays the routing logic for SAS in normal state for INVITE messages received from the external proxy:

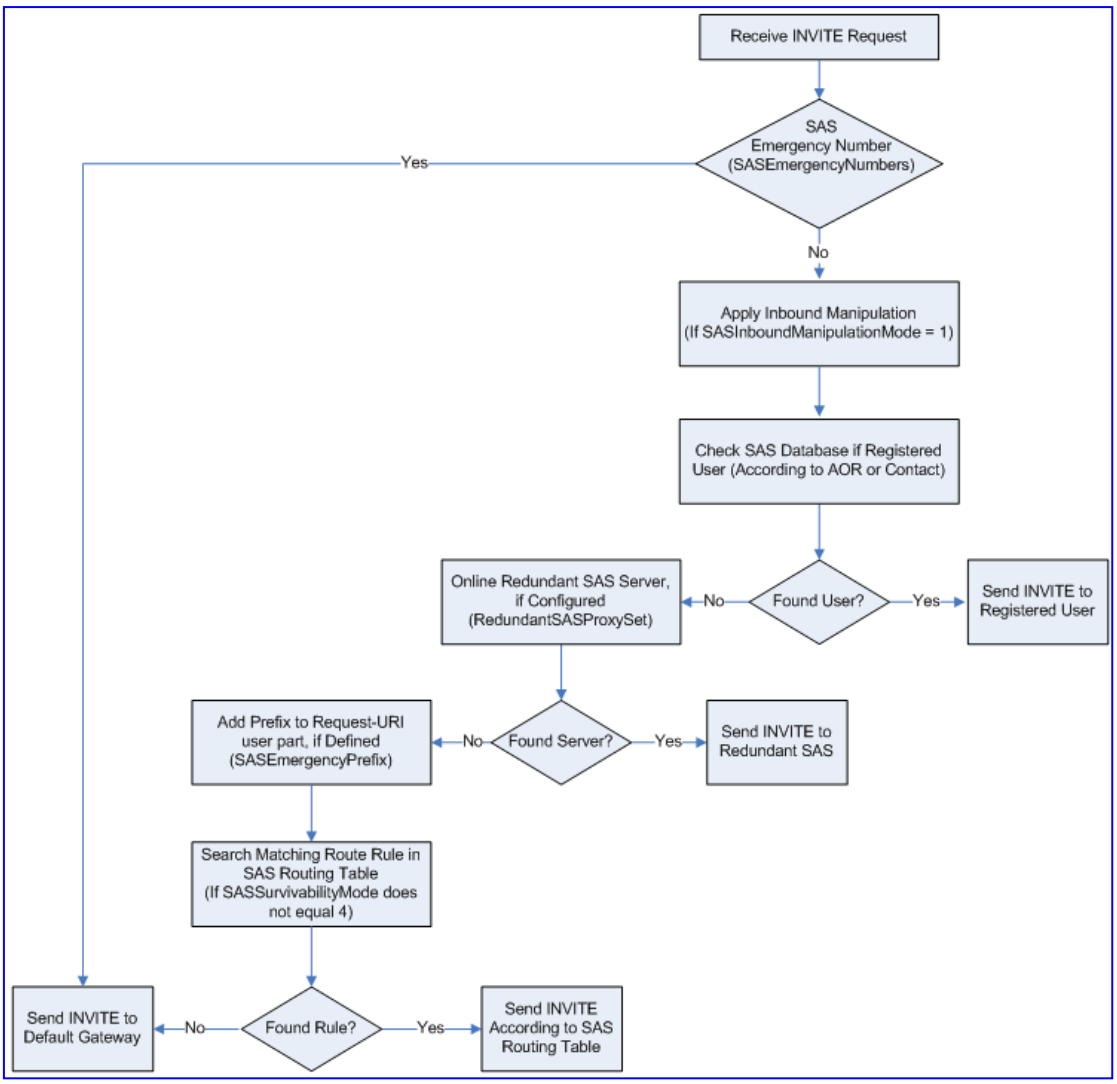
Figure 21-6: Flowchart of INVITE from Primary Proxy in SAS Normal State



21.1.2.2 SAS Routing in Emergency State

The flowchart below shows the routing logic for SAS in emergency state:

Figure 21-7: Flowchart for SAS Emergency State



21.2 SAS Configuration

SAS supports various configuration possibilities, depending on how the device is deployed in the network and the network architecture requirements. This section provides step-by-step procedures on configuring the SAS application, using the device's Web interface.

The SAS configuration includes the following:

- General SAS configuration that is common to all SAS deployment types (see 'General SAS Configuration' on page 480)
- SAS outbound mode (see 'Configuring SAS Outbound Mode' on page 483)
- SAS redundant mode (see 'Configuring SAS Redundant Mode' on page 484)
- Gateway and SAS applications deployed together (see 'Configuring Gateway Application with SAS' on page 484)
- Optional, advanced SAS features (see 'Advanced SAS Configuration' on page 488)

21.2.1 General SAS Configuration

This section describes the general configuration required for the SAS application. This configuration is applicable to all SAS modes.

21.2.1.1 Enabling the SAS Application

Before you can configure SAS, you need to enable the SAS application on the device. Once enabled, the device's Web interface provides the SAS pages for configuring SAS.

Note: The SAS application is available only if the device is installed with the SAS Software Upgrade Key. If your device is not installed with the SAS feature, contact your AudioCodes representative.

➢ **To enable the SAS application:**

1. Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** > **Applications Enabling**).
2. From the 'SAS Application' drop-down list, select **Enable**.

Figure 21-8: Enabling SAS

3. Click **Submit**.
4. Save the changes to the flash memory with a device reset; after the device resets, the SAS menu appears and you can now begin configuring the SAS application.

21.2.1.2 Configuring Common SAS Parameters

The procedure below describes how to configure SAS settings that are common to all SAS modes. This includes various SAS parameters as well as configuring the Proxy Set for the SAS proxy (if required). The SAS Proxy Set ID defines the address of the UAs' external proxy.

➢ **To configure common SAS settings:**

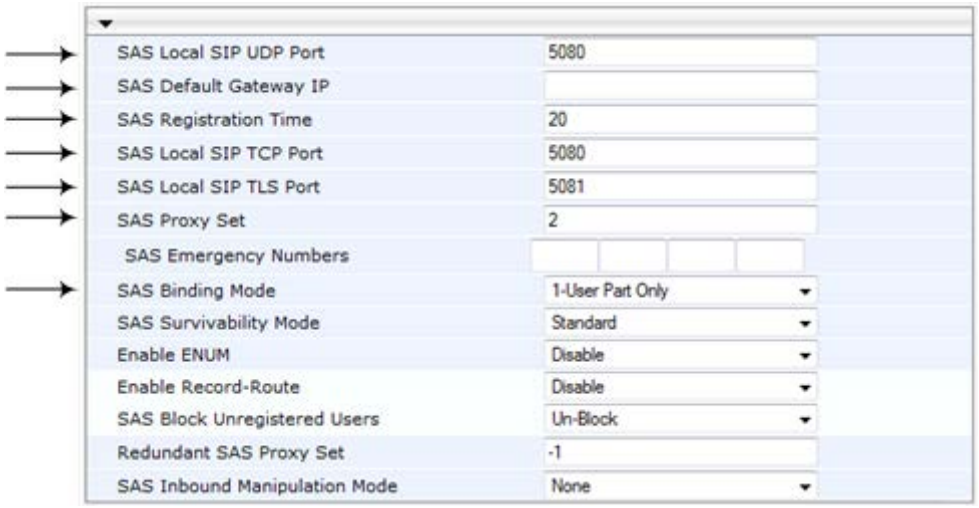
1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Stand Alone Survivability**).
2. Define the port used for sending and receiving SAS messages. This can be any of the following port types:
   - UDP port - defined in the 'SAS Local SIP UDP Port' field
   - TCP port - defined in the 'SAS Local SIP TCP Port' field
   - TLS port - defined in the 'SAS Local SIP TLS Port' field

Note: This SAS port must be different than the device's local gateway port (i.e., that defined for the 'SIP UDP/TCP/TLS Local Port' parameter in the 'SIP General Parameters page page - **Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**).

3. In the 'SAS Default Gateway IP' field, define the IP address and port (in the format *x.x.x.x:port)* of the device (i.e., Gateway application). Note that the port of the device is defined by the parameter 'SIP UDP Local Port' (refer to the note in Step 2 above).
4. In the 'SAS Registration Time' field, define the value for the SIP Expires header, which is sent in the 200 OK response to an incoming REGISTER message when SAS is in emergency state.
5. From the 'SAS Binding Mode' drop-down list, select the database binding mode:
   - **0-URI:** If the incoming AOR in the REGISTER request uses a 'tel:' URI or 'user=phone', the binding is done according to the Request-URI user part only. Otherwise, the binding is done according to the entire Request-URI (i.e., user and host parts - user@host).
   - **1-User Part Only:** Binding is done according to the user part only.

   You must select **1-User Part Only** in cases where the UA sends REGISTER messages as SIP URI, but the INVITE messages sent to this UA include a Tel URI. For example, when the AOR of an incoming REGISTER is sip:3200@domain.com, SAS adds the entire SIP URI (e.g., sip:3200@domain.com) to its database (when the parameter is set to '0-URI'). However, if a subsequent Request-URI of an INVITE message for this UA arrives with sip:3200@10.1.2.3 user=phone, SAS searches its database for "3200", which it does not find. Alternatively, when this parameter is set to '1-User Part Only', then upon receiving a REGISTER message with sip:3200@domain.com, SAS adds only the user part (i.e., "3200") to its database. Therefore, if a Request-URI of an INVITE message for this UA arrives with sip:3200@10.1.2.3 user=phone, SAS can successfully locate the UA in its database.

Figure 21-9: Configuring Common Settings



6. In the 'SAS Proxy Set' field, enter the Proxy Set used for SAS. The SAS Proxy Set must be defined only for the following SAS modes:
   - **Outbound mode:** In SAS normal state, SAS forwards REGISTER and INVITE messages received from the UAs to the proxy servers defined in this Proxy Set.
   - **Redundant mode and only if UAs don't support homing:** SAS sends keep-alive messages to this proxy and if it detects that the proxy connection has resumed, it ignores the REGISTER messages received from the UAs, forcing them to send their messages directly to the proxy.

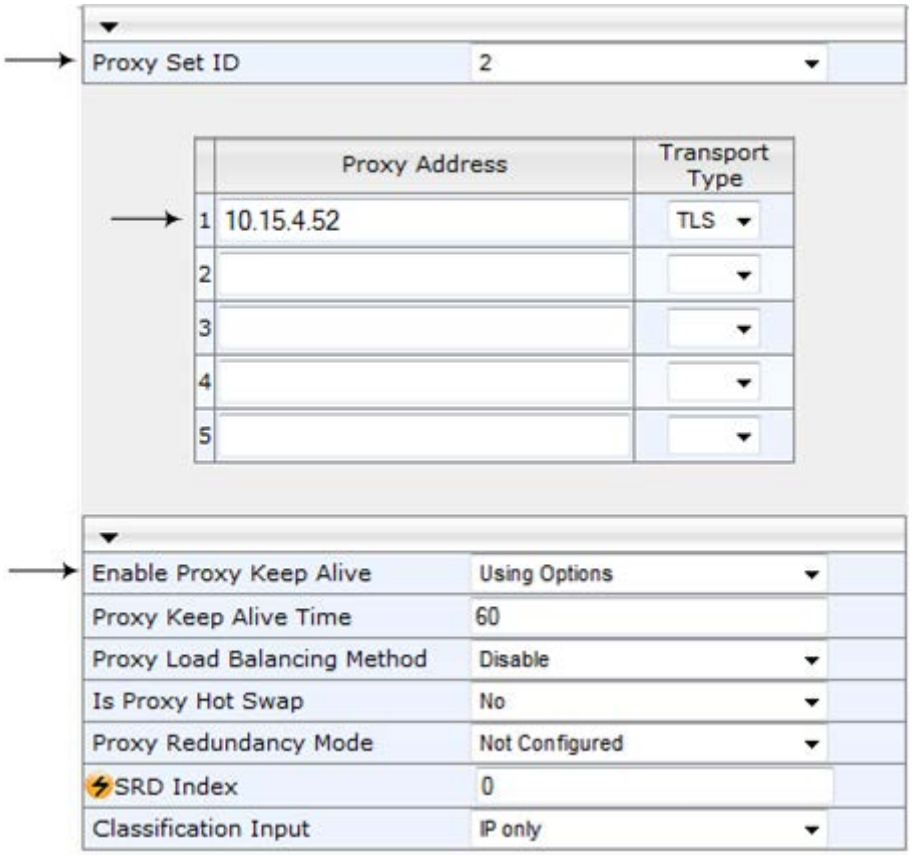
   If you define a SAS Proxy Set ID, you must configure the Proxy Set as described in Step 8 below.
7. Click **Submit** to apply your settings.
8. If you defined a SAS Proxy Set ID in Step 6 above, then you must configure the SAS Proxy Set ID:
   a. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Networks** > **Proxy Set Table**).
   b. From the 'Proxy Set ID' drop-down list, select the required Proxy Set ID.

Notes:
- The selected Proxy Set ID number must be the same as that specified in the 'SAS Proxy Set' field in the 'SAS Configuration page (see Step 6).
- Do not use Proxy Set ID 0.

   c. In the 'Proxy Address' field, enter the IP address of the external proxy server.

d. From the 'Enable Proxy Keep Alive' drop-down list, select **Using Options**. This instructs the device to send SIP OPTIONS messages to the proxy for the keep-alive mechanism.

Figure 21-10: Defining UAs' Proxy Server



e. Click **Submit** to apply your settings.

21.2.2 Configuring SAS Outbound Mode

This section describes how to configure the SAS outbound mode. These settings are in addition to the ones described in 'Configuring Common SAS Parameters' on page 481.

Note: The VoIP CPEs (such as IP phones or residential gateways) need to be defined so that their proxy and registrar destination addresses and ports are the same as that configured for the device's SAS IP address and SAS local SIP port. In some cases, on the UAs, it is also required to define SAS as their outbound proxy, meaning that messages sent by the UAs include the host part of the external proxy, but are sent (on Layer 3/4) to the IP address / UDP port of SAS.

➢ **To configure SAS outbound mode:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Stand Alone Survivability**).
2. From the 'SAS Survivability Mode' drop-down list, select **Standard**.
3. Click **Submit**.

21.2.3 Configuring SAS Redundant Mode

This section describes how to configure the SAS redundant mode. These settings are in addition to the ones described in 'Configuring Common SAS Parameters' on page 481.

Note: The VoIP CPEs (such as IP phones or residential gateways) need to be defined so that their primary proxy is the external proxy, and their redundant proxy destination addresses and port is the same as that configured for the device's SAS IP address and SAS SIP port.

➢ **To configure SAS redundant mode:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Stand Alone Survivability**).
2. From the 'SAS Survivability Mode' drop-down list, select one of the following, depending on whether the UAs support homing (i.e., they always attempt to operate with the primary proxy, and if using the redundant proxy, they switch back to the primary proxy whenever it's available):
   - **UAs support homing:** Select **Always Emergency**. This is because SAS does not need to communicate with the primary proxy of the UAs; SAS serves only as the redundant proxy of the UAs. When the UAs detect that their primary proxy is available, they automatically resume communication with it instead of with SAS.
   - **UAs do not support homing:** Select **Ignore REGISTER**. SAS uses the keep-alive mechanism to detect availability of the primary proxy (defined by the SAS Proxy Set). If the connection with the primary proxy resumes, SAS ignores the messages received from the UAs, forcing them to send their messages directly to the primary proxy.
3. Click **Submit**.

21.2.4 Configuring Gateway Application with SAS

If you want to run both the Gateway and SAS applications on the device, the configuration described in this section is required. The configuration steps depend on whether the Gateway application is operating with SAS in outbound mode or SAS in redundant mode.

Note: The Gateway application must use the same SAS operation mode as the SIP UAs. For example, if the UAs use the SAS application as a redundant proxy (i.e., SAS redundancy mode), then the Gateway application must do the same.

21.2.4.1 Gateway with SAS Outbound Mode

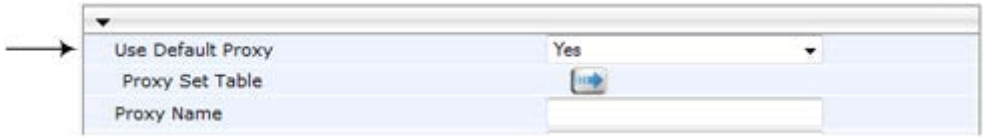
The procedure below describes how to configure the Gateway application with SAS outbound mode.

➢ **To configure Gateway application with SAS outbound mode:**

1. Define the proxy server address for the Gateway application:
   a. Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Proxy & Registration**).

b. From the 'Use Default Proxy' drop-down list, select **Yes**.

Figure 21-11: Enabling Proxy Server for Gateway Application



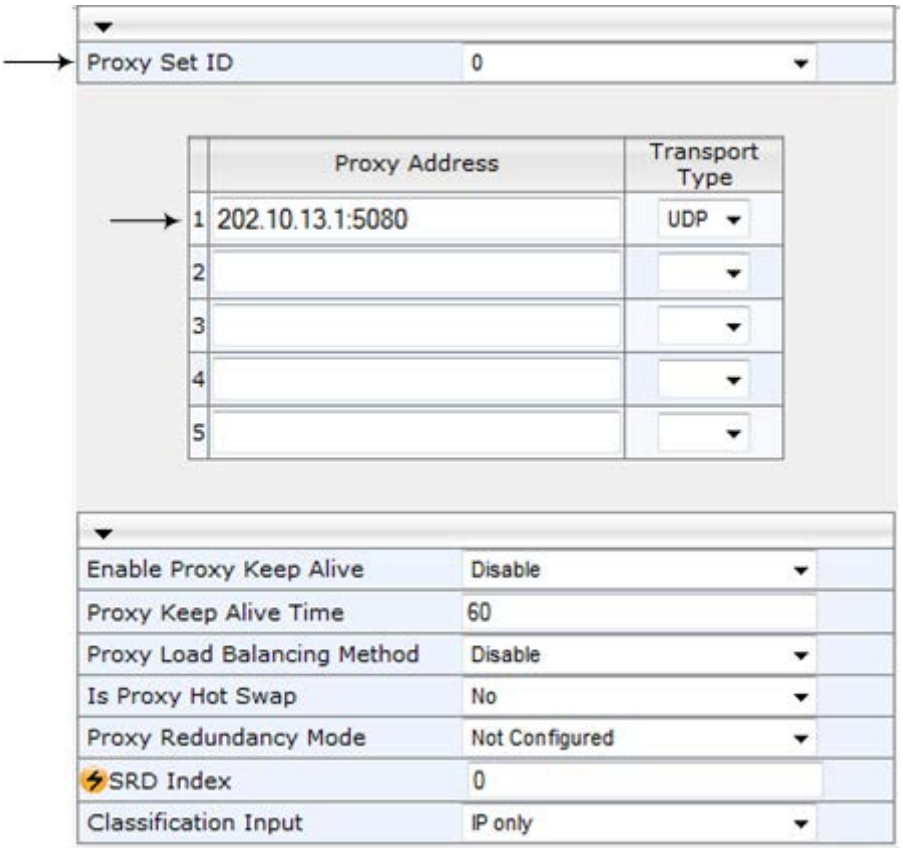
c. Click **Submit**.

d. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **Proxy Sets** Table).

e. From the 'Proxy Set ID' drop-down list, select **0**.

f. In the first 'Proxy Address' field, enter the IP address and port of the device (in the format *x.x.x.x:port*). This is the port as defined in the 'SAS Local UDP/TCP/TLS Port' field (see 'Configuring Common SAS Parameters' on page 481).

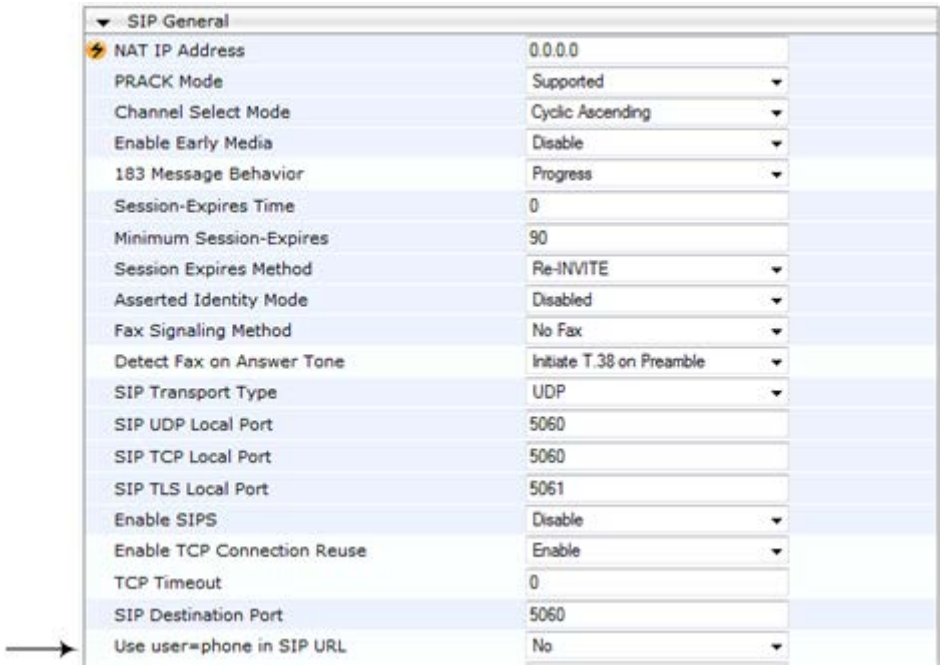
Figure 21-12: Defining Proxy Server for Gateway Application



g. Click **Submit**.

2. Disable use of user=phone in SIP URL:
   a. Open the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **General Parameters**).
   b. From the 'Use user=phone in SIP URL' drop-down list, select **No**. This instructs the Gateway application not to use *user=phone* in the SIP URL and therefore, REGISTER and INVITE messages use SIP URI. (By default, REGISTER messages are sent with *sip uri* and INVITE messages with *tel uri*.)

Figure 21-13: Disabling user=phone in SIP URL



   c. Click **Submit**.

21.2.4.2 Gateway with SAS Redundant Mode

The procedure below describes how to configure the Gateway application with SAS redundant mode.

➢ **To configure Gateway application with SAS redundant mode:**

1. Define the proxy servers for the Gateway application:
   a. Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Proxy & Registration**).
   b. From the 'Use Default Proxy' drop-down list, select **Yes**.

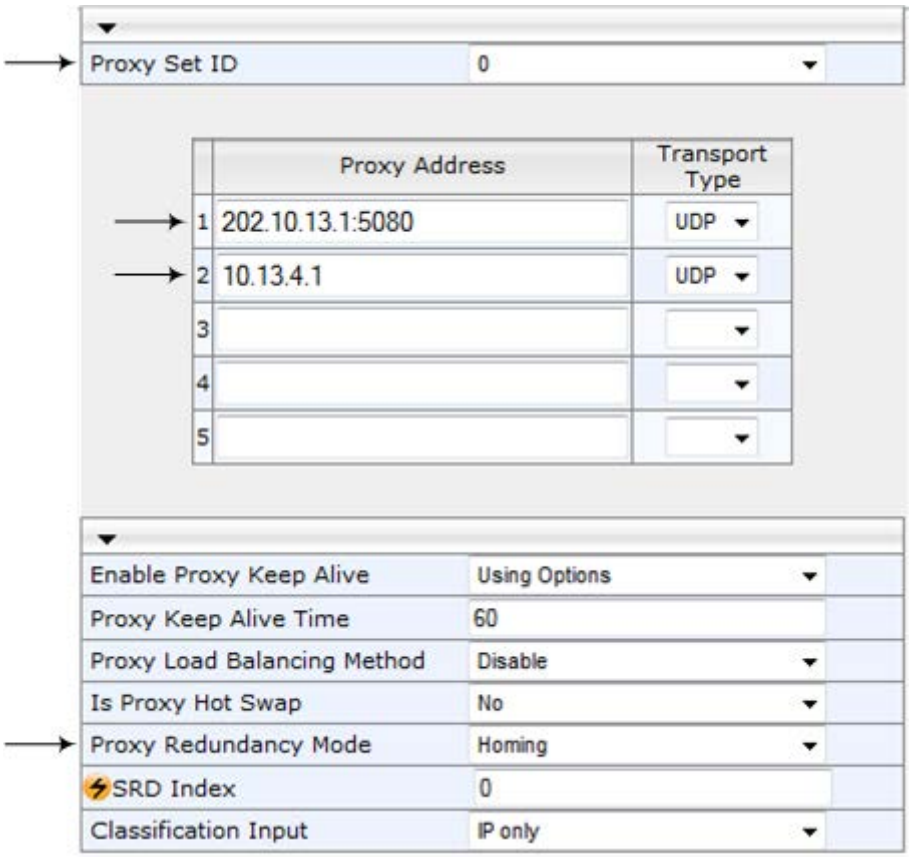
Figure 21-14: Enabling Proxy Server for Gateway Application

   c. Click **Submit**.
   d. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **Proxy Sets Table**).
   e. From the 'Proxy Set ID' drop-down list, select **0**.
   f. In the first 'Proxy Address' field, enter the IP address of the external proxy server.

g. In the second 'Proxy Address' field, enter the IP address and port of the device (in the format *x.x.x.x:port*). This is the same port as defined in the 'SAS Local UDP/TCP/TLS Port' field (see 'Configuring Common SAS Parameters' on page 481).

h. From the 'Proxy Redundancy Mode' drop-down list, select **Homing**.

Figure 21-15: Defining Proxy Servers for Gateway Application



i. Click **Submit**.

2. Disable the use of *user=phone* in the SIP URL:

a. Open the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **General Parameters**).

b. From the 'Use user=phone in SIP URL' drop-down list, select **No**. This instructs the Gateway application not to use *user=phone* in SIP URL and therefore, REGISTER and INVITE messages use SIP URI. (By default, REGISTER messages are sent with *sip uri* and INVITE messages with *tel uri*.)

c. Click **Submit**.

21.2.5 Advanced SAS Configuration

This section describes the configuration of advanced SAS features that can be optionally implemented in your SAS deployment:

- Manipulating incoming SAS Request-URI user part of REGISTER message (see 'Manipulating URI user part of Incoming REGISTER' on page 488)
- Manipulating destination number of incoming SAS INVITE messages (see 'Manipulating Destination Number of Incoming INVITE' on page 489)
- Defining SAS routing rules based on the IP2IP Routing table (see 'SAS Routing Based on IP2IP Routing Table' on page 492)
- Blocking unregistered SAS UA's (see 'Blocking Calls from Unregistered SAS Users' on page 496)
- Defining SAS emergency calls (see 'Configuring SAS Emergency Calls' on page 497)
- Adding SIP Record-Route header to INVITE messages (see 'Adding SIP Record-Route Header to SIP INVITE' on page 498)
- Replacing SIP Contact header (see 'Replacing Contact Header for SIP Messages' on page 499)

21.2.5.1 Manipulating URI user part of Incoming REGISTER

There are scenarios in which the UAs register to the proxy server with their full phone number (for example, "976653434"), but can receive two types of INVITE messages (calls):

- INVITEs whose destination is the UAs' full number (when the call arrives from outside the enterprise)
- INVITES whose destination is the last four digits of the UAs' phone number ("3434" in our example) when it is an internal call within the enterprise

Therefore, it is important that the device registers the UAs in the SAS registered database with their extension numbers (for example, "3434") in addition to their full numbers. To do this, you can define a manipulation rule to manipulate the SIP Request-URI user part of the AOR (in the To header) in incoming REGISTER requests. Once manipulated, it is saved in this manipulated format in the SAS registered users database in addition to the original (un-manipulated) AOR.

For example: Assume the following incoming REGISTER message is received and that you want to register in the SAS database the UA's full number as well as the last four digits from the right of the SIP URI user part:

```
REGISTER sip:10.33.38.2 SIP/2.0
Via: SIP/2.0/UDP 10.33.4.226:5050;branch=z9hG4bKac10827
Max-Forwards: 70
From: <sip: 976653434@10.33.4.226>;tag=1c30219
To: <sip: 976653434@10.33.4.226>
Call-ID: 16844@10.33.4.226
CSeq: 1 REGISTER
Contact: <sip: 976653434@10.10.10.10:5050>;expires=180
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,
UPDATE
Expires: 180
User-Agent: Audiocodes-Sip-Gateway-/v.
Content-Length: 0
```

After manipulation, SAS registers the user in its database as follows:

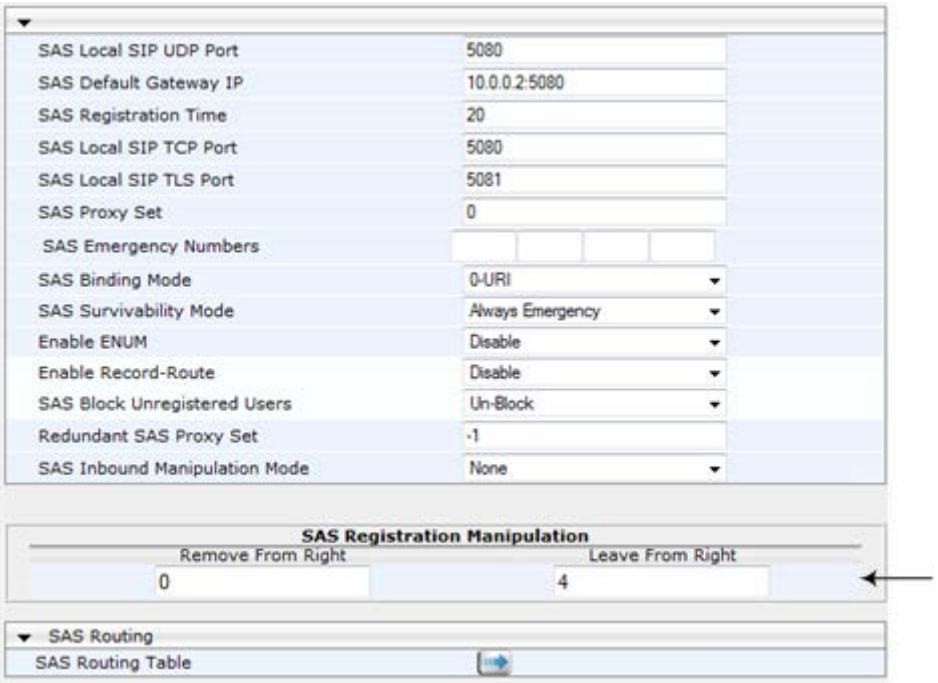
- **AOR:** 976653434@10.33.4.226
- **Associated AOR:** 3434@10.33.4.226 (after manipulation, in which only the four digits from the right of the URI user part are retained)
- **Contact:** 976653434@10.10.10.10

The procedure below describes how to configure the manipulation example scenario above (relevant *ini* parameter is SASRegistrationManipulation):

➢ **To manipulate incoming Request-URI user part of REGISTER message:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Stand Alone Survivability**).
2. In the SAS Registration Manipulation table, in the 'Leave From Right' field, enter the number of digits (e.g., "4") to leave from the right side of the user part. (The Leave From Right' field defines the number of digits to retain from the right side of the user part; all other digits in the user part are removed.)

Figure 21-16: Manipulating User Part in Incoming REGISTER



3. Click **Submit**.

21.2.5.2 Manipulating Destination Number of Incoming INVITE

You can define a manipulation rule to manipulate the destination number in the Request-URI of incoming INVITE messages when SAS is in emergency state. This is required, for example, if the call is destined to a registered user but the destination number in the received INVITE is not the number assigned to the registered user in the SAS registration database. To overcome this and successfully route the call, you can define manipulation rules to change the INVITE's destination number so that it matches that of the registered user in the database. This is done using the IP to IP Inbound Manipulation table.

For example, in SAS emergency state, assume an incoming INVITE has a destination number "7001234" which is destined to a user registered in the SAS database as "552155551234". In this scenario, the received destination number needs to be

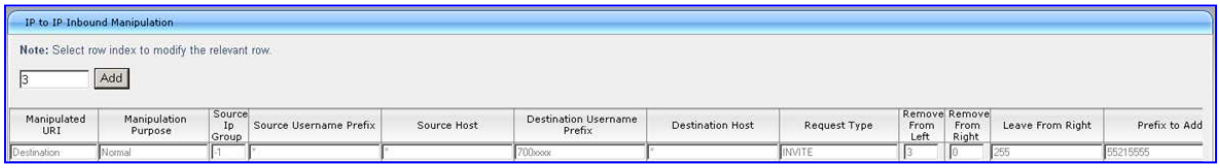
manipulated to the number "552155551234". The outgoing INVITE sent by the device then also contains this number in the Request-URI user part.

In normal state, the numbers are not manipulated. In this state, SAS searches the number 552155551234 in its database and if found, it sends the INVITE containing this number to the UA.

➢ **To manipulate destination number in SAS emergency state:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Stand Alone Survivability**).
2. From the 'SAS Inbound Manipulation Mode' (*SASInboundManipulationMode*) drop-down list, select **Emergency Only**.
3. Click **Submit**; the **SAS Inbound Manipulation Mode Table** button appears on the page.
4. Click this button to open the IP to IP Inbound Manipulation page.

Figure 21-17: Manipulating INVITE Destination Number



The figure above displays a manipulation rule for the example scenario described above whereby the destination number "7001234" is changed to "552155551234":

- 'Manipulated URI' field: **Destination**
- 'Destination Username Prefix' field: "700xxxx"
- 'Request Type' field: **INVITE**
- 'Remove From Left' field: "3"
- 'Prefix to Add' field: "55215555"

5. Add your SAS manipulation rule as desired. See the table below for descriptions of the parameters.
6. Click **Apply** to save your changes.

Notes: The following fields of the IP to IP Inbound Manipulation table are not applicable to SAS and must be left at their default values:

- **Additional Manipulation** - default is "0"
- **Manipulation Purpose** - default is Normal
- **Source IP Group** - default is "-1"

Table 21-1: SAS IP to IP Inbound Manipulation Parameters

Parameter	Description
Matching Characteristics	
Is Additional Manipulation **[IsAdditionalManipulation]**	Determines whether additional SIP URI user part manipulation is done for the table entry rule listed directly above it. ▪ **[0]** 0 = Regular manipulation rule (not done in addition to the rule above it). ▪ **[1]** 1 = If the above row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified

Parameter	Description
	by this rule. **Note:** Additional manipulation can only be performed on a different SIP URI (either source or destination) to the rule configured in the row above (defined by the parameter ManipulatedURI).
Manipulation Purpose **[ManipulationPurpose]**	Defines the purpose of the manipulation: ▪ **[0]** Normal = Inbound manipulations affect the routing input and source and/or destination number (default). ▪ **[1]** Routing input only = Inbound manipulations affect the routing input only, retaining the original source and destination number. ▪ **[2]** Shared Line = Used for the Shared-Line Appearance feature. This manipulation is for registration requests to change the destination number of the secondary extension numbers to the primary extension.
Source IP Group **[SrcIpGroup]**	Defines the IP Group from where the incoming INVITE is received. For any Source IP Group, enter the value -1.
Source Username Prefix **[SrcUsernamePrefix]**	Defines the prefix of the source SIP URI user name (usually in the From header). For any prefix, enter the asterisk '*' symbol (default). **Note:** The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 374.
Source Host **[SrcHost]**	Defines the source SIP URI host name - full name (usually in the From header). For any host name, enter the asterisk '*' symbol (default).
Destination Username Prefix **[DestUsernamePrefix]**	Defines the prefix of the destination SIP URI user name (usually in the Request-URI). For any prefix, enter the asterisk '*' symbol (default). **Note:** The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 374.
Destination Host **[DestHost]**	Defines the destination SIP URI host name - full name (usually in the Request URI). For any host name, enter the asterisk '*' symbol (default).
Request Type **[RequestType]**	Defines the SIP request type to which the manipulation rule is applied. ▪ **[0]** All = all SIP messages (default) ▪ **[1]** INVITE = all SIP messages except REGISTER and SUBSCRIBE ▪ **[2]** REGISTER = only SIP REGISTER messages ▪ **[3]** SUBSCRIBE = only SIP SUBSCRIBE messages ▪ **[4]** INVITE and REGISTER = all SIP messages except SUBSCRIBE ▪ **[5]** INVITE and SUBSCRIBE = all SIP messages except REGISTER
Manipulated URI **[ManipulatedURI]**	Determines whether the source or destination SIP URI user part is manipulated. ▪ **[0]** Source = Manipulation is done on the source SIP URI user part. (default) ▪ **[1]** Destination = Manipulation is done on the destination SIP URI user part.
Operation Manipulation Rule (when match occurs in characteristics)	

Parameter	Description
Remove From Left **[RemoveFromLeft]**	Defines the number of digits to remove from the left of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".
Remove From Right **[RemoveFromRight]**	Defines the number of digits to remove from the right of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "j".
Leave From Right **[LeaveFromRight]**	Defines the number of characters that you want retained from the right of the user name.
Prefix to Add **[Prefix2Add]**	Defines the number or string that you want added to the front of the user name. For example, if you enter 'user' and the user name is "john", the new user name is "userjohn".
Suffix to Add **[Suffix2Add]**	Defines the number or string that you want added to the end of the user name. For example, if you enter '01' and the user name is "john", the new user name is "john01".

21.2.5.3 SAS Routing Based on IP2IP Routing Table

SAS routing based on rules configured in the SAS Routing table is applicable for SAS in the following states:

- SAS in normal state, if the SASSurvivabilityMode parameter is set to 4
- SAS in emergency state, if the SASSurvivabilityMode parameter is not set to 4

The SAS routing rule destination can be an IP Group, IP address, Request-URI, or ENUM query.

The IP2IP Routing Table page allows you to configure up to 120 SAS routing rules (for Normal and Emergency modes). The device routes the SAS call (received SIP INVITE message) once a rule in this table is matched. If the characteristics of an incoming call do not match the first rule, the call characteristics is then compared to the settings of the second rule, and so on until a matching rule is located. If no rule is matched, the call is rejected.

When SAS receives a SIP INVITE request from a proxy server, the following routing logic is performed:

a. Sends the request according to rules configured in the IP2IP Routing table.

b. If no matching routing rule exists, the device sends the request according to its SAS registration database.

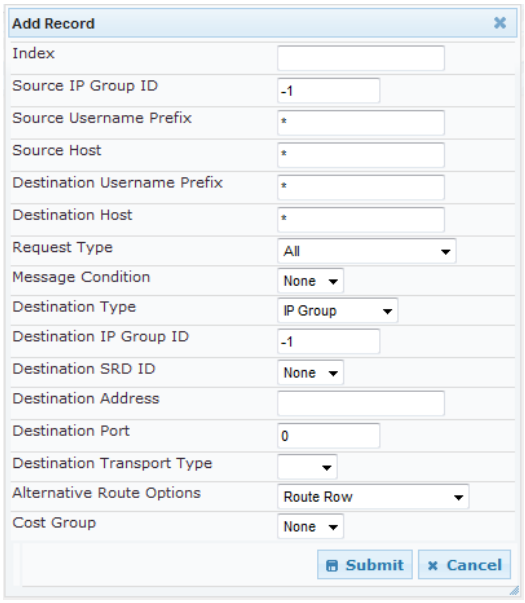
c. If no routing rule is located in the database, the device sends the request according to the Request-URI header.

Note: The IP2IP Routing table can also be configured using the *ini* file table parameter IP2IPRouting (see 'Configuration Parameters Reference' on page 752).

➢ **To configure the IP2IP Routing table for SAS:**

1. In the SAS Configuration page, click the **SAS Routing Table** button; the IP2IP Routing Table page appears.
2. Click **Add**.

Figure 21-18: IP2IP Routing Page



3. Configure the rule according to the table below.
4. Click **Submit** to apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Note: The following parameters are not applicable to SAS and should be ignored: Source IP Group ID, Destination IP Group ID, Destination SRD ID, and Alternative Route Options.

Table 21-2: SAS IP2IP Routing Table Parameters

Parameter	Description
Matching Characteristics	
Source IP Group ID **[IP2IPRouting_SrcIPGroupID]**	Selects the IP Group from where the IP-to-IP call originated. Typically, the IP Group of an incoming SIP dialog is determined (or classified) using the 'Classification' table (see "Configuring Classification Table" on page 452). If not used (i.e., any IP Group), simply leave the field empty. The default is -1.
Source Username Prefix **[IP2IPRouting_SrcUsernamePrefix]**	Defines the prefix of the user part of the incoming SIP dialog's source URI (usually the From URI). The default is "*". **Note:** The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 374.

Parameter	Description
Source Host **[IP2IPRouting_SrcHost]**	Defines the host part of the incoming SIP dialog's source URI (usually the From URI). If this rule is not required, leave the field empty. To denote any host name, use the asterisk (*) symbol. The default is "*".
Destination Username Prefix **[IP2IPRouting_DestUsernamePrefix]**	Defines the prefix of the incoming SIP dialog's destination URI (usually the Request URI) user part. If this rule is not required, leave the field empty. To denote any prefix, use the asterisk (*) symbol. The default is "*". **Note:** The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 374.
Destination Host **[IP2IPRouting_DestHost]**	Defines the host part of the incoming SIP dialog's destination URI (usually the Request-URI). If this rule is not required, leave the field empty. The asterisk (*) symbol can be used to denote any destination host. The default is "*".
Request Type **[IP2IPRouting_RequestType]**	Defines the SIP dialog request type of the incoming SIP dialog. ▪ **[0]** All (default) ▪ **[1]** INVITE ▪ **[2]** REGISTER ▪ **[3]** SUBSCRIBE ▪ **[4]** INVITE and REGISTER ▪ **[5]** INVITE and SUBSCRIBE ▪ **[6]** OPTIONS
Message Condition **[IP2IPRouting_MessageCondition]**	Selects a Message Condition rule. To configure Message Condition rules, see "Configuring Condition Rules" on page 456.
Operation Routing Rule (when match occurs in characteristics)	
Destination Type **[IP2IPRouting_DestType]**	Determines the destination type to which the outgoing SIP dialog is sent. ▪ **[0]** IP Group (default) = The SIP dialog is sent to the IP Group's Proxy Set (SERVER-type IP Group) or registered contact from the database (if USER-type IP Group). ▪ **[1]** Dest Address = The SIP dialog is sent to the address configured in the following fields: 'Destination SRD ID', 'Destination Address', 'Destination Port', and 'Destination Transport Type'. ▪ **[2]** Request URI = The SIP dialog is sent to the address indicated in the incoming Request-URI. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence. ▪ **[3]** ENUM = An ENUM query is sent to include the destination address. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence. ▪ **[4]** Hunt Group = Used for call center survivability. For more information, see "Call Survivability for Call Centers" on page 440.

Parameter	Description
Destination IP Group ID **[IP2IPRouting_DestIPGroupID]**	Defines the IP Group ID to where you want to route the call. The SIP dialog messages are sent to the IP address defined for the Proxy Set associated with this IP Group. If you select an IP Group, it is unnecessary to configure a destination IP address (in the 'Destination Address' field). However, if both parameters are configured, then the IP Group takes precedence. If the destination IP Group is of USER type, the device searches for a match between the Request-URI (of the received SIP dialog) to an AOR registration record in the device's database. The SIP dialog is then sent to the IP address of the registered contact. The default is -1. **Notes:** ▪ This parameter is only relevant if the parameter 'Destination Type' is set to 'IP Group'. However, regardless of the settings of the parameter 'Destination Type', the IP Group is still used - only for determining the IP Profile or outgoing SRD. If neither IP Group nor SRD are defined in this table, the destination SRD is determined according to the source SRD associated with the Source IP Group (configured in the IP Group table, see "Configuring IP Groups" on page 212). If this table does not define an IP Group but only an SRD, then the first IP Group associated with this SRD (in the IP Group table) is used. ▪ If the selected destination IP Group ID is type SERVER, the request is routed according to the IP Group addresses. ▪ If the selected destination IP Group ID is type USER, the request is routed according to the IP Group specific database (i.e., only to registered users of the selected database). ▪ If the selected destination IP Group ID is ANY USER (**[-2**]), the request is routed according to the general database (i.e., any matching registered user).
Destination SRD ID **[IP2IPRouting_DestSRDID]**	Defines the SRD ID. The default is None. **Note:** The destination IP Group must belong to the destination SRD, if both are configured in this table.
Destination Address **[IP2IPRouting_DestAddress]**	Defines the destination IP address (or domain name, e.g., domain.com) to where the call is sent. **Notes:** ▪ This parameter is applicable only if the parameter 'Destination Type' is set to 'Dest Address' **[1].** ▪ When using domain names, enter a DNS server IP address or alternatively, define these names in the 'Internal DNS Table' (see "Configuring the Internal SRV Table" on page 138).
Destination Port **[IP2IPRouting_DestPort]**	Defines the destination port to where the call is sent.
Destination Transport Type **[IP2IPRouting_DestTransportType]**	Defines the transport layer type for sending the call: ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When this parameter is set to -1, the transport type is determined by the parameter SIPTransportType.

Parameter	Description
Alternative Route Options **[IP2IPRouting_AltRouteOptions]**	Determines whether this routing rule is the main routing rule or an alternative routing rule (to the rule defined directly above it in the table). ▪ **[0]** Route Row (default) = Main routing rule - the device first attempts to route the call to this route if the incoming SIP dialog's input characteristics matches this rule. ▪ **[1]** Alt Route Ignore Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route regardless of the incoming SIP dialog's input characteristics. ▪ **[2]** Alt Route Consider Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route only if the incoming SIP dialog matches this routing rule's input characteristics. **Notes:** ▪ The alternative routing entry (**[1]** or **[2]**) **must be defined in the next consecutive table entry index** to the Route Row entry (i.e., directly below it). For example, if Index 4 is configured as a Route Row, Index 5 must be configured as the alternative route. ▪ For IP-to-IP alternative routing, configure SBC alternative routing reasons upon receipt of 4xx, 5xx, and 6xx SIP responses (see "Configuring Alternative Routing Reasons" on page 462). However, if no response, ICMP, or a SIP 408 response is received, the device attempts to use the alternative route even if no entries are configured in the 'SBC Alternative Routing Reasons' table. ▪ Multiple alternative route entries can be configured (e.g., Index 1 is the main route - Route Row - and indices 2 through 4 are configured as alternative routes).
Cost Group **[IP2IPRouting_CostGroup]**	Reserved for future use.

21.2.5.4 Blocking Calls from Unregistered SAS Users

To prevent malicious calls (for example, Service Theft), it is recommended to configure the feature for blocking SIP INVITE messages received from SAS users that are not registered in the SAS database. This applies to SAS in normal and emergency states.

➢ **To block calls from unregistered SAS users:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS Stand Alone Survivability**).
2. From the 'SAS Block Unregistered Users' drop-down list, select **Block**.
3. Click **Submit** to apply your changes.

21.2.5.5 Configuring SAS Emergency Calls

You can configure SAS to route emergency calls (such as 911 in North America) directly to the PSTN (through its FXO interface or E1/T1 trunk). Therefore, even during a communication failure with the external proxy, enterprise UAs can still make emergency calls.

You can define up to four emergency numbers, where each number can include up to four digits. When SAS receives a SIP INVITE (from a UA) that includes one of the user-defined emergency numbers in the SIP user part, it forwards the INVITE directly to the default gateway (see 'SAS Routing in Emergency State' on page 479). The default gateway is defined in the 'SAS Default Gateway IP' field, and this is the device itself. The device then sends the call directly to the PSTN.

This feature is applicable to SAS in normal and emergency states.

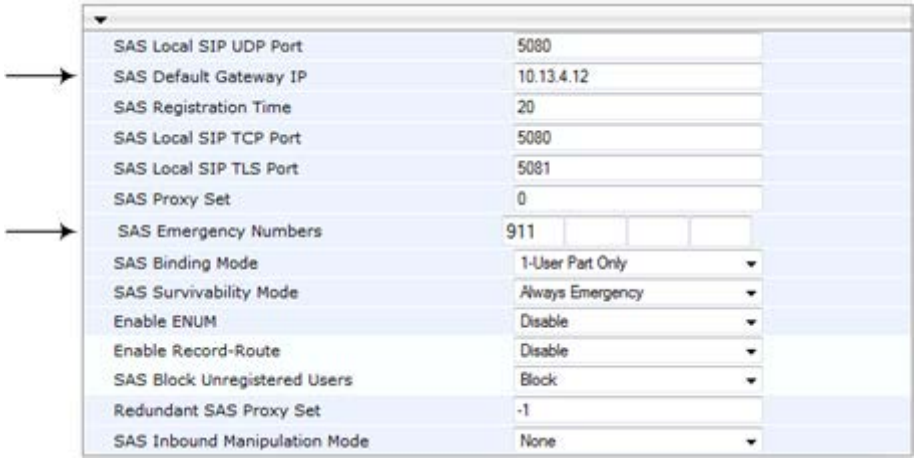
➢ **To configure SAS emergency numbers:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Stand Alone Survivability**).
2. In the 'SAS Default Gateway IP' field, define the IP address and port (in the format *x.x.x.x:port*) of the device (Gateway application).

Note: The port of the device is defined in the 'SIP UDP/TCP/TLS Local Port' field in the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**).

3. In the 'SAS Emergency Numbers' field, enter an emergency number in each field box.

Figure 21-19: Configuring SAS Emergency Numbers



4. Click **Submit** to apply your changes.

21.2.5.6 Adding SIP Record-Route Header to SIP INVITE

You can configure SAS to add the SIP Record-Route header to SIP requests (e.g. INVITE) received from enterprise UAs. SAS then sends the request with this header to the proxy. The Record-Route header includes the IP address of the SAS application. This ensures that future requests in the SIP dialog session from the proxy to the UAs are routed through the SAS application. If not configured, future request within the dialog from the proxy are sent directly to the UAs (and do not traverse SAS). When this feature is enabled, the SIP Record-Route header includes the URI "lr" parameter, indicating loose routing, as shown in the following example:

```
Record-Route: <sip:server10.biloxi.com;lr>
```

Notes:

- This feature is applicable only to SAS outbound mode.
- This feature can also be enabled using the SASEnableRecordRoute *ini* file parameter.

➢ **To enable the Record-Route header:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Stand Alone Survivability**).
2. From the 'Enable Record-Route' drop-down list, select **Enable**.
3. Click **Submit** to apply your changes.

21.2.5.7 Replacing Contact Header for SIP Messages

You can configure SAS to change the SIP Contact header so that it points to the SAS host. Therefore, this ensures that in the message, the top-most SIP Via header and the Contact header point to the same host.

Notes:

- This feature is applicable only to SAS outbound mode.
- The device may become overloaded if this feature is enabled, as all incoming SIP dialog requests traverse the SAS application.

Currently, this feature can only be configured using the SASEnableContactReplace *ini* file parameter.

- **[0]** (default): Disable - when relaying requests, SAS adds a new Via header (with the IP address of the SAS application) as the top-most Via header and retains the original Contact header. Thus, the top-most Via header and the Contact header point to different hosts.
- **[1]:** Enable - SAS changes the Contact header so that it points to the SAS host and therefore, the top-most Via header and the Contact header point to the same host.

21.3 Viewing Registered SAS Users

You can view all the users that are registered in the SAS registration database. This is displayed in the 'SAS/SBC Registered Users page, as described in 'Viewing SAS/SBC Registered Users' on page 722. The maximum number of users that can be registered in the database is 600.

Note: Despite the maximum number of SAS users, you can increase this capacity by implementing the SAS Cascading feature, as described in 'SAS Cascading' on page 500.

21.4 SAS Cascading

The SAS Cascading feature allows you to increase the number of SAS users above the maximum supported by the SAS gateway. This is achieved by deploying multiple SAS gateways in the network. For example, if the SAS gateway supports up to 600 users, but your enterprise has 1,500 users, you can deploy three SAS gateways to accommodate all users: the first SAS gateway can service 600 registered users, the second SAS gateway the next 600 registered users, and the third SAS gateway the rest (i.e., 300 registered users).

In SAS Cascading, the SAS gateway first attempts to locate the called user in its SAS registration database. Only if the user is not located, does the SAS gateway send it on to the next SAS gateway according to the SAS Cascading configuration.

There are two methods for configuring SAS Cascading. This depends on whether the users can be identified according to their phone extension numbers:

- **SAS Routing Table:** If users can be identified with unique phone extension numbers, then the SAS Routing table is used to configure SAS Cascading. This SAS Cascading method routes calls directly to the SAS Gateway (defined by IP address) to which the called SAS user is registered.

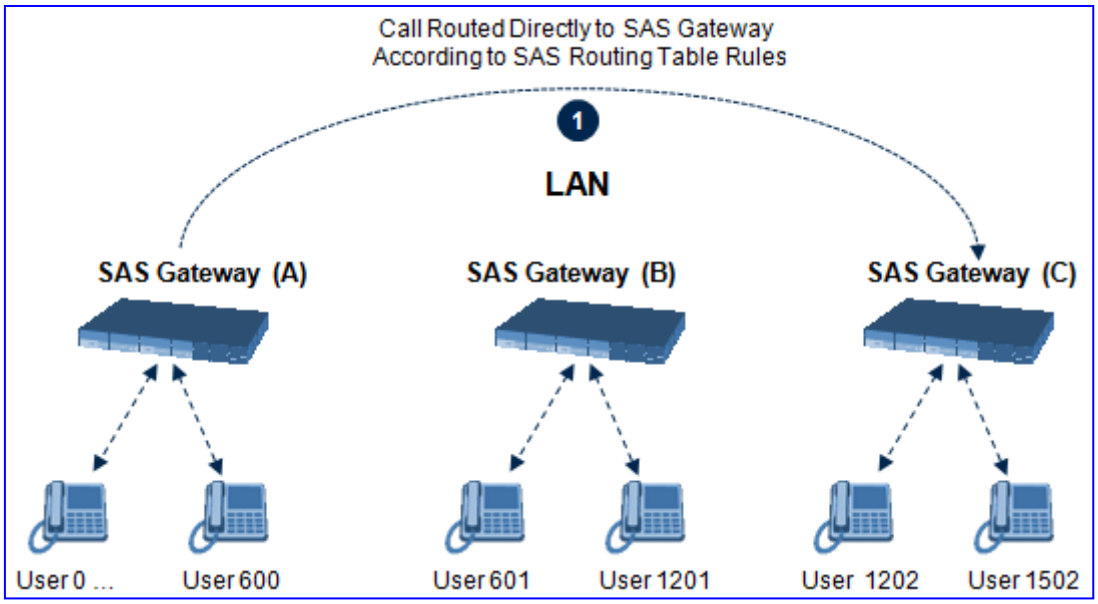
  The following is an example of a SAS Cascading deployment of users with unique phone extension numbers:

  - users registered to the first SAS gateway start with extension number “40”
  - users registered to the second SAS gateway start with extension number “20”
  - users registered to the third SAS gateway start with extension number “30”

  The SAS Routing table rules for SAS Cascading are created using the destination (called) extension number prefix (e.g., “30”) and the destination IP address of the SAS gateway to which the called user is registered. Such SAS routing rules must be configured at each SAS gateway to allow routing between the SAS users. The routing logic for SAS Cascading is similar to SAS routing in Emergency state (see the flowchart in 'SAS Routing in Emergency State' on page 479). For a description on the SAS Routing table, see 'SAS Routing Based on SAS Routing Table' on page 492.

  The figure below illustrates an example of a SAS Cascading call flow configured using the SAS Routing table. In this example, a call is routed from SAS Gateway (A) user to a user on SAS Gateway (B).

Figure 21-20: SAS Cascading Using SAS Routing Table - Example

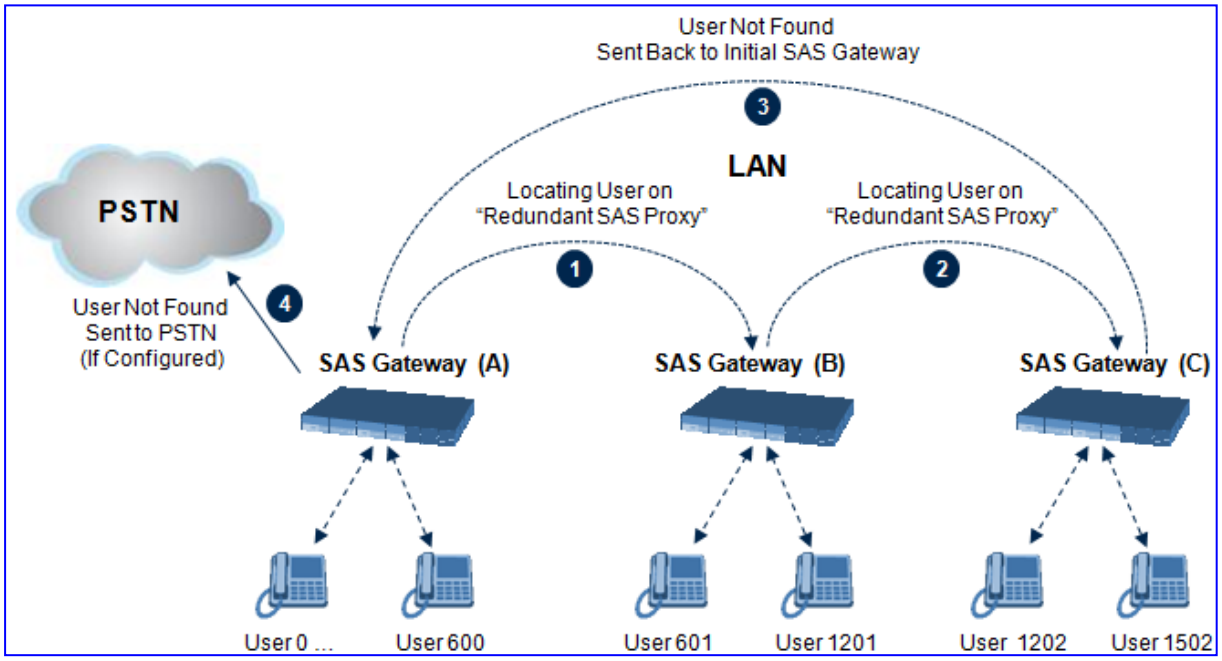


- **SAS Redundancy mode:** If users cannot be distinguished (i.e., associated to a specific SAS gateway), then the SAS Redundancy feature is used to configure SAS Cascading. This mode routes the call in a loop fashion, from one SAS gateway to the next, until the user is located. Each SAS gateway serves as the redundant SAS gateway ("redundant SAS proxy server") for the previous SAS gateway (in a one-way direction). For example, if a user calls a user that is not registered on the same SAS gateway, the call is routed to the second SAS gateway, and if not located, it is sent to the third SAS gateway. If the called user is not located on the third (or last) SAS gateway, it is then routed back to the initial SAS gateway, which then routes the call to the default gateway (i.e., to the PSTN).

  Each SAS gateway adds its IP address to the SIP via header in the INVITE message before sending it to the next ("redundant") SAS gateway. If the SAS gateway receives an INVITE and its IP address appears in the SIP via header, it sends it to the default gateway (and not to the next SAS gateway), as defined by the SASDefaultGatewayIP parameter. Therefore, this mode of operation prevents looping between SAS gateways when a user is not located on any of the SAS gateways.

  The figure below illustrates an example of a SAS Cascading call flow when configured using the SAS Redundancy feature. In this example, a call is initiated from a SAS Gateway (A) user to a user that is not located on any SAS gateway. The call is subsequently routed to the PSTN.

Figure 21-21: SAS Cascading Using SAS Redundancy Mode - Example



Reader's Notes

22 Configuring the IP Media Parameters

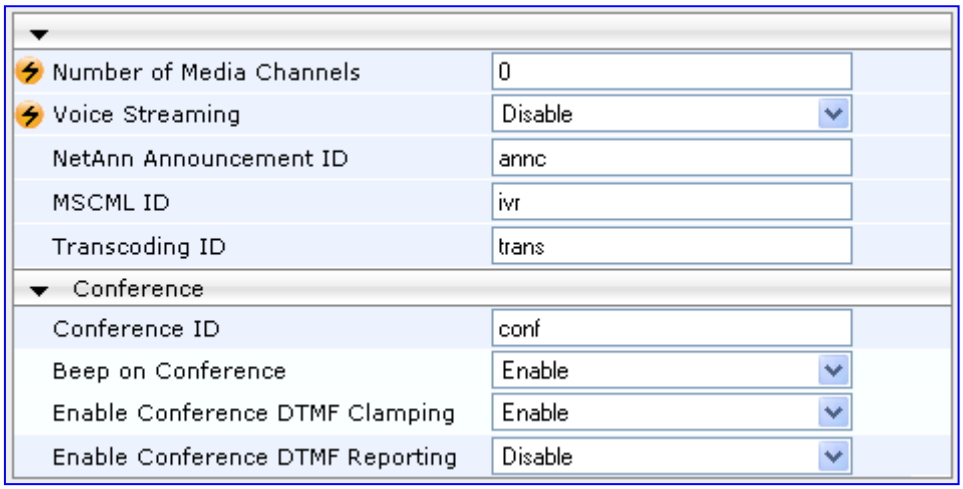
The IP Media Settings page allows you to configure IP media parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 752.

Note: This page appears only if your device is installed with the relevant Software Upgrade Key (see 'Loading Software Upgrade Key' on page 696).

➢ **To configure the IP media parameters:**

1. Open the IP Media Settings page (**Configuration** tab > **VoIP** menu > **IP Media** submenu > **IP Media Settings**).

Figure 22-1: IP Media Settings Page



2. Configure the parameters as required.
3. Click Submit to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' **on page 680.**

22.1 Transcoding using Third-Party Call Control

The device supports transcoding using a third-party call control Application server. This support is provided by the following:

- Using RFC 4117 (see 'Using RFC 4117' on page 504)
- Using RFC 4240 - NetAnn Conferencing (see Using RFC 4240 - NetAnn 2-Party Conferencing on page 504)

Note: Transcoding can also be implemented using the IP-to-IP (IP2IP) application and SBC application.

22.1.1 Using RFC 4117

The device supports RFC 4117 - Transcoding Services Invocation in the Session Initiation Protocol (SIP) Using Third Party Call Control (3pcc) - providing transcoding services (i.e., acting as a transcoding server). This is used in scenarios where two SIP User Agents (UA) would like to establish a session, but do not have a common coder or media type. When such incompatibilities are found, the UAs need to invoke transcoding services to successfully establish the session. Note that transcoding can also be performed using NetAnn, according to RFC 4240.

To enable the RFC 4117 feature, the parameter EnableRFC4117Transcoding must be set to 1 (and the device must be reset).

The 3pcc call flow is as follows: The device receives from one of the UAs, a single INVITE with an SDP containing two media lines. Each media represents the capabilities of each of the two UAs. The device needs to find a match for both of the media, and opens two channels with two different media ports to the different UAs. The device performs transcoding between the two voice calls.

In the example below, an Application Server sends a special INVITE that consists of two media lines to perform transcoding between G.711 and G.729:

```
m=audio 20000 RTP/AVP 0
c=IN IP4 A.example.com
m=audio 40000 RTP/AVP 18
c=IN IP4 B.example.com
```

22.1.2 Using RFC 4240 - NetAnn 2-Party Conferencing

Transcoding bridges (or translates) between two remote *network* locations, each of which uses a different coder and/or a different DTMF and fax transport types. The device supports IP-to-IP transcoding. It creates a transcoding call that is similar to a dial-in, two-party conference call. The SIP URI in the INVITE message is used as a transcoding service identifier. The transcoding identifier is configured using the 'Transcoding ID' parameter (TranscodingID) in the IP Media Settings page (see 'Configuring the IP Media Parameters' on page 503)..

It is assumed that the device is controlled by a third-party, Application server (or any SIP user agent) that instructs the device to start an IP transcoding call by sending two SIP INVITE messages with SIP URI that includes the transcoding identifier name. For example:

```
Invite sip:trans123@audiocodes.com SIP/2.0
```

The left part of the SIP URI includes the transcoding ID (the default string is ‘trans’) and is terminated by a unique number (123). The device immediately sends a 200 OK message in response to each INVITE.

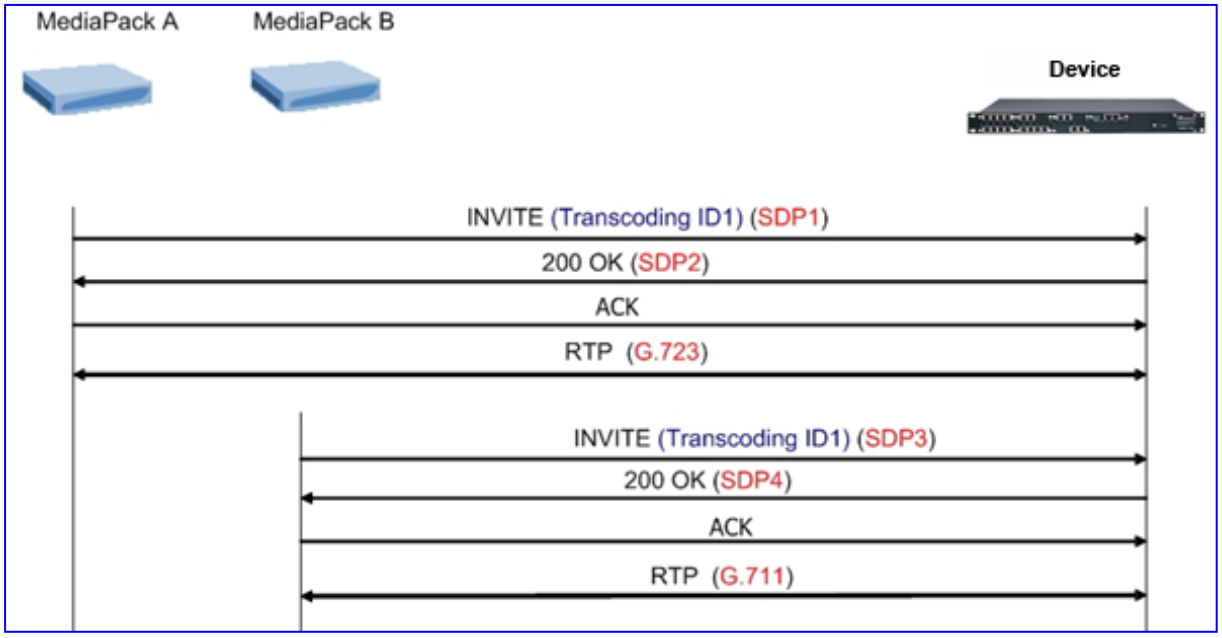
Each of the transcoding SIP call participants can use a different VoIP coder and a different DTMF transport type, negotiated with the device using common SIP negotiation.

Sending a BYE request to the device by any of the participants, terminates the SIP session and removes it from the Transcoding session. The second BYE from the second participant ends the transcoding session and releases its resources.

The device uses two media (DSP) channels for each call, thereby reducing the number of available transcoding sessions to half of the defined value for MediaChannels. To limit the number of resources for transcoding, use the 'Number of Media Channels' parameter (MediaChannels) in the IP Media Settings page (see 'Configuring the IP Media Parameters' on page 503). For example, if 'Number of Media Channels' is set to "40", only 20 transcoding sessions are available.

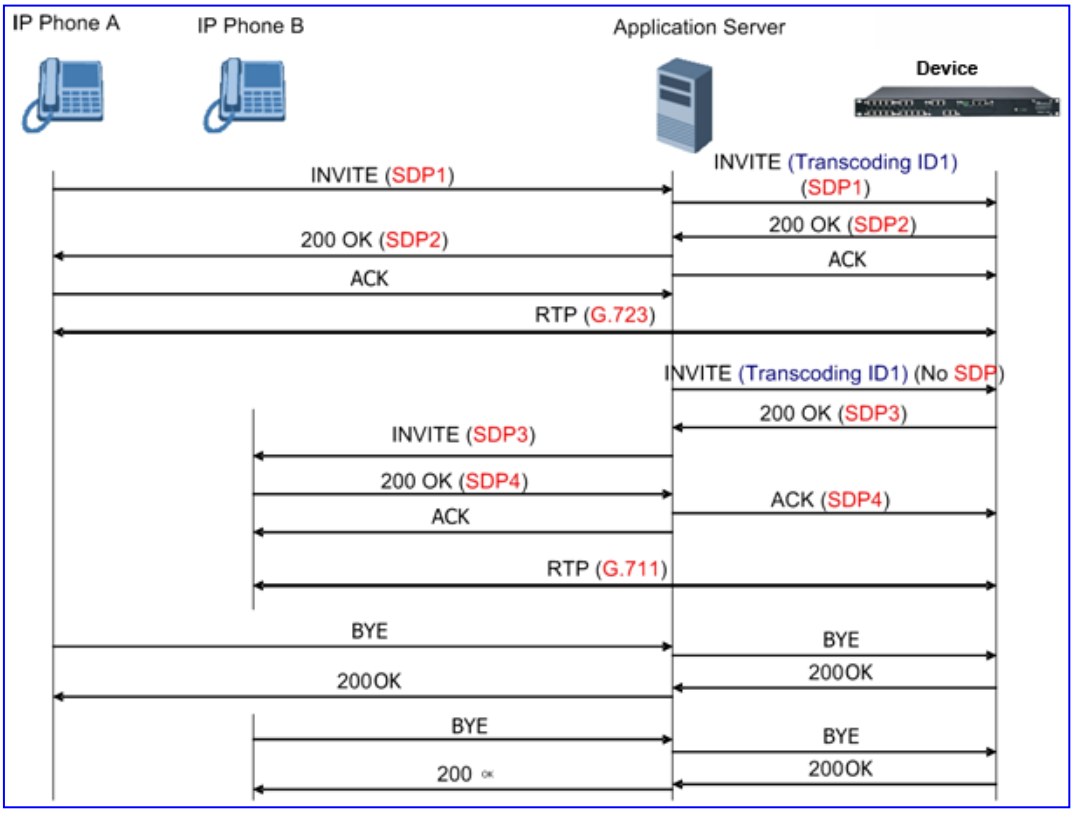
The figure below illustrates an example of a direct connection to a device:

Figure 22-2: Direct Connection (Example)



The figure below illustrates an example of implementing an Application server:

Figure 22-3: Application Server



22.2 Overview

This section provides information on the device's media server capabilities:

- Multi-party conferencing (see 'Conference Server' on page 506)
- Playing and recording Announcements (see 'Announcement Server' on page 521)
- IP-to-IP Transcoding (see 'IP-to-IP Transcoding' on page 504)
- Voice XML Interpreter (see Voice XML Interpreter on page 540)

The device conference, transcoding, announcement and media server applications can be used separately, each on a different platform, or all on the same device. The SIP URI name in the INVITE message is used to identify the resource (media server, conference or announcement) to which the SIP session is addressed.

The number of DSP channels that are allocated for IP conferences, transcoding and IP announcements is determined by the parameter MediaChannels. Other DSP channels can be used for PSTN media server.

The device's SIP implementation is based on the decomposition model described in the following IETF Internet-Drafts:

- "A Multi-party Application Framework for SIP" (draft-ietf-sipping-cc-framework-06)
- "Models for Multi Party Conferencing in SIP" (draft-ietf-sipping-conferencing-framework-05)
- "A Framework for Conferencing with the Session Initiation Protocol (SIP)" (RFC 4353)
- "Basic Network Media Services with SIP" (RFC 4240)
- "Media Server Control Markup Language (MSCML) and Protocol" (draft-vandyke-mscml-06)

Note: To use the device's advanced Announcement capabilities, it's essential that the *ini* file parameter AMSProfile be set to 1.

22.2.1 Conference Server

The device supports dial-in, multi-party conferencing. In conference applications, the device functions as a centralized conference bridge. In ad-hoc or prearranged conferences, users 'invite' the conference bridge. The conference bridge mixes the media and sends it to all participants.

The device supports the following interfaces for conferencing:

- Simple, according to NetAnn (see 'Simple Conferencing (NetAnn)' on page 507)
- Advanced, according to MSCML (see 'Advanced Conferencing (MSCML)' on page 509)

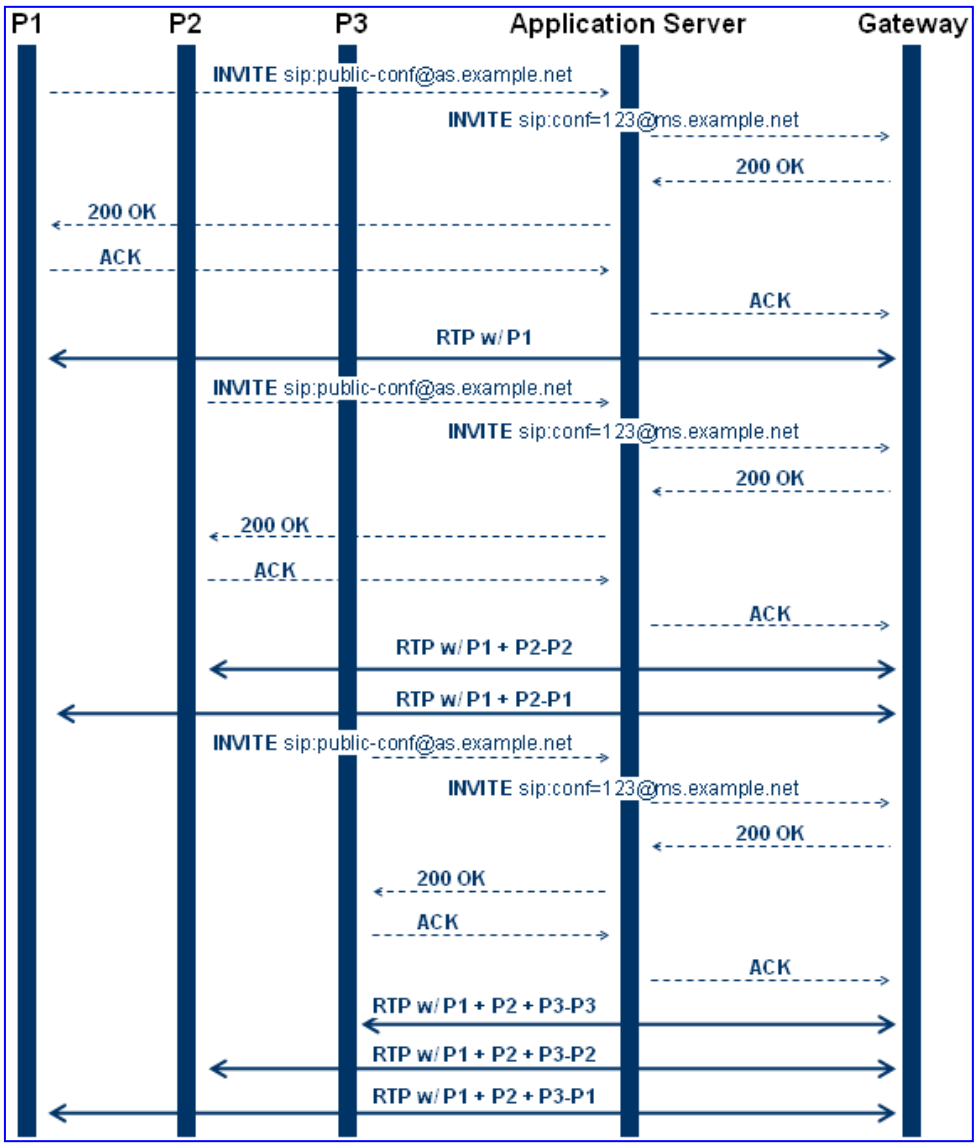
Note: The conference application is a special order option.

22.2.1.1 Simple Conferencing (NetAnn)

22.2.1.1.1 SIP Call Flow

A SIP call flow for simple conferencing is shown below:

Figure 22-4: Simple Conferencing SIP Call Flow



22.2.1.1.2 Creating a Conference

The device creates a conference call when the first user joins the conference. To create a conference, the Application Server sends a regular SIP INVITE message to the device. The User Part of that Request-URI includes both the Conference Service Identifier (indicating that the requested Media Service is a Conference) and a Unique Conference Identifier (identifying a specific instance of a conference).

```
INVITE sip: conf100@audiocodes.com SIP/2.0
```

By default, a request to create a conference reserves three resources on the device. It is possible to reserve a larger number of resources in advance by adding the number of required participants to the User Part of the Request-URI. For example, '6conf100' reserves six resources for the duration of the conference. If the device can allocate the requested number of resources, it responds with a 200 OK.

The Conference Service Identifier can be set using the 'Conference ID' parameter (ConferenceID) in the IP Media Settings page (see 'Configuring the IP Media Parameters' on page 503). By default, it is set to 'conf'.

22.2.1.1.3 Joining a Conference

To join an existing conference, the Application Server sends a SIP INVITE message with the same Request-URI as the one that created the conference. Each conference participant can use a different coder negotiated with the device using usual SIP negotiation.

If more than the initially requested number of participants try to join the conference (i.e., four resources were reserved and a fifth INVITE is received) and the device has an available resource, the request is granted.

If an INVITE to join an existing conference is received with a request to reserve a larger number of participants than initially requested, it is granted if the device has available resources. A request for a smaller number of participants is not granted as this may create a situation where existing legs would need to be disconnected.

The maximal number of participants in a single conference is 60. The maximal number of participants that actually participate in the mix at a given time is three (the loudest legs).

The Application Server can place a participant on Hold/Un-hold by sending the appropriate SIP Re-INVITE on that participant dialog.

22.2.1.1.4 Terminating a Conference

The device never disconnects an existing conference leg. If a BYE is received on an existing leg, it is disconnected, but the resource is still saved if the same leg (or a different one) wants to re-join the conference. This logic occurs only for the initial number of reserved legs.

For example:

1. INVITE reserves three legs.
2. A, B, and C join the conference.
3. A disconnects.
4. A joins (guaranteed).
5. D joins.
6. A disconnects.
7. A joins (not guaranteed).

Sending a BYE request to the device terminates the participant's SIP session and removes it from the conference. The final BYE from the last participant ends the conference and releases all conference resources.

22.2.1.1.5 PSTN Participants

Adding PSTN participants is done by performing a loopback from the IP side (the device's IP address is configured in the Outbound IP Routing Table). If the destination phone number in the incoming call from the PSTN is equal to the Conference Service Identifier and Unique Conference Identifier, the participant joins the conference.

A PSTN participant uses two DSP channels (caused by the IP loopback).

22.2.1.2 Advanced Conferencing (MSCML)

22.2.1.2.1 Creating a Conference

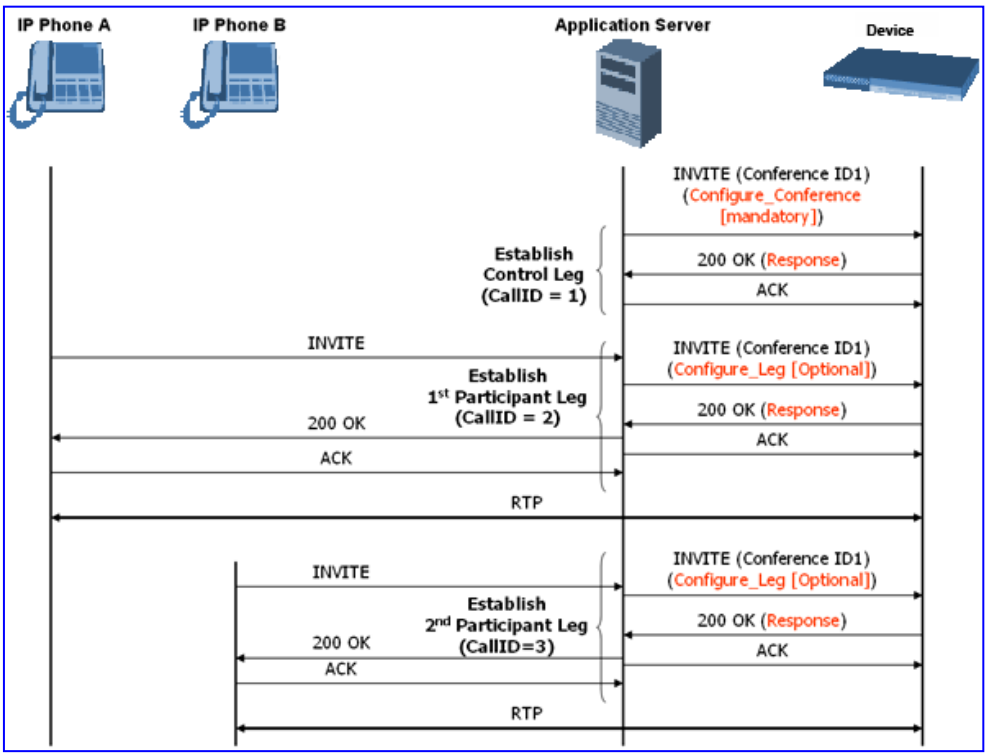
The device creates a conference call when the first INVITE is received from the Application Server (same as NetAnn). The Unique Conference Identifier is used to join participants to the same conference. This first INVITE must include a <configure_conference> MSCML request body. If this body is not included, a simple conference is established. This first leg is the Control Leg, which is different from a regular Participant Leg. The Control Leg is used to perform operations for the whole conference.

The MSCML response to the first INVITE is sent in the 200 OK SIP response. If no error occurs, the response is:
<response request="configure_conference" code="200" text="OK"/>.

The <configure_conference> can include the following attributes:

- **Id:** identification number of the MSCML request. This is used to correlate between MSCML requests and responses.
- **Reservedtalkers:** defines the maximum number of talker legs. As the device does not support “listener only” legs, this actually sets the maximum number of participants in the conference. The device reserves this number of participants for the entire duration of the conference. If a participant leg decides to leave the conference by issuing a BYE, the resource is not freed, thereby allowing that same leg (or a new one) to join at any stage.
- **Reserveconfmedia:** determines if Media Services such as Play or Record can be applied to the conference. If set to Yes, the device reserves the necessary amount of resources to play an announcement to the whole conference or record the whole conference. The Application Server can change the value of reserveconfmedia during an existing conference. By default, reserveconfmedia is set to Yes.

Figure 22-5: Advanced Conferencing SIP Call Flow



22.2.1.2.2 Joining a Conference

To join an existing conference, the Application Server sends a SIP INVITE message with the same Request-URI as the one that created the conference. The INVITE message may include a <configure_leg> MSCML request body. If not included, defaults are used for that leg attributes.

The <configure_leg> can include the following attributes:

- **Id:** identification number of the MSCML request. This is used to correlate between MSCML requests and responses.
- **Type:** Talker / Listener. If set to Listener, the incoming RTP from that leg does not participate in the conference mix. The default is Talker.
- **Mixmode:**
  - Full: RTP from this leg participates in the mix (default).
  - Mute: RTP from this leg is not participating in the mix.
  - Private: RTP from this leg can only hear participants within a conference team (<teammate>) to which it belongs (see below).

The <configure_team> element enables clients to create personalized mixes for scenarios where the standard mixmode settings do not provide sufficient control. The <configure_team> element is a child of <configure_leg>. The <configure_team> element, containing one or more <teammate> elements, specifies those participants that should be present in this participant's personalized mix. The <configure_team> element supports several commands: set, add, remove, and query.

The participants are identified in the <teammate> elements by their IDs that are assigned in their <configure_leg> element. The team configuration is implicitly symmetric, i.e. if participant A defines participant B as its team member, implicitly participant B defines participant A as its team member.

A "coaching" example:

Table 22-1: MSCML Conferencing with Personalized Mixes

Participant	ID	Team Members	Mixmode	Hears
Supervisor	"supervisor"	Agent	Private	Customer and Agent
Agent	"agent"	Supervisor	Full	Customer and Supervisor
Customer	"customer"		Full	Agent

This scenario is established as follows:

1. Conference is created on the control leg with <configure_conference>.
2. Coach leg joins and issues:

```
<configure_leg id="supervisor" mixmode="private"/>
```

3. Agent leg joins and issues:

```
<configure_leg id="agent">
         <configure_team action="set">
           <teammate id="supervisor"/>
         </configure_team>
       </configure_leg>
```

4. Customer joins and issues:

```
<configure_leg id="customer"/>
```

22.2.1.2.3 Modifying a Conference

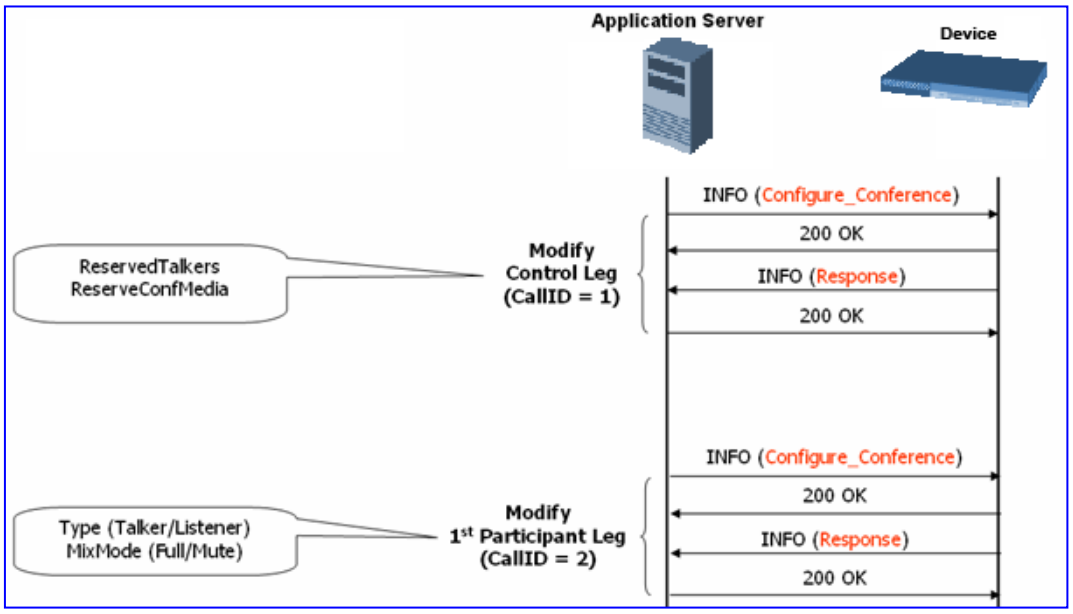
To modify an existing conference, INFO messages are used. Each INFO message carries an MSCML request. The MSCML response is included in an INFO message back from the device to the Application Server. It is possible to modify an entire conference (by issuing requests on the Control Leg) or only a certain participant (by issuing requests on that specific leg).

To modify the entire conference, a <configure_conference> MSCML request body is sent in an INFO message on the Control Leg SIP dialog. Using this request, the Application Server can modify the following attributes:

- **Reservedtalkers:** If the Application Server sets a number that is lower than the initial number requested in the INVITE, then the request is not granted. If the number is higher than the initial number, the device sends a success response in the response INFO.
- **Reserveconfmedia:** If the necessary resources for applying Media Services on the entire conference were reserved in advance, then by setting reserveconfmedia to Yes, it is reserved. If set to No, the device can free the resource.

To modify a certain Participant Leg, a <configure_leg> MSCML request body is sent in an INFO message on that leg SIP dialog. Using this request, the Application Server can modify any of the attributes defined for the <configure_leg> request.

Figure 22-6: Modifying a Conference - SIP Call Flow

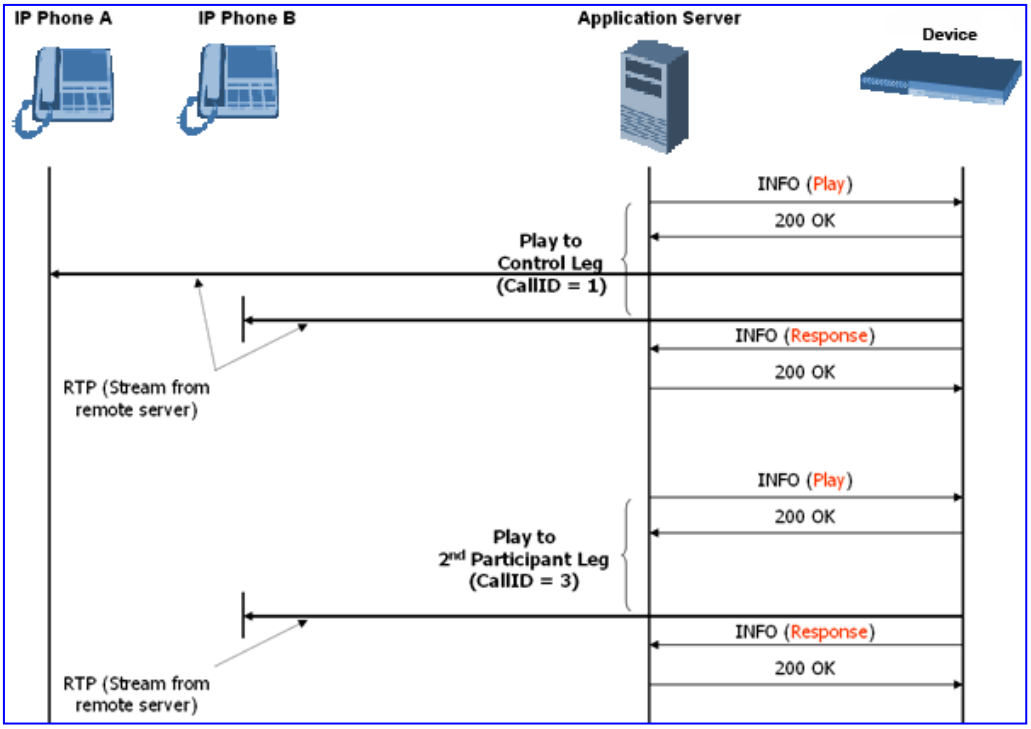


22.2.1.2.4 Applying Media Services on a Conference

The Application Server can issue a Media Service request (<play>, <playcollect>, or <playrecord>) on either the Control Leg or a specific Participant Leg. For a Participant Leg, all three requests are applicable. For the Control Leg, the <playcollect> is not applicable as there is no way to collect digits from the whole conference.

When issuing a Media Service on the Control Leg, it affects all Participant Legs in the conference, e.g., play an announcement. When issuing a Media Service on a Participant Leg, it affects the specific leg only.

Figure 22-7: Applying Media Services on a Conference - SIP Call Flow



22.2.1.2.5 Active Speaker Notification

After an advanced conference is established, the Application Server can subscribe to the device to receive notifications of the current set of active speakers in a conference at any given moment. This feature is referred to as *Active Speaker Notification* (ASN) and is designed according to the MSCML standard. Notifications provide information on the number of active participants and their details.

The notifications are sent unsolicited at specific intervals requested by the application and only when a change in the number of active conference speakers occurs. If a change in the speakers list occurs, the server issues an INVITE to the specific SIP UA, and then transfers the call to the UA.

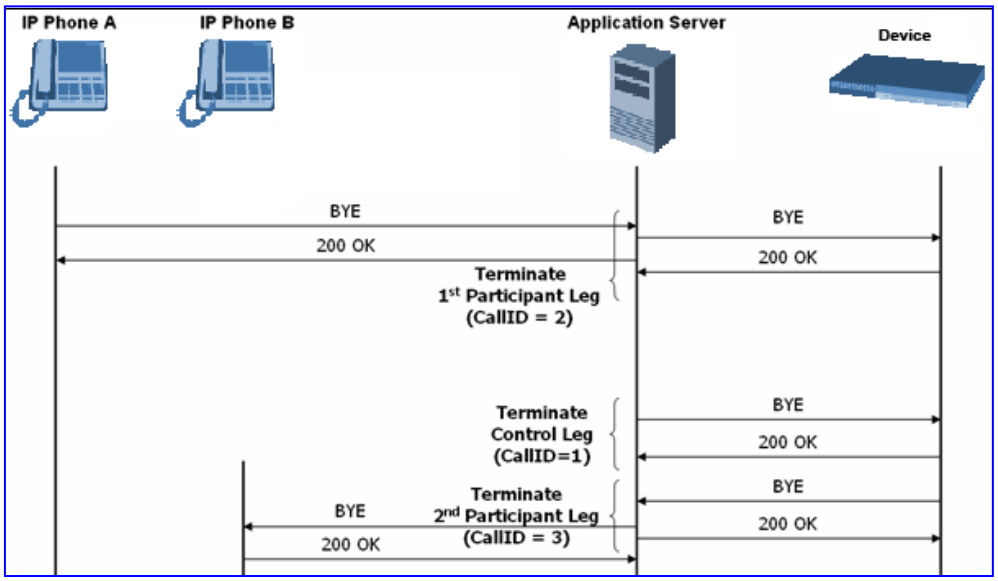
Event notifications are sent in SIP INFO messages, as shown in the example below of XML Response Generated for ASN:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
<notification>
<conference uniqueID="3331" numtalkers="1">
<activetalkers>
<talker callID="9814266171512000193619@10.8.27.118"/>
</activetalkers>
</conference>
</notification>
</MediaServerControl>
```

22.2.1.2.6 Terminating a Conference

To remove a leg from a conference, the Application Server issues a SIP BYE request on the selected dialog representing the conference leg. The Application Server can terminate all legs in a conference by issuing a SIP BYE request on the Control Leg. If one or more participants are still in the conference when the device receives a SIP BYE request on the Control Leg, the device issues SIP BYE requests on all of the remaining conference legs to ensure a clean up of the legs.

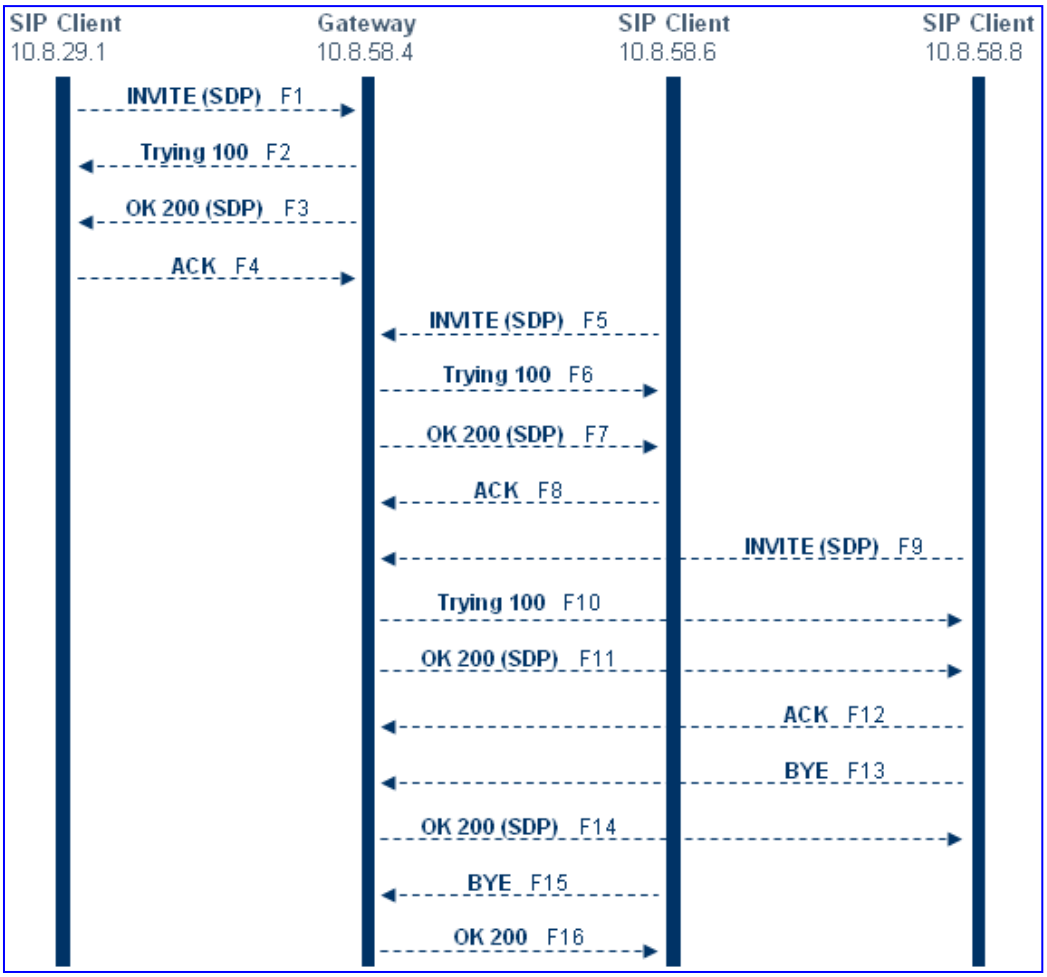
Figure 22-8: Terminating a Conference - SIP Call Flow



22.2.1.3 Conference Call Flow Example

The call flow, shown in the following figure, describes SIP messages exchanged between the device (10.8.58.4) and three conference participants (10.8.29.1, 10.8.58.6 and 10.8.58.8).

Figure 22-9: Conference Call Flow Example



1. **SIP MESSAGE 1:** 10.8.29.1:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
Max-Forwards: 70
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Contact: <sip:100@10.8.29.1>
Supported: em,100rel,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.5.40A.010.006
Content-Type: application/sdp
Content-Length: 216
v=0
o=AudiocodesGW 663410 588654 IN IP4 10.8.29.1
```

```
s=Phone-Call
c=IN IP4 10.8.29.1
t=0 0
m=audio 6000 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

2. **SIP MESSAGE 2:** 10.8.58.4:5060() -> 10.8.29.1:5060()

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40A.010.006
Content-Length: 0
```

3. **SIP MESSAGE 3:** 10.8.58.4:5060 -> 10.8.29.1:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Contact: <sip:10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40A.010.006
Content-Type: application/sdp
Content-Length: 216
v=0
o=AudiocodesGW 820775 130089 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7160 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

4. SIP MESSAGE 4: 10.8.29.1:5060 -> 10.8.58.4:5060

```
ACK sip:10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacbUrWtRo
Max-Forwards: 70
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 ACK
Contact: <sip:100@10.8.29.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.5.40A.010.006
Content-Length: 0
```

5. SIP MESSAGE 5: 10.8.58.6:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 INVITE
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.4.60A.005.009
Content-Type: application/sdp
Content-Length: 313
v=0
o=AudiocodesGW 702680 202680 IN IP4 10.8.58.6
s=Phone-Call
c=IN IP4 10.8.58.6
t=0 0
m=audio 6000 RTP/AVP 4 8 0 110 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:110 AMR/8000/1
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

6. SIP MESSAGE 6: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
```

```
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40A.010.006
Content-Length: 0
```

7. SIP MESSAGE 7: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 INVITE Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40A.010.006
Content-Type: application/sdp
Content-Length: 236
v=0 o=AudiocodesGW 886442 597756 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7150 RTP/AVP 4 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

8. SIP MESSAGE 8: 10.8.58.6:5060 -> 10.8.58.4:5060

```
ACK sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacRRRZPXN
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 ACK
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.4.60A.005.009
Content-Length: 0
```

9. SIP MESSAGE 9: 10.8.58.8:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
```

```
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.4.60A.005.009
Content-Type: application/sdp Content-Length: 236
v=0
o=AudiocodesGW 558246 666026 IN IP4 10.8.58.8
s=Phone-Call
c=IN IP4 10.8.58.8
t=0 0 m=audio 6000 RTP/AVP 4 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

10. SIP MESSAGE 10: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40A.010.006
Content-Length: 0
```

11. SIP MESSAGE 11: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40A.010.006
Content-Type: application/sdp
Content-Length: 236
v=0
o=AudiocodesGW 385533 708665 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7140 RTP/AVP 4 96
a=rtpmap:4 g723/8000
```

```
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

12. SIP MESSAGE 12: 10.8.58.8:5060 -> 10.8.58.4:5060

```
ACK sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKacisqqyow
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 ACK
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.5.40A.010.006
Content-Length: 0
```

13. SIP MESSAGE 13: 10.8.58.8:5060 -> 10.8.58.4:5060

```
BYE sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKackSIyGww
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 2 BYE
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.5.40A.010.006
Content-Length: 0
```

14. SIP MESSAGE 14: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKackSIyGww
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 2 BYE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40A.010.006
Content-Length: 0
```

15. SIP MESSAGE 15: 10.8.58.6:5060 -> 10.8.58.4:5060

```
BYE sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacQypxnvl
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 2 BYE
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.5.40A.010.006
Content-Length: 0
```

16. SIP MESSAGE 16: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacQypxnvl
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 2 BYE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40A.010.006
Content-Length: 0
```

22.2.2 Announcement Server

The device supports playing and recording of announcements (local Voice Prompts or HTTP streaming) and playing of Call Progress Tones over the IP network. Three different methods are available for playing and recording announcements:

- NetAnn for playing a single announcement (see 'NetAnn Interface' on page 521)
- MSCML for playing single or multiple announcements and collecting digits (see 'MSCML Interface' on page 522)

22.2.2.1 NetAnn Interface

The device supports playing announcements using NetAnn format (according to RFC 4240).

22.2.2.1.1 Playing a Local Voice Prompt

To play a single local Voice Prompt, the Application Server (or any SIP user agent) sends a regular SIP INVITE message with SIP URI that includes the NetAnn Announcement Identifier name. For example:

```
INVITE sip:annc@audiocodes.com; play=file://12 SIP/2.0
```

The left part of the SIP URI includes the string 'annc'. In the example above, the device starts playing announcement number 12 from the internal Voice Prompts file (file:// and http://localhost formats are supported). The NetAnn Announcement Identifier string is configured using the *ini* file (parameter NetAnnAnncID) or Web interface (see 'Configuring the IPmedia Parameters' on page 503). Sending a BYE request terminates the SIP session and stops the playing of the announcement. If the played Voice Prompt reaches its end, the device initiates a BYE message to notify the Application Server that the session has ended.

22.2.2.1.2 Playing using HTTP/NFS Streaming

To play a single announcement via HTTP or NFS streaming, the Application Server (or any SIP user agent) sends a regular SIP INVITE message with SIP URI that includes the NetAnn Announcement Identifier name. For example:

```
INVITE sip:annc@ac.com;
play=http://server.net/gem/Hello.wav   SIP/2.0
```

The left part of the SIP URI includes the string 'annc' terminated by the IP address of the HTTP server, and the name and path of the file to be played. In the example above, the device starts playing the 'Hello.wav' file that resides in the folder 'server.net/gem'. The NetAnn Announcement Identifier string is configured using the *ini* file (parameter NetAnnAnncID) or Web interface (see 'Configuring the IPmedia Parameters' on page 503). Sending a BYE request terminates the SIP session and stops the playing of the announcement. If the played announcement reaches its end, the device initiates a BYE message to notify the Application Server that the session is ended.

Notes:

- A 200 OK message is sent only after the HTTP connection is successfully established and the requested file is found. If the file isn't found, a 404 Not Found response is sent.
- To use NFS, the requested file system should be first mounted by using the NFS Servers table, see 'Configuring the NFS Settings' on page 141.

22.2.2.1.3 Supported Attributes

When playing announcements, the following attributes are available:

- **Repeat:** defines the number of times the announcement is repeated. The default value is 1. The valid range is 1 to 1000, or -1 (i.e., repeats the message forever).
- **Delay:** defines the delay (in msec) between announcement repetitions. The default value is 0. The valid range is 1 to 3,600,000.
- **Duration:** defines the total duration (in msec) the announcement(s) are played. The default value is 0 (i.e., no limitation). The valid range is 1 to 3,600,000.

For example:

```
INVITE sip:annc@ac.com;
play=http://server.net/gem/Hello.wav; repeat=5;delay=10000 SIP/2.0
```

22.2.2.2 MSCML Interface

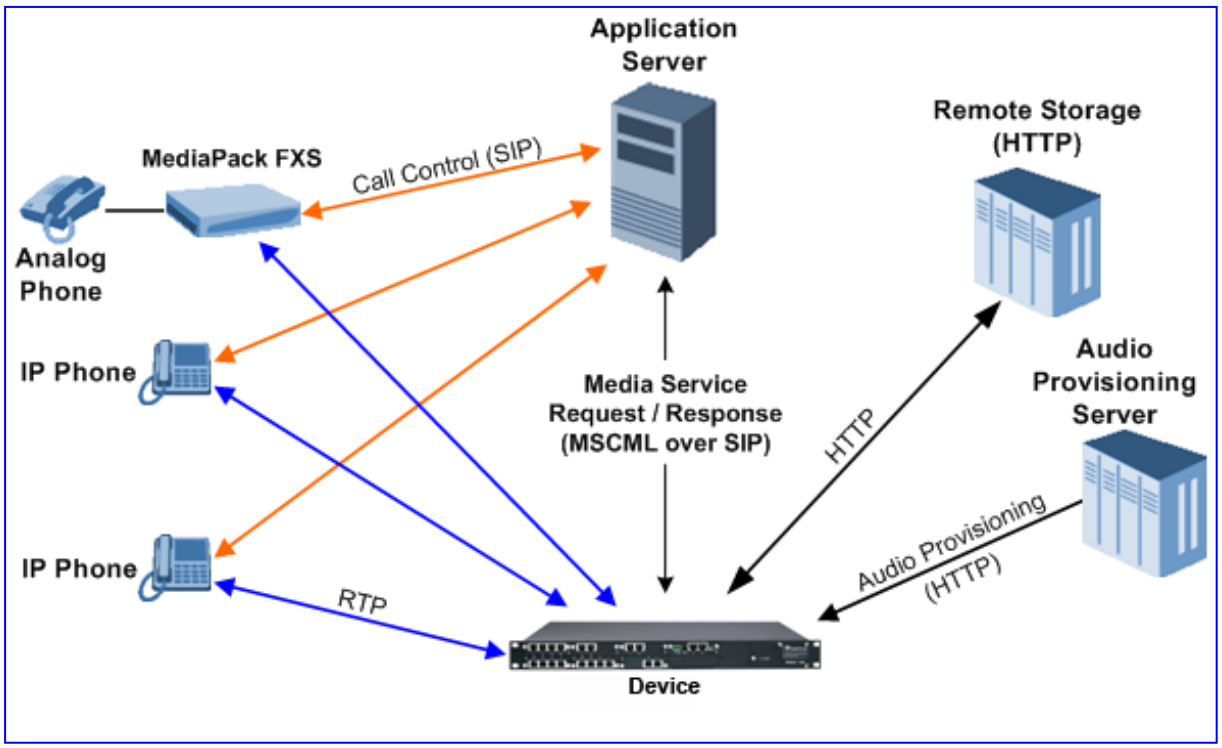
Media Server Control Markup Language (MSCML), according to IETF RFC 5022 is a protocol used in conjunction with SIP to provide advanced announcements handling. MSCML is implemented by adding an XML body to existing SIP INFO messages. Only a single message body (containing a single request or response) is allowed per message.

The device supports all the Interactive Voice Response (IVR) requirements for playing announcements, collecting digits, and recording (Play, PlayCollect, and PlayRecord).

Note: MSCML is only supported on devices operating with 128-MByte RAM size.

The following figure illustrates standard MSCML application architecture:

Figure 22-10: MSCML Architecture



The architecture comprises the following components:

- **device:** Operating independently, the device controls and allocates its processing resources to match each application's requirements. Its primary role is to handle requests from the Application server for playing announcements and collecting digits.
- **Application Server:** An application platform that controls the call signaling. It interfaces with the device using MSCML. It instructs the media server to play announcements, collect digits and record voice streams.
- **Audio Provisioning Server (APS):** The APS is used for offline generation of .dat files of audio packages including audio files, audio sequences, and different languages for variable announcement playing. These can be later loaded to the device.
- **Remote Storage:** An HTTP server that contains less-frequently used voice prompts for playback and to which voice stream recording is performed.
- **IP Phones / MediaPack:** Client applications.

22.2.2.2.1 Operation

The APS server can be used to generate two files - the audio package as a VP.dat file, and an XML file (segments.xml) that contains indices to the announcements stored on the VP.dat file for playing announcements. These two files can be loaded to the device using the Web interface.

An alternative method uses the AutoUpdate mechanism as described in the *Product Reference Manual*. Both the vp.dat and segments.xml files that were previously created using the APS should be located on an external storage server (HTTP, FTP). At startup, the device fetches the files from the remote storage. By using the AutoUpdate mechanism, the device periodically checks if new files are posted to the remote server and fetches these files.

The Application server communicates with the device using MSCML Requests (sent by the Application server), as shown in the example below:

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <request>
      ... request body ...
    </request>
  </MediaServerControl>
```

The device uses MSCML Responses (i.e., sent by the device) to reply to the Application server, as shown in the example below:

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <response>
      ... response body ...
    </response>
  </MediaServerControl>
```

To start an MSCML IVR call, the Application server (or any SIP user agent) sends a regular SIP INVITE message with a SIP URI that includes the MSCML Identifier name. For example:

```
INVITE sip:ivr@audiocodes.com SIP/2.0
```

The left part of the SIP URI includes the MSCML Identifier string 'ivr', which can be configured using the *ini* file (parameter MSCMLID) or Web interface (see 'Configuring the IPmedia Parameters' on page 503).

After a call is established, SIP INFO messages are used to carry MSCML requests and responses. An INFO message that carries an MSCML body is identified by its content-type header that is set to 'application/mediaservercontrol+xml'.

Note that IVR requests are not queued. Therefore, if a request is received while another is in progress, the device stops the first operation and executes the new request. The device generates a response message for the first request and returns any data collected up to that point. If an application is required to stop a request in progress, it issues a <Stop> request. This request also causes the device to generate a response message.

The device supports basic IVR functions of playing announcements, collecting DTMF digits, and voice stream recording. These services are implemented using the following Request and Response messages:

- <Play> for playing announcements
- <PlayCollect> for playing announcements and collecting digits
- <PlayRecord> for playing announcements and recording voice
- <Stop> for stopping the playing of an announcement

The device sends a Response to each Request that is issued by the Application server.

The <Play>, <PlayCollect>, and <PlayRecord> messages are composed of two sections: Attributes and a Prompt block (the request can contain several different Prompt blocks). The Attributes section includes several request-specific parameters. The Prompt block section itself is also composed of two sections: prompt-specific parameters and audio segments (audio / variable). The (optional) prompt-specific parameters include:

- *locale*: defines the language in which the prompt block is played (supported for local files only). For more information on language usage, refer to the *Audio Provisioning Server User's Manual* (LTRT-971xx).
- *baseurl*: defines a URL address that functions as a prefix to all audio segment URLs in the Prompt block.

The Prompt block contains references to one or more audio segments. The following audio segment types are available:

- **Physical Audio Segments:** These are physical audio files that are located either locally (on-blade) or on an external HTTP server. If the file is located on-blade, the reference to it is by using one of the following syntaxes:

  'file:///x', 'file:////x' or 'http://localhost/x'

  Where *x* stands for the file identifier (the ID or alias given by the APS server for local files; or the file's URL in for HTTP streaming).
- **Variables:** These are audio segments whose value is determined at run time. They are defined in the request as a <type, subtype, value> tuple. The device transforms the variable data to voice. To support variable playing, APS server support is mandatory. Available variable types are (subtypes in parenthesis): date (DMY - day month year; MDY - month day year - default), duration, month, money (USD), number (crd, ord), digit (gen, ndn) silence, string, time (t12, t24) and weekday.
  It is also possible to store audio files that are required to play supported types of phrases (e.g., dates and times) on an off-board system. This is beneficial in scenarios where the device's on-board storage limit has been reached, and thus, additional languages and audio can be stored off-board.
- **Sequences:** These are audio segments that consist of physical audio files and variables. These sequences can be defined using the APS server.

22.2.2.2.2 Operating with Audio Bundles

Voice prompts can be played from the device's local memory where they are stored as Audio bundles. An audio bundle is composed of a .dat file and an .xml file containing the information to properly parse the .dat file. Audio bundles are created through the APS and are then stored on a server supporting NFS or HTTP.

Uploading a Bundle to the Device

The audio bundle can be uploaded through HTTP. For more information, see the relevant Automatic Update chapter in the *Product Reference Manual*.

To upload a voice bundle to the device, the following *ini* file parameters should be set:

```
APSEnabled = 1
AMSProfile = 1
VpFileUrl = 'url-dat-file/dat-file'
APSSegmentsFileUrl = 'url-xml-file/xml-file''
```

Where *url-dat-file* and *url-xml-file* relate to the location of the relevant .dat and .xml files, and *dat-file* and *xml-file* relate to the file names, as shown in the example below:

```
APSEnabled = 1
AMSProfile = 1
VpFileUrl = 'http://10.50.2.1/dat_files/vp.dat'
APSSegmentsFileUrl = 'http://10.4.2.5/segments/segments.xm'
```

You can upload a bundle to the device using one of the following methods:

- Loading an ini file as described above, and then resetting the device (hard reset). Optionally, you can configure parameters through Web interface or using SNMP, and then burn parameters to flash and reset the device through Web or SNMP (soft reset).
- Adding the following *ini* file parameter to periodically upload the .dat and .xml files:

```
AutoUpdateFrequency = 100         // updating is performed every
100 minutes.
```

For more information, refer to Automatic Updates in the *Product Reference Manual*.

- Using SNMP to trigger an immediate upload of the files by setting acSysActionSetAutoUpdate to true.

Note: When uploading files through HTTP, if the names of the file that are already loaded to the device and the file intended to be uploaded are the same, time stamps of the old file and the new file should differ.

You can be notified on the outcome of an operation in two ways:

- Syslog messages – Informative Syslog messages are supplied when the operation has succeeded or failed. On operation failure, resort to first analyzing these messages.
- SNMP traps - Similar messages are also supplied via SNMP traps. For more information refer to the *Product Reference Manual*.

22.2.2.2.3 Playing Announcements

A <Play> request is used to play an announcement to the caller. Each <Play> request contains a single Prompt block and the following request-specific parameters:

- *id*: an optional random number used to synchronize request and response.
- *promptur*l: a specific audio file URL that is used in addition to the references in the Prompt block. This audio file is the first to be played.

An example of an MSCML <Play> Request that includes local and streaming audio files as well as variables is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
    <request>
          <play id=”123”>
                <prompt>
                      <audio url="http://localhost/1"/>
                      <variable type="digits" value="284"/>
                      <variable type="silence" value="1"/>
                      <audio url="http://10.3.0.2/aa.wav"/>
                <audiourl="nfs://10.3.0.3/prov_data/bb.wav"/>
                </prompt>
          </play>
    </request>
</MediaServerControl>
```

22.2.2.2.4 Playing Announcements and Collecting Digits

The <PlayCollect> request is used to play an announcement to the caller and to then collect entered DTMF digits. The play part of the <PlayCollect> request is identical to the <Play> request. The collect part includes an expected digit map. The collected digits are continuously compared to the digit map. Once a match is found, the collected digits are sent in a <PlayCollect> response. The digit map should be in MGCP format (the type value must be set to ‘mgcpdigitmap’).

For example:

```
<regex type="mgcpdigitmap" value="([0-1]xxx)">
</regex>
```

Each <PlayCollect> request contains the following request-specific parameters in addition to the Prompt block (all parameters are optional):

- *id*: an optional random number used to synchronize request and response.
- *prompturl:* a specific audio file URL that is used in addition to the references in the prompt block. This audio file is the first to be played.
- *barge*: if set to 'NO', DTMF digits received during announcement playback are ignored. If set to 'YES', DTMF digits received during announcement playback stop the playback and start the digit collection phase.
- *firstdigittimer:* defines the amount of time (in milliseconds) the user does not enter any digits, after which a response is sent indicating timeout.
- *interdigittimer:* defines the amount of time (in milliseconds) the user does not enter any digits after the first DTMF digit is received, after which a response is sent indicating timeout.
- *extradigittimer:* used to enable the following:
  - Detection of command keys (ReturnKey and EscapeKey).
  - Not report the shortest match. MGCP Digitmap searches for the shortest possible match. This means that if a digitmap of (123 | 1234) is defined, once the user enters 123, a match is found and a response is sent. If ExtraDigitTimer is defined, the match can also be 1234 because the device waits for the next digits. To use ExtraDigitTimer, it must be defined in the request and you must add a "T" to the Digitmap (for example, 'xxT'). The ExtraDigitTimer is only used when a match is found. Before a match is found, the timer used is the InterDigitTimer. Therefore, if the ExtraDigitTimer expires, a "match" response reason is reported -- never a "timeout".
- *maxdigits:* defines the maximum number of collected DTMF digits after which the device sends a response.
- *cleardigits:* defines whether or not the device clears the digit buffer between subsequent requests.
- *returnkey*: defines a specific digit (including '*' and '#') which (when detected during a collection) stops the collection and initiates a response (that includes all digits collected up to that point) to be sent.
- *escapekey:* defines a specific digit (including '*' and '#') which (when detected during a collection) stops the collection and initiates a response (with no collected digits) to be sent.

An example is shown below of an MSCML <PlayCollect> Request that includes a sequence with variables and an MGCP digit map:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <playcollect id="6379" barge="NO" returnkey="#">
                      <prompt>
                            <audio url="http://localhost/1">
                            <variable type="silence" value="1"/>
                            <variable type="date" subtype="mdy"
value="20041210"/>
                      </prompt>
                      <regex type="mgcpdigitmap" value="([0-
```

```
1]xxx)">
                        </regex>
                    </playcollect>
            </request>
</MediaServerControl>
```

An example is shown below of an MSCML <PlayCollect> Response:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <response request="playcollect" id="6478" code="200"
text="OK" digits="4563">
          </response>
</MediaServerControl>
```

22.2.2.2.5 Playing Announcements and Recording Voice

The <PlayRecord> request is used to play an announcement to the caller and to then record the voice stream associated with that caller. The play part of the <PlayRecord> request is identical to the <Play> request. The record part includes a URL to which the voice stream is recorded. This URL refers to anHTTP server.

Each <PlayRecord> request contains the following request-specific parameters in addition to the Prompt block (all parameters except 'recurl' are optional):

- *id*: an optional random number used to synchronize request and response.
- *prompturl:* a specific audio file URL that is used in addition to the references in the prompt block. This audio file is the first to be played.
- *barge*: if set to 'NO', DTMF digits received during announcement playback are ignored. If set to 'YES', DTMF digits received during announcement playback stop the playback and start the recording phase.
- *cleardigits:* defines whether or not the device clears the digit buffer between subsequent requests.
- *escapekey:* defines a specific digit (including '*' and '#') which (when detected during any phase) stops the request and initiates a response.
- *recurl*: the URL on the external storage server to which the RTP stream is sent for recording. This is a mandatory parameter.
- *mode:* defines if the recording 'overwrites' the existing file or 'appends' to it.
- *initsilence:* defines how long to wait for initial speech input before terminating the recording. This parameter may take an integer value in milliseconds.
- *endsilence:* defines how long the device waits after speech has ended to stop the recording. This parameter may take an integer value in milliseconds.
- *duration*: the total time in milliseconds for the entire recording. Once this time expires, recording stops and a response is generated.
- *recstopmask:* defines a digit pattern to which the device compares digits detected during the recording phase. If a match is found, recording stops and a response is sent.

An example is shown below of an MSCML <PlayRecord> Request:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <playrecord id="75899" barge="NO"
Recurl=nfs://10.11.12.13/save/recordings/11.wav>
                      <prompt>
                            <audio url="nfs://100.101.102.103/45">
                            <variable type="date" subtype="mdy"
value="20041210"/>
                      </prompt>
                </playrecord>
          </request>
</MediaServerControl>
```

An example is shown below of an MSCML <PlayRecord> Response:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <response request=”playrecord” id=”75899” code=”200”
text=”OK” reclength=”15005”>
          </response>
</MediaServerControl>
```

22.2.2.2.6 Stopping the Playing of an Announcement

The Application server issues a <stop> request when it requires that the device stops a request in progress and not initiate another operation. The only (optional) request-specific parameter is id.

The device refers to a SIP re-INVITE message with hold media (c=0.0.0.0) as an implicit <Stop> request. The device immediately terminates the request in progress and sends a response.

An example is shown below of an MSCML <Stop> Request:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <stop id="123">
                </stop>
          </request>
</MediaServerControl>
```

22.2.2.2.7 Relevant Parameters

The following parameters (described in 'IP Media Parameters' on page 999) are used to configure the MSCML:

- AmsProfile = 1 (mandatory)
- AASPackagesProfile = 3 (mandatory)
- VoiceStreamUploadMethod = 1 (mandatory)
- EnableVoiceStreaming = 1 (mandatory)
- MSCMLID (default=”ivr”)

- AmsPrimaryLanguage (default=”eng”)
- AmsSecondaryLanguage (default=”heb”)
- When using AutoUpdate:
  - VPFileURL
  - APSSegmentsFileUrl
  - AutoUpdateFrequency / AutoUpdatePredefinedTime

22.2.2.2.8 Signal Events Notifications

The device supports Signal Events Notifications as defined in RFC 4722/5022 - MSCML. MSCML defines event notifications that are scoped to a specific SIP dialog or call leg. These events allow a client to be notified of various call progress signals. Subscriptions for call leg events are performed by sending an MSCML <configure_leg> request on the desired SIP dialog. Call leg events may be used with the MSCML conferencing or IVR services. Using the Signal Notifications, the device can report the following events:

Table 22-2: Reportable Events

Type	Subtype
AMD	▪ voice ▪ automata ▪ silence ▪ unknown
CPT	▪ SIT-NC ▪ SIT-IC ▪ SIT-VC ▪ SIT-RO ▪ busy ▪ reorder
FAX	▪ CED ▪ CNG ▪ modem

Below is an example:

```
<?xml version="1.0"?>
 <MediaServerControl version="1.0">
  <request>
   <configure_leg>
    <subscribe>
     <events>
      <signal type="amd" report="yes"/>
     </events>
    </subscribe>
   </configure_leg>
  </request>
 </MediaServerControl>


<?xml version="1.0"?>
 <MediaServerControl version="1.0">
  <notification>
   <signal type="amd" subtype="voice"/>
  </notification>
</MediaServerControl>
```

22.2.2.3 Voice Streaming

The voice streaming layer provides you with the ability to play and record different types of files while using an NFS or HTTP server.

22.2.2.3.1 Voice Streaming Features

The following subsections summarizes the Voice Streaming features supported on HTTP and NFS servers, unless stated otherwise.

Basic Streaming Play

You may play a .wav, .au or .raw file from a remote server using G.711 coders.

Supported File Formats

The voice streaming layer provides support for .wav, .au, and .raw file formats. The maximum supported header size of the file is 150 bytes.

In .wav format, only mono mode and supported/known coders are supported. The maximum number of the non-data, non-fmt chunks can be up to 5.

Play from Offset

You may play a .wav, .au or .raw file from a given offset within the file. Offset can be both positive and negative relative to the files length. A negative offset relates to an offset from the end of the file.

Remote File Systems

You may configure up to 16 remote file systems to operate with the system through NFS mounting.

Using Proprietary Scripts

You may use cgi or servlet scripts released with the version for recording to a remote HTTP server using the POST or PUT method.

Dynamic HTTP URLs

Voice streaming supports dynamic HTTP URLs. The following terminology is used:

- **Static audio content:** Traditional audio file URLs containing references to specific files (.wav, .au or .raw). For example: http://10.50.0.2/qa/GOSSIP_ENG.wav
- **Dynamic audio content:** URLs referencing to cgi scripts or servlets. For example: http://10.50.0.2/cgi/getaudio.cgi?filename=DEFAULT_GREETING.raw&offset=0

In the case of dynamic URLs, the device performs the GET command with the supplied URL and as a result, the servlet or cgi script on the Web server is invoked. The Web server responds by sending a GET response containing the audio.

The URL format can be as follows (RFC 1738 URLs, section 3.3):

```
http://<host>:<port>/<path>?<searchpart>
```

where,

- **:<port>** is optional.
- **<path>** is a path to a server-side script.
- **<searchpart>** is of the form: key=value**[&key**=value]*

Note: At least one key=value pair is required.

Another example of a dynamic URL is shown below:

```
http://MyServer:8080/prompts/servlet?action=play&language=eng&file
=welcome.raw&format=1
```

(See also RFC 2396 URI: Generic Syntax.)

The servlet or cgi script can respond by sending a complete audio file or a portion of an audio file. The device skips any .wav or .au file header that it encounters at the beginning of the response. The device does not attempt to use any information in the header. For example, the device does not use the coder from the header. Note however, that the coder may be supplied through Web or *ini* file parameters.

Play LBR Audio File

You may play a file using low bit rate coders for .wav and .raw files.

Note: This feature is relevant for both NFS and HTTP.

Basic Record

You may record a .wav, .au or .raw files to a remote server using G.711 coders.

Note: This feature is relevant for both NFS and HTTP.

Remove DTMF Digits at End of Recording

You may configure a recording to remove the DTMF received at the end, indicating an end of a recording.

Note: This feature is relevant for both NFS and HTTP.

Record Files Using LBR

You may record a file using low bit rate coders for .wav and .raw files.

Notes: This feature is relevant for both NFS and HTTP.

Modifying Streaming Levels Timers

Several parameters enable the user to control streaming level timers for NFS and HTTP and also the number of data retransmission when using NFS as the application layer protocol:

- **General command timeout – ServerRespondTimeout:** Defines the maximal time a command or respond may be delayed. This relates both to HTTP commands (GET, PUT, POST, HEAD etc.) and to NFS commands (create, lookup, read, write etc.).
- **Recording packet overruns timer – StreamingRecordingOverRunTimeout:** An

overrun condition is one in which the device sends data to the server but does not receive responds from the server acknowledging that it received the data. Overruns relate to recording data to a remote server and result with “holes” in the recording. The streaming level aborts sessions containing consecutive overruns as derived from this timer. You may set the timer to longer periods than the default value, thereby enabling the device to be more "tolerant" to overrun conditions.

- **Playing packet underruns – StreamingPlayingUnderRunTimeout:** An underrun condition is one in which the device does not supply the DSPs with sufficient data, thus "starving" the DSPs. Underruns relate to playing data from a server to the device where, due to environmental conditions (usually network problems), the data is not passed quickly enough. This condition results with broken data passed to the user. The streaming level aborts sessions containing consecutive underruns as derived from this timer. You may set the timer to longer periods than the default value, thereby enabling the device to be more "tolerant" to underrun conditions.
- **NFS command retransmission – NFSClientMaxRetransmission:** Defines the number of times an NFS command is retransmitted when the server side does not respond. By default, the value is set to 0 and not used - instead, the number of retransmissions is derived from the server response timeout parameter and the current Recovery Time Objective (RTO) of the system.

22.2.2.3.2 Using File Coders with Different Channel Coders

The tables in the following subsections describe the support for different combinations of file coders (used for recording or playing a file) and channel coders (used when opening a voice channel).

The following abbreviations are used in the subsequent tables:

- **LBR:** Low Bit Rate Coder
- **PCMU:** G.711 µ-law coder
- **PCMA:** G.711 A-law coder
- **WB:** Linear PCM 16KHZ Wide Band Coder

Note: When recording with an LBR type coder, it is assumed that the same coder is used both as the file coder and the channel coder. Combinations of different LBR coders are currently not supported.

Playing a File

The table below lists the device's support of channel coders and file coders for playing a file.

Table 22-3: Coder Combinations - Playing a File

File Coder	File Type											
	.wav				.au				.raw			
	Channel Coder				Channel Coder				Channel Coder			
	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB
PCMA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
PCMU	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
LBR	No	No	Yes	No	No	No	No	No	No	No	Yes	Yes

Recording a File

The table below lists the device's support of channel coders and file coders for recording a file.

Table 22-4: Coder Combinations - Recording a File

File Coder	File Type											
	WAV				AU				RAW			
	Channel Coder				Channel Coder				Channel Coder			
	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB
PCMA	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No
PCMU	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No
LBR	No	No	Yes	No	No	No	No	No	No	No	No	No

22.2.2.3.3 Maximum Concurrent Playing and Recording

For details on maximum concurrent playing and recording, refer to the *Release Notes*.

22.2.2.3.4 LBR Coders Support

The following table describes the different low bit rate (LBR) coders and their support for .wav, .au, and .raw files.

Note: Coder support depends on the specific DSP template version installed on the device.

Table 22-5: LBR Coders and File Extension Support

Coder	.wav file	.raw file	.au file
G.726 (Rate 16)	Yes	Yes	No
G.726 (Rate 24)	Yes	Yes	No
G.726 (Rate 32)	Yes	Yes	No
G.726 (Rate 40)	Yes	Yes	No
G.723.1 (Rate 5.3)	Yes	Yes	No
G.723.1 (Rate 6.3)	Yes	Yes	No
G.729	Yes	Yes	No
GSM FR	Yes	Yes	No
MS GSM	Yes	Yes	No
GSM EFR	Yes	Yes	No
AMR (Rate 4.75)	No	Yes	No
AMR (Rate 5.15)	No	Yes	No

Coder	.wav file	.raw file	.au file
AMR (Rate 5.9)	No	Yes	No
AMR (Rate 6.7)	No	Yes	No
AMR (Rate 7.4)	No	Yes	No
AMR (Rate 7.95)	No	Yes	No
AMR (Rate 10.2)	No	Yes	No
AMR (Rate 12.2)	No	Yes	No
QCELP (Rate 8)	No	Yes	No
QCELP (Rate 13)	No	Yes	No

22.2.2.3.5 HTTP Recording Configuration

The HTTP record method (PUT or POST) is configured using the following offline *ini* parameter:

```
// 0=post (default), 1=put
VoiceStreamUploadMethod = 1
```

The default value is shown below:

```
VoiceStreamUploadPostUri =
"/audioupload/servlet/AcAudioUploadServlet"
```

Note: The PUT method disregards this string.

22.2.2.3.6 Supported HTTP Servers

The following is a list of HTTP servers that are known to be compatible with AudioCodes voice streaming under Linux™:

- **Apache:** cgi scripts are used for recording and supporting dynamic URLs.
- **Jetty:** servlets scripts are used for recording and supporting dynamic URLs.
- **Apache tomcat:** using servlets.

Tuning the Apache Server

It is recommended to perform the following modifications in the http.conf file located in the apache conf/ directory:

- Define PUT script location: Assuming the put.cgi file is included in this package, add the following line for defining the PUT script (script must be placed in the cgi-bin/ directory):

```
Script PUT /cgi-bin/put.cgi
```

- Create the directory /the-apache-dir/perl (for example, /var/www/perl) and copy the

CGI script to this directory. In the script, change the first line from c:/perl/bin/perl to your perl executable file (this step is required only if mod_perl is not included in your Apache installation).

- Keep-alive parameters: the following parameters must be set for correct operation with multiple POST requests:

```
KeepAlive On
MaxKeepAliveRequests 0 (unlimited amount)
```

- Using mode perl, fix the mod_perl to the following:

```
<IfModule mod_perl.c>
<Location /cgi-bin>
 SetHandler perl-script
   PerlResponseHandler ModPerl::Registry
   Options +ExecCGI
   PerlOptions +ParseHeaders
   Order allow,deny
   Allow from all
</Location>
</IfModule>
```

- Apache MPM worker: it is recommended to use the Multi-Processing Module implementing a hybrid multi-threaded multi-process Web server. The following configuration is recommended:

```
<IfModule worker.c>
ThreadLimit       64
StartServers         2
ServerLimit       20000
MaxClients        16384
MinSpareThreads    100
MaxSpareThreads     250
ThreadsPerChild     64
MaxRequestsPerChild  16384
</IfModule>
```

22.2.2.3.7 Common Troubleshooting

Always inspect the Syslog for any problem you may encounter; in many cases, the cause appears there.

Table 22-6: Troubleshooting

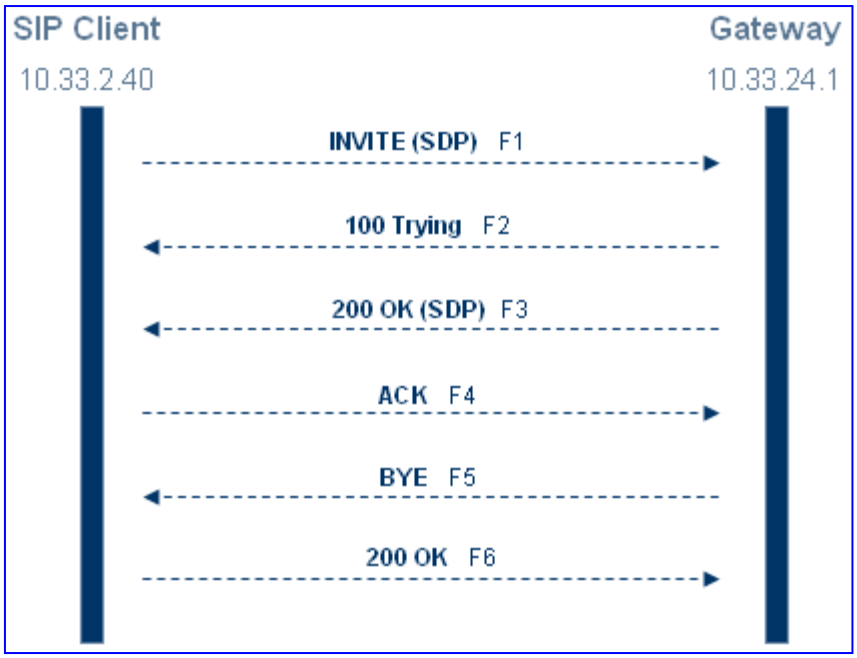
Problem	Probable Cause	Corrective Action
General Voice Streaming Problems		
Attempts to perform voice streaming operations results in each Syslog containing the string: 'VS_STACK_NOT_ACTIVE'.	Voice streaming is not enabled.	Enable voice streaming by loading an *ini* file containing this entry: EnableVoiceStreaming = 1
HTTP Voice Streaming Problems		

Problem	Probable Cause	Corrective Action
The last half-second of an announcement is not played, or a record operation terminates abnormally and the Syslog displays the following: 'VSReceiveFromNetwork: VS_CONNECTION_WITH_SERVER_LOST'. (The problem has been experienced with Apache version 2.0.50 on Solaris 9.)	The Web server is closing the virtual circuit at unexpected times.	Increase the Apache KeepAliveTimeout config parameter. Try to increase it to 30 seconds or longer than the longest announcement or expected record session.

22.2.2.4 Announcement Call Flow Example

The call flow, shown in the following figure, describes SIP messages exchanged between the device (10.33.24.1) and a SIP client (10.33.2.40) requesting to play local announcement #1 (10.8.25.17) using AudioCodes proprietary method.

Figure 22-11: Announcement Call Flow



1. **SIP MESSAGE 1:** 10.33.2.40:5060 -> 10.33.24.1:5060

```
INVITE sip:annc@10.33.24.1;play=http://10.3.0.2/hello.wav;repeat=2
SIP/2.0
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
Max-Forwards: 70
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE
Contact: <sip:103@10.33.2.40>
Supported: em,100rel,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Type: application/sdp
Content-Length: 215
```

```
v=0
o=AudiocodesGW 377662 728960 IN IP4 10.33.41.52
s=Phone-Call
c=IN IP4 10.33.41.52
t=0 0
m=audio 4030 RTP/AVP 4 0 8
a=rtpmap:4 g723/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=ptime:30
a=sendrecv
```

2. **SIP MESSAGE 2:** 10.33.24.1:5060 -> 10.33.2.40:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40.010.006D
Content-Length: 0
```

3. **SIP MESSAGE 3:** 10.33.24.1:5060 -> 10.33.2.40:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE Contact: <sip:10.33.24.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.5.40.010.006D
Content-Type: application/sdp
Content-Length: 165
v=0
o=AudiocodesGW 355320 153319 IN IP4 10.33.24.1
s=Phone-Call
c=IN IP4 10.33.24.1
t=0 0
m=audio 7170 RTP/AVP 0
a=rtpmap:0 pcmu/8000
a=ptime:20
a=sendrecv
```

4. **SIP MESSAGE 4:** 10.33.2.40:5060 -> 10.33.24.1:5060

```
ACK sip:10.33.24.1 SIP/2.0
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKacnNUEeKP
Max-Forwards: 70
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 ACK
Contact: <sip:103@10.33.2.40>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Length: 0
```

5. **SIP MESSAGE 5:** 10.33.24.1:5060 -> 10.33.2.40:5060

```
BYE sip:103@10.33.2.40 SIP/2.0
Via: SIP/2.0/UDP 10.33.24.1;branch=z9hG4bKacFhtFbFR
Max-Forwards: 70
From: <sip:annc@10.33.24.1>;tag=1c1528117157
To: <sip:103@10.33.2.40>;tag=1c2917829348
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 BYE
Contact: <sip:10.33.24.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000
MSBG/v.5.40.010.006D
Content-Length: 0
```

6. **SIP MESSAGE 6:** 10.33.2.40:5060 -> 10.33.24.1:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.24.1;branch=z9hG4bKacFhtFbFR
From: <sip:annc@10.33.24.1>;tag=1c1528117157
To: <sip:103@10.33.2.40>;tag=1c2917829348
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 BYE
Contact: <sip:103@10.33.2.40>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Length: 0
```

22.2.3 Voice XML Interpreter

The device supports Voice Extensible Markup Language (VoiceXML) version 2.0. VXML is an XML-based scripting language used to prompt and collect information from callers. A VXML-based script may be used to control many types of interactive voice response (IVR) activities, including playing recorded announcements, collecting DTMF digits, recording a caller's voice, recognizing speech (i.e., automatic speech recognition or ASR), and synthesizing speech (i.e., text-to-speech or TTS). Its major goal is to bring the advantages of Web-based development and content delivery to interactive voice response applications.

Note: Currently, ASR and TTS are not supported.

22.2.3.1 Features

VoiceXML offers the following features:

- VXML uses the AMS for enhanced audio features (i.e., playing prompts on a remote server, synthesized variables, enhanced digit patterns capabilities, different languages).
- Supports DTMF recognition.
- Supports recording of audio for later playback.
- Speech recognition: subscriber's speech is compared with voice grammars residing on an external speech server that is directed using the MRCP protocol) with matching words or phrases are returned as text strings.
- Text-to-Speech (TTS): regular text written in the IVR script is translated to speech and played to the user (the translation itself is done by an external server that is directed using the MRCP protocol).

22.2.3.2 Feature Key

VXML is protected by a Software Upgrade Key. However, if the device's Software Upgrade Key doesn't contain VXML, then VXML support is disabled. In such circumstances, upon trying to activate a VXML script, a Syslog message appears notifying the user that the VXML task was disabled. In addition, when the Software Upgrade Key doesn't contain VXML, the EnableVXML *ini* file parameteris ignored (i.e., although set to 1, VXML remains disabled).

22.2.3.3 VXML Scripts

Conceptually, there are two different types of VXML scripts that can be used (simultaneously or only one) by the device:

- **Dynamic scripts:** This script is downloaded as needed for an individual call and usually contains customized content for that particular call. When a call arrives, the device accesses a remote Web server to download a script. Once the script is downloaded. it's parsed, executed, and cleaned up at the end of the call.
- **Static scripts:** This script represents an application that can be used across many different callers. An example of such an application might be a drug prescription refill service where a prompt is played to the caller, the prescription number is obtained from the caller as speech or DTMF digits, and this data is then saved to an off-board database.

There are ramifications in using both these types of scripts. A dynamic script can be customized for each caller, but has to be downloaded and parsed for every call. However, static scripts are loaded once (although the system checks periodically for updates), parsed once, and then is re-used for each call. This results in better performance and using of fewer resources per call, because each call uses only what it needs of the parent script, and doesn't need its own copy of the full script.

Scripts are loaded initially through an INVITE message from the SIP Call Agent to the device. If the script is a static script, the device checks whether the file has already been loaded, and if so, it uses the existing script. Otherwise, the script is loaded to the device. Dynamic scripts are always loaded when requested.

A VXML script can trigger another VXML script to be loaded. An example of this is the VXML <goto> element, which can cause a transition to a different form in the same script, or to a completely different script. The script that is loaded as a result of the execution of the first script can be either dynamic or static. If the second script is a static script, the device checks whether it has already been loaded to the device and references that copy if it exists. Otherwise, the second script is loaded, parsed, and executed.

There are multiple ways in which VXML scripts may be loaded to the device. These include automatic update for static scripts (which allows for a script to be loaded using a remote HTTP server) or HTTP for dynamic scripts. Refer to the appropriate sections for additional details.

The device can activate a VXML script using the VXMLID parameter in the Request-URI user part only, upon receipt of a regular INVITE message. For example:

```
Request-URI = <VXMLID>http://mydomain.com/myscript.cgi@host;
```

This is in addition to invoking VXML scripts on the receipt of SIP Request-URIs such as:

```
<VXMLID>@host;voicexml= http://...
```

This feature is supported for IP-to-Tel and Tel-to-IP calls. For specified dialed phone numbers, the user part can be manipulated by adding a VXML script path. For example, upon receipt of the INVITE request, INVITE sip:100@myhost, the device can be configured to manipulate (using the IP to Tel Manipulation table) the Request-URI user part to voicexml=http://myhost.com/script.cgi@myhost.

22.2.3.4 Proprietary Extensions

To provide the functionality intended by the VXML specification and to extend the functionality of the VXML specification, some proprietary extensions have been included in the AudioCodes VXML Interpreter. These extensions are discussed in the following sections and are intended to enable a VXML script to make use of the advanced audio capabilities provided by the device.

22.2.3.4.1 Record

As the device doesn't provide the ability to record ‘on-board’, it is necessary to record a caller’s speech by streaming the audio to either an external NFS server. There are two additional attributes for the VXML <record> element that can be used to specify the off-board file name as well as the streaming mechanism for recording speech.

- “dest” attribute for <record>, which refers to a fully specified URL. An example of this is the following script:

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns=\"http://www.w3.org/2001/vxml\">
<form id="form1">
   <record name="msg" finalsilence="3000ms" maxtime="60s"
dtmfterm="true"
dest="http://192.168.1.2/recordings/greetings/callersspeech.wav">
      <audio src= "http://192.168.1.2/prompts/recordprompt.wav"/>
      <filled>
           <audio src = “http://192.168.1.2/prompts/confirm.wav”/>
           <audio src= "http://192.168.1.2/
greetings/callersspeech.wav"/>
         <exit/>
      </filled>
       <noinput>
           <audio src=
"http://192.168.1.2/prompts/recordprompt2.wav "/>
            <reprompt/>
        </noinput>
   </record>
  </form>
</vxml>
```

In this example, the “dest” attribute of the <record> element specifies that the caller’s speech must be streamed with HTTP to the system with IP address 192.168.1.2, and stored in a file called “callersspeech.wav” on that system.

- The “destexpr” attribute provides an alternative to the "dest" attribute. The “destexpr” attribute is evaluated during runtime to determine where to store the caller’s speech. The following is an example script illustrating its usage:

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns=\"http://www.w3.org/2001/vxml\">
<var name=”recordpath” expr =
“’http://192.168.1.2/recordings/greetings/’”/>
<form id="form1">
   <record name="msg" finalsilence="3000ms" maxtime="60s"
dtmfterm="true" destexpr="recordpath + ‘callersspeech.wav’">
      <audio src= "http://192.168.1.2/prompts/recordprompt.wav"/>
      <filled>
           <audio src = “http://192.168.1.2/prompts/confirm.wav”/>
           <audio expr= " recordpath + ‘callersspeech.wav’"/>
         <exit/>
      </filled>
       <noinput>
           <audio src=
"http://192.168.1.2/prompts/recordprompt2.wav"/>
            <reprompt/>
        </noinput>
   </record>
  </form>
</vxml>
```

22.2.3.4.2 Audio Extensions

The device provides a rich set of functionality for building and playing announcements using recorded audio files. This functionality includes the ability to play certain types of phrases such as date, time, and number based upon a specific languages grammar rules.

The files used to build the announcements can be stored on the device, or can be stored off-board on an external file system.

To take advantage of the advanced announcement capabilities provided by the device, the AudioCodes resident VXML Interpreter provides some extensions to the VXML <audio> element. These extensions are discussed in the following sections.

For more information on provisioning audio for the device, refer to the *Audio Provisioning Server (APS) User's Manual*.

Local Audio

While not a true extension, it's possible to play audio files that reside on-board a device. The following is an example of how such an audio file can be referenced using a VXML <audio> element.

```
<audio src = "http://localhost/123"/>
```

This reference directs the VXML software to play the audio segment marked with identifier '123'.

Using this method of access, the advanced audio structures defined by the AudioCodes Audio Provisioning Server (APS) can be referenced. While these various structures are outside the scope of the current document, they include sets, sequences, and multi-language variables. For more information on these advanced audio structures, refer to the *Audio Provisioning Server (APS) User's Manual*.

Say-as Tag for the Audio Element

While the VXML <say-as> tag is typically used as a directive to a text-to-speech engine in association with a VXML <prompt> element, the AudioCodes resident VXML Interpreter allows the <say-as> tag to also be used with the <audio> element. In this context, the <say-as> tag directs the VXML Interpreter to play phrases such as dates and times using provisioned audio files. The following is an example of an <audio> element using the <say-as> extension:

```
<audio> <say-as interpret-as="date"> 20080704 </say-as> </audio>
```

This example assumes that the device has been provisioned with the appropriate audio to play this example. The <audio> element in the example directs the VXML Interpreter to announce the date "July 4th, 2008".

The following table lists the supported phrase types, any valid subtypes for the phrases, the expected input format for each phrase type, and any notes for the various phrase types.

Table 22-7: Say-as Phrase Types

Say-as Token	Variable Type	Variable Subtype	Variable Input Format	Note
date	date	None supported	yyyymmdd	Dates are always announced according to the grammar rules of the language.
duration	duration	None supported	The input is up to 10 digits, with the value representing the duration in seconds.	Duration is always announced as hours, minutes, and seconds.

Say-as Token	Variable Type	Variable Subtype	Variable Input Format	Note
currency	money	Three-character ISO currency code. There is a specific set of currencies supported by the device, which are documented in Audio Provisioning Server (APS) User's Manual: Audio Files.	The input is up to 10 digits.	The number is converted appropriately to the currency in question. A value of 1234 in US Dollars, for example, is spoken as 12 dollars and 34 cents. The same input as Yen would be 1 thousand, 234 Yen.
number	number	cardinal	Up to 10 digits.	Integer
number:cardinal	number	cardinal	Up to 10 digits.	
number:ordinal	number	ordinal	Up to 10 digits.	Range of supported ordinal numbers varies by language, as to whether a certain ordinal number is supported by the language.
number:digits	digits	generic	A string of up to 64 digits including 0-9, * and #.	
telephone	digits	generic	A string of up to 64 digits including 0-9, * and #.	
telephone:ndn	digits	North American DN	Must be 10 digits 0-9.	
telephone:gen	digits	generic directory number	A string of up to 64 digits including 0-9, *, and #.	
time	time	t24	hhmm	24 hour time.
time:t12	time	t12	hhmm	
time:t24	time	t24	hhmm	

Below are two examples that direct the device to announce the cardinal number 1000.

```
<audio> <say-as interpret-as="number"> 1000 </say-as> </audio>
<audio> <say-as interpret-as="number:cardinal"> 1000 </say-as>
</audio>
```

In the example below, the device is directed to announce the string as a North American directory number. The output is "eight hundred, five five five, one two one two". A few moments of silence are inserted at the points in the phrase indicated by commas.

```
<audio> <say-as interpret-as="telephone:ndn"> 8005551212 </say-as>
</audio>
```

In the example below, the device outputs the announcement “one million two hundred twenty thousand seven hundred dollars and fifteen cents”:

```
<audio> <say-as interpret-as=”currency:usd”> 122070015 </say-as>
</audio>
```

Supplying Values to Provisioned Variables

As mentioned previously, the APS provides the capability to provision several types of advanced audio structures, including multi-language variables. A multi-language variable is an instance of one of the supported phrase types such as date and time. The APS assigns a numeric segment identifier to each variable, and the value for the variable can be provided at runtime. VXML doesn't define any capability for passing a value to a variable, therefore, the AudioCodes VXML Interpreter provides an extension to support this capability.

Below is an example that demonstrates this capability. Assume that a variable of type “date” has been provisioned on the APS, and the variable has been assigned segment identifier 17.

```
<audio src=”http://localhost/17?var=20080120”/>
```

In this example, the device outputs the date “January 20th, 2008”.

Supplying Selector Values to Provisioned Variables and to Say-as Phrases

Another concept supported by the device is “selectors”. A selector is a keyword and value pair that is used by the device software to build announcements. There can be many combinations of keywords and values used for selectors, but the keyword “lang” and a language code are especially useful because this pair of tokens can be used to vary the language for announcements. For example, to announce the date from the previous example in French, below is syntax using the previous example along with a selector that builds the French announcement:

```
<audio src=”http://localhost/17?var=20070620&sel=lang=fr”/>
```

More than one selector in an <audio> element can exist. In the example below, the language is French and gender selector with the value “female” is also specified:

```
<audio
src=”http://localhost/17?var=20070620&sel=lang=fr&gender=F”/>
```

Selectors can also be useful in combination with <say-as> elements. For example, the following illustrates making the same announcement from the previous example using a <say-as> element:

```
<audio src=”?sel=lang=fr&gender=F”> <say-as interpret-as=”date”>
20070620 </say-as> </audio>
```

For more information on available selectors, refer to the *Audio Provisioning Server (APS) User's Guide*.

22.2.3.4.3 Language Identifier Support

The AudioCodes resident VXML engine supports language identifiers as specified by RFC 3066. However, when accessing audio resident on the device using the proprietary extensions described earlier, the country code portion of the identifier is ignored. In addition, the language code portion of the identifier supports the languages listed in the table below.

Table 22-8: Support for Language Code Portion of Identifier

Language Code	Language
bd	Belgian Dutch
ca	Catalan
cs	Czech
de	German
el	Greek
en	English
es	Spanish
eu	Basque
fr	French
gl	Gallegan
he	Hebrew
hi	Hindi
it	Italian
ja	Japanese
ko	Korean
ms	Malay
nl	Netherlands Dutch
pt	Portuguese
ru	Russian
sw	Swedish
th	Thai
tl	Tagalog
tr	Turkish
vi	Vietnamese
yu	Cantonese
zh	Mandarin

22.2.3.5 Combining <audio> Elements

The VXML specification supports multiple <audio> elements nested within other elements such as prompts. An example demonstrating this functionality which includes the AudioCodes extensions is useful to show how multiple components can be combined to create a single announcement.

The following example shows how an announcement can be constructed that says "Welcome to Acme Corporation. Today's date is June 20th, 2010. Today's special is large widgets, two for ten dollars." For the sake of the example, assume the following:

- "Welcome to Acme Corporation" and "Today's special is large widgets, two for ten dollars" are stored on an external file system and is played using HTTP streaming.
- "Today's date is" is a recording provisioned on the APS as segment 99.
- "June 20th, 2008" is a multi-language variable announcement made up of multiple recordings provisioned on the APS.

```
<prompt bargein="false">
   <audio src = "http://192.168.1.2/announcements/welcome.wav"/>
   <audio src = "http://localhost/99"/>
   <audio src="?sel=lang=en"> <say-as interpret-as="date">
20080620 </say-as> </audio>
   <audio src =
"http://192.168.1.2/announcements/todaysspecial.wav"/>
</prompt>
```

22.2.3.6 Notes Regarding Non-compliant Functionality

The AudioCodes resident VXML Interpreter doesn't asynchronously throw events as described in the VXML Specification. For example, if the clear element tries to clear an element within a script and that element does not exist, the VXML Specification specifies that the Interpreter should throw an "error.badfetch" event. In contrast, the AudioCodes Interpreter logs an appropriate error to the syslog for the device and the script exits. The VXML Interpreter behaves similarly for software errors in general such as running out of memory resources, trying to access non-existent audio files, etc. The impact of not throwing events asynchronously can be minimized by carefully testing all code paths for a VXML script before its deployment.

22.2.3.7 Supported Elements and Attributes

The following status legend should be referenced for all tables in the following subsections:

- **NS:** Not Supported
- **PS:** Partially Supported
- **S:** Supported

22.2.3.7.1 VoiceXML Supported Elements and Attributes

Table 22-9: VoiceXML Supported Elements and Attributes

Element	Parameter	Max Size	Shadow Variable	Status	Comments
<assign>				S	
	name	64		S	
	expr	128		S	
<audio>				S	The AudioCodes audio element has proprietary extensions in addition to attributes from the standard to support on-board audio variables.

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	src	256		S	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
	expr	128		S	
	caching			Ignored	1.0 VXML attribute not present in VXML 2.0.
<block>				S	
	name	32		S	
	expr	128		S	
	cond	64		S	
<catch>				S	
	event	64		S	
	count	numeric		S	
	cond	128		S	
<choice>				S	
	dtmf			S	
	accept			NS	
	next	256		S	
	expr	128		S	
	event	64		S	
	eventexpr	128		S	
	message			NS	
	messageexpr			NS	
	fetchaudio			NS	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
<clear>				S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	namelist	4 * 32		S	
<disconnect>				S	
<else>				S	
<elseif>				S	
	cond	128		S	
<enumerate>				NS	
<error>				S	
	count	numeric field		S	
	cond	128		S	
<exit>				S	
	expr	128		S	
	namelist	4 * 32		S	
<field>				S	
	name	32		S	
	expr	128		S	
	cond	128		S	
	type	enum		PS	Built-in grammars are supported for recognition against fields, but the match isn't spoken as the built-in type in text-to-speech.
	slot			NS	Default value is the variable name, thus, slot is not needed.
	modal	true/false		S	
		64	name$.utterance	S	
		enum	name$.inputmode	S	
		64	name$.interpretation	S	
		numeric	name$.confidence	S	
<filled>				S	
	mode			S	
	namelist	4 * 32		S	
<form>				S	
	id	32		S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	scope	enum		S	
<goto>				S	
	next	256		S	
	expr	128		S	
	nextitem	32		S	
	expritem	128		S	
	fetchaudio			NS	
	fetchtimeout			NS	
	fetchhint			NS	
	maxage			NS	
	maxstale			NS	
<grammar>				S	
	version			S*	For voice grammars, this is passed to the speech recognition engine.
	xml:lang	5		S*	For voice grammars, this is passed to the speech recognition engine.
	mode			S	
	root			S	
	tag			S	
	xml:base			NS	
	src	256		S	
	scope	enum		S*	In this release, a document scope grammar isn't active in a dialog scope form.
	type	enum		PS	Built-in grammars are supported for recognition against fields, but the match is not spoken as the built-in type in text-to-speech.
	weight	numeric		S*	For voice grammars, this is passed to the speech recognition engine.
	fetchtimeout			NS	Voice grammars are maintained on the speech recognition server, not on device, thus this set of attributes that control caching of grammar doesn't apply.
	fetchhint			NS	Voice grammars are maintained on the speech recognition server, not on device, thus this set of attributes that control caching of grammar doesn't apply.
	maxage			NS	Voice grammars are maintained on the speech recognition server, not on device, thus this set of attributes that

Element	Parameter	Max Size	Shadow Variable	Status	Comments
					control caching of grammar doesn't apply.
	maxstale			NS	Voice grammars are maintained on the speech recognition server, not on device, thus this set of attributes that control caching of grammar doesn't apply.
<help>				S	
	count	numeric field		S	
	cond	128		S	
<if>				S	
	cond	128		S	
<initial>				NS	The initial element and all its attributes aren't supported in this release.
	name			NS	
	expr			NS	
	cond			NS	
<link>				S	
	next	256		S	
	expr	128		S	
	event	65		S	
	eventexpr	128		S	
	message			NS	
	messageexpr			NS	
	dtmf	31		S	
	fetchaudio			NS	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
<log>				S	
	label	32		S	
	expr	128		S	
<menu>				S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	id	32		S	
	scope	enum		S	
	dtmf	true/false		S	
	accept			NS	It's not obvious how to instruct the speech recognition engine that approximate matches are acceptable.
<noinput>				S	
	count	numeric		S	
	cond	128		S	
<nomatch>				S	
	count	numeric field		S	
	cond	128		S	
<object>				S	
	name	*		S	Since objects are developed for proprietary purposes as needed, attribute sizes aren't listed.
	expr			S	
	cond			S	
	classid			S	
	codebase			S	
	codetype			S	
	data			S	
	type			S	
	archive			S	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
<option>				S	
	dtmf	31		S	
	accept			NS	It's not obvious how to instruct the speech recognition engine that approximate matches are acceptable.
	value	32		S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
<param>				S	
	name	32		S	
	expr	128		S	
	value	128		S	
	valuetype	enum		S	
	type	128		S	
<prompt>				S	
	bargein	true/false		S	
	bargeintype			PS	Speech barge-in is supported, but not hotword.
	cond	128		S	
	count	numeric		S	
	timeout	numeric		S	
	xml:lang	5			
	xml:base	256			
<property>				S	
	name	32		S	
	value	128		S	
<record>				S	
	name	32		S	
	expr	128		S	
	cond	128		S	
	modal			NS	Grammars are not supported, thus, modal doesn't apply.
	beep	true/false		S	Requires that a user-defined tone be added to the system. For an example, see 'Example of UDT 'beep' Tone Definition' on page 562. Refer to the Auxilary Files section for additional details regarding creating user-defined tones.
	maxtime	time value		S	
	finalsilence	time value		S	
	dtmfterm			NS	DTMF and voice grammars aren't supported for record, but the termchar property can be used to terminate recordings.

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	type			NS	The recorded audio format is specified by the file extension in the dest or destexpr attribute.
	dest	256		S*	Not part of the standard, either this attribute or destexpr are needed to specify the remote URL where the recorded audio is stored.
	destexpr	128		S*	Refer to previous item.
		numeric	name$.duration	S	
			name$.size	NS	As recorded audio is not stored onboard the device size is not available.
		1	name$.termchar	S	
		true/false	name$.maxtime	S	
<reprompt>				S	
<return>				S	
	event	64		S	
	eventexpr	128		S	
	message			NS	
	messageexpr			NS	
	namelist	4 * 32		S	
<script>				NS	The script element and all of its attributes are not supported.
	src			NS	
	charset			NS	
	fetchtimeout			NS	
	fetchhint			NS	
	maxage			NS	
	maxstale			NS	
<subdialog>				S*	Playing a prompt from a sub-dialog element is not supported in this release.
	name	32		S	
	expr	128		S	
	cond	128		S	
	namelist	4 * 32		S	
	src	256		S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	srcexpr	128		S	
	method	enum		S	
	enctype			NS	
	fetchaudio			NS	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
<submit>				S	
	next	256		S	
	expr	128		S	
	namelist	4 * 32		S	
	method	enum		S	
	enctype			NS	
	fetchaudio			NS	
	fetchtimeout			NS	
	Fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	Maxage			NS	
	maxstale			NS	
<throw>				S	
	Event	64		S	
	eventexpr	128		S	
	message			NS	
	messageexpr			NS	
<transfer>				S	
	Name			S	
	Expr			NS	
	Cond			S	
	Dest			NS	Only numbers.

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	destexpr			S	
	Bridge			S	Only blind transfer supported (false).
	type			S	Only blind transfer supported (blind).
	connecttimeout			NS	
	maxtime			NS	
	transferaudio			NS	
	Aai			NS	
	Aaiexpr			NS	
			name$.duration	S	
			name$.inputmode	S	
			name$.utterance	S	
<value>				S	
	expr	128		S	
<var>				S	
	name	32		S	
	expr	128		S	
<transfer>	Name			S	
	Expr			NS	
	Cond			S	
	Dest			NS	Only numbers.
	destexpr			S	
	Bridge			NS	Only Bridge = false
	type			NS	Only type = blind
	connecttimeout			NS	
	maxtime			NS	
	transferaudio			NS	
	Aai			NS	
	Aaiexpr			NS	
			name$.duration	S	
			name$.inputmode	S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
			name$.utterance	S	
<value>				S	
	expr	128		S	
<var>				S	
	name	32		S	
	expr	128		S	
<vxml>				S	

22.2.3.7.2 SRGS and SSML Support

Note that elements associated with either the Speech Recognition Grammar Specification (SRGS) or Speech Synthesis Markup Language (SSML) are used to control the behavior of a remote speech engine for either speech recognition or text-to-speech. These elements would be passed from the VXML interpreter to the remote speech engine and are outside the scope of VXML.

22.2.3.7.3 VoiceXML Supported Properties

Table 22-10: VoiceXML Supported Properties

Platform Properties	Status	Equivalent *ini* file parameter or Notes
Recognizer		
confidencelevel	S	VxmlConfidenceLevel
Sensitivity	S	VxmlSensitivityLevel
speedvsaccuracy	S	VxmlSpeedVsAccuracy
Completetimeout	S	VxmlCompleteTimeout
incompletetimeout	S	VxmlInCompleteTimeout
maxspeechtimeout	S	VxmlMaxSpeechTimeout
DTMF Recognizer		
Interdigittimeout	S	VxmlInterDigitTimeout
Termtimeout	S	VxmlTermTimeout. Note that the system default is not 0 as directed in the specification for the protocol, but 3 seconds. This is to ensure digit collection functions correctly.
Termchar	S	VxmlTermChar
Prompt and Collect		
Bargein	S	VxmlBargeinAllowed

Platform Properties	Status	Equivalent *ini* file parameter or Notes
Bargeintype	NS	Regular speech vs hotword bargein
Timeout	S	VxmlNoInputTimeout
Fetching		
Audiofetchhint	NS	
Audiomaxage	NS	
Audiomaxstale	NS	
documentfetchhint	NS	
documentmaxage	NS	
documentmaxstale	NS	
grammarfetchhint	NS	
Grammarmaxage	NS	
Objectfetchhint	NS	
Objectmaxage	NS	
Objectmaxstale	NS	
Scriptfetchhint	NS	
Scriptmaxage	NS	
Scriptmaxstale	NS	
Fetchaudio	NS	
Fetchaudiodelay	NS	
fetchaudiominimum	NS	
Fetchtimeout	NS	
Miscellaneous		
Inputmodes	S	VxmlSystemInputModes. Note that the system default is 0 (DTMF) vs 2 (Voice and DTMF) as specified in the specification. This is because the majority of systems are expected to use DTMF collection and local or streamed announcements as opposed to text-to-speech and speech recognition.
Universals	NS	Universal grammars and behaviors such as help, cancel, and exit. Default is none.
Maxnbest	NS	Size of last result array

22.2.3.7.4 VoiceXML Variables and Events

Table 22-11: VoiceXML Variables and Events

Variable/Event Name	Status	Notes
Standard Session Variables		

Variable/Event Name	Status	Notes
session.connection.local.uri	S	
session.connection.remote.uri	S	
session.connection.protocol.name	S	
session.connection.protocol.version	S	The version is "2" (instead of "2.0").
session.connection.redirect	S	Redirect reason and screening information contains underscore "_" (instead of white space) between words.
session.connection.aai	S	
session.connection.originator	NS	
Standard Application Variables		
application.lastresult$	S	The application.lastresult variables array is one element deep.
application.lastresult$**[i].**confidence	S	
application.lastresult$**[i].**utterance	S	
application.lastresult$**[i].**inputmode	S	
application.lastresult$**[i].**interpretation	S	
Pre-defined Events **Note:** while throwing and catching events from scripts are supported, throwing events asynchronously from within the interpreter (e.g., an event.badfetch) is currently not supported.		
catch	S	
connection.disconnect.hangup	NS	
connection.disconnect.transfer	NS	
exit	S	
help	S	
noinput	S	
nomatch	S	
maxspeechtimeout	S	
error.badfetch	PS	In most cases, the conditions that would cause this event are recognized during script parsing, thus, the script loading fails.
error.badfetch.http.response_code	NS	
error.badfetch.protocol.response_code	NS	
error.semantic	PS	
error.noauthorization	NS	
error.noresource	NS	
error.unsupported.builtin	NS	
error.unsupported.format	NS	
error.unsupported.language	NS	

Variable/Event Name	Status	Notes
error.unsupported.objectname	NS	Unsupported elements are recognized during initial parsing, thus, the script isn't executed, and no events are thrown.
error.unsupported.element	NS	Unsupported elements are recognized during initial parsing, thus, the script isn't executed, and no events are thrown.
Transfer Events		
connection.disconnect.hangup	NS	
connection.disconnect.transfer	NS	
Transfer Errors		
error.connection.noauthorization	NS	
error.connection.baddestination	NS	
error.connection.noroute	NS	
error.connection.noresource	NS	
error.connection.protocol.nnn	NS	
error.unsupported.transfer.blind	NS	
error.unsupported.transfer.bridge	NS	
error.unsupported.uri	NS	

22.2.3.7.5ECMAScript Support

The following table describes the ECMAScript support that the AudioCodes resident VXML engine provides. As shown in the example below, all operands and operators in an expression must be separated by one or more ECMAScript whitespace characters.

```
<var name="orange" expr=”var1 + 7"/>
```

Below is an example of incorrect formatting (i.e., not supported):

```
<var name="orange" expr=”var1+7"/>
```

Table 22-12: ECMAScript Support

Operand/Operator	Examples	Status	Note
Whitespace chars	tab, vertical tab, form feed, and space	S	
Arithmetic Operators	+, ++, -, --, *, /, %	S	
Logical Operators	&&, ||, !	S	
Assignment Operators	=, +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, >>>=	S	
Bitwise Operators	&, ^, |, ~, <<, >>, >>>	S	

Operand/Operator	Examples	Status	Note
Comparison Operators	==, !=, >, >=. <. <=	S	
String Operators	+, +=	S	
Entity Reference Mapping The following is supported / required: **Operator Entity Reference** < &lt <= <= > >= >= >= && && Support for the '≤' and '≥' entities is currently not available.		S	
Null Literals	null	S	Section 7.8.1, ECMA-262 3rd Edition December, 1999
Boolean Literals	true, false	S	Section 7.8.2, ECMA-262 3rd Edition December, 1999
Numeric Literals		S	Section 7.8.3, ECMA-262 3rd Edition December, 1999
String Literals		S	Section 7.8.4, ECMA-262 3rd Edition December, 1999

22.2.3.8 Example of UDT 'beep' Tone Definition

The following is an example definition for 'beep' tone used for the <record> element:

```
#record beep tone
[CALL PROGRESS TONE #1]
Tone Type=202
Low Freq [Hz]=430
High Freq [Hz]=0
Low Freq Level [-dBm]=13
High Freq Level [-dBm]=0
First Signal On Time [10msec]=100
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
Default Duration [msec]=350
```

22.2.3.9 Limitations and Restrictions

The maximal length of the VXML file is 65536 bytes.

Part V

Data-Router Settings

This part provides general system configurations.

This part describes the data-router configurations and includes the following main areas:

- WAN settings - see 'WAN Access' on page 567
- Firewall and ACL - see 'Firewall and ACL' on page 581
- Quality of Service - see 'QoS' on page 595
- Virtual Private Network - see 'VPN' on page 605
- Data services - see 'Data Services' on page 615
- Data routing - see 'Data Routing' on page 623
- Objects and rules - see 'Objects and Rules' on page 629
- Network connections - see 'Configuring Network Connections' on page 633

Before you begin configuring the data functionality, you should familiarize yourself with working with the data-related configuration pages, as described in 'Getting Acquainted with Data Configuration Pages' on page 565.

Notes:

- Virtual Routing and Forwarding (VRF) can only be configured using the device's CLI. This allows multiple instances of a routing table to co-exist within the same router at the same time. The device's VRF feature allows interfaces to be clustered into a VRF to provide segregated routing domains. The VRF feature uses the device's single physical router as multiple logical routers (up to 32). Each VRF is associated with its own routing table. When creating fully separated logical routers on the same physical router, every interface can be mapped to a specified VRF and static routes can be added to it. The main CLI command for configuring VRF is **ip vrf**. **Note:** Some features are available only on the default, unnamed, VRF. These include, amongst others, BGP, OSPF, RIP, Management interfaces (Web, CLI and SNMP), and SIP (when using the device's VoIP component). For a complete list of features supported only on the default VRF, please contact AudioCodes. For a detailed description of CLI configuration, refer to the *MSBG Series CLI Reference Guide*.
- IPSec tunneling can only be configured using the device's CLI. This is configured using the **crypto** CLI commands. For a detailed description of CLI configuration, refer to the *MSBG Series CLI Reference Guide*.
- The device supports the monitoring of traffic traversing its LAN ports (i.e., Port Mirroring). This includes monitoring of egress and/or ingress traffic. This feature is useful for analyzing traffic or debugging network problems. The CLI commands, **port monitor** and **show data port-monitoring** are used for displaying this information.

23 Getting Acquainted with Data Configuration Pages

This section provides you with an overview on working with the data-routing configuration pages.

23.1 Working with Tables

Throughout the data section of the Web interface, various configuration icons are provided in the configuration tables. The figure below displays a typical example of such a table, where each row defines an entry in the table:

Figure 23-1: Working with Tables

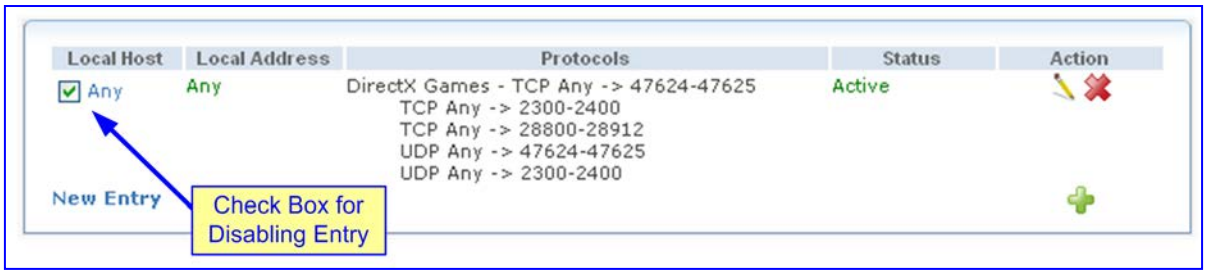
The Action' column provides icons for performing various configuration actions, as described in the table below:

Table 23-1: Description of Table Action Icons

Icon	Name	Description
	New	Adds an entry (row) in the table.
	Edit	Edits an entry in the table.
	Remove	Deletes an entry from the table.
	Download	Downloads a file.
	Copy	Copies an item to the clipboard.
	Move Up	Moves an entry one place up in the table.
	Move Down	Moves an entry one place down in the table.

Once you have added an entry in a table, you can later disable the entry by clearing the check box corresponding to the entry. For example, you can temporarily disable an access rule, by clearing the check box, as shown below.

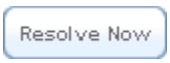
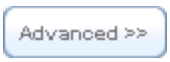
Figure 23-2: Checkbox for Temporarily Disabling Entry



23.2 Using the Configuration Buttons

Throughout the Data section of the Web interface, various buttons appear in the configuration pages, as described in the table below:

Table 23-2: Description of the Main Configuration Buttons

Button	Name	Description
	OK	Applies and saves the settings.
	Apply	Applies the settings.
	Cancel	Disregards your settings.
	Refresh	Refreshes the page, updating the configuration values or status.
	**Resolve Now**	Verifies or validates a setting, for example, that a valid Web site exists at a specified URL.
	Close	Closes the page and returns you to the previously opened page.
	**Advanced**	Shows additional (advanced) fields on the page.
	Basic	Hides the advanced fields (if previously shown by clicking the **Advanced** button).

The above table describes the main configuration buttons. Buttons specific to a certain configuration is explained later in the relevant section.

24 WAN Access

The **WAN Access** menu allows you to configure your WAN (Internet) connection. When subscribing to a broadband service, your Internet Service Provider (ISP) should provide you information regarding the method by which you are connecting to the Internet.

The device can connect to the broadband service (WAN), using one of the following methods:

- Ethernet interface (copper or fiber cable), using connection methods such as automatic IP address allocation or Point-to-Point Protocol over Ethernet (PPPoE). For more information, see 'Configuring Ethernet WAN Interface' on page 567.
- T1 WAN connection through a dual T1 line interface (according to ANSI T1.403-1999). The device uses its dual T1 WAN Data Service Unit/Channel Service Unit (DSU/CSU) port interface to transmit and receive data using IP over Point-to-Point Protocol (PPP), IP over High-Level Data Link Control (HDLC), or IP over Multilink Point-to-Point Protocol (MLPPP) framing. For more information, see 'Configuring T1 WAN Interface' on page 569.
- Symmetric High-Speed Digital Subscriber Line (SHDSL) connection using copper wire pairs. The SHDSL WAN connection supports up to four copper wire pairs according to G.991.2, acting as a remote-terminal CPE device. Both ATM and EFM modes are supported. In the ATM mode, a variety of protocols are supported, including PPPoE, PPPoA, and RFC 2684 in both bridged (Ethernet-over-ATM) and routed (IP-over-ATM) variants. In the EFM mode, the SHDSL port functions as a logical Ethernet device. For more information, see 'Configuring SHDSL WAN Interface' on page 576.

Note: The supported WAN connection methods depend on the installed Software Upgrade Key. For installing a Software Upgrade Key, refer to 'Loading Software Upgrade Key' on page 696.

24.1 Configuring Ethernet WAN Interface

For Ethernet WAN interface, you can configure the device to use one of the following methods for connecting to the Internet:

- Manual IP address Ethernet connection
- Automatic IP address Ethernet connection
- Point-to-Point Protocol over Ethernet (PPPoE)
- Point-to-Point Tunneling Protocol (PPTP)
- Layer 2 Tunneling Protocol (L2TP)

The type of connection should be as instructed by your ISP. If you do not have an Internet connection or if you want to disable all existing connections, select **No Internet Connection** from the 'Connection Type' drop-down list.

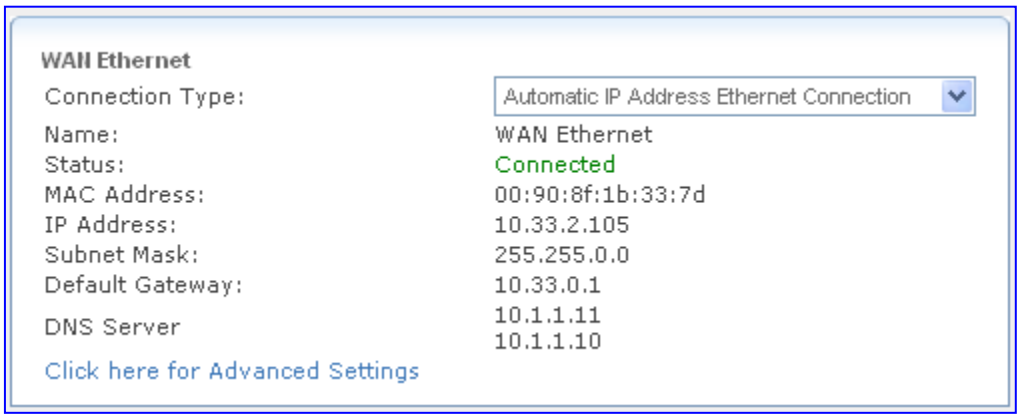
Once you have defined the basic WAN connection type (listed above and described in the procedure below), you can configure advanced settings such as the WAN port's physical attributes (link speed and duplex mode) and routing. The advanced configuration parameters can be accessed using one of the following methods:

- Clicking the **Click here for Advanced Settings** link as described in Step 3 below
- Clicking the **Edit** icon in the Connections list, as described in 'Editing Existing Connections' on page 671

➢ **To configure the Internet (WAN) connection:**

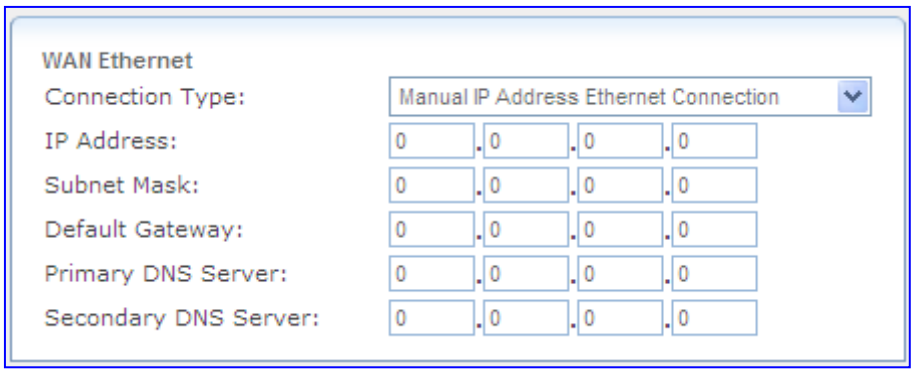
1. Open the Settings page (**Configuration** tab > **Data** menu > **WAN Access** submenu > **Settings**); the following page appears:

Figure 24-1: WAN Access



2. From the 'Connection Type' drop-down list, select the required WAN connection type:
   - 'Automatic IP Address Ethernet Connection' (see figure above)
   - 'Manual IP Address Ethernet Connection':

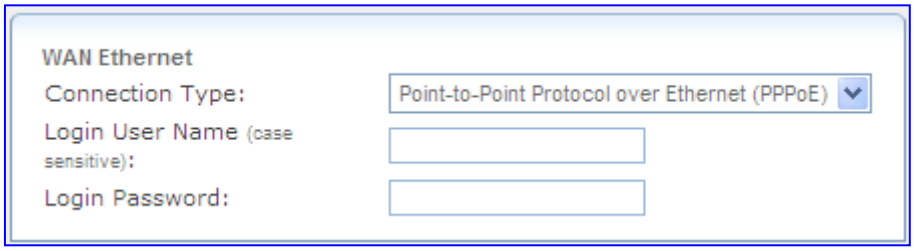
Figure 24-2: Manual WAN Connection



Specify the following:
- ♦ IP address
- ♦ Subnet mask
- ♦ Default gateway
- ♦ Primary DNS Server
- ♦ Secondary DNS Server

   - 'Point-to-Point Protocol over Ethernet (PPPoE)':

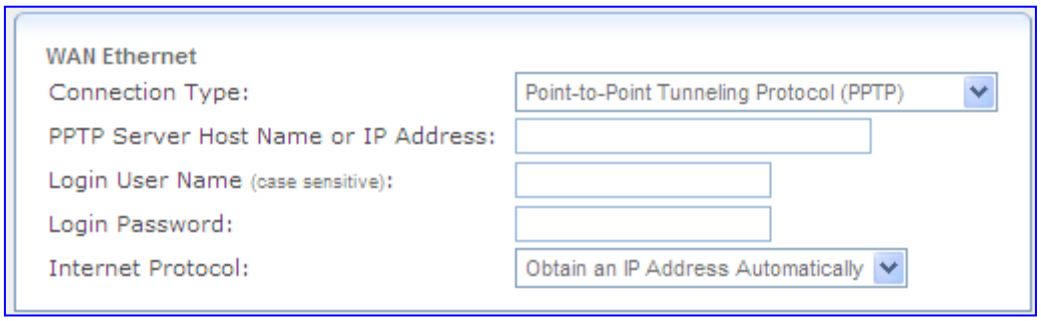
Figure 24-3: PPPoE WAN Connection



Specify the following:
- ♦ Login user name
- ♦ Login password

- 'Point-to-Point Tunneling Protocol (PPTP)'**:**

Figure 24-4: PPTP WAN Connection Type



Specify the following:

- PPTP Server Host Name or IP Address
- Login User Name
- Login Password
- Internet Protocol - select the method used by your ISP for assigning an IP address.

- 'Layer 2 Tunneling Protocol (L2TP)':

Figure 24-5: L2TP WAN Connection Type

Specify the following:

- L2TP Server Host Name or IP Address
- Login User Name
- Login Password
- Internet Protocol - select the method used by your ISP for assigning an IP address

3. For advanced WAN settings, click the **Click here for Advanced Settings** link. For more information on configuring advanced settings, see 'Editing Existing Connections' on page 671.

24.2 Configuring T1 WAN Interface

The T1 WAN interface can be configured to transmit/receive data using IP over Point-to-Point Protocol (PPP), IP over High-Level Data Link Control (HDLC), or IP over Multilink Point-to-Point Protocol (also referred to as ML-PPP, MP, MPPP, MLP, or Multilink) framing. The WAN connection must contain a physical interface (T1) and a Layer-2 protocol (PPP, HDLC, or ML-PPP). The physical interface provides the actual bandwidth between the device and the network provider. The Layer 2 protocol defines how the data is packaged and presented on the physical interface. Layer 2 protocols must be configured to match the protocol provided on the circuit.

Notes:

- Each physical T1 link is configured separately.
- Local-line loopback on the T1 WAN interface can only be configured using the device's CLI. In this loopback, packets from the Tx line interface connect to the Rx line interface. The maximum time of this loopback is enabled and configured using the **loopback** CLI command.
- Bit error rate (BER) testing on the T1 WAN interface for the far-end transmission link can only be configured using the device's CLI. The BER Test is done by generating and detecting both pseudorandom and repeating bit patterns. The test is enabled using the **ber-test** CLI command.
- For more information on CLI configuration, refer to the *MSBG Series CLI Reference Guide*.

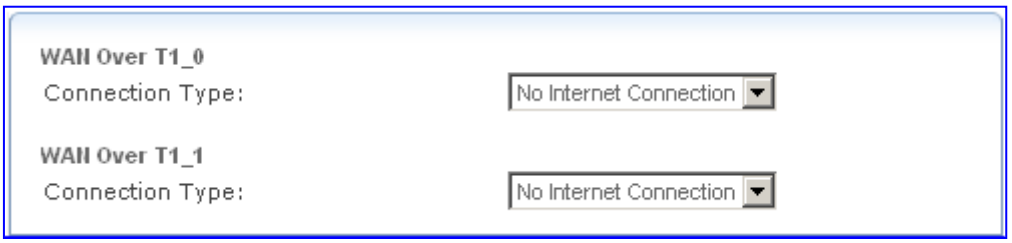
24.2.1 PPP over T1 WAN

The procedure below describes how to configure PPP over T1 WAN interface.

➢ **To configure PPP over T1 WAN interface:**

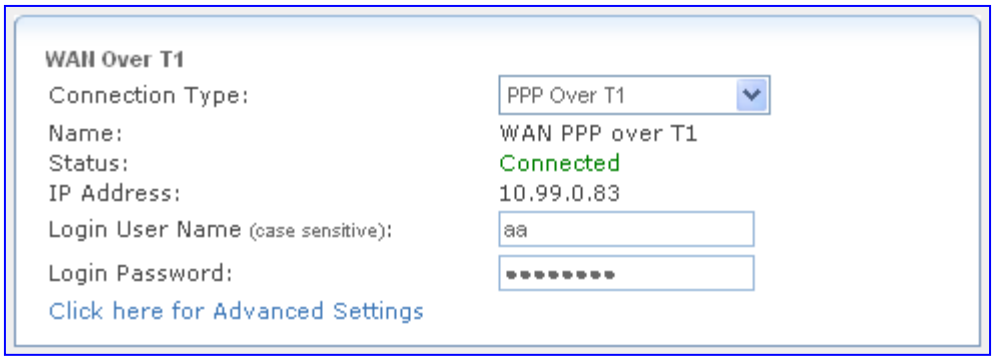
1. Open the WAN Access page (**Configuration** tab > **Data** menu > **WAN Access** > **Settings**).

Figure 24-6: WAN Access Page for T1 WAN Interface



2. From the 'Connection Type' drop-down list, select "PPP Over T1"; the page refreshes, displaying the relevant parameters:

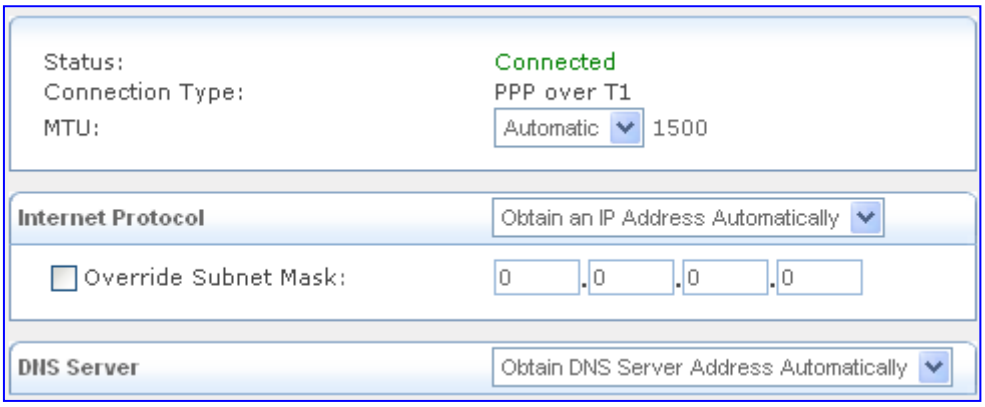
Figure 24-7: PPP Over T1



3. In the 'Login User Name' and 'Login Password' fields, enter the username and password provided by your Internet Service Provider (ISP), and then click **OK**.
4. Click the **Click here for Advanced Settings** link; the **General** tab page is displayed. The **General** tab allows you to define an arbitrary name for the connection (in the 'Name' field), view various statistical information of the connection, and disable the connection (by clicking **Disable**).

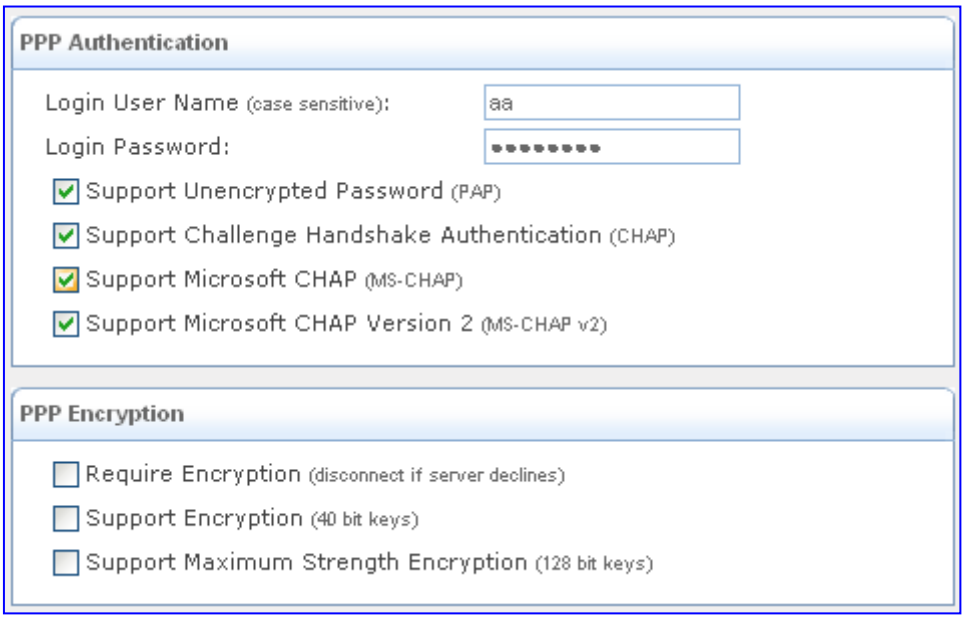
5. Click the **Settings** tab, and then from the 'Internet Protocol' drop-down list, select the IP address method used for the PPP link. This can be obtained automatically from the ISP or defined manually (by selecting 'Use the Following IP Address').

Figure 24-8: Settings Tab for PPP over T1



6. Click the **PPP** tab, and then select the supported PPP authentication and encryption methods. (You can also change the PPP login username and password.)

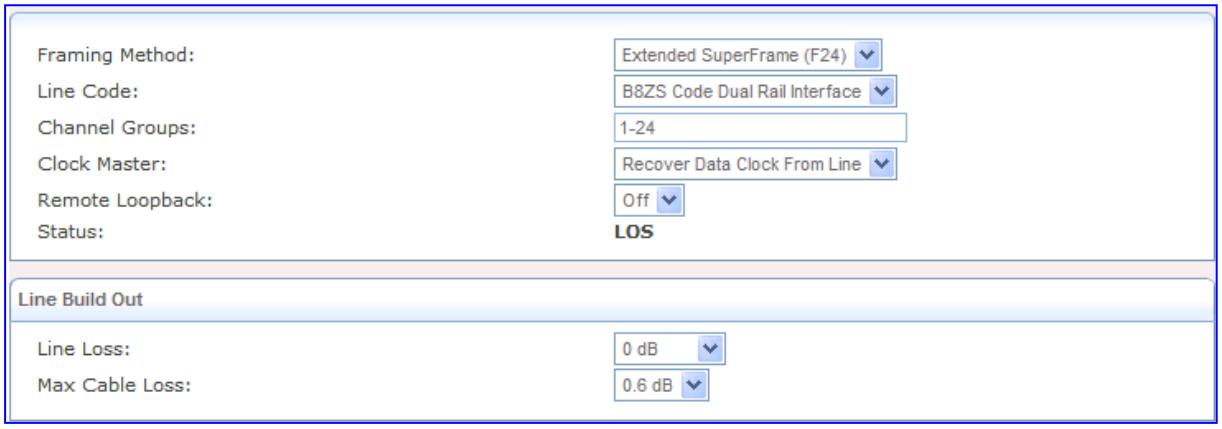
Figure 24-9: PPP Tab



7. Click the **T1** tab, and then define the following as instructed by your ISP:
   - Framing Method
   - Line Code
   - Channel Groups: define the active T1 data channels. This can be Full T1 (1.e., 1-24) or Fractional T1 (e.g., 1-3,5,8-12,24)
   - Clock Master
   - Line Build Out - pulse shape of the T1 analog interface:
     - Line Loss (pulse shape of the T1 analog interface): the Line CI code, as defined by ANSI T1.403 Annex H.

- Max Cable Loss: the maximum customer cable loss, as defined by ANSI T1.403 Annex H

Figure 24-10: T1 Tab



The 'Status' field displays the status of the T1 WAN interface:

- "Not available" (wait to sync)
- "Active" (sync)
- "LOS" (Red alarm – loss of signal)
- "LOF" (Red alarm – OOF)
- "AIS"
- "RAI" (Yellow alarm)

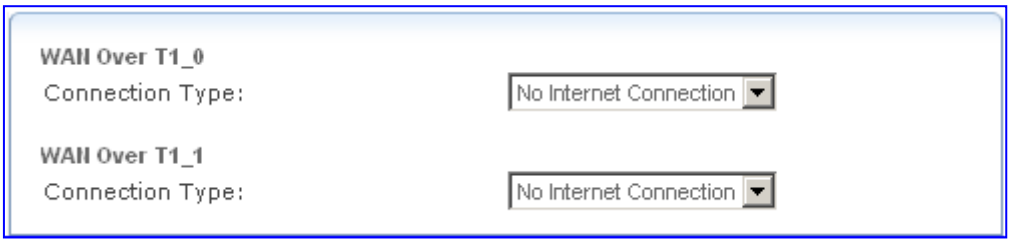
24.2.2 HDLC over T1 WAN

The procedure below describes how to configure HDLC over T1 WAN interface.

➢ **To configure HDLC over T1 WAN interface:**

1. Open the WAN Access page (**Configuration** tab > **Data** menu > **WAN Access** > **Settings**).

Figure 24-11: WAN Access Page for T1 WAN Interface



2. From the 'Connection Type' drop-down list, select "HDLC Over T1"; the page refreshes, displaying the relevant parameters:

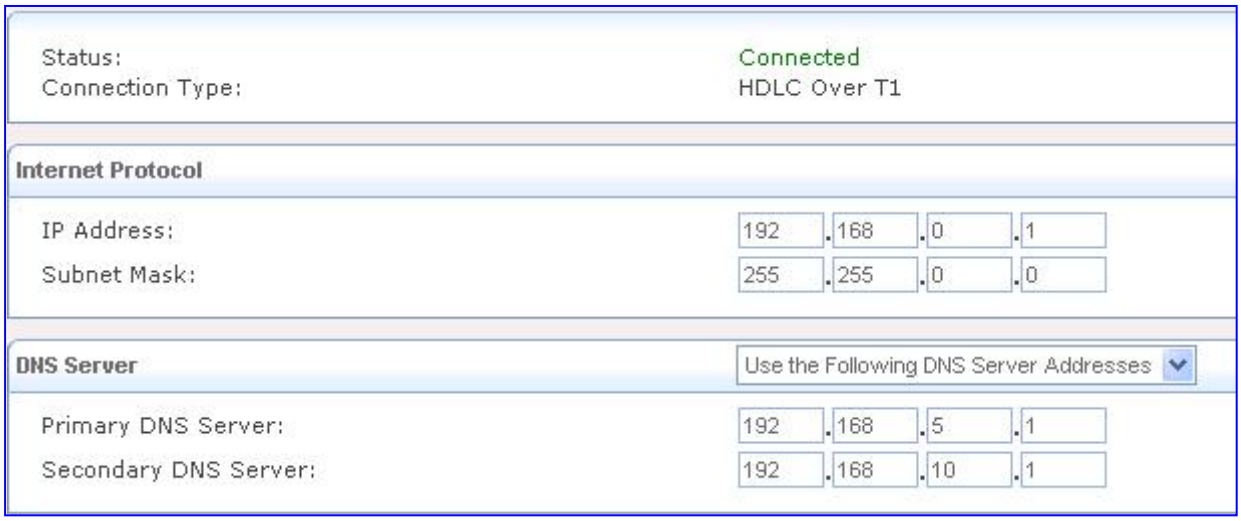
Figure 24-12: HDLC Over T1

3. In the 'IP Address' and 'Subnet Mask' fields, enter the IP address supplied by your ISP

for this connection, and then click **OK**.

4. Click the **Click here for Advanced Settings** link; the **General** tab page is displayed. The **General** tab allows you to define an arbitrary name for the connection (in the 'Name' field), view various statistical information of the connection, and disable the connection (by clicking **Disable**).
5. Click the **Settings** tab, and then from the 'Internet Protocol' drop-down list, select the IP address method used for the link. This can be obtained automatically from the ISP or defined manually (by selecting 'Use the Following IP Address').

Figure 24-13: Settings Tab HDLC over T1



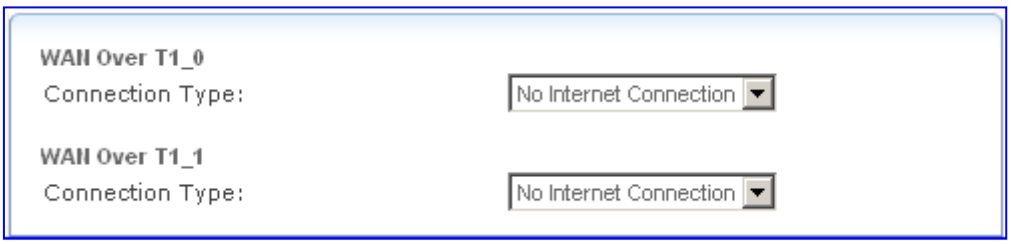
24.2.3 ML-PPP over T1 WAN

The procedure below describes how to configure ML-PPP over T1 WAN interface.

➢ **To configure ML-PPP over T1 WAN interface:**

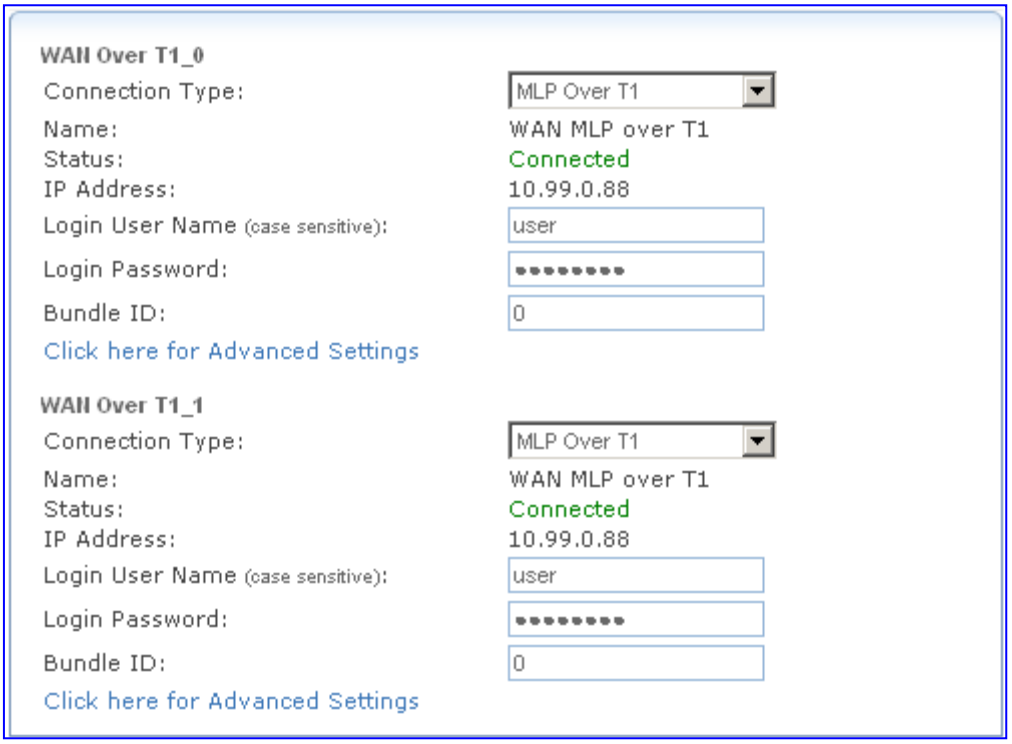
1. Open the WAN Access page (**Configuration** tab > **Data** menu > **WAN Access** > **Settings**).

Figure 24-14: WAN Access Page for T1 WAN Interface



2. From the 'Connection Type' drop-down list, select "MLP Over T1"; the page refreshes, displaying the relevant parameters:

Figure 24-15: MLP over T1 WAN



Note: For each physical T1 link to which you want to add the logical bundle, perform the steps below.

   a. In the 'Login User Name' and 'Login Password' fields, enter the username and password respectively, provided by your ISP.
   b. In the 'Bundle ID' field, enter an arbitrary number that is common to all physical T1 links on the same bundle.
   c. Click **OK**.

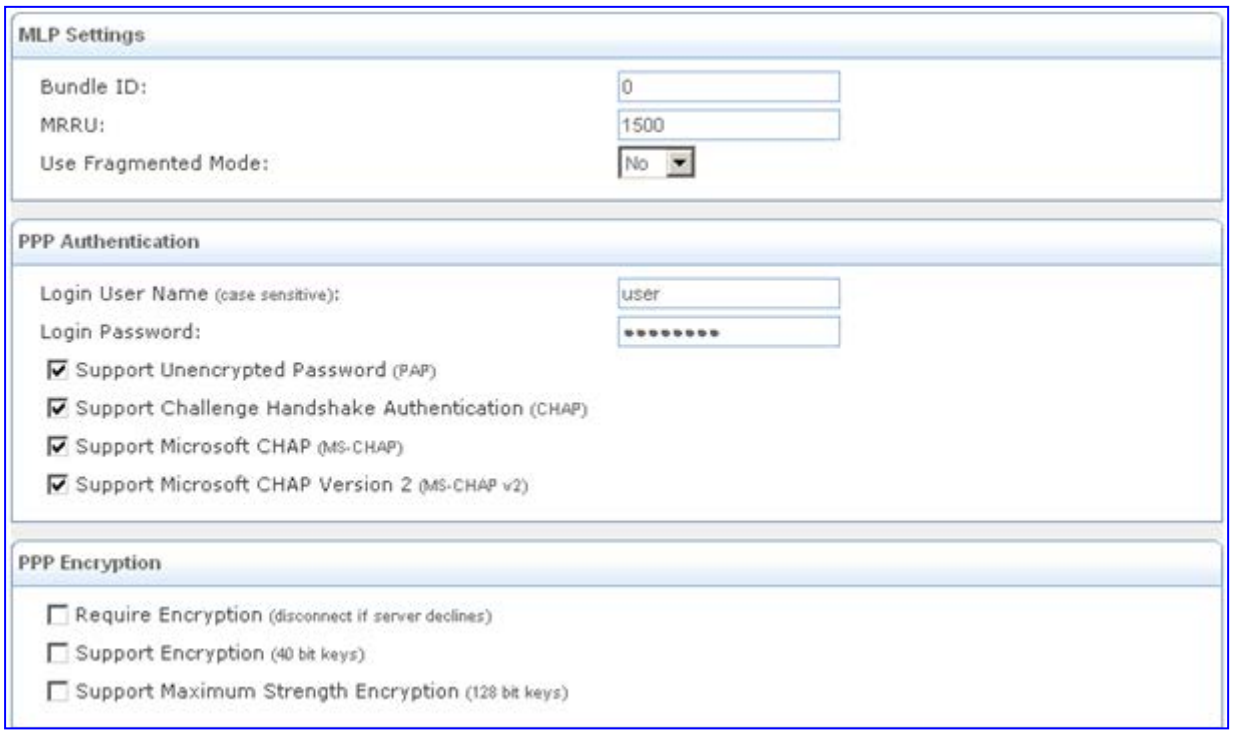
3. Click the **Click here for Advanced Settings** link; the **General** tab page appears. The **General** tab allows you to define an arbitrary name for the connection (in the 'Name' field), view various statistical information of the connection, and disable the connection (by clicking **Disable**).

4. Click the **Settings** tab, and then from the 'Internet Protocol' drop-down list, select the IP address method used for the PPP link. This can be obtained automatically from the ISP or defined manually (by selecting 'Use the Following IP Address').

Figure 24-16: Settings Tab for MLP over T1

5. Click the **PPP** tab.

Figure 24-17: PPP Tab for MLP over T1 WAN



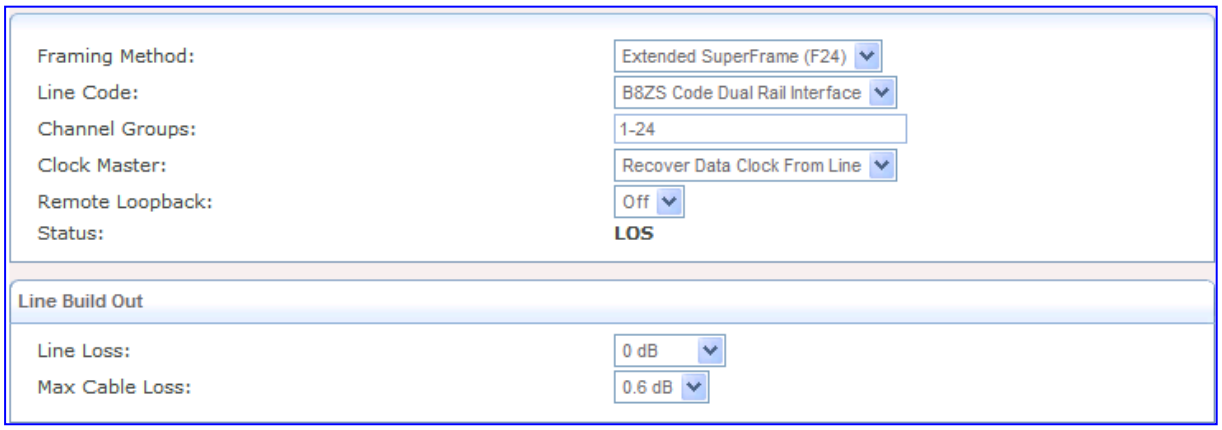
a. Select the supported PPP authentication and encryption methods. (You can also change the PPP login username and password.)

b. Configure the MLP parameters:

- Bundle ID: the logical bundle identifier
- MRRU: the Maximum Reconstructed Receive Unit value
- Use Fragmented Mode: whether to use a fragmented mode:
  - ✓ Yes: each packet is fragmented per the bandwidth ratio of the physical links in the bundle or non fragmented mode
  - ✓ No: each packet is sent as a whole on a single link while alternating between physical T1 links in the bundle

6. Click the **T1** tab, and then define the following as instructed by your ISP:
   - Framing Method
   - Line Code
   - Channel Groups: define the active T1 data channels. This can be Full T1 (1.e., 1-24) or Fractional T1 (e.g., 1-3,5,8-12,24)
   - Clock Master
   - Line Build Out - pulse shape of the T1 analog interface:
     - Line Loss (pulse shape of the T1 analog interface): Line CI code, as defined by ANSI T1.403 Annex H
     - Max Cable Loss: maximum customer cable loss, as defined by ANSI T1.403 Annex H

Figure 24-18: T1 Tab



The Status field displays the status of the T1 WAN interface:
- "Not available" (wait to sync)
- "Active" (sync)
- "LOS" (Red alarm – loss of signal)
- "LOF" (Red alarm – OOF)
- "AIS"
- "RAI" (Yellow alarm)

24.3 Configuring SHDSL WAN Interface

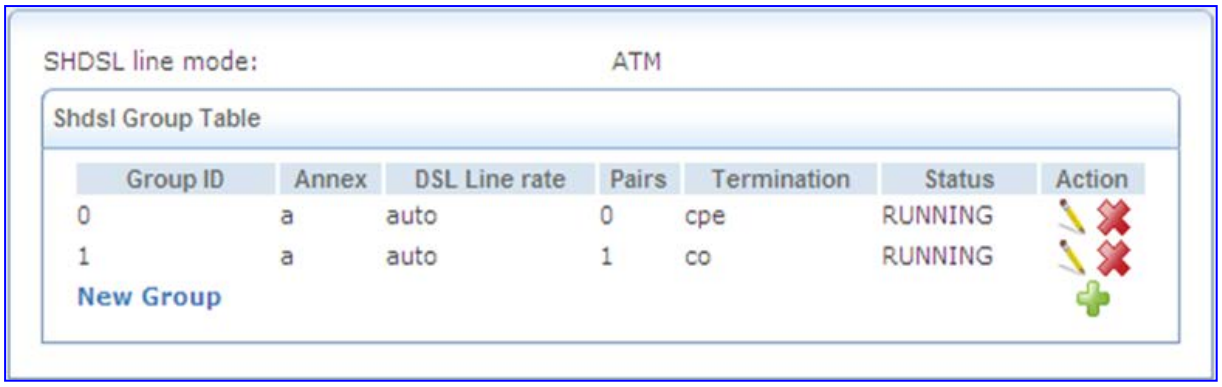
The SHDSL WAN interface can be set up using a variety of protocol configurations. The WAN connection must contain a physical interface specification (SHDSL pairs and grouping information) and one or more Layer-2 protocols (PPPoE, PPPoA, RFC 2684 ETHoA, or IPoA). This configuration must match the network setup provided by your Internet Service Provider (ISP).

Multiple wire-pairs are bonded into a single broadband access link using G.991.2 multiple-pair (also known as "m-pair") technology when the transmission control is ATM, or by 802.3ah PMD Aggregation Function (PAF) when using Ethernet in the First Mile (EFM).

➢ **To configure the SHDSL WAN interface:**

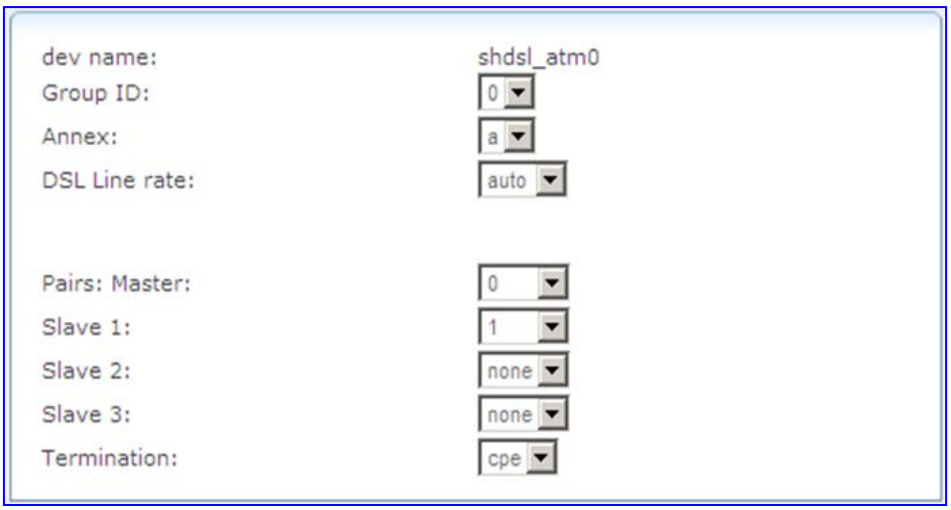
1. Obtain the connection information from your ISP, including the following data:
   - Transmission control type (ATM or EFM)
   - Line rate (automatic or specific range)
   - Regional annex (A/F or B/G)
   - Number of copper wire-pairs used and the order (i.e., one pair is the master, the others are slaves)
   - VPI and VCI values
   - IP encapsulation (PPPoE, PPPoA, ETHoA, IPoA) and variant (LLC-SNAP or VC multiplexing)
   - ATM service class information (CBR/VBR/UBR and data rates)
2. Open the SHDSL Line Settings page (**Configuration** tab > **Data** menu > **WAN Access** > **SHDSL Line Settings**).
3. Select ATM or EFM mode of operation. Note that any change to this setting erases all SHDSL configuration; after switching modes, save the configuration to flash and reset the device.

Figure 24-19: SHDSL Line Mode Page



4. Click **New Group** to add a new SHDSL wire.

Figure 24-20: Adding a New Group



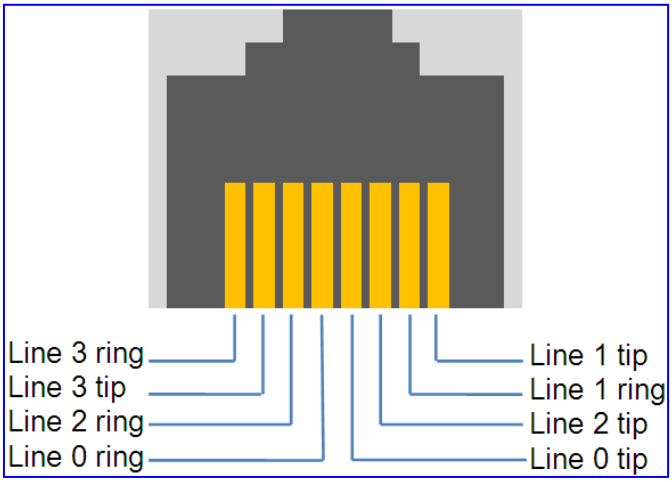
Note: When using EFM, only one wire-pair group can be defined.

5. Configure the annex, line rate, and pair numbers as provided by your ISP, and then click **OK**.

Note: Central Office (CO) termination mode is available for diagnosis only; the device cannot be used as a full-featured DSLAM or LTU.

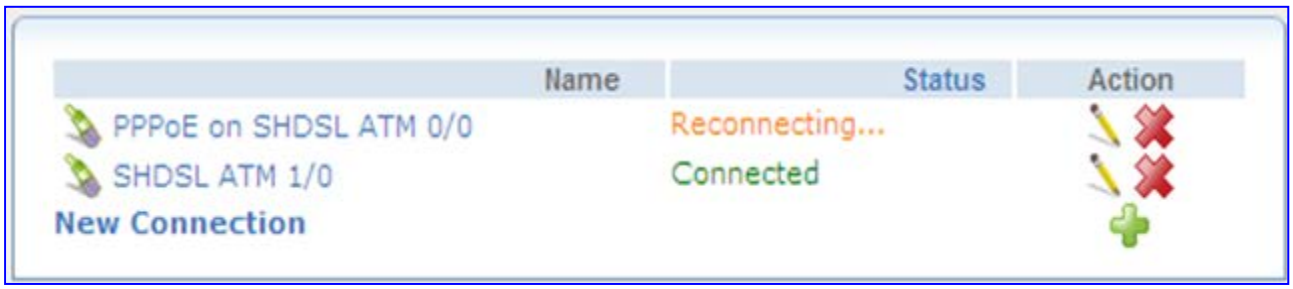
6. Connect the SHDSL cabling as required. SHDSL wire pairs are numbered 0 to 3, according to the following RJ-45 connector pinouts:

Figure 24-21: SHDSL RJ-45 Wire Pinouts



7. Click the **SHDSL Line Settings** link to refresh the status display and wait for the line to synchronize. Ensure that the group status displays "RUNNING" before proceeding to the next step.
8. If EFM mode was selected, skip the following steps and proceed to 'Configuring Ethernet WAN Interface' on page 567.
9. Open the Protocol Interface Settings page (**Configuration** tab > **Data** menu > **WAN Access** > **Protocol Interface Settings**); the current ATM interface list is displayed.

Figure 24-22: Protocol Interface Settings Page

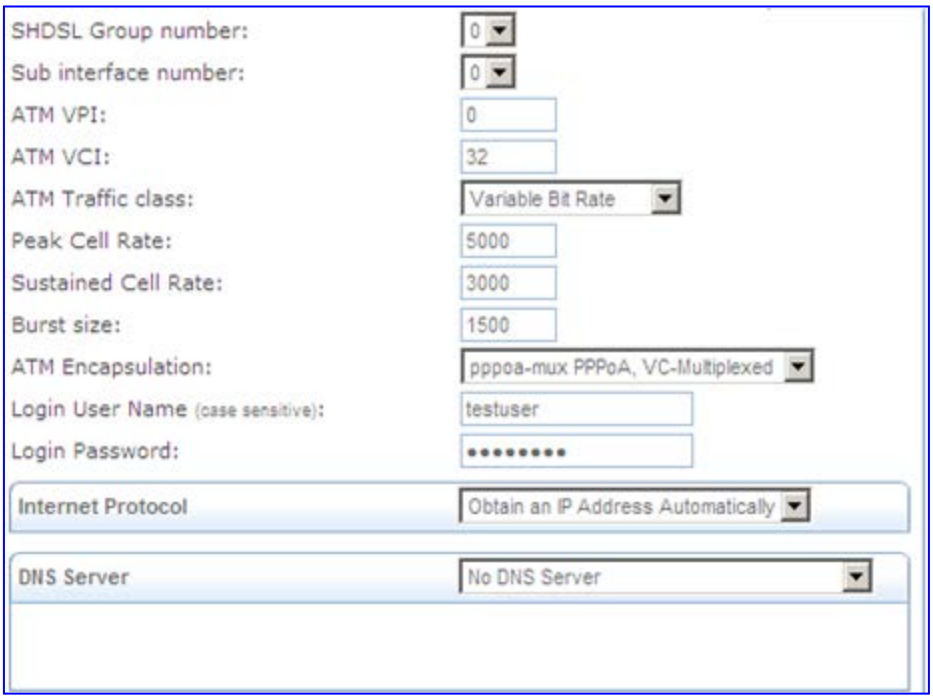


10. Click **New Connection**, select the Internet Connection option, and then click **Next**.

Figure 24-23: Choosing Internet Connection Type

11. Select the required IP encapsulation method, and then click **Next**.

Figure 24-24: Configuring Internet Connection



a. Select a unique sub-interface number for the new connection.
b. Configure the VPI, VCI, encapsulation variant and class-of-service parameters as provided by your ISP. Note that the VPI/VCI combination must be unique in an SHDSL group.
c. If required by your ISP, configure the IP addressing parameters (IP address, network mask, DNS server information); otherwise, use the default setting "Obtain an IP address automatically".
d. For PPPoE and PPPoA links, enter the user name and password for the connection.
e. Click **Next** and then **Finish** to complete creating the connection.

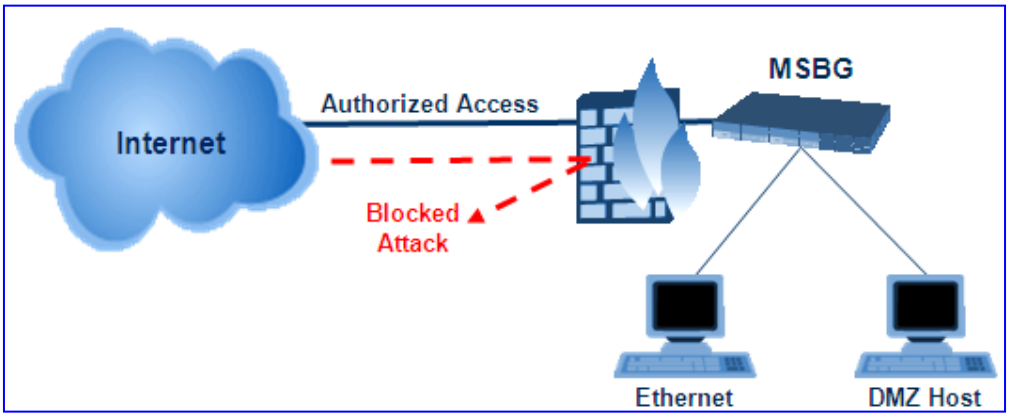
12. Repeat as necessary for any additional ATM interfaces.Reader's Notes

Reader's Notes

25 Firewall and ACL

The **Firewall and ACL** menu allows you to configure various security applications. The device's security suite includes comprehensive and robust security services: Stateful Packet Inspection Firewall, user authentication protocols and password protection mechanisms. These features together allow users to connect their computers to the Internet and simultaneously be protected from the security threats of the Internet. The device's firewall has been pre-configured to provide optimum security (see the figure below).

Figure 25-1: Device's Firewall (Example)



The device's firewall provides both security and flexibility. It provides a managed, professional level of network security while enabling the safe use of interactive applications, such as Internet gaming and video-conferencing. Additional features, including surfing restrictions and access control, can also be easily configured locally by the user through a user-friendly Web-based interface, or remotely by a service provider. The firewall supports advanced filtering, designed to allow comprehensive control over the firewall's behavior. You can define specific input and output rules, control the order of logically similar sets of rules and make a distinction between rules that apply to WAN and LAN network devices.

The **Firewall and ACL** menu includes the following items:

- **General Security:** allows you to choose the security level for the firewall (see 'Configuring General Security Settings' on page 582).
- **LAN Restrictions:** allows you to restrict access from the device's LAN network to the Internet (see 'Configuring LAN Restrictions' on page 583).
- **Port Forwarding:** allows you to enable access from the Internet to specified services provided by computers in the network and special Internet applications (see 'Configuring Port Forwarding' on page 584).
- **DMZ Host:** allows you to configure a LAN host to receive all traffic arriving at your device, which does not belong to a known session (see 'Configuring DMZ Host' on page 586).
- **Port Triggering:** allows you to define port triggering entries to dynamically open the firewall for specific protocols or ports (see 'Configuring Port Triggering' on page 586).
- **Web Restrictions:** allows you to block LAN access to specified hosts or Web sites on the Internet (see 'Configuring Website Restrictions' on page 588).
- **NAT:** allows you to manually control the translation of network addresses and ports (see 'Configuring NAT' on page 589).
- **Access Lists:** allows you to define firewall settings and rules (see 'Configuring the Access List' on page 591).
- **Advanced Filtering:** allows you to assign Access List rules to the device's LAN/WAN interfaces (see 'Configuring Advanced Filtering' on page 594).

25.1 Configuring General Security Settings

The **General Security** item allows you to easily configure the device's basic security settings. The firewall regulates the flow of data between the enterprise's network and the Internet. Both incoming and outgoing data are inspected and then either accepted (allowed to pass through) or rejected (barred from passing through) according to the configurable set of rules.

The firewall rules specify what types of services available on the Internet may be accessed from the enterprise's network and what types of services available in the enterprise's network may be accessed from the Internet. Each request for a service that the firewall receives, whether originating in the Internet or from a computer in the enterprise's network is checked against the set of firewall rules to determine whether the request should be allowed to pass through the firewall. If the request is permitted to pass, then all subsequent data associated with this request (a "session") is also allowed to pass, regardless of its direction.

For example, when you point your Web browser to a Web page on the Internet, a request is sent out to the Internet for this page. The device's firewall identifies the request type and origin—HTTP and a specific PC in your enterprise's network, in this case. Unless you have configured access control to block requests of this type from this computer, the firewall allows this request to pass out onto the Internet. When the Web page is returned from the Web server the firewall associates it with this session and allows it to pass, regardless of whether HTTP access from the Internet to the enterprise's network is blocked or permitted. Therefore, it is the origin of the request, not subsequent responses to this request that determines whether a session can be established or not.

These services include Telnet, HTTP, HTTPS, DNS, IMAP, POP3 and SMTP. The list of allowed services at 'Maximum Security' mode can be edited in the Access Control page. Note that some applications (such as some Internet messengers and Peer-To-Peer client applications) tend to use these ports if they cannot connect with their own default ports. When applying this behavior, these applications will not be blocked outbound, even at Maximum Security Level.

➢ **To configure basic security:**

1. Click the **General Security** item (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **General Security**); the following page appears:

Figure 25-2: Configuring General Security

2. Select one of the pre-defined security levels.

Note: Selecting the 'Minimum Security' option may expose the enterprise's network to significant security risks, and therefore, should only be used if necessary.

3. Select the 'Block IP Fragments' check box to protect your network from a common type of hacker attack that could make use of fragmented data packets to sabotage your network. Note that VPN over IPSec and some UDP-based services make legitimate use of IP fragments. Therefore, you need to allow IP fragments to pass into the enterprise's network to make use of these select services.
4. Click **OK** to save your settings.

25.2 Configuring LAN Restrictions

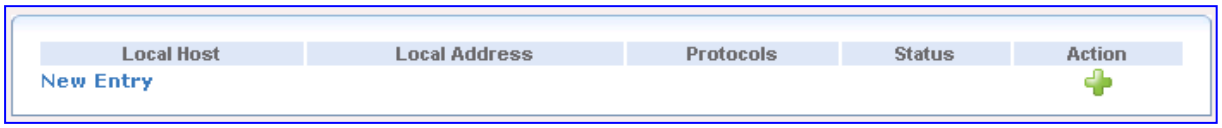
The **LAN Restrictions** item allows you to define restriction rules on the types of requests that may pass from the LAN network to the Internet, and thus, may block traffic flowing in both directions. It can also be used for allowing specific services when maximum security is configured. You may want to block specific computers within the home network (or even the whole network) from accessing certain services on the Internet. For example, you may want to prohibit one computer from surfing the Web and the whole network from receiving incoming e-mail (by blocking their outgoing requests to POP3 servers on the Internet). There are numerous services you should consider blocking, such as popular game and file sharing servers. For example, if you want to make sure that your employees do not put your business at risk from illegally traded copyright files, you may want to block several popular P2P and file sharing applications.

Note: When Web Filtering is enabled, HTTP services cannot be blocked by Access Control.

➢ **To configure LAN restrictions rule:**

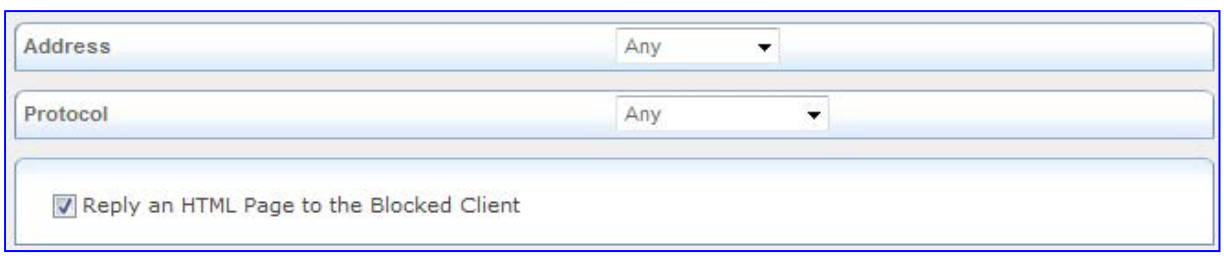
1. Click the **LAN Restrictions** item (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **LAN Restrictions**); the following page appears:

Figure 25-3: Configuring LAN Restriction Rules



2. Click the **New** icon; the following page appears:

Figure 25-4: Adding an Access Control Rule



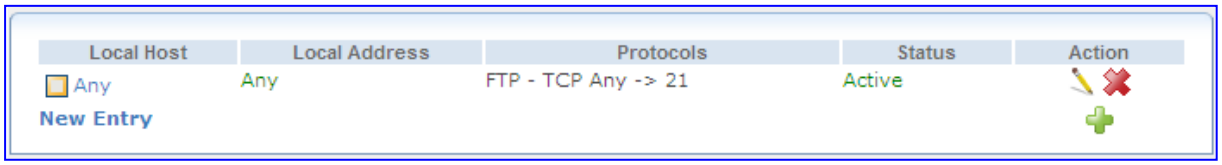
3. From the 'Address' drop-down list, specify the computer or group of computers on which you would like to apply the access control rule. Select an address or a name from the list to apply the rule on the corresponding host, or **Any** to apply the rule on all the device's LAN hosts. If you want to add a new address, select **User Defined**, and then follow the sequence to add a new Network Object, representing the new host (see 'Configuring Network Objects' on page 630).
4. From the 'Protocol' drop-down list, select or specify the type of protocol. Selecting **Show All Services** expands the list of available protocols. Select a protocol or add a new one by selecting **User Defined**, and then follow the sequence to add a new Service representing the protocol (see 'Configuring Protocols' on page 629).
5. Select the 'Reply an HTML Page to the Blocked Client' check box to display the following message to the client: "Access Denied – this computer is not allowed to surf the WAN. Please contact your admin.". When this check box is cleared, the client's packets are simply ignored and no notification is issued.
6. Click **OK** to save your changes; the LAN restriction rule is displayed in the LAN Restriction list.

You can disable a LAN restriction rule to make a service available without having to delete the rule. This may be useful if you wish to make the service temporarily available and expect to reinstate the restriction in the future.

➢ **To disable a LAN restriction rule:**

- In the LAN Restriction list, clear the check box corresponding to the rule that you want to disable.

Figure 25-5: Disabled LAN Restrictions - Cleared Check Box



25.3 Configuring Port Forwarding

By default, the device blocks all external users from connecting to or communicating with your network. Therefore, the system is safe from hackers who may try to intrude on the network and damage it. However, you may want to expose your network to the Internet in certain limited and controlled ways to enable some applications to work from the LAN (game, voice and chat applications, for example) and to enable Internet-access to servers in the home network. The Port Forwarding feature supports both of these functionalities.

The **Port Forwarding** item enables you to define the applications that require special handling by the device. This allows you to select the application's protocol (or add a new protocol) and the local IP address of the computer that will be using or providing the service.

Similarly, you can grant Internet users access to servers inside your home network, by identifying each service and the PC that will provide it. This is useful, for example, if you want to host a Web server inside your home network. When an Internet user points a browser to the device's WAN IP address, the device forwards the incoming HTTP request to your Web server.

Note that as the device has one external (WAN) IP address, different applications can be assigned to your LAN computers, however each type of application is limited to one computer (IP address). Therefore, to overcome this, you can add additional public IP addresses to port forwarding rules, which you must first obtain from your ISP, and enter into the 'NAT IP Addresses Pool' (see 'NAT' on page 589).

Additionally, port forwarding enables you to redirect traffic to a different port instead of the one to which it was designated. For example, you have a Web server running on your PC

on port 8080 and you want to grant access to this server to anyone who accesses the device via HTTP. To accomplish this, you have to define a port forwarding rule for the HTTP service, with the PC's IP or host name, as well as specify 8080 in the 'Forward to Port' field. All incoming HTTP traffic is now forwarded to the PC running the Web server on port 8080.

When setting a port forwarding service, you must ensure that the port is not already in use by another application, which may stop functioning. A common example is when using SIP signaling in Voice over IP—the port used by the device's VoIP application (5060) is the same port on which port forwarding is set for LAN SIP agents.

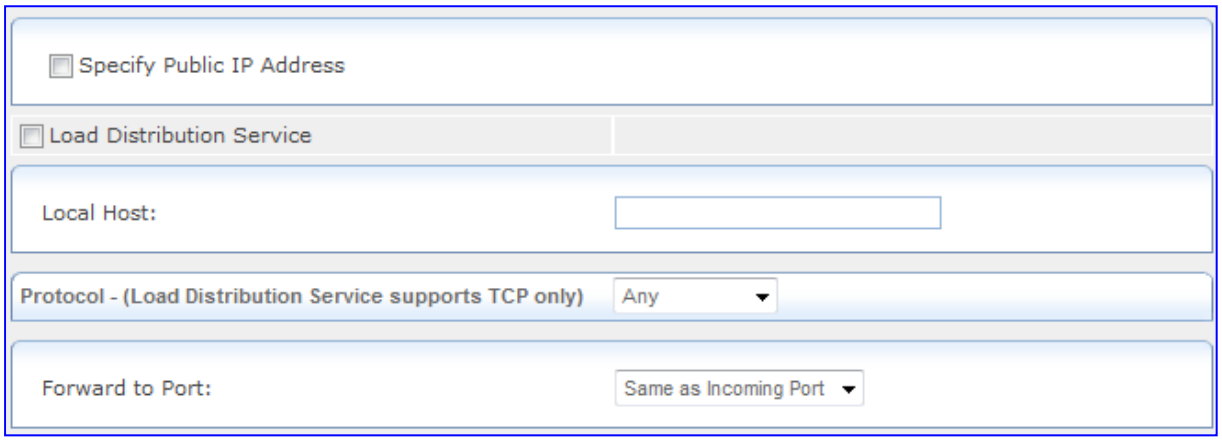
➢ **To configure a port forwarding service:**

1. Click the **Port Forwarding** item (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **Port Forwarding**); the following page appears:

Figure 25-6: Configuring Port Forwarding

2. Click the **New Entry** link; the following page appears:

Figure 25-7: Adding Port Forwarding Rule



3. Select the 'Specify Public IP Address' check box if you want to apply this rule on the device's non-default IP address, defined in the 'NAT page (see 'Configuring NAT' on page 589) and then in the 'Public IP Address' field, enter the additional external IP address.
4. In the 'Local Host' field, enter the host name or IP address of the computer that will provide the service (the "server"). Note that unless an additional external IP address has been added, only one LAN computer can be assigned to provide a specific service or application.
5. From the 'Protocol' drop-down list, select **Any** or specify the type of protocol. Add a new one by selecting **User Defined** and then add a new Service representing the protocol (see 'Configuring Protocols' on page 629).
6. From the 'Forward to Port' drop-down list, select **Specify** and define a port to redirect traffic to a different port. By default, the device forwards traffic to the same port as the incoming port.
7. Click **OK** to save your changes; the main Port Forwarding page displays a summary of the rule that you added.

25.4 Configuring DMZ Host

The DMZ (Demilitarized) Host feature allows a single local computer to be exposed to the Internet. You can designate a DMZ host for the following scenario examples:

- You wish to use a special-purpose Internet service, such as an on-line game or video conferencing program that is not present in the Port Forwarding list and for which no port range information is available.
- You are not concerned with security and wish to expose one computer to all services without restriction.

Warning: A DMZ host is not protected by the firewall and may be vulnerable to attack. Designating a DMZ host may also put other computers in the home network at risk. When designating a DMZ host, you must consider the security implications and protect it if necessary.

For an incoming request for access to a service in the home network, such as a Web server, the device forwards this request to the DMZ host (if one is designated) unless the service is being provided by another PC in the home network (assigned in Port Forwarding), in which case that PC receives the request instead.

➢ **To designate a local computer as a DMZ Host:**

1. Click the **DMZ Host** item (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **DMZ Host**); the following page appears:

Figure 25-8: Defining a DMZ Host

2. Select the check box, and then enter the local IP address of the computer that you want to designate as a DMZ host. Note that only one LAN computer may be a DMZ host at any time.
3. Click **OK** to save the settings.

25.5 Configuring Port Triggering

Port triggering can be used for dynamic port forwarding configuration. By setting port triggering rules, you can allow inbound traffic to arrive at a specific LAN host, using ports different than those used for the outbound traffic. This is called port triggering since the outbound traffic triggers to which ports inbound traffic is directed. For example, consider a gaming server that is accessed using UDP protocol on port 2222. The gaming server responds by connecting the user using UDP on port 3333 when starting gaming sessions. In such a case, you must use port triggering, since this scenario conflicts with the following default firewall settings:

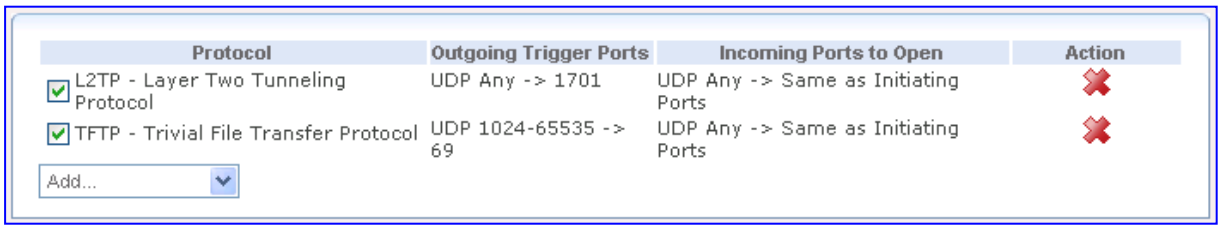
- The firewall blocks inbound traffic, by default.
- The server replies to the device's IP address, and the connection is not sent back to your host, since it is not part of a session.

To solve this you need to define a Port Triggering entry, which allows inbound traffic on UDP port 3333, only after a LAN host generated traffic to UDP port 2222. This will result in accepting the inbound traffic from the gaming server and sending it back to the LAN Host which originated the outgoing traffic to UDP port 2222.

➢ **To configure port triggering:**

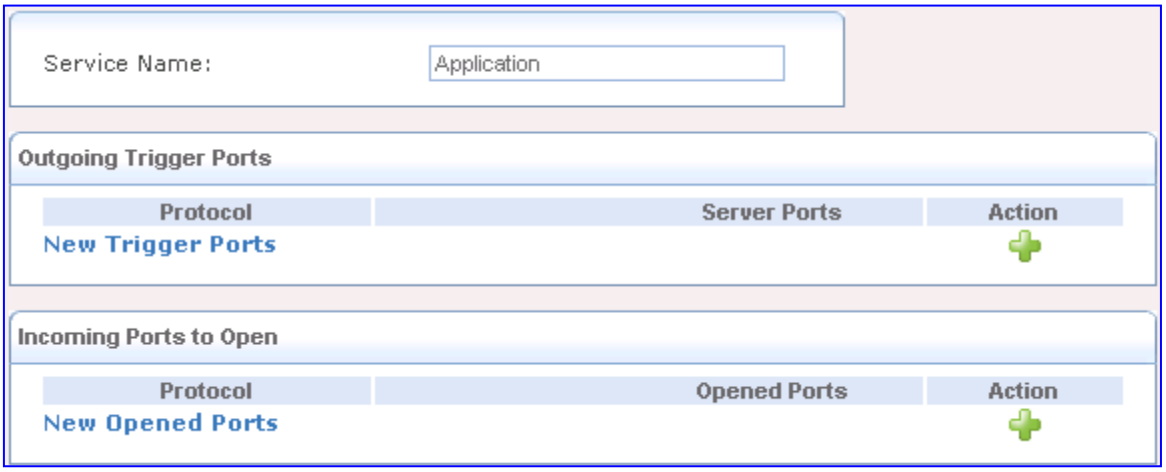
1. Click the **Port Triggering** item (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **Port Triggering**); the following page appears:

Figure 25-9: Configuring Port Triggering



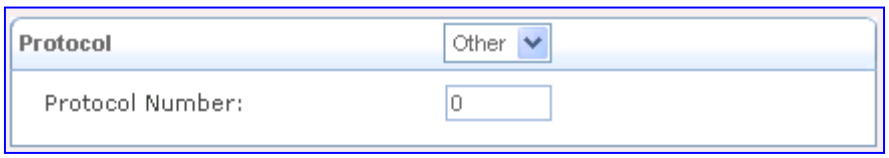
2. From the drop-down list, you can select a pre-configured service by selecting **Show All Services**, and then from the refreshed drop-down list, selecting a service. If you want to define your own service, select **User-Defined**; the following page appears:

Figure 25-10: Editing Port Triggering Rule



3. In the 'Service Name', enter a name for the service (e.g. "game_server"), and then under the 'Outgoing Trigger Ports' group, click the **New Trigger Ports** link; the following page appears:

Figure 25-11: Defining Trigger Ports



4. From the 'Protocol' drop-down lists, select the protocol (e.g., 'UDP'); the 'Source Ports' and 'Destination Ports' fields appear.
5. Leave the 'Source Ports' value at its default 'Any'.
6. In the 'Destination Ports' drop-down list, select **Single**; a field for entering the destination port appears (enter the destination port, e.g., "2222").
7. Click **OK** to save the settings.
8. Under the 'Incoming Ports to Open' group, click the **New Opened Ports** link, and then configure the incoming ports by repeating steps 4 through 7, but entering values for incoming ports.

25.6 Configuring Website Restrictions

You can configure the device to block specific Internet Web sites so that they cannot be accessed from computers in the home network. Moreover, restrictions can be applied to a comprehensive and automatically-updated table of sites to which access is not recommended.

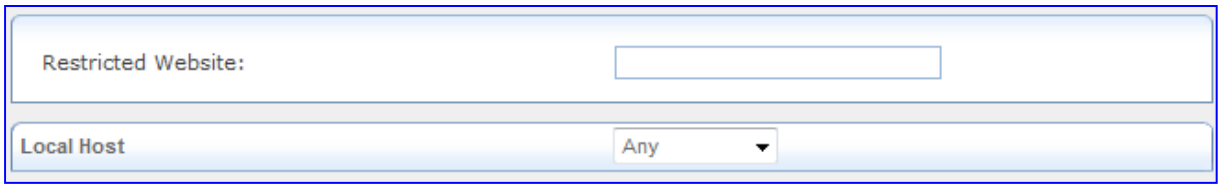
➢ **To block access to a web site:**

1. Click the **Web Restrictions** item (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **Web Restrictions**); the following page appears:

Figure 25-12: Configuring Website Restrictions

2. Click the **New Entry** link; the following page appears:

Figure 25-13: Adding a Restricted Website



3. In the 'Restricted Website' field, enter the URL (or part of the URL) that you want to make inaccessible from your home network (all Web pages within this URL are also be blocked). If the URL has multiple IP addresses, the device resolves all additional addresses and automatically adds them to the restrictions table.
4. From the 'Local Host' drop-down list, specify the computer or group of computers to which you want to apply the Web site restriction. Select an address or a name from the list to apply the rule on the corresponding host, or 'Any' to apply the rule on all the device's LAN hosts. If you want to add a new address, select **User Defined** and add a new Network Object representing the new host (see 'Configuring Network Objects' on page 630).
5. Click **OK** to save the settings; you are returned to the previous page while the device attempts to find the site. 'Resolving...' appears in the Status column while the site is being located (the URL is 'resolved' into one or more IP addresses).
6. Click **Refresh** to update the status if necessary. If the site is successfully located, then 'Resolved' appears in the status bar, otherwise 'Hostname Resolution Failed' appears.

25.7 Configuring NAT

The device features a configurable Network Address Translation (NAT) and Network Address Port Translation (NAPT) mechanism, allowing you to control the network addresses and ports of packets routed through the device. When enabling multiple computers on your network to access the Internet using a fixed number of public IP addresses, you can define static NAT/NATP rules which map (translate) LAN IP addresses (LAN computers) to NAT IP addresses and/or ports.

By default, the device operates in NAPT routing mode (see 'Configuring Network Connections' on page 633). The NAT/NAPT mechanism is useful for managing Internet usage in your LAN, or complying with various application demands. For example, you can assign your primary LAN computer with a single NAT IP address to assure its permanent connection to the Internet. Another example is when an application server with which you wish to connect such as a security server, requires that packets have a specific IP address – you can define a NAT rule for that address.

For example, if you have LAN IP addresses of 192.168.1.10 to 192.168.15 (i.e., six PC's), and you have obtained from your ISP the NAT IP addresses 192.168.71.12 through 192.168.71.15 (i.e., four NAT addresses), you can map the six LAN IP addresses to the four NAT IP addresses. This would mean that only four of the six LAN computers may have WAN access at the same time. You can also ensure that a computer always has access to the Internet, by defining a new rule mapping its LAN IP address to one of the NAT IP address (and excluding this NAT IP address from the second rule for the other computers).

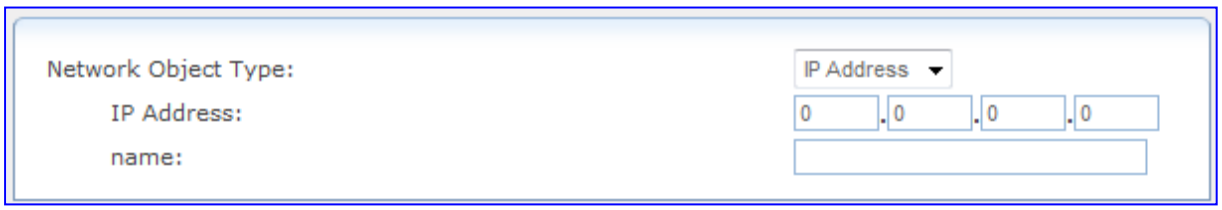
➢ **To configure NAT:**

1. Click the **NAT** item (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **NAT**); the following page appears:

Figure 25-14: Configuring NAT

2. Define additional public IP addresses obtained from your ISP as your NAT IP addresses (the primary IP address used by the WAN device for dynamic NAPT must not be added):
   a. Under the 'NAT IP Address Pool' group, click the **New IP Pool** link; the following page appears:

Figure 25-15: Defining Public IP Address

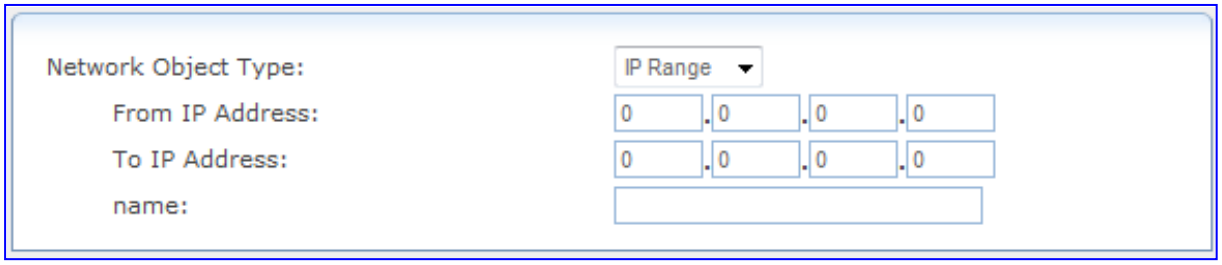


   b. From the 'Network Object Type' drop-down list, select between 'IP Address', 'IP Subnet' or 'IP Range', and then enter the information respectively.

c. Click **OK** to save the settings; you are returned to the main page.

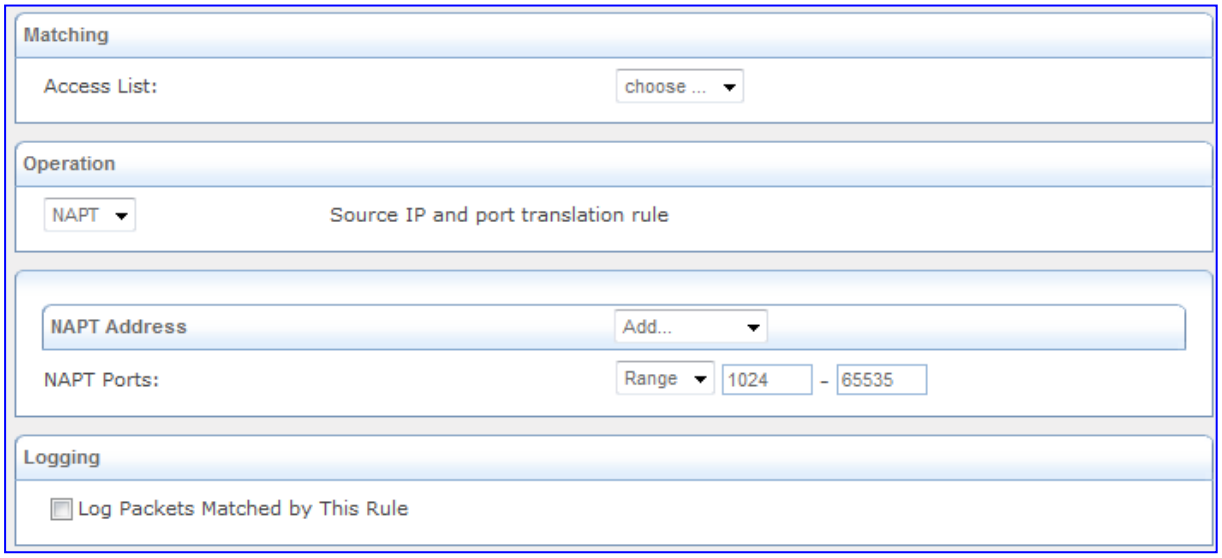
3. Defines a NAT LAN IP address pool:
   a. Under the 'NAT LAN IP Address Pool' group, click the **New IP Pool** link; the following page appears:

Figure 25-16: NAT LAN IP Address Pool



   b. From the 'Network Object Type' drop-down list, select between 'IP Range' or 'IP Subnet', and then enter the information respectively.
   c. Click **OK** to save the settings; you are returned to the main page.
4. Define a new NAT/NAPT rule:
   a. Under the 'NAT/NAPT Rule Sets' group, click the **New Entry** link; the following page appears:

Figure 25-17: Defining NAT/NAPT Rule



This page is divided into two main groups: 'Matching' and 'Operation'. The Matching' group defines the LAN addresses to be translated to the external addresses, which are defined in the 'Operation' group.

5. Configure the 'Matching' group parameters to define characteristics of the packets matching the rule.
   a. Access list: choose the matching Access List rule, if configured (see 'Configuring the Access List' on page 591).
6. Configure the 'Operation' group parameters to define the operation that will be applied to the IP addresses matching the criteria defined above. The operations available are NAT or NAPT:
   - **NAT:** The NAT address into which the original IP address is translated. The drop-down list displays all of your available NAT addresses/ranges, from which you can select an entry. If you want to add a single address or a sub-range from the given pool/range, select **User Defined** to add a new Network Object representing the new host (see 'Configuring Network Objects' on page 630).

- **NAPT:** The NAPT address into which the original IP address is translated. The drop-down list displays all of your available NAPT addresses/ranges, from which you can select an entry. If you want to add a single address or a sub-range from the given pool/range, select **User Defined** to add a new Network Object representing the new host (see 'Configuring Network Objects' on page 630). Enter a single port or select **Range** to enter a range of ports.

7. Select the 'Log Packets Matched by This Rule' check box to log the first packet from a connection that was matched by this rule.

25.8 Configuring the Access List

The Access list is designed to allow comprehensive control over the firewall's behavior. You can define specific set of rules for ingress (inbound) and egress (outbound) traffic and control the order of logically similar sets of rules. These rules can later be assigned to the device's LAN and/or WAN interfaces (see 'Configuring Advanced Filtering').

Notes:

- Only one Access List group can be assigned to an interface. Therefore, ensure that your Access List group includes all the required rules that you want to later assign to a specific interface.
- You can enable the reporting (through Syslog messages) of Access List rules that are set to "log", whenever traffic matching the Access List is encountered. This is enabled using the *ini* file parameter, EnableSecSyslog.

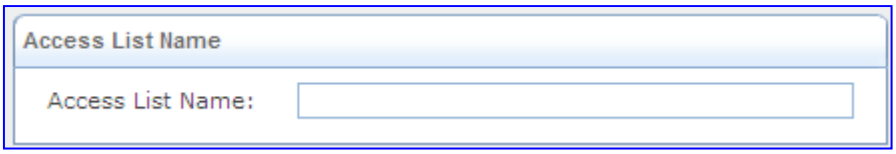
➢ **To configure Access List rules:**

1. Open the **Access Lists** table (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **Access List**); the following appears:

Figure 25-18: Access Lists Table

2. Add a new Access List group name:
   a. Click the **New ACL** link; the 'Access List Name page appears.

Figure 25-19: Defining Access List Name



   b. In the 'Access List Name' field, enter a name for the Access List rule group, and then click **OK**; the Access Lists table re-appears, displaying the newly added Access List rule group.
3. Add rules to the Access List group:

a. Under the Access List name that you defined (in Step 2), click the **New Entry** link; the following page appears for defining a rule:

Figure 25-20: Adding an Access List Rules

The Matching' and 'Operation' groups define the operation to be executed when matching conditions apply.

b. Configure the 'Matching' parameters to define characteristics of the packets matching the rule:

- **Source Address:** Specify the source address (i.e., computers) of packets sent or received by the device. Select an address or a name from the list to apply the rule on the corresponding host, or **Any** to apply the rule on all the device's LAN hosts. If you want to add a new address, select **User Defined** to add a new Network Object representing the new host (see 'Configuring Network Objects' on page 630).
- **Destination Address:** Destination address of packets sent or received by the device. This address can be configured in the same manner as the source address.
- **Protocol:** Specify a traffic protocol. Selecting **Show All Services** expands the list of available protocols. Select a protocol or add a new one by selecting **User Defined** to add a new Service representing the protocol (see 'Configuring Protocols' on page 629).
- **DSCP:** Select this check box to display two DSCP fields, which enable you to specify a hexadecimal DSCP value and its mask assigned to the packets matching the priority rule.
- **Priority:** Select this check box to display a drop-down list in which you can select a priority level assigned to the packets matching the priority rule.

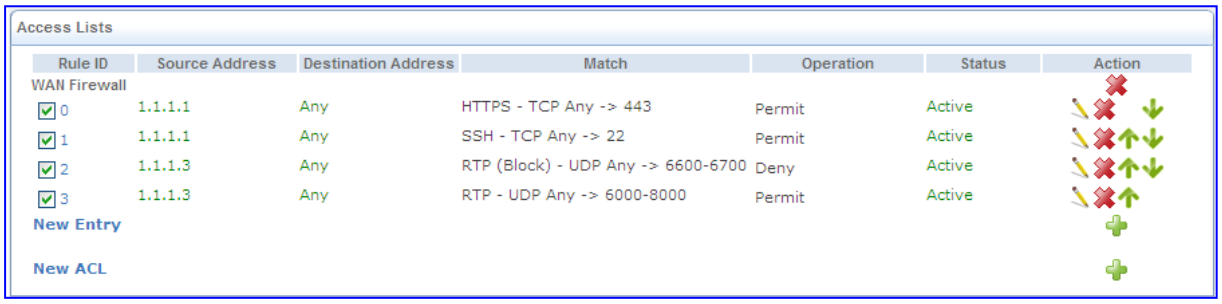
c. Configure the 'Operation' parameters to define the action that the rule performs:

- **Permit Established:** Allow access to packets that match the criteria defined. The data transfer session is handled using Stateful Packet Inspection (SPI), meaning that other packets matching this rule are automatically allowed access.
- **Permit:** Allow access to packets that match the criteria defined, without keeping track of the data transfer session state.
- **Deny:** Deny access to packets that match the source and destination IP addresses and service ports defined above.

d. Select the 'Log Packets Matched by This Rule' check box to log the first packet from a connection that was matched by this rule.

4. Click **OK** to save your changes; the Access Lists table re-appears, displaying the defined rules under the Access List Rule ID:

Figure 25-21: Added Access List Rules



The order of the rules appearing under a specific Access List name represents both the order in which they are defined and the sequence by which they are applied. You can change this order, by using the **Move Up** and **Move Down** icons.

5. Assign the Access List rule ID to the required LAN or WAN interface (see 'Configuring Advanced Filtering').

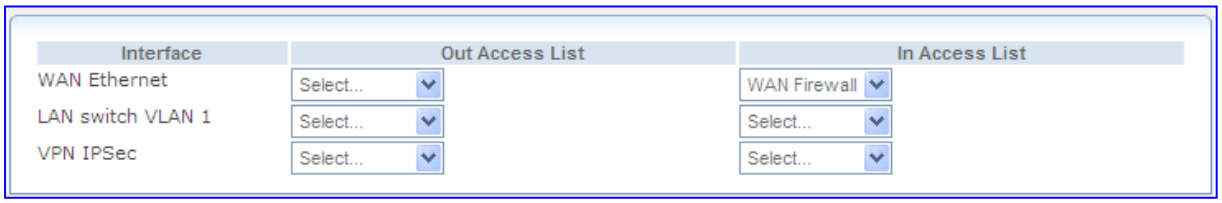
25.9 Configuring Advanced Filtering

The Advanced Filtering allows you to assign Access List rules (defined in 'Configuring the Access List' on page 591) to the device's LAN and/or WAN interfaces.

➢ **To assign Access List rules to the device's LAN/WAN interfaces:**

1. Click the **Advanced Filtering** item (**Configuration** tab > **Data** menu > **Firewall and ACL** submenu > **Advanced Filtering**); the following page appears listing the device's LAN/WAN interfaces and a corresponding drop-down list containing the defined Access list rules (in 'Configuring the Access List' on page 591):

Figure 25-22: Advanced Filtering



2. From the drop-down list corresponding to the interface to which you want to apply an Access List rules, select the required Access List group name.
3. Click **OK**.

26 QoS

The device's Quality of Service (QoS) provides the capability to provide better service to selected network traffic. This is achieved by shaping the traffic and processing higher priority traffic before lower priority traffic.

The **QoS** menu provides the following items:

- **General QoS** (see 'Configuring General QoS Settings' on page 595)
- **Match Rules** (see 'Configuring Matching Rules' on page 596)
- **Traffic Shaping** (see 'Configuring Traffic Shaping' on page 599)
- **DSCP Settings** (see 'Configuring DSCP Settings' on page 602)
- **802.1p Settings** (see 'Configuring 802.1p Settings' on page 604)

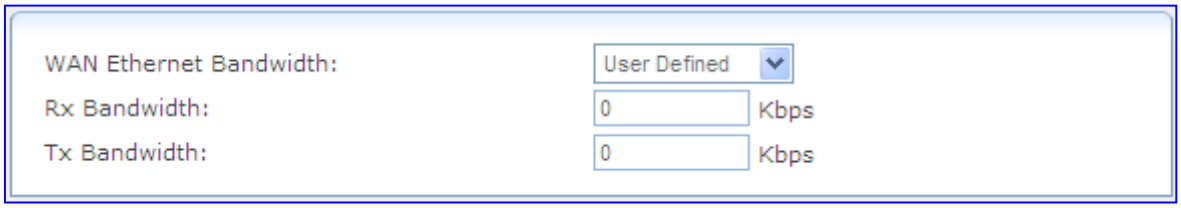
26.1 Configuring General QoS Settings

The **General QoS** item allows you to configure your WAN bandwidth.

➢ **To configure the device's WAN bandwidth:**

1. Click the **General QoS** item (**Configuration** tab > **Data** menu > **QoS** submenu > **General QoS**); the following page appears:

Figure 26-1: Configuring General WAN Bandwidth



2. From the 'WAN Devices Bandwidth (Rx/Tx)' drop-down list, select the required Rx/Tx bandwidth. If you do not see an appropriate entry, select **User Defined**, and then enter your Tx and Rx bandwidths in the fields below:
   - 'Rx Bandwidth': defines the device's Internet traffic receiving rate (in Kbps).
   - 'Tx Bandwidth': defines the device's outbound transmission rate (in Kbps).

Notes:

- For correct QoS performance, ensure that the bandwidth values are correct.
- For T1 WAN interface, the maximum Tx bandwidth per T1 physical link is 1.544 Mbps.

26.2 Configuring Matching Rules

Matching of packet rules allows you to manage and avoid traffic congestion by defining inbound and outbound priority rules for each element on your device. These rules determine the priority assigned to the packets traveling through the element. QoS parameters (DSCP marking and packet priority) are set per packet on an application basis. You can set QoS parameters using flexible rules, according to the following parameters:

- Source/destination IP address, MAC address or host name
- Device
- Source/destination ports
- Limit the rule for specific days and hours

The device supports two priority marking methods for packet prioritization:

- DSCP (see 'Configuring DSCP Settings' on page 602).
- 802.1p Priority (see 'Configuring 802.1p Settings' on page 604).

The matching of packets by rules is connection-based, known as Stateful Packet Inspection (SPI), using the same connection-tracking mechanism used by the device's firewall. Once a packet matches a rule, all subsequent packets with the same attributes receive the same QoS parameters, both inbound and outbound. A packet can match more than one rule. Therefore:

- The first class rule has precedence over all other class rules (scanning is stopped once the first rule is reached).
- The first traffic-priority (classless) rule has precedence over all other traffic-priority rules.
- There is no prevention of a traffic-priority rule conflicting with a class rule. In this case, the priority and DSCP setting of the class rule (if given) takes precedence.

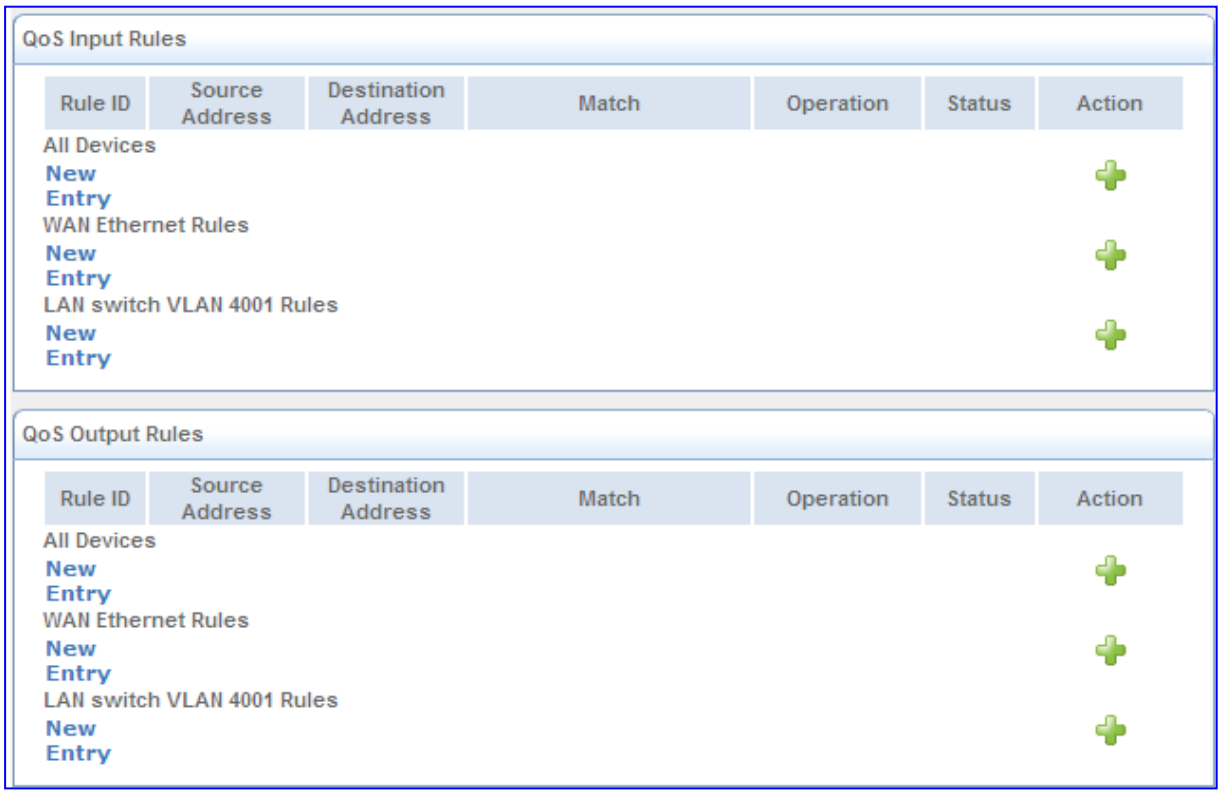
Tx Traffic Shaping classes can be assigned to Matching Rules. These classes are defined in 'Configuring Traffic Shaping' on page 599.

Connection-based QoS also allows inheriting QoS parameters by some of the applications that open subsequent connections. For instance, you can define QoS rules on SIP, and the rules also apply to both control and data ports (even if the data ports are unknown). This feature applies to all applications that have ALG in the firewall such as SIP, MSN Messenger/Windows Messenger, Port Triggering applications (see 'Configuring Port Triggering' on page 586), PPTP, and IPSec.

➢ **To define matching-of-packet rules:**

1. Click the **Match Rules** item (**Configuration** tab > **Data** menu > **QoS** submenu > **Match Rules**); the following page appears:

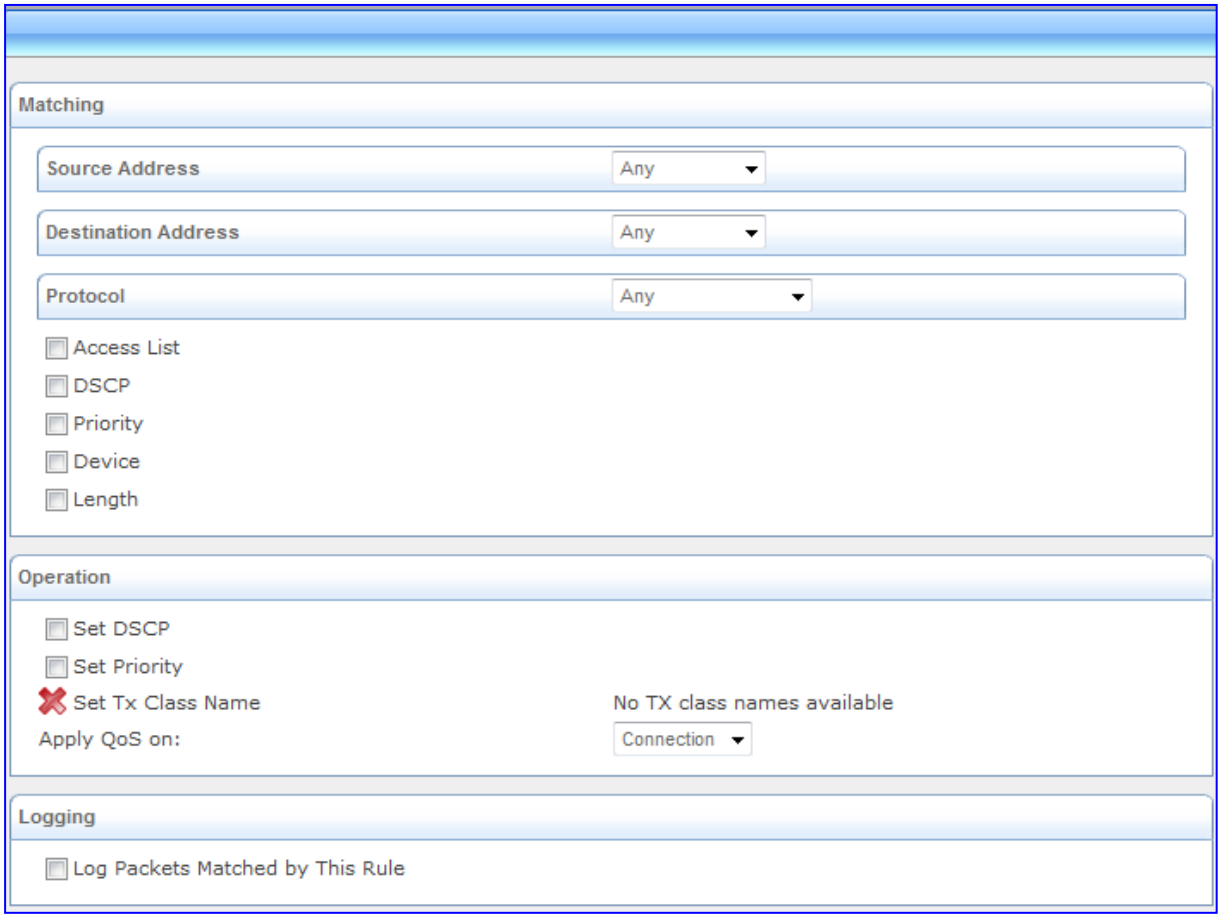
Figure 26-2: Configuring Traffic Priority



This page is organized into two groups - 'QoS Input Rules' and 'QoS Output Rules' - for prioritizing inbound and outbound traffic, respectively. Each group lists all the devices on which rules can be set. You can set rules on all devices using the 'All Devices' group.

2. Click the **New Entry** link corresponding to the traffic direction (i.e., 'QoS Input Rules' or 'QoS Output Rules') and the device on which to set the rule; the following page appears:

Figure 26-3: Adding a Traffic Priority Rule



3. Under the 'Matching' group, define characteristics of the packets matching the QoS rule:
   - **Source Address:** source address of packets sent or received by the device. The drop-down list allows you to specify a computer (address or a name) or group of computers on which you want to apply the rule. Select **Any** to apply the rule to all the device's LAN hosts. To add a new address, select **User Defined** and then add a new Network Object representing the new host (see 'Configuring Network Objects' on page 630).
   - **Destination Address:** destination address of packets sent or received by the device. This address can be configured in the same manner as the source address.
   - **Protocol:** You may also specify a traffic protocol. Selecting **Show All Services** from the drop-down lists expands the list of available protocols. Select a protocol or add a new one by selecting **User Defined** and then add a new Service representing the protocol (see 'Configuring Protocols' on page 629).
   - **Access List:** Select this check box to display a drop-down list from which you can select an Access List group (defined in 'Configuring the Access List' on page 591) to which the packets are assigned.
   - **DSCP:** Select this check box to display two DSCP fields, which enable you to specify a hexadecimal DSCP value and its mask assigned to the packets matching the priority rule.
   - **Priority:** Select this check box to display a drop-down list from which you can select a priority level assigned to the packets matching the priority rule.

- **Device:** Select this check box to display a drop-down list from which you can select a network device on which the packet-rule matching is performed.
- **Length:** Select this check box if you want to specify the length of packets or the length of their data portion.

4. Under the 'Operation' group, define the following operation/s on packets that match the priority rule:
   - **Set DSCP:** Select this check box if you want to change the DSCP value (hexadecimal) on packets matching the rule, prior to routing them further.
   - **Set Priority:** Select this check box if you want to change a priority (where zero is the lowest and seven the highest) of the packets matching the rule. Each priority level is assigned a default queue number, where Queue 0 has the lowest priority. The device's QoS supports up to eight queues. The matching between a priority level and a queue number can be edited in the '802.1p Settings page (see 'Configuring 802.1p Settings' on page 604).
   - **Set Tx Class Name:** Select the check box and then from the drop-down list, select the defined Tx Class.
   - **Apply QoS on:** Select whether to apply QoS on a connection or just the first packet. When applying on a connection, the data transfer session is handled using Stateful Packet Inspection (SPI). This means that other packets matching this rule are automatically allowed to access, and the same QoS scheme is applied to them.
5. Under the 'Logging' group, select the 'Log Packets Matched by This Rule' to log the first packet from a connection that was matched by this rule.
6. Click **OK** to save your changes.

The order of appearance of the rules represents both the order in which they were defined and the sequence by which they are applied. You may change this order, by using the **Move Up** and **Move Down** icons.

26.3 Configuring Traffic Shaping

Traffic shaping allows you to define Tx (transmission) traffic classes for all the LAN and WAN interfaces. These traffic shaping classes can later be assigned to matching packet priority rules (defined in 'Configuring Matching Rules' on page 596).

Traffic Shaping allows you to manage and avoid congestion where a high speed LAN meets limited broadband bandwidth. A user may have, for example, a 100 Mbps Ethernet LAN with a 100 Mbps WAN interface router. The router may communicate with the ISP using a modem with a bandwidth of 2 Mbps. This typical configuration makes the modem, having no QoS module, the bottleneck.

The router sends traffic as fast as it is received, while its well-designed QoS algorithms are left unused. Traffic shaping limits the bandwidth of the router, artificially forcing the router to be the bottleneck. A traffic shaper is essentially a regulated queue that accepts uneven and/or bursty flows of packets and transmits them in a steady, predictable stream so that the network is not overwhelmed with traffic. While Traffic Priority allows basic prioritization of packets, Traffic Shaping provides more sophisticated definitions, such as the following:

- Bandwidth limit for each device
- Bandwidth limit for classes of rules
- Prioritization policy
- TCP serialization on a device

The bandwidth of a device can be divided to reserve constant portions of bandwidth to predefined traffic types. Such a portion is known as a Traffic Class. When not used by its predefined traffic type, or owner (for example VoIP), the bandwidth is available to all other

traffic. However when needed, the entire class is reserved solely for its owner. Moreover, you can limit the maximum bandwidth that a class can use even if the entire bandwidth is available. When a shaping class is first defined for a specific traffic type, two shaping classes are created. The second class is the 'Default Class', which is responsible for all the packets that do not match the defined shaping class, or any other classes that may be defined on the device.

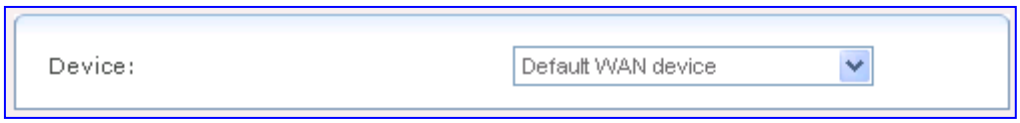
➢ **To configure traffic shaping:**

1. Click the **Traffic Shaping** item (**Configuration** tab > **Data** menu > **QoS** submenu > **Traffic Shaping**); the following page appears:

Figure 26-4: Configuring Traffic Shaping

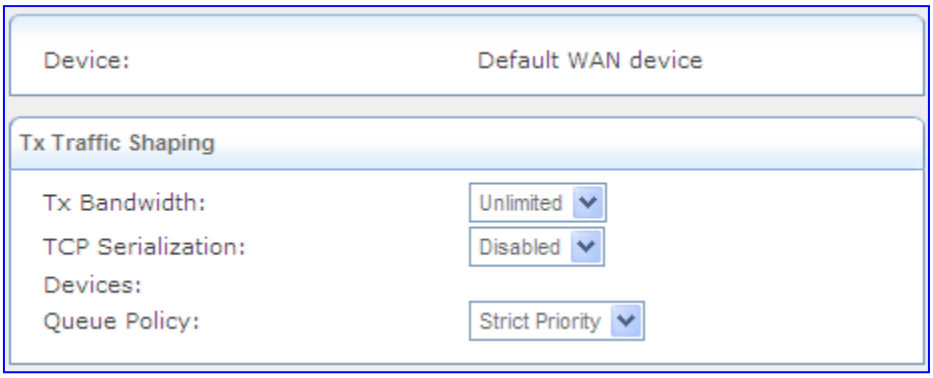
2. Click the **New Entry** link; the following page appears.

Figure 26-5: Adding Device for Traffic Shaping



3. From the 'Device' drop-down list, select the interface for which you want to shape the traffic. If you want to apply the settings on all LAN devices, select **Default LAN Device**.
4. Click **OK**; the following page appears:

Figure 26-6: Defining Device Traffic Shaping



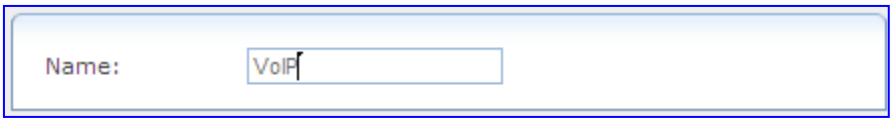
5. From the 'Tx Bandwidth' drop-down list, select the device's bandwidth transmission rate limit. If you want to specify a TX bandwidth, see Step 8.
6. From the 'TCP Serialization' drop-down list, select whether TCP Serialization is enabled or disabled. You can enable TCP Serialization for active voice calls only or for all traffic. If you select **Enable**, the 'Maximum Delay' field appears for defining the maximal allowed transmission time frame (in milliseconds) of a single packet. Any packet that requires a longer transmission time is fragmented to smaller sections. This avoids transmission of large, bursty packets that may cause delay or jitter for real-time traffic such as VoIP. If you insert a delay value in milliseconds, the delay in number of bytes is automatically updated on refresh.
7. From the 'Queue Policy' drop-down list, select the Tx traffic queueing method. This can be based on a shaping class (Class Based) or on the pre-defined priority levels (Strict Priority). Note that when based on shaping class, the class's bandwidth requirements are met regardless of the priority, and only excess bandwidth is given to traffic with a higher priority. However, when unlimited bandwidth is selected for the Tx

traffic, the queue policy can only be based on the pre-defined priority levels (i.e., Strict Priority).

8. If you want to specify a TX bandwidth, select **Specify** from the 'Tx Bandwidth' drop-down list and then enter the maximum Tx bandwidth.

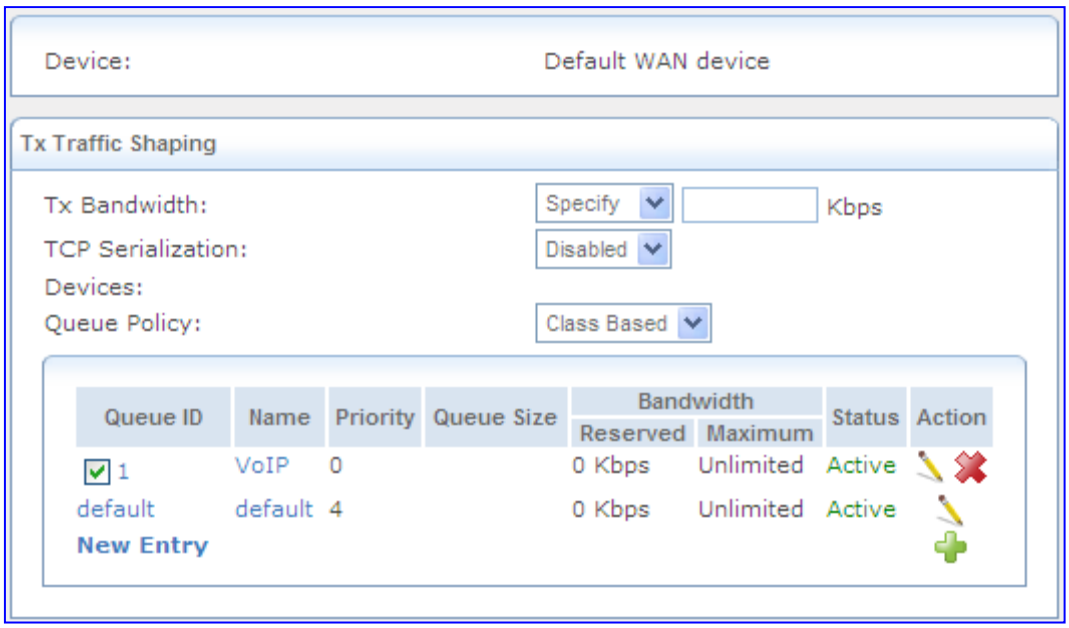
   a. In the table, click the New **Entry** link; the following page appears:

Figure 26-7: Adding Tx Shaping Class



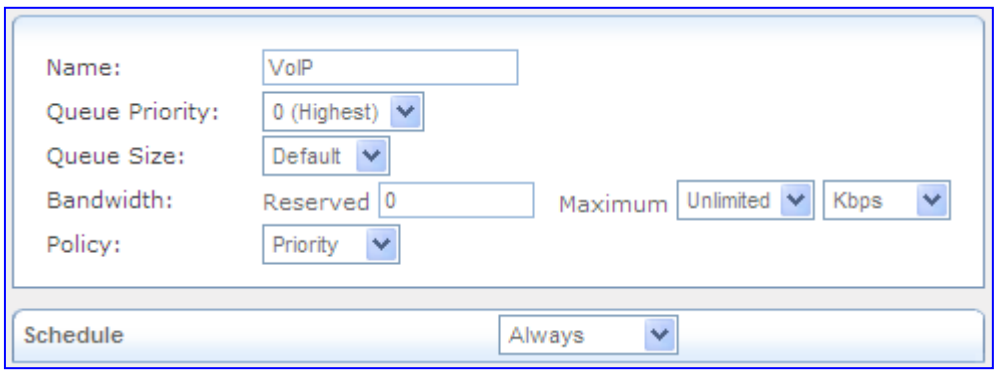
   b. In the 'Name' field, enter a new Tx traffic shaping class name (e.g., Class A), and then click **OK** to save the settings; the class is added to the table.

Figure 26-8: Class Name Added to Table



   c. Click the newly added class name; the following page appears:

Figure 26-9: Defining Shaping Class



   d. Configure the following fields:

   - **Name:** Name of the class.
   - **Queue Priority:** Priority level of the class (where zero is the highest and seven the lowest).
   - Queue Size:
   - **Bandwidth:** Reserved transmission bandwidth in kilobits per second. You

can limit the maximum allowed bandwidth by selecting **Specify** and then defining the Kbits/s.

- **Policy:** Class policy determines the policy of routing packets inside the class:
  - ✓ **Priority:** Priority queuing utilizes multiple queues, so that traffic is distributed among queues based on priority. This priority is defined according to packet's priority, which can be defined explicitly by a DSCP value or by a 802.1p value.
  - ✓ **FIFO:** The "First In, First Out" priority queue. This queue ignores any previously-marked priority that packets may have.
  - ✓ **Fairness:** The fairness algorithm ensures no starvation by granting all packets a certain level of priority.
  - ✓ **RED:** The Random Early Detection algorithm utilizes statistical methods to drop packets in a "probabilistic" way before queues overflow. Dropping packets slows a source down enough to keep the queue steady and reduces the number of packets that would be lost when a queue overflows and a host is transmitting at a high rate.
  - ✓ **WRR:** Weighted Round Robin utilizes a process scheduling function that prioritizes traffic according to the pre-defined 'Weight' parameter of a traffic's class. This level of prioritizing provides more flexibility in distributing bandwidth between traffic types, by defining additional classes within a parent class.
- **Schedule:** By default, the class is always active. However, you can configure schedular rules to define time segments during which the class may be active.

e. Click **OK**.

9. Click **OK**.

26.4 Configuring DSCP Settings

The **DSCP Settings** item defines Differentiated Services Code Point (DSCP). Differentiated Services (Diffserv) is a Class of Service (CoS) model that enhances best-effort Internet services by differentiating traffic by users, service requirements and other criteria. Packets are specifically marked, allowing network nodes to provide different levels of service, as appropriate for voice calls, video playback or other delay-sensitive applications, via priority queuing or bandwidth allocation, or by choosing dedicated routes for specific traffic flows.

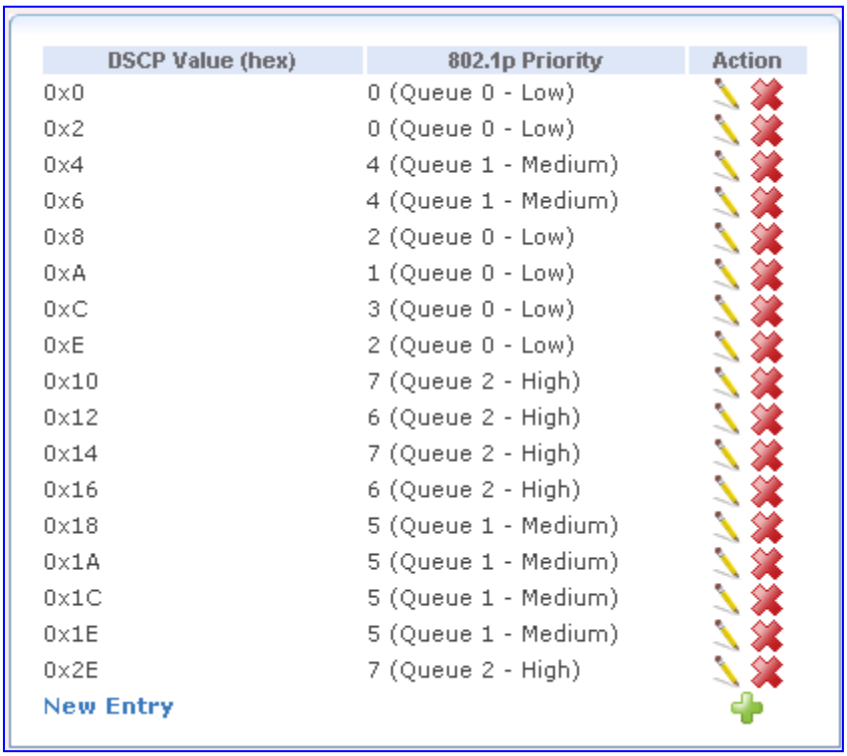
Diffserv defines a field in IP packet headers referred to as DSCP. Hosts or routers passing traffic to a Diffserv-enabled networks typically mark each transmitted packet with an appropriate DSCP. The DSCP markings are used by Diffserv network routers to appropriately classify packets and to apply particular queue handling or scheduling behavior.

The device provides a table of predefined DSCP values, which are mapped to 802.1p priority marking method. You can edit or delete any of the existing DSCP settings, as well as add new entries. Each DSCP value is assigned a default queue number as a part of its 802.1p priority settings. The device's QoS supports up to eight queues, where Queue 0 has the lowest priority.

➢ **To add, edit or delete DSCP settings:**

1. Click the **DSCP Settings** item (**Configuration** tab > **Data** menu > **QoS** submenu > **DSCP Settings**); the following page appears:

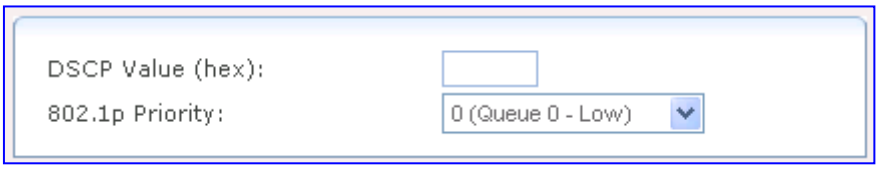
Figure 26-10: Configuring DSCP Settings



DSCP Value (hex)	802.1p Priority	Action
0x0	0 (Queue 0 - Low)	
0x2	0 (Queue 0 - Low)	
0x4	4 (Queue 1 - Medium)	
0x6	4 (Queue 1 - Medium)	
0x8	2 (Queue 0 - Low)	
0xA	1 (Queue 0 - Low)	
0xC	3 (Queue 0 - Low)	
0xE	2 (Queue 0 - Low)	
0x10	7 (Queue 2 - High)	
0x12	6 (Queue 2 - High)	
0x14	7 (Queue 2 - High)	
0x16	6 (Queue 2 - High)	
0x18	5 (Queue 1 - Medium)	
0x1A	5 (Queue 1 - Medium)	
0x1C	5 (Queue 1 - Medium)	
0x1E	5 (Queue 1 - Medium)	
0x2E	7 (Queue 2 - High)	
New Entry		

2. To edit an existing entry, click the corresponding **Edit** icon. To add a new entry, click the **New Entry** link. In both cases, the following page appears.

Figure 26-11: Defining DSCP to 802.1p Priority Mapping



3. In the 'DSCP Value (hex)' field, enter a hexadecimal number for the DSCP value.
4. From the '802.1p Priority' drop-down list, select an 802.1p priority level (each priority level is mapped to low/medium/high priority).
5. Click **OK** to save the settings.

Note: The DSCP value overriding the priority of incoming packets with an unassigned value (priority 0, assumed to be a no-priority-set) is "0x0.

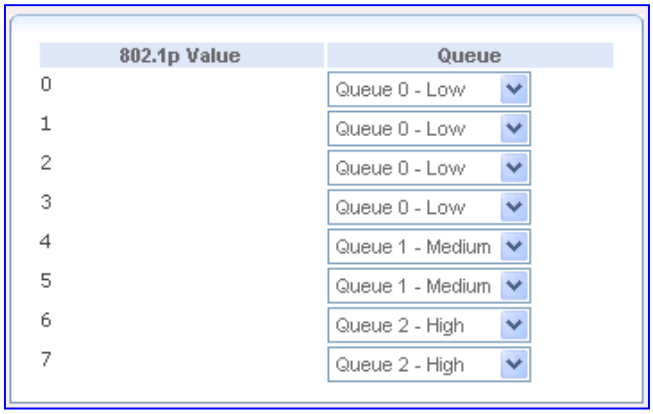
26.5 Configuring 802.1p Settings

The IEEE 802.1p priority marking method is a standard for prioritizing network traffic at the data link/Mac sub-layer. 802.1p traffic is simply classified and sent to the destination, with no bandwidth reservations established. The 802.1p header includes a 3-bit prioritization field, which allows packets to be grouped into eight levels of priority (0-7), where level 7 is the highest. In addition, the device maps these eight levels to priority queues, where Queue 0 has the lowest priority. The device's QoS supports up to eight queues. By default, the higher the level and queue values, the higher priority they receive. Therefore, the more critical the traffic, the higher priority level and queue number it should receive.

➢ **To change the mapping between a priority value and a queue value:**

1. Click the **802.1p Settings** item (**Configuration** tab > **Data** menu > **QoS** submenu > **802.1p Settings**); the following page appears:

Figure 26-12: Configuring 802.1p Settings



2. From the corresponding drop-down list, select the desired level.
3. Click **OK**.

27 VPN

The **VPN** menu allows you to configure Virtual Private Networking (VPN) over the Internet, and includes the following items:

- **IPSec** (see 'Configuring IPsec' on page 605)
- **PPTP** (see 'Configuring PPTP Server' on page 608)
- **L2TP** (see 'Configuring L2TP Server' on page 609)

27.1 Configuring IPSec

Internet Protocol Security (IPSec) is a series of guidelines for the protection of Internet Protocol (IP) communications. It specifies procedures for securing private information transmitted over public networks. The IPSec protocols include:

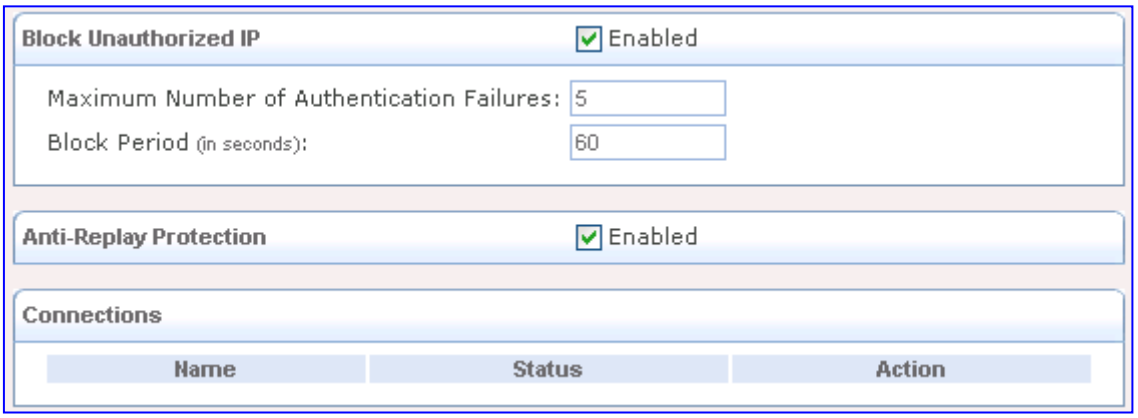
- AH (Authentication Header) provides packet-level authentication.
- ESP (Encapsulating Security Payload) provides encryption and authentication.
- IKE (Internet Key Exchange) negotiates connection parameters, including keys for the other two services.

Services supported by the IPSec protocols (AH, ESP) include confidentiality (encryption), authenticity (proof of sender), integrity (detection of data tampering), and replay protection (defense against unauthorized resending of data). IPSec also specifies methodologies for key management. Internet Key Exchange (IKE), the IPSec key management protocol, defines a series of steps to establish keys for encrypting and decrypting information; it defines a common language on which communications between two parties is based. Developed by the Internet Engineering Task Force (IETF), IPSec and IKE together standardize the way data protection is performed, thus making it possible for security systems developed by different vendors to interoperate.

➢ **To configure IPSec:**

1. Click the **IPSec** item (**Configuration** tab > **Data** menu > **VPN** submenu > **IPSec**); the following page appears:

Figure 27-1: Configuring VPN IPSec



The Connections' group displays a list of IPSec connections (to create an IPSec connection, see 'Configuring Network Connections' on page 633).

2. To block unauthorized IP to the device, do the following:
   a. Select the 'Block Unauthorized IP' check box.
   b. In the 'Maximum Number of Authentication Failures' field, enter the maximum number of packets to authenticate before blocking the origin's IP address.

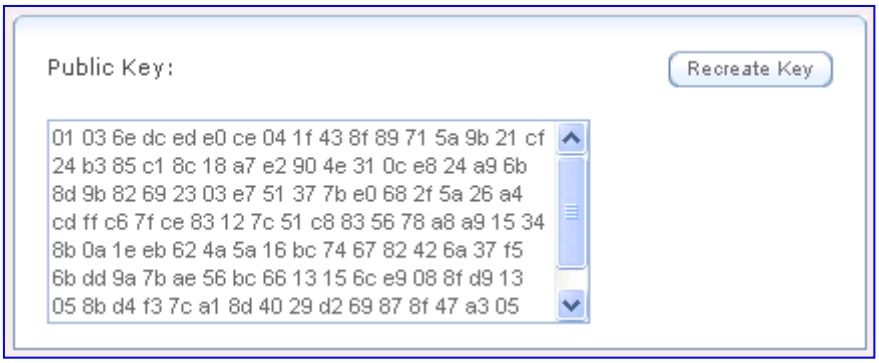
c. In the 'Block Period' field, enter the time frame during which the device drops packets from an unauthorized IP address.

3. To enable dropping of packets that are recognized (by their sequence number) as already been received, select the 'Anti-Replay Protection' check box.

4. Configure the device's IPSec public keys:

a. Click the **Settings** button; the following page appears.

Figure 27-2: Recreating IPSec Public Key



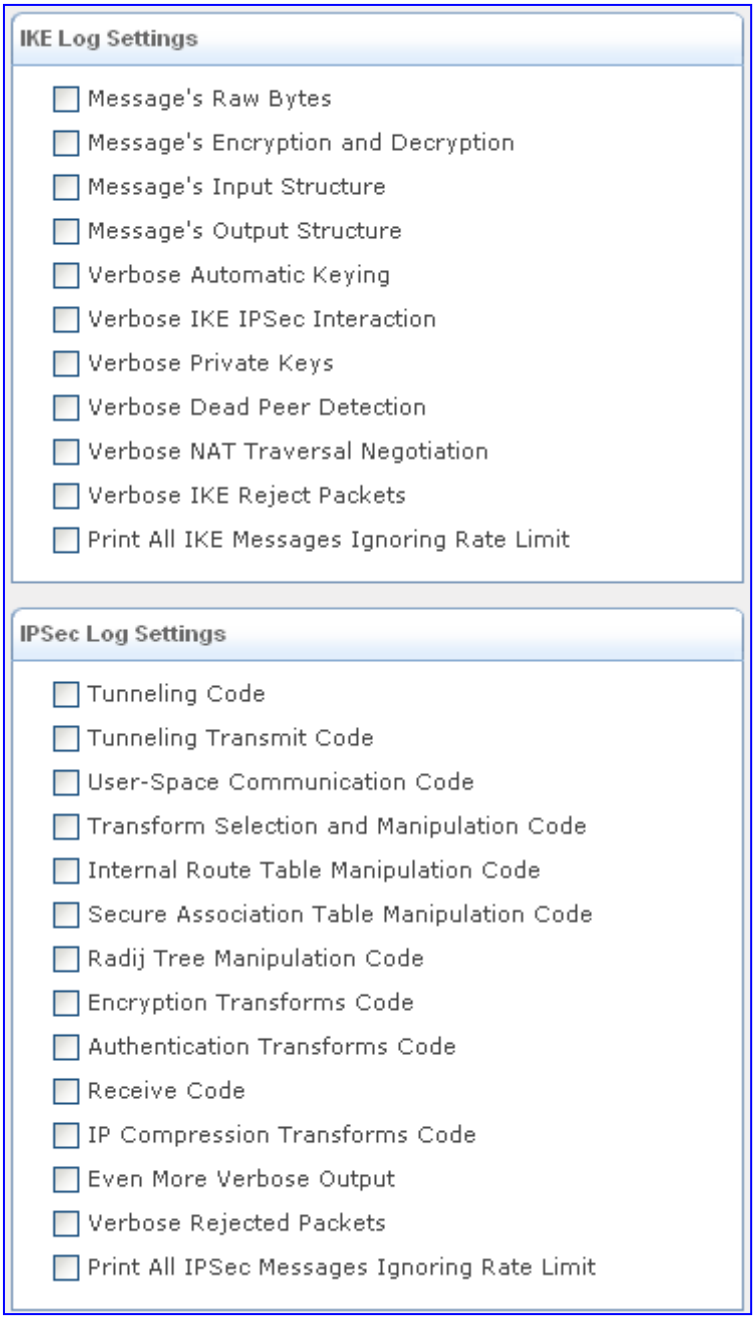
b. Click the **Recreate Key** button to recreate the public key, or the **Refresh** button to refresh the displayed key.

c. Click **Close**; you are returned to the previous page.

5. Configure the IPSec log display for identifying and analyzing the history of the IPSec package commands, attempts to create connections, etc:

a. Click the **Log Settings** button; the following page appears.

Figure 27-3: IPSec Log Settings



b. Select the check boxes relevant to the information you want the IPSec log to record.

c. Click **OK** to save the settings.

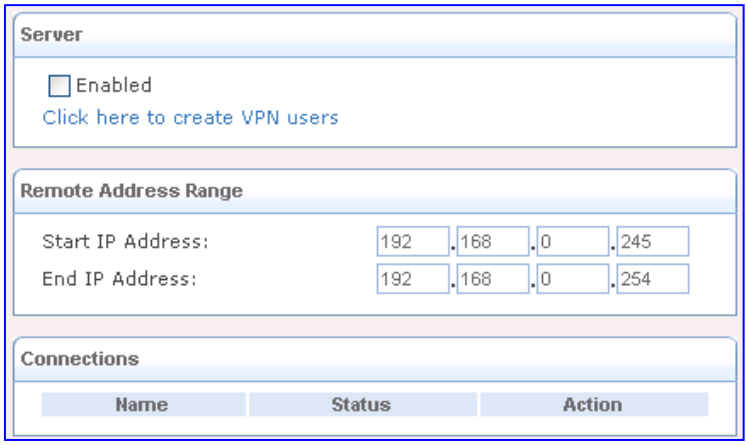
27.2 Configuring PPTP Server

The device can act as a Point-to-Point Tunneling Protocol Server (PPTP Server), accepting PPTP client connection requests.

➢ **To configure PPTP:**

1. Click the **PPTP** item (**Configuration** tab > **Data** menu > **VPN** submenu > **PPTP**); the following page appears:

Figure 27-4: Configuring VPN PPTP Server



2. Under the 'Server' group, perform the following:
   a. Select the 'Enabled' check box to enable the PPTP feature. Note that checking this box creates a PPTP server, but does not define remote users.
   b. Click the **Click here to create VPN users** link to define remote users that are granted access to your home network (see 'Creating VPN Users' on page 610).
   c. Click the **Advanced** button to display additional parameters and configure the following:
      - In the 'Max Idle Time to Disconnect in Seconds' field, specify the amount of idle time (during which no data is sent or received) that should elapse before the device disconnects a PPTP connection.
      - Select the 'Authentication Required' check box to enable PPTP to use authentication, and then select the algorithms the server may use when authenticating its clients.
      - Select the 'Encryption Required' check box to enable PPTP to use encryption, and then select the algorithms the server may use when encrypting data.
      - From the 'MPPE Encryption Mode' drop-down list, select the Microsoft Point-to-Point Encryption mode.
3. Under the 'Remote Address Range' group, in the 'Start IP Address' and 'End IP Address' fields, specify the range of IP addresses that are granted by the PPTP server to the PPTP client.

Note: The server settings must be compatible with the client settings, described in 'Configuring Network Connections' on page 633.

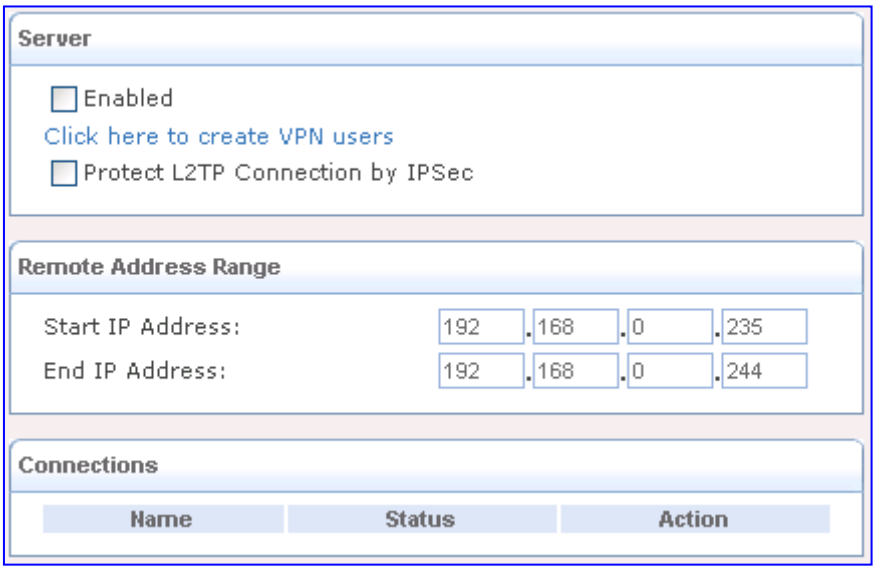
27.3 Configuring L2TP Server

The device can act as a Layer 2 Tunneling Protocol Server (L2TP Server), accepting L2TP client connection requests.

➢ **To configure L2PT:**

1. Click the **L2TP** item (**Configuration** tab > **Data** menu > **VPN** submenu > **L2TP**); the following page appears:

Figure 27-5: Configuring VPN L2TP Server



2. Under the 'Server' group, perform the following:
   a. Select the 'Enabled' check box to enable the L2TP feature. Note that checking this box creates a L2TP server (if not yet created with the wizard), but does not define remote users.
   b. Click the **Click here to create VPN users** link to define remote users that are granted access to your home network (see 'Creating VPN Users' on page 610).
   c. Select the 'Protect L2TP Connection by IPSec' to secure the L2TP connection by the IP Security (IPSec) protocol. When enabled, the following entry appears:
      - **Create Default IPSec Connection:** When creating an L2TP Server with the connection wizard, a default IPSec connection is created to protect it. If you wish to disable this feature, clear this option. However, if L2TP protection is enabled by IPSec (see previous entry), you must provide an alternative, active IPSec connection for users to be able to connect. When this feature is enabled, the following entry appears.
        ✓ **L2TP Server IPSec Shared Secret:** change the IPSec shared secret provided when the connection was created.
   d. Click the **Advanced** button, and then configure the following:
      - In the 'Max Idle Time to Disconnect in Seconds' field, specify the amount of idle time (during which no data is sent or received) that should elapse before the device disconnects an L2TP connection.
      - Select the 'Authentication Required' check box to enable L2TP to use authentication, and then select the algorithms the server may use when authenticating its clients.
      - Select the 'Encryption Required' check box to enable L2TP to use encryption and then select the algorithms the server may use when encrypting data.

- From the 'MPPE Encryption Mode' drop-down list, select the Microsoft Point-to-Point Encryption mode: Stateless or Stateful.

3. Under the 'Remote Address Range' group, in the 'Start IP Address' and 'End IP Address' fields, specify the range of IP addresses that are granted by the L2TP server to the L2TP client.

Note: The server settings must be compatible with the client settings, described in 'Configuring Network Connections' on page 633.

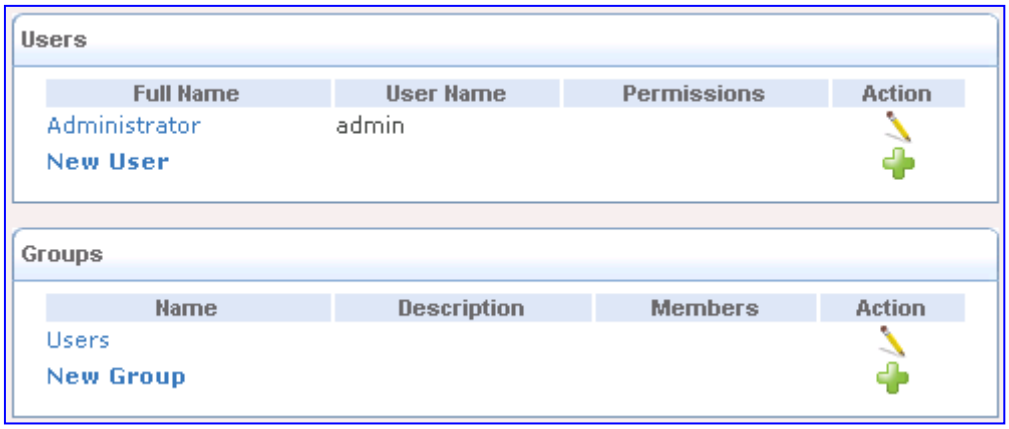
27.4 Creating VPN Users

The procedure below describes how to add users and user groups, after clicking the **Click here to create VPN users** link when configuring a PPTP server and L2TP server (see 'Configuring PPTP Server' on page 608 and 'Configuring L2TP Server' on page 609 respectively). You can also group users according to your preferences. The "Administrator" user is the pre-defined default user.

➢ **To add a user:**

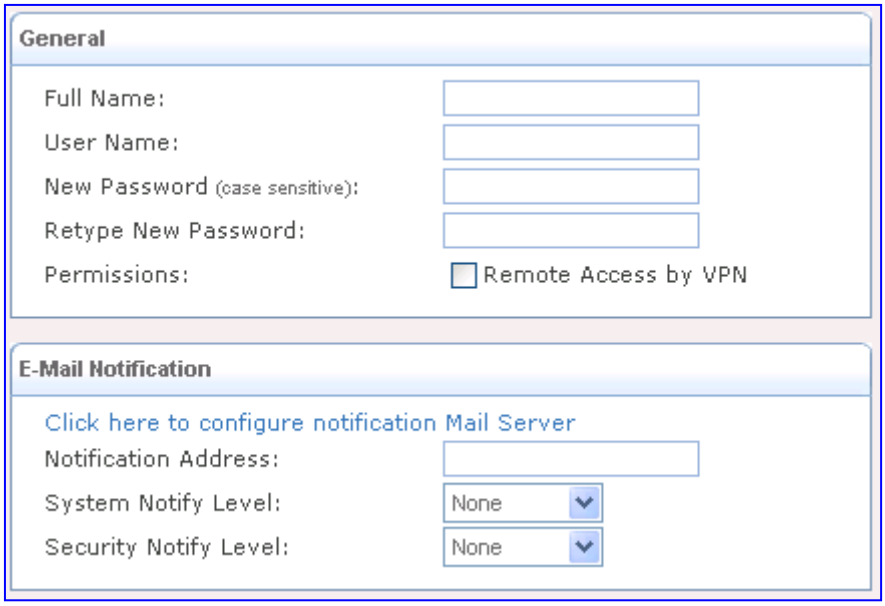
1. After clicking the **Click here to create VPN users** link, the following page appears:

Figure 27-6: Adding Users



2. Under the 'Users' group, click the **New User** link; the following page appears:

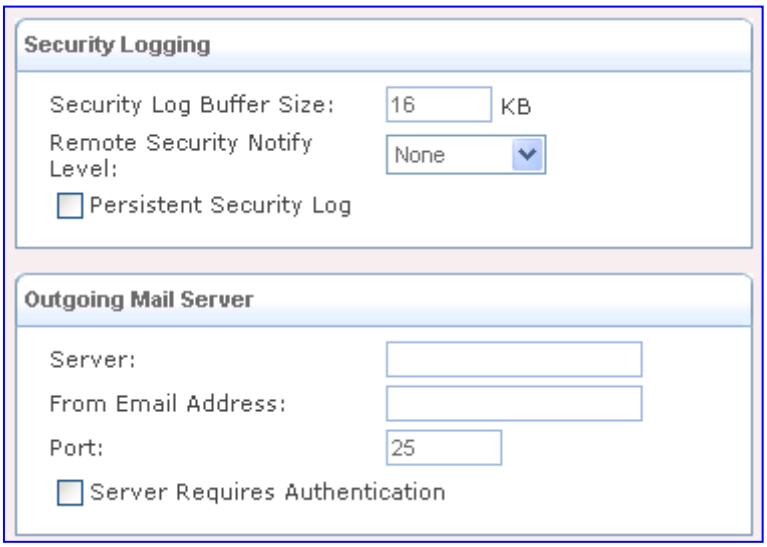
Figure 27-7: Adding a New User



3. Under the 'General' group, configure the following parameters:
   a. **Full Name:** remote user's full name.
   b. **User Name:** name that a user uses to access your network.
   c. **New Password:** user's password.
   d. **Retype New Password:** if a new password is assigned, type it again to verify its correctness.
   e. **Primary Group:** this check box only appears after a user is defined, enabling you to assign the user to a primary group.
   f. **Permissions:** select the 'Remote Access by VPN' check box to grant remote access to the device using the VPN protocol.
4. Under the 'E-Mail Notification' group, you can use e-mail notification to receive indications of system events for a predefined severity classification. The available types of events are 'System' or 'Security' events. The available severity of events are 'Error', 'Warning', 'Information', and 'Debug'. If the 'Information' level is selected, the user receives notification of the 'Information', 'Warning' and 'Error' events. If the 'Warning' level is selected, the user receives notification of the 'Warning' and 'Error' events, and so on.
   a. Ensure that you have configured an outgoing mail server. Click the **Click here to configure notification Mail Server** link to configure the outgoing mail server (see Step 5).
   b. Notification Address: user's e-mail address.
   c. Select the 'System' and 'Security' notification levels in the 'System Notify Level' and 'Security Notify Level' drop-down lists respectively.

5. Configure an outgoing mail server:

Figure 27-8: Defining Outgoing Mail Server



   a. **Server:** enter the hostname of your outgoing (SMTP) server.
   b. **From Email Address:** each email requires a 'from' address and some outgoing servers refuse to forward mail without a valid 'from' address for anti-spam considerations.
   c. **Port:** enter the port that is used by your outgoing mail server.
   d. **Server Requires Authentication:** if your outgoing mail server requires authentication, select this check box and enter your user name and password in the 'User Name' and 'Password' fields respectively.

6. Under the 'Security Logging' group, configure the following parameters:
   - **Security Log Buffer Size:** size of the security log buffer in Kilobytes.
   - **Remote Security Notify Level:** remote security notification level - None, Error, Warning, and Information.
   - **Persistent Security Log:** select this check box to save the security log to the flash memory.

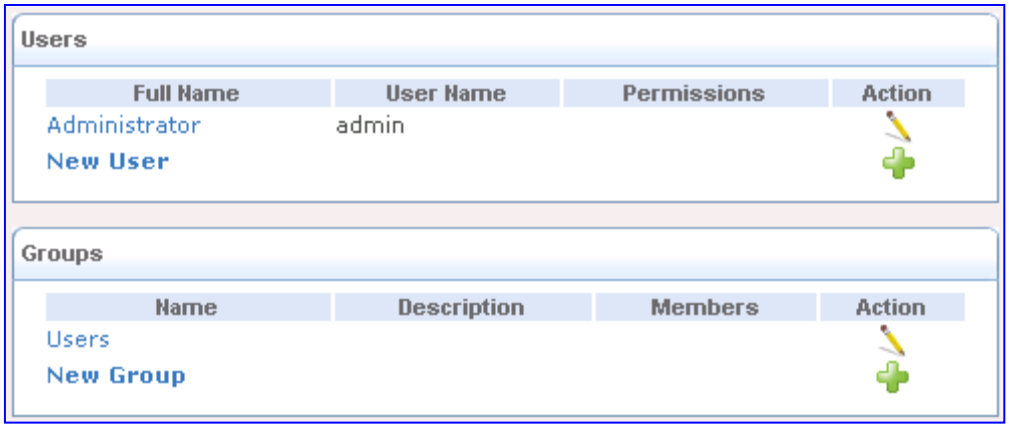
7. Click **OK** to save your settings.

You can assemble your defined users into different groups, based on different criteria, for example, home users versus office users. By default, new users are added to the default group "Users".

➢ **To add a user group:**

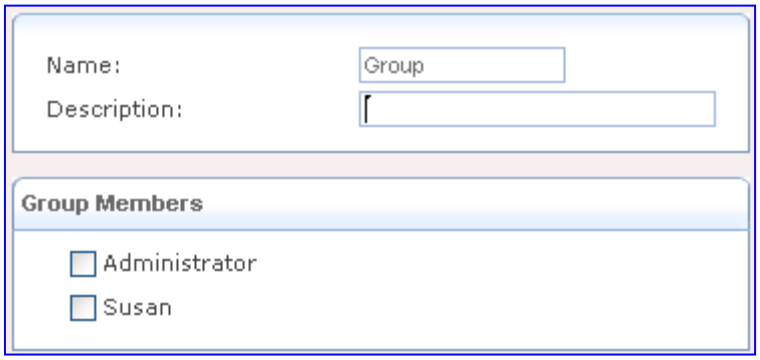
1. After clicking the **Click here to create VPN users** link, the following page appears:

Figure 27-9: Adding Users



2. Under the 'Groups' section, click the **New Group** link; the following page appears:

Figure 27-10: Adding a User Group



3. In the 'Name' field, enter a name for the group.
4. In the 'Description' field, enter a short description for the group (optional).
5. Under the Group Members section, select the users that you want to assign to this group. A user can belong to more than one group.
6. Click **OK** to save your settings.

Reader's Notes

28 Data Services

The **Data Services** submenu allows you to configure various services (applications), and includes the following menus:

- **DDNS** (see 'Configuring DDNS' on page 615)
- **DNS Server** (see 'Configuring DNS Server' on page 617)
- **DHCP Server** (see 'Configuring DHCP Server' on page 618)

28.1 Configuring DDNS

The Dynamic DNS (DDNS) service enables you to alias a dynamic IP address to a static hostname, allowing your computer to be more easily accessible from various locations on the Internet. Typically, when you connect to the Internet, your service provider assigns an unused IP address from a pool of IP addresses, and this address is used only for the duration of a specific connection. Dynamically assigning addresses extends the usable pool of available IP addresses, whilst maintaining a constant domain name. When using the DDNS service, each time the IP address provided by your ISP changes, the DNS database changes accordingly to reflect the change. In this way, even though your IP address changes often, your domain name remains constant and accessible.

To use the DDNS feature, you must first obtain a DDNS account. For example, you can open a free account at http://www.dyndns.com/account/create.html . When applying for an account, you will need to specify a user name and password.

Use the DDNS feature to define different static host names for each of your WAN connections. Moreover, you can define more than one static host name for each WAN connection, by simply repeating the following procedure for the same connection.

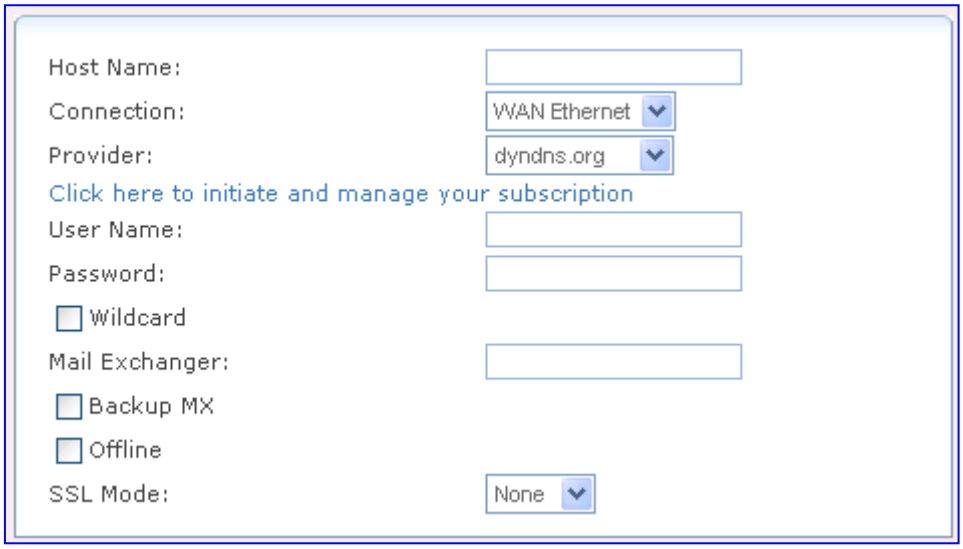
➢ **To create a dynamic DNS:**

1. Click the **DDNS** item (**Configuration** tab > **Data** menu > **Data Services** submenu > **DDNS**); the following page appears:

Figure 28-1: Configuring Dynamic DNS (DDNS) Services

2. Click the **New Dynamic DNS Entry** link to add a new DDNS entry; the following page appears:

Figure 28-2: Adding a DDNS



3. In the 'Host Name' field, enter your full DDNS domain name.
4. From the 'Connection' field, select the connection to which you want to couple the DDNS service. The DDNS service only uses the selected device, unless failover is enabled. In this case, the failed-to device is used instead (assuming its route rules consent), until the selected device is up again.
5. From the 'Provider', select your DDNS service provider; the page displays parameters required by the selected provider. To open the selected provider's account creation Web page, click the link **Click here to initiate and manage your subscription**.
6. The parameters described below are available if you select the provider dyndns (in Step 5), which includes all available parameters.
   - **User Name:** enter your DDNS user name.
   - **Password:** enter your DDNS password.
   - **Wildcard:** select this check box to enable use of special links such as http://www.<your host>.dyndns.com.
   - **Mail Exchanger:** enter your mail exchange server address to redirect all e-mails arriving at your DDNS address to your mail server.
   - **Backup MX:** select this check box to designate the mail exchange server to be a backup server.
   - **Offline:** if you wish to temporarily take your site offline (prevent traffic from reaching your DDNS domain name), check this box to enable redirection of DNS requests to an alternative URL, predefined in your DDNS account. The availability of this feature depends on your account's level and type of service.
   - **SSL Mode:** secured DDNS services are accessed using HTTPS. Upon connection, the device validates the DDNS server's certificate. Use this entry to choose the certificate's validation method.
     - **None:** do not validate the server's certificate.

- **Chain:** validate the entire certificate chain. If you select this option, the 'Validate Time' drop-down lists appears to validate the certificate's expiration time. If the certificate has expired, the connection terminates immediately.
- **Direct:** ensures that the server's certificate is directly signed by the root certificate. If selected, the 'Validate Time' drop-down lists appears for validation of the certificate's expiration time, as described above.

7. Click **OK**.

28.2 Configuring DNS Server

Domain Name System (DNS) provides a service that translates domain names into IP addresses and vice versa. The device's DNS server is an auto-learning DNS, which means that when a new computer is connected to the network the DNS server learns its name and automatically adds it to the DNS table. Other network users may immediately communicate with this computer using either its name or its IP address. In addition, your device's DNS:

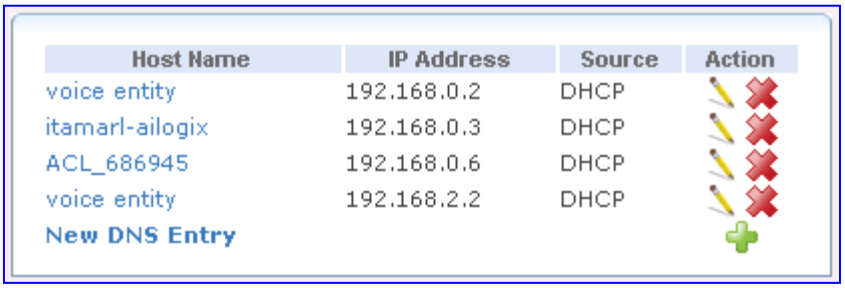
- Shares a common database of domain names and IP addresses with the DHCP server
- Supports multiple subnets within the LAN simultaneously
- Automatically appends a domain name to unqualified names.
- Allows new domain names to be added to the database (using the Web interface)
- Permits a computer to have multiple host names
- Permits a host name to have multiple IPs (needed if a host has multiple network cards)

The DNS server does not require configuration. However, you may wish to view the list of computers known by the DNS, edit the host name or IP address of a computer on the list, or manually add a new computer to the list.

➢ **To create a DNS entry:**

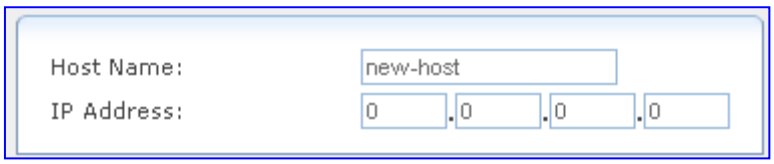
1. Click the **DNS Server** item (**Configuration** tab > **Data** menu > **Data Services** submenu > **DNS Server**); the following page appears:

Figure 28-3: Configuring a DNS Server



2. Click the **New DNS Entry** link; the following page appears:

Figure 28-4: Adding a DNS Server



3. In the 'Host Name' field, enter the computer's host name.
4. In the 'IP Address' field, enter the computer's IP address.
5. Click **OK** to save the settings.

28.3 Configuring DHCP Server

Your device's Dynamic Host Configuration Protocol (DHCP) server makes it possible to easily add computers that are configured as DHCP clients to the home network. It provides a mechanism for allocating IP addresses and delivering network configuration parameters to such hosts. The device's default DHCP server is the LAN bridge. A client (host) sends out a broadcast message on the LAN requesting an IP address for itself. The DHCP server then checks its list of available addresses and leases a local IP address to the host for a specific period of time and simultaneously designates this IP address as `taken'. At this point the host is configured with an IP address for the duration of the lease. The host can choose to renew an expiring lease or let it expire. If it chooses to renew a lease then it also receives current information about network services, as it did with the original lease, allowing it to update its network configurations to reflect any changes that may have occurred since it first connected to the network. If the host wishes to terminate a lease before its expiration it can send a release message to the DHCP server, which then makes the IP address available for use by others.

Note: By default, the device's DHCP server is enabled. Therefore, when connecting the device to your enterprise's LAN, the device responds to DHCP requests and consequently distributes IP addresses (instead of your Enterprise's DHCP server, if exists).

Your device's DHCP server:

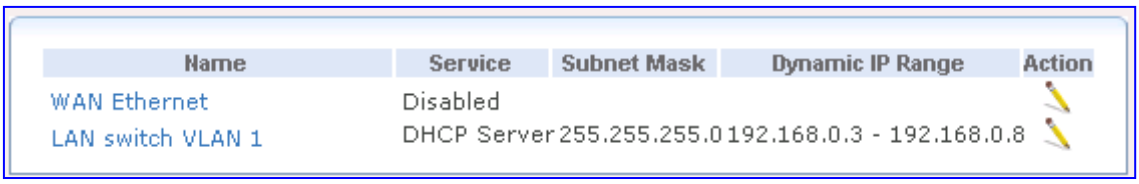
- Displays a list of all DHCP host devices connected to the device
- Defines the range of IP addresses that can be allocated in the LAN
- Defines the length of time for which dynamic IP addresses are allocated
- Provides the above configurations for each LAN device and can be configured and enabled/disabled separately for each LAN device
- Can assign a static lease to a LAN PC so that it receives the same IP address each time it connects to the network, even if this IP address is within the range of addresses that the DHCP server may assign to other computers
- Provides the DNS server with the host name and IP address of each PC that is connected to the LAN

The device can also act as a DHCP relay, escalating DHCP responsibilities to a WAN DHCP server. In this case, the device acts merely as a router, while its LAN hosts receive their IP addresses from a DHCP server on the WAN. With the device's optional Zero Configuration Technology feature, the IP Auto Detection method detects statically-defined IP addresses in addition to the device's DHCP clients. It learns all the IP addresses on the LAN, and integrates the collected information with the database of the DHCP server. This allows the DHCP server to issue valid leases, thus avoiding conflicting IP addresses used by other computers in the network.

➢ **To configure DHCP:**

1. Click the **DHCP Server** item (**Configuration** tab > **Data** menu > **Data Services** submenu > **DHCP Server**); the following page appears:

Figure 28-5: Configuring DHCP Server

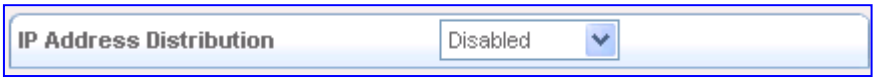


The page displays a summary of the services currently being provided by the DHCP server.

Note: If a device is listed as "Disabled" in the 'Service' column, then DHCP services are not being provided to hosts connected to the network through that device. This means that the device does not assign IP addresses to these computers, which is useful if you wish to work with static IP addresses only.

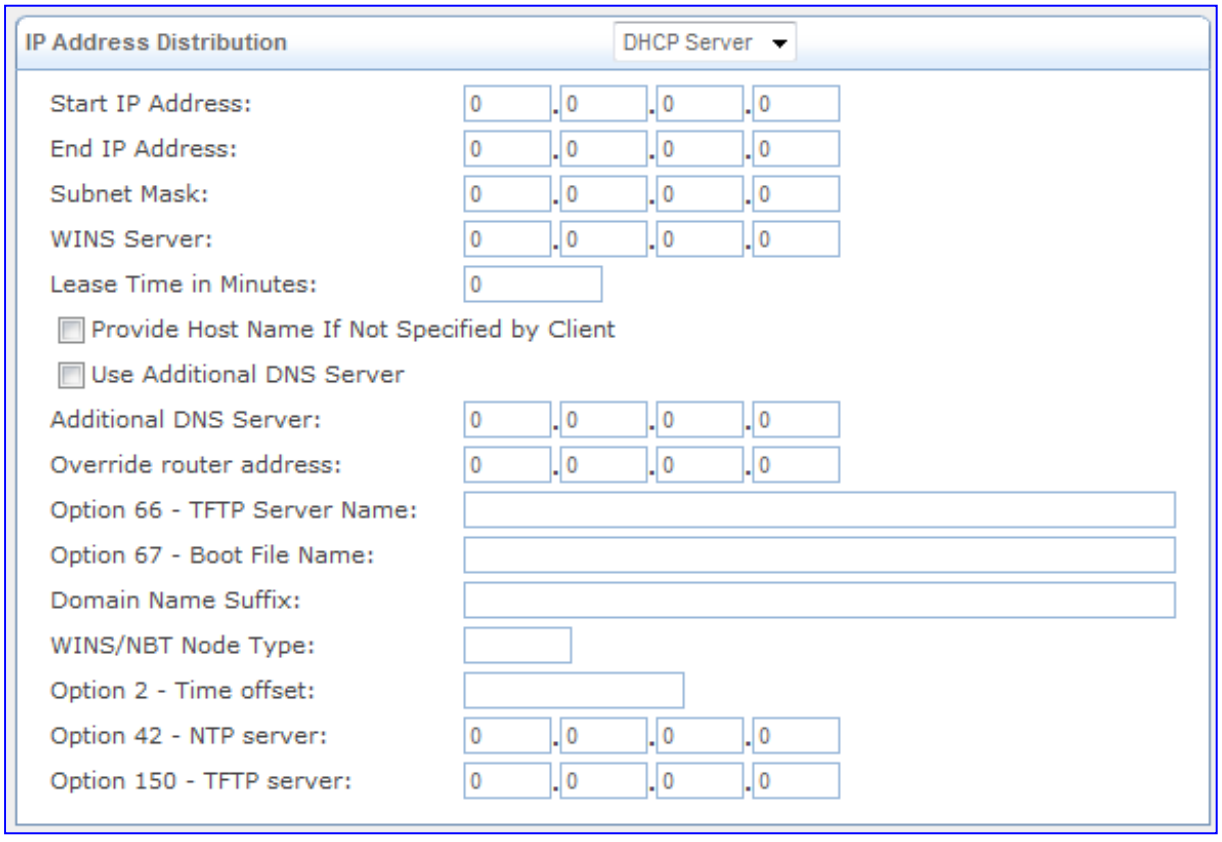
2. Click the **Edit** icon corresponding to the required interface; the following page appears:

Figure 28-6: Defining IP Distribution Type



3. From the 'IP Address Distribution' drop-down list, choose either 'DHCP Server', 'DHCP Relay' (or 'Disabled' if you want to disable DHCP).
4. If you selected 'DHCP Server', the following fields appear:

Figure 28-7: Defining DHCP Server Parameters



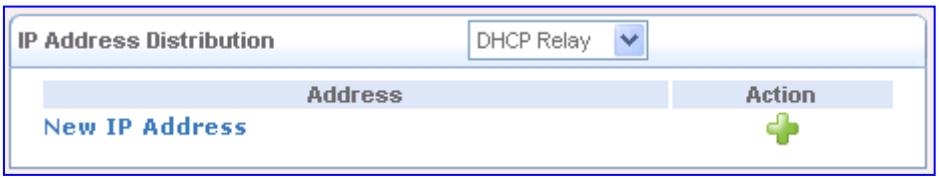
- **Start IP Address:** First IP address that can be assigned to a LAN host. Since the device's default IP address is 192.168.1.1, this address must be 192.168.1.2 or greater.
- **End IP Address:** Last IP address in the range that can be used to automatically assign IP addresses to LAN hosts.
- **Subnet Mask:** Mask used to determine to what subnet an IP address belongs (e.g.,255.255.0.0).

- **WINS Server:** The Windows Internet Naming Service (WINS) Server IP address that is given in a lease.
- **Lease Time In Minutes:** Each LAN host is assigned an IP address by the DHCP server for a this amount of time, when it connects to the network. When the lease expires, the server determines if the computer has disconnected from the network. If it has, the server may reassign this IP address to a newly-connected computer. This feature ensures that IP addresses that are not in use become available for other computers on the network.
- **Provide Host Name If Not Specified by Client:** If the DHCP client does not have a host name, the device automatically assigns one.
- **Use Additional DNS Server:** The additional DNS server address allows the network administrator to provision the DHCP clients with another DNS server (other than the device itself).
- **Override router address: :** IP address for Option 3 of the DHCP message (RFC 2132 section 3.5).
- **Option 66 - TFTP Server Name:** This option is used to identify a TFTP server.
- **Option 67 - Boot File Name:** This option is used to identify the boot file name.
- **Domain Name Suffix:** String for Option 15 of the DHCP message (RFC 2132 section 3.17).
- **WINS/NBT Node Type:** Numeric value for Option 46 of the DHCP message (RFC 2132 section 8.7).
- **Option 2 - Time offset:** Specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC). The offset is expressed as a two's complement 32-bit integer. A positive offset indicates a location east of the zero meridian and a negative offset indicates a location west of the zero meridian. The code for the time offset option is 2, and its length is 4 octets.
- **Option 42 - NTP server:** Network Time Protocol (NTP) Servers option specifies a list of IP addresses indicating NTP servers available to the client. Servers should be listed in order of preference. The code for this option is 42. Its minimum length is 4, and the length must be a multiple of 4.
- **Option 150 - TFTP server:** DHCP option (RFC 2132) contains one or more IPv4 addresses that the client may use. The current use of this option is for downloading configuration from a VoIP server via TFTP; however, the option may be used for purposes other than contacting a VoIP configuration server.

a. Click **OK** to save the settings.

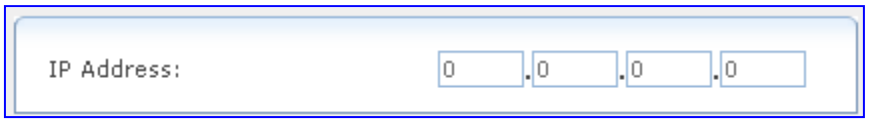
5. If you selected 'DHCP Relay', the following appears:

Figure 28-8: Defining DHCP Relay (DHCP for LAN Bridge)



a. Click the **New IP Address** link; the following page appears:

Figure 28-9: Defining DHCP Server's IP Address



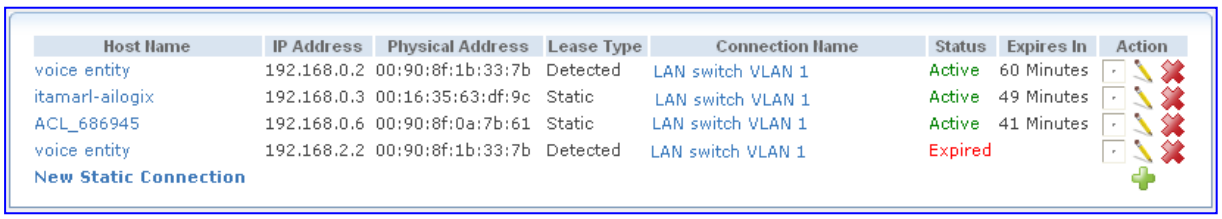
b. In the 'IP Address' field, enter the IP address of the DHCP server.
c. Click **OK** to save the settings, and then click **OK** again.
d. You now need to configure the WAN to operate in routing mode, which is necessary for DHCP relay:
   a. Click the **Connections** item (**Configuration** tab > **Data Settings** menu > **Data System** > **Connections**).
   b. Click the **Edit** icon corresponding to the 'WAN Ethernet' connection.
   c. Select the **Routing** tab and then from the 'Routing Mode' drop-down list, select **Route**.
   d. Click **OK** to save the settings.

You can also view a list of computers currently recognized by the DHCP server:

➢ **To view a list of computers currently recognized by the DHCP server and to add a new computer with a static IP address:**

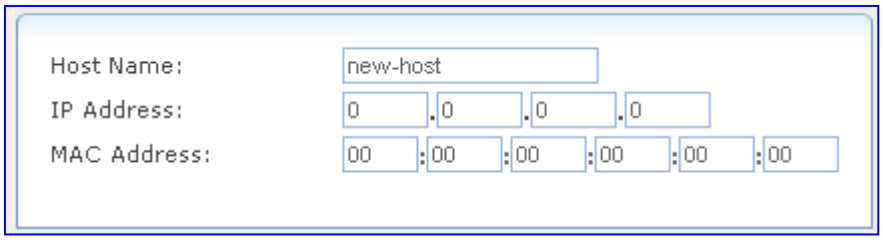
1. Click the **DHCP Server** item (**Configuration** tab > **Data Settings** menu > **Services** > **DHCP Server**).
2. Click the **Connection List** button located at the bottom of the page; the following page appears.

Figure 28-10: Computers Recognized by DHCP Server



3. To define a new connection with a fixed IP address, click the **New Static Connection** link; the following page appears:

Figure 28-11: Defining New Static Connection (IP Address)



4. Enter a host name for this connection.
5. Enter the fixed IP address that you would like to have assigned to the computer.
6. Enter the MAC address of the computer's network card.

Note: A device's fixed IP address is actually assigned to the specific network card's (NIC) MAC address installed on the LAN computer. If you replace this network card then you must update the device's entry in the DHCP Connections list with the new network card's MAC address.

7. Click **OK** to save the settings.

29 Data Routing

The **Data Routing** submenu allows you to configure the device's routing rules, and includes the following items:

- **General Routing** (see 'Configuring General Routing Settings' on page 623)
- **Dynamic Routing** (see 'Configuring Dynamic Routing' on page 627)

29.1 Configuring General Routing

You can choose to setup your device to use static or dynamic routing. Dynamic routing automatically adjusts how packets travel on the network, whereas static routing specifies a fixed routing path to destinations. The **Data Routing** item allows you to add, edit and delete routing rules from the routing table.

This page also allows you to add or edit the device's default route devices by changing their metric value.

The device supports platforms with multiple physical WAN devices (ports), which can be used for traffic load balancing, failover, and various routing policies. The multiple WAN features may also be used to define multiple logical devices (e.g. PPTP VPN, PPPoE) on device's with a single WAN port.

- **Load balancing:** traffic load (bandwidth) is distributed between two WAN interfaces. Load balancing uses the IP pairs technique, in which traffic between a pair of source and destination IP addresses is routed to the same WAN device for a certain time frame. A router load balancing on a per-destination basis uses the parallel routes in a round-robin fashion, and forwards an entire destination-based flow in each pass.
- **DSCP-based policy routing:** you may specify that traffic matching a certain DSCP value is routed to a specific device. This is useful for routing different types of data to different WAN devices. It is also useful if you want to segregate the voice traffic from the data traffic over two lower-cost broadband circuits in an effort to have better voice quality.
- **Failover:** traffic is routed to an active WAN device in case its current WAN device fails, ensuring connectivity. This transfer of traffic is done regardless of DSCP-based policy routing rules. An example scenario is Inbound Failover where if a connection fails and its IP is no longer accessible, the device notifies the other party to use a different IP, using Dynamic DNS. The device supports the following types of failover:
  - Full Link Redundancy: two or more active WAN devices, usually with equal speed must be configured. A device set as the default gateway functions as the main device, while the other one(s) work in the backup mode. This means that if one of the devices fails, the next one take its place. In this configuration, the devices can also work in conjunction with the Load Balancing feature to enhance the WAN throughput.
  - Rollover Connection: during uptime, a rollover device is kept inactive. This is usually a slow link, for example, a dialup. When all other failover devices lose connectivity, the rollover device becomes active automatically, and may keep the same IP as the main device. This allows the use of a slow connection as a backup to the main fast connection. When a failover device regains connectivity, the rollover device becomes inactive again. Note that if dialup is done by demand, activating the backup device may take a noticeable amount of time. The failover process consists of three phases:
    a. Detection – performed using a DNS test.
    b. Action – when a DNS test fails, the failover process simply removes the route records of the failed connection. This enables you to reach the desired failover behavior by configuring the device's routing rules correctly.

c. Recover – during failover, tests continue to run on the failed connection. When a test succeeds, the connection recovers its route records.

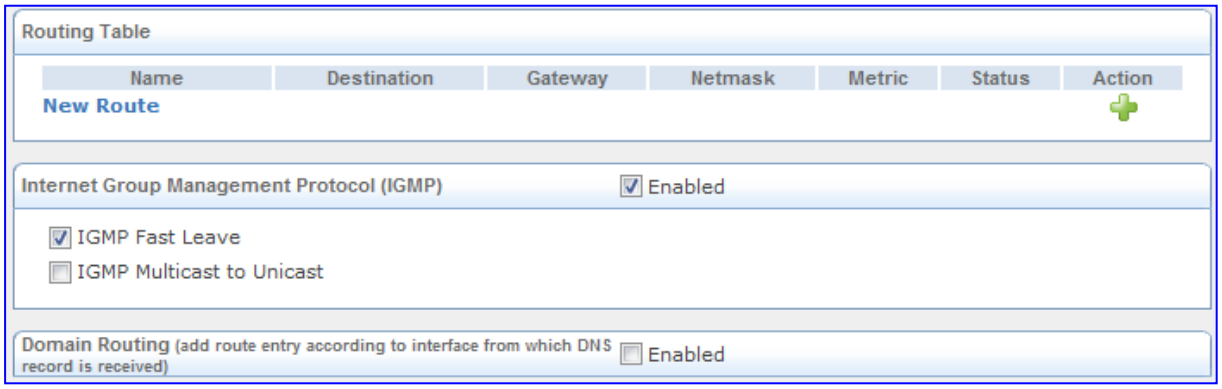
Notes:

- Only default route devices can participate in load balancing.
- DSCP-based policy routing takes precedence over load balancing. If most of the traffic falls under the DSCP-based policy routing rules, it is then forwarded accordingly, regardless of the load balancing. Load balancing, in this case, is by best-effort load balancing, and balances the remaining traffic not directed by the DSCP-based policy routing rules.

➢ **To configure general routes:**

1. Click the **General Routing** item (**Configuration** tab > **Data** menu > **Data Routing** submenu > **General Routing**); the following page appears:

Figure 29-1: Configuring General Routing

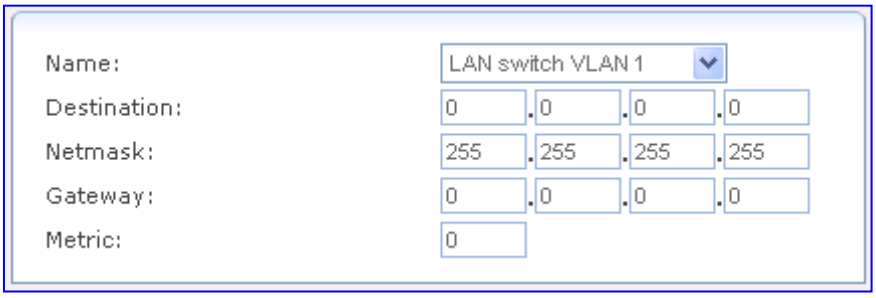


To view additional parameters on the page, click the **Advanced** button.

2. To add a new routing rule:

a. Under the 'Routing Table' group, click the **New Route** link; the following page appears:

Figure 29-2: Adding a Routing Rule



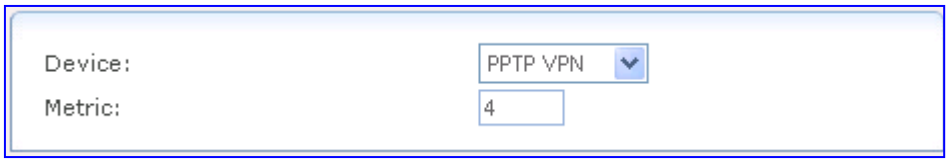
b. Define the following fields:

- **Name:** select the network device.
- **Destination:** enter the destination host, subnet address, network address, or default route. The destination for a default route is 0.0.0.0.
- **Netmask:** network mask is used in conjunction with the destination to determine when a route is used.
- **Gateway:** enter the device's IP address.
- **Metric:** measurement of a route's preference. Typically, the lowest metric is the most preferred route. If multiple routes have the same metric value, the

default route is the first in the order of appearance.

3. To add or edit a default route:
   a. Under the 'Default Routes' group, click the required device whose default route you want to edit or click the **New Default Route** link to add a default route; the following page appears:

Figure 29-3: Editing the Default Route



   b. If adding a default route, From the 'Device' drop-down list, select the WAN device.
   c. In the 'Metric' field, enter a value for the metric route preference.
   d. Click **OK** to save the settings.

Notes:

- To add an additional (logical) default route device, you must first define a new WAN device that has an IP address.
- Although multiple devices may be configured as default routes, only one serves as the default route—the one with the lowest metric value, or, if metric values are identical, the first in order.

4. To enable load balancing between multiple WAN devices:
   a. Under the 'Load Balancing' group, select the 'Enabled' check box; the load balancing table appears.
   b. Select the devices on which load balancing is performed by checking their respective check boxes.
   c. Click the **Edit** icon corresponding to the device for which you want to modify the weight in the balancing procedure (determines the ratio of IP pairs provided to each device); the following page appears:

Figure 29-4: Defining Load Balancing

   d. In the 'Weight' field, enter the numeric ratio to represent the weight of the device.
   e. Click **OK** to save the settings.
5. To add a DSCP-based policy route:
   a. Under the 'DSCP-Based Policy Routing' group, click the **New Route** link; the following page appears:

Figure 29-5: Adding DSCP-Based Route

   b. From the 'Device' drop-down list, select the network device.

c. In the 'DSCP' field, specify the DSCP value. All traffic matching this DSCP value is routed to the selected device.

d. Click **OK** to save the settings.

Note: The DSCP-based policy routing ensures that specified traffic is routed via a certain WAN device, but if this WAN device is defined as the default route, other traffic may also be routed through it. If you want your device to be dedicated to transmitting only traffic matching the DSCP value you specified, you must clear the check box corresponding to the default route for that device.

6. To enable failover between multiple WAN devices:

a. Under the 'Failover' group, select the 'Enabled' check box; the failover table appears under the group.

b. Click the **Add Device** link; the following page appears:

Figure 29-6: Defining Failover between WAN Devices

c. From the 'Device' drop-down list, select the WAN device to configure as failover.

d. Select the 'Rollover Connection' check box to configure the WAN device as a rollover connection type of failover.

e. Select the 'Use DNS Lookup to Check Connectivity' check box to enable periodic connectivity check using a DNS query, and then in the 'DNS Lookup Host' field, enter the URL that the periodic check queries.

f. Click **OK** to save the settings.

7. To enable Internet Group Management Protocol (IGMP) multicasting:

a. Under the 'Internet Group Management Protocol (IGMP)' group, select the 'Enabled' check box. When a host sends a request to join a multicast group, the device listens and intercept the group's traffic, forwarding it to the subscribed host. The device keeps record of subscribed hosts. When a host requests to cancel its subscription, the device queries for other subscribers and stops forwarding the multicast group's traffic after a short timeout.

- **IGMP Fast Leave:** if a host is the only subscriber, the device stops forwarding traffic to it immediately upon request (i.e., no query delay).
- **IGMP Multicast to Unicast:** the device converts incoming multicast data stream into unicast format to route it to the specific LAN host that requested the data. Therefore, the device prevents flooding the rest of the LAN hosts with irrelevant multicast traffic.

8. To add a route entry according to the interface from which DNS record is received, under the 'Domain Routing' group, select the 'Enabled' check box. When the device's DNS server receives a reply from an external DNS server, it adds a routing entry for the IP address of the reply through the device from which it arrived. This means that future packets from this IP address are routed through the device from which the reply arrived.

9. Click **OK** to save the settings.

29.2 Configuring BGP and OSPF

The BGP and OSPF feature is an implementation of two routing protocols used to deliver up-to-date routing information to a network or a group of networks, called Autonomous System.

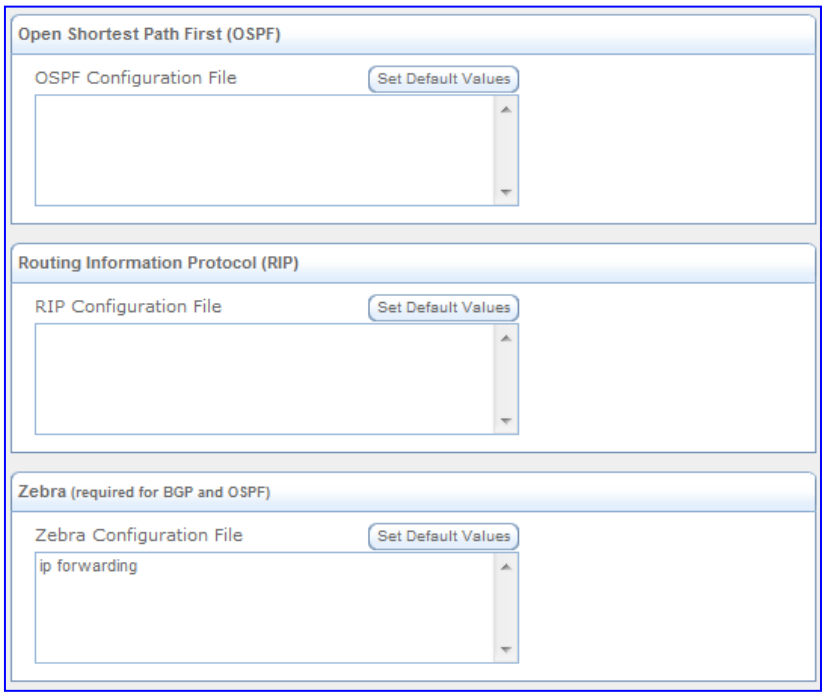
- **Border Gateway Protocol (BGP):** The main routing protocol of the Internet. It is used to distribute routing information among Autonomous Systems (for more information, refer to the protocol's RFC at http://www.ietf.org/rfc/rfc1771.txt).
- **Open Shortest Path First Protocol (OSPF):** An Interior Gateway Protocol (IGP) used to distribute routing information within a single Autonomous System (for more information, refer to the protocol's RFC at http://www.ietf.org/rfc/rfc2328.txt). The feature's routing engine is based on the Quagga GNU routing software package. By using the BGP and OSPF protocols, this routing engine enables the device to exchange routing information with other routers within and outside an Autonomous System.

If the OSPF daemon is activated, the device starts sending the 'Hello' packets to other routers to create adjacencies. After determining the shortest path to each of the neighboring routers, Zebra updates the routing table according to the network changes. If the BGP deamon is activated, the device starts to advertise routes it uses to other BGP-enabled network devices located in the neighboring Autonomous System(s). The BGP protocol uses TCP as its transport protocol. Therefore, the device first establishes a TCP connection to routers with which it communicates. KeepAlive messages are sent periodically to ensure the liveness of the connection. When a change in the routing table occurs, the device advertises an Update message to its peers. This update message adds a new route or removes the unfeasible one from their routing table.

➢ **To configure dynamic routing:**

1. Click the **Dynamic Routing** item (**Configuration** tab > **Data** menu > **Data Routing** submenu > **Dynamic Routing**); the following page appears:

Figure 29-7: Page Displaying Area for Configuration File



2. Create a configuration file for the protocol daemon and also for Zebra. Zebra is Quagga's IP routing management daemon which provides kernel routing table

updates, interface lookups, and redistribution of routes between the routing protocols. To view examples of the configuration files, browse to http://www.quagga.net/docs/quagga.pdf.

3. Enter the configuration files in their respective code fields. Alternatively, click the **Set Default Values** button located to the right of each code field. The default values displayed are as follows:

   - **BGP:**

     ```
     !router bgp <AS number>  ; The exclamation mark is Quagga's
     comment character. The router bgp string is a command that
     activates the BGP daemon. The exclamation mark emphasizes
     that the command must be followed by an exact Autonomous
     System's ID number.
     log syslog ;  instructs the daemon to send its log messages
     to the system log.
     ```

   - **OSPF:**

     ```
     router ospf   ; activates the OSPF daemon
     log syslog
     ```

   - **Zebra:**

     ```
     interface eth1  ; instructs the daemon to query and update
     routing information via a specific WAN device
     log syslog
     ```

4. Click **OK** to save the settings.

30 Objects and Rules

The **Objects and Rules** submenu allows you to configure objects and rules. Once defined, they can later be used in other configurations (e.g., in Access List rules). This submenu includes the following items:

- **Protocols** (see 'Configuring Protocols' on page 629)
- **Network Objects** (see 'Configuring Network Objects' on page 630)

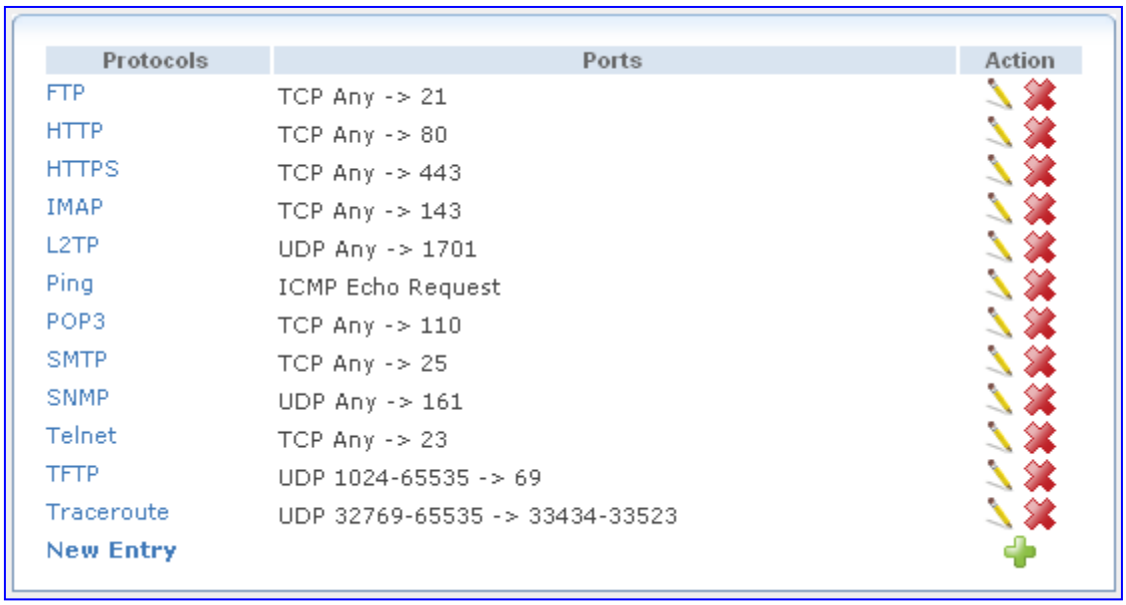
30.1 Configuring Protocols

The **Protocols** item displays a list of preset and user-defined applications and common port settings. You can use protocols in various security features such as Port Forwarding. You can add new protocols to support new applications or edit existing ones according to your needs.

➢ **To define a protocol:**

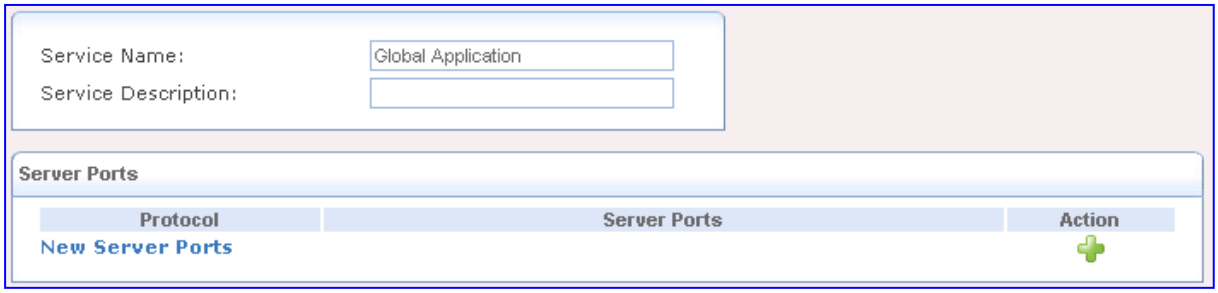
1. Click the **Protocols** item (**Configuration** tab > **Data** menu > **Objects and Rules** submenu > **Protocols**); the following page appears:

Figure 30-1: Viewing Pre-defined Protocols



2. Click the **Advanced** button for a complete list of the supported protocols.
3. Click the **New Entry** link; the following page appears:

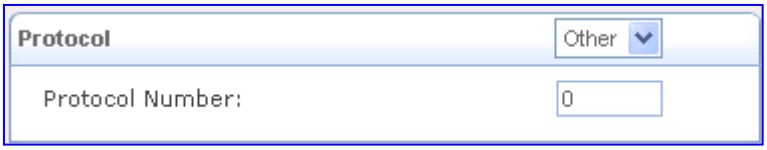
Figure 30-2: Adding a Service Protocol



4. In the 'Service Name' field, enter a name for the service protocol.

5. In the 'Service Description' field, enter a brief description of this service.
6. Click the **New Server Ports** link; the following page appears:

Figure 30-3: Defining Service Server Ports



7. From the 'Protocol' drop-down list, select any of the protocols available, or add a new one by selecting **Other**.
8. Enter the relevant information, and then click **OK** to save the settings.

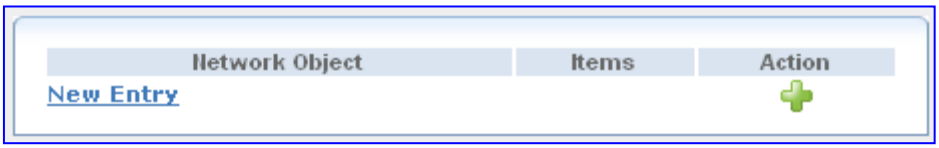
30.2 Configuring Network Objects

Network Objects is a method used to logically define a set of LAN hosts according to specific criteria such as MAC address, IP address, or host name. Defining such a group can assist when configuring system rules. For example, network objects can be used when configuring security filtering such as IP address, host name, or MAC address filtering. You can use network objects to apply security rules based on host names instead of IP addresses. This may be useful, since IP addresses change from time to time. It is also possible to define network objects according to MAC addresses, making rule application more persistent against network configuration settings. Moreover, the device supports DHCP Options 60, 61, and 77. DHCP Option 60 enables application of security and QoS rules on a network object according to its unique vendor class ID. For example, a vendor's IP telephone can be identified and applied with specific QoS priority rules.

➢ **To define a network object:**

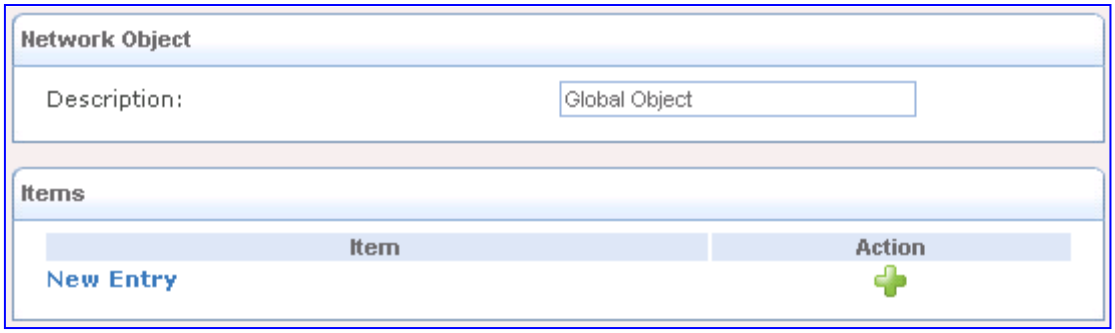
1. Click the **Network Objects** item (**Configuration** tab > **Data** menu > **Objects and Rules** submenu > **Network Objects**); the following page appears:

Figure 30-4: Configuring Network Objects



2. Click the **New Entry** link; the following page appears:

Figure 30-5: Defining Name for Network Object



3. In the 'Description' field, enter a name for the network object.

4. Click the **New Entry** link; the following page appears:

Figure 30-6: Defining Network Object Type

5. From the 'Network Object Type' drop-down list, select a network object type; the page displays the respective fields for entering the relevant information. The group definition can be according to one of the following:
   - IP Address: enter an IP address common to the group.
   - IP Subnet: enter a subnet IP address and a subnet mask.
   - IP Range: enter first and last IP addresses in the range.
   - MAC Address: enter a MAC address and mask.
   - Host Name: enter a host name common to the group.
   - DHCP Option: enter a vendor class ID.
6. Click **OK** to save the settings.

Reader's Notes

31 Configuring Network Connections

The device supports various network connections, both physical and logical. The **Data System > Connections** item enables you to configure the various parameters for your physical connections, the LAN and WAN, and create new connections, using tunneling protocols over existing connections such as PPP and VPN.

Every network connection in the device can be configured as one of three types: WAN, LAN or DMZ (Demilitarized). For example, you may define that a LAN connection on the device operates as a WAN network. This means that all hosts in this LAN are referred to as WAN computers, both by computers outside the device and by the device itself. WAN and firewall rules may be applied, such as on any other WAN network. Another example is that a network connection can be defined as a DMZ network. Although the network is physically inside the device, it functions as an unsecured, independent network for which the device merely acts as a router.

Note: When defining a network connection as a DMZ network:

- Change the connection's routing mode to 'Route'.
- Add a routing rule on your external gateway (which may be with your ISP) informing of the DMZ network behind the device.

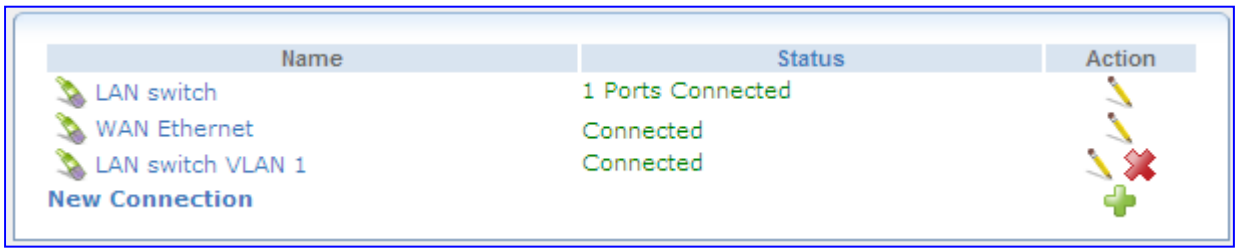
You can configure the following network connections:

- WAN (Internet connection):
  - Point-to-Point Protocol over Ethernet
  - Ethernet Connection
  - Point-to-Point Tunneling Protocol
  - Layer 2 Tunneling Protocol
  - Dynamic Host Configuration Protocol
  - Manual IP Address Configuration
  - 3G Cellular Modem (USB Modem)
- Virtual Private Network over the Internet:
  - Layer 2 Tunneling Protocol over Internet Protocol Security
  - Layer 2 Tunneling Protocol Server
  - Point-to-Point Tunneling Protocol Virtual Private Network
  - Point-to-Point Tunneling Protocol Server
  - Internet Protocol Security
  - Internet Protocol Security Server
- LAN Ethernet switch
- Advanced connections:
  - LAN Bridging
  - VLAN Interface
  - Internet Protocol over Internet Protocol
  - General Routing Encapsulation

➢ **To access the Network Connection list table:**

- Click the **Connections** item (**Configuration** tab > **Data** menu > **Data System** submenu > **Connections**); the following page appears:

Figure 31-1: Configuring Network Connections



The page displays the configured WAN and LAN connections:

- **LAN Switch:** represents all the device's ports.
- **WAN Ethernet:** connects the device to another network either directly or through an external modem.

31.1 Network Connection Wizard

The logical network connections can easily be created using the Connection wizard. This wizard consists of a series of Web pages, intuitively structured to gather all the information needed to create a logical connection. The procedure for configuring a connection using the wizard is described below.

➢ **To create a connection using the wizard:**

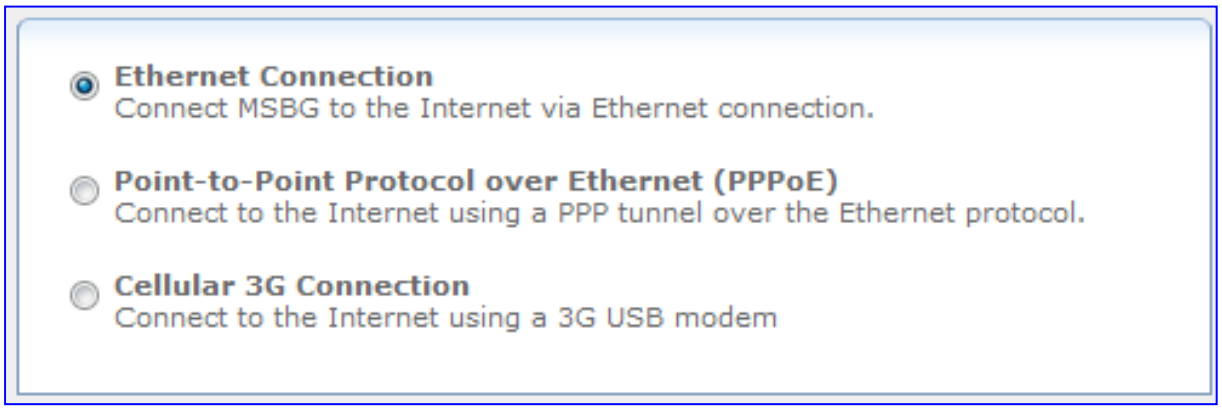
1. In the Connections page, click the **New** icon; the wizard appears.

Figure 31-2: Defining a New Connection

2. Select whether you want to configure an Internet connection, a VPN connection, or advanced connections:
   - **For configuring an Internet connection:**

a. Select the **Internet Connection** option, and then click **Next**; the following wizard page appears:

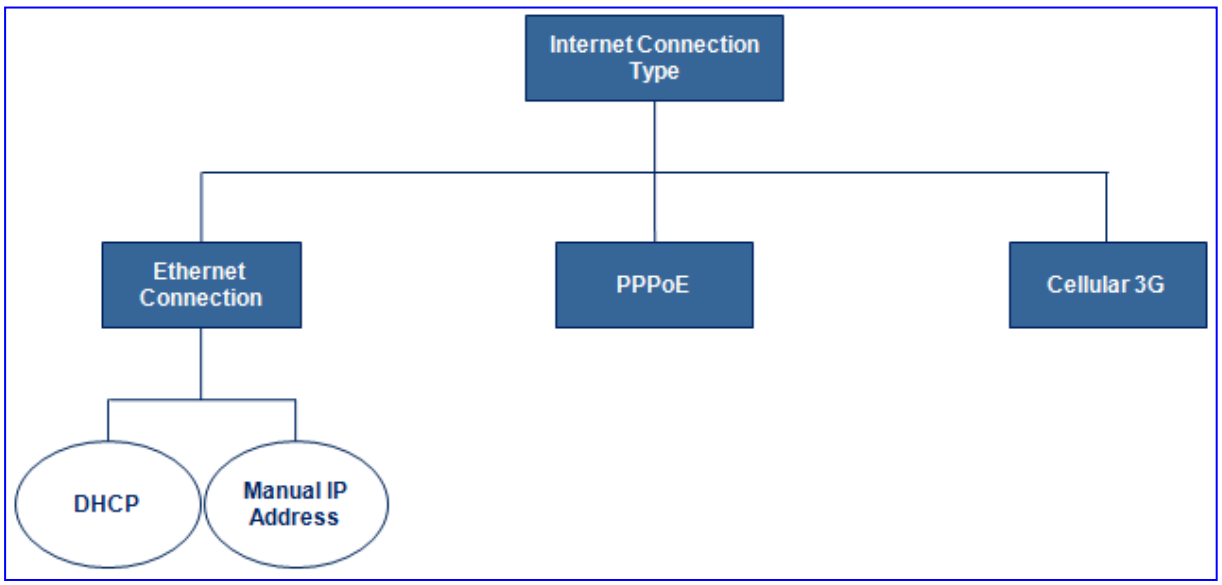
Figure 31-3: Defining Internet Connection Type



b. Select the required Internet connection type, click **Next**, and then follow the instructions provided by the wizard.

The tree-like structure of the Internet Connection options are shown below:

Figure 31-4: Internet Connection Types

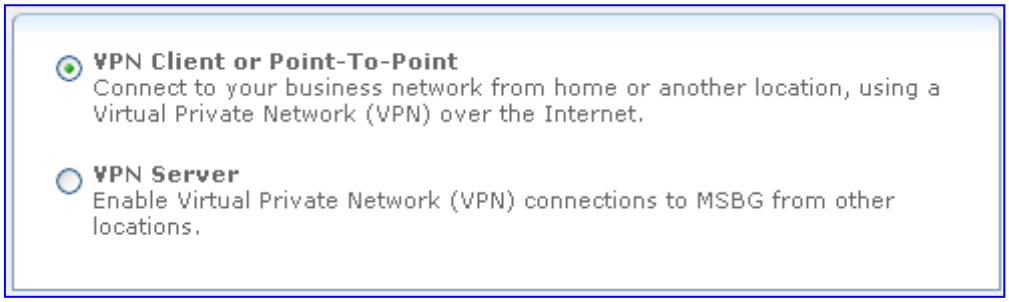


- **Point-to-Point Protocol over Ethernet (PPPoE):** relies on two widely accepted standards, PPP and Ethernet. PPPoE enables your home network PCs that communicate on an Ethernet network to exchange information with PCs on the Internet. PPPoE supports the protocol layers and authentication widely used in PPP and enables a point-to-point connection to be established in the normally multipoint architecture of Ethernet. A discovery process in PPPoE determines the Ethernet MAC address of the remote device in order to establish a session.
- **Ethernet connection:** configures the physical WAN Ethernet connection. It is the most basic method intended for connections that do not require user name and password to connect to the Internet. The IP address can be assigned automatically using a DHCP server or manually defined.
- **Cellular 3G:** The device can connect to the internet using 3G cellular USB modem. This can serve as the primary internet connection or the backup connection in case the primary connection fails.

- **For configuring a VPN-over-Internet connection:**

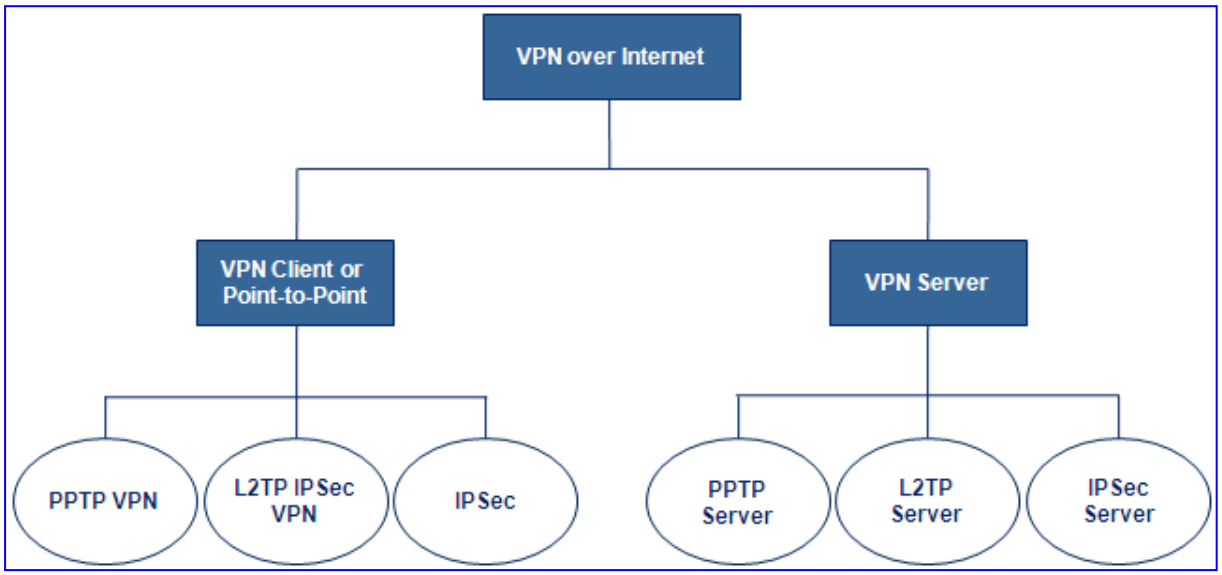
a. Select the **Connect to a Virtual Private Network over the Internet** option, and then click **Next**; the following wizard page appears:

Figure 31-5: Defining Virtual Private Network over Internet



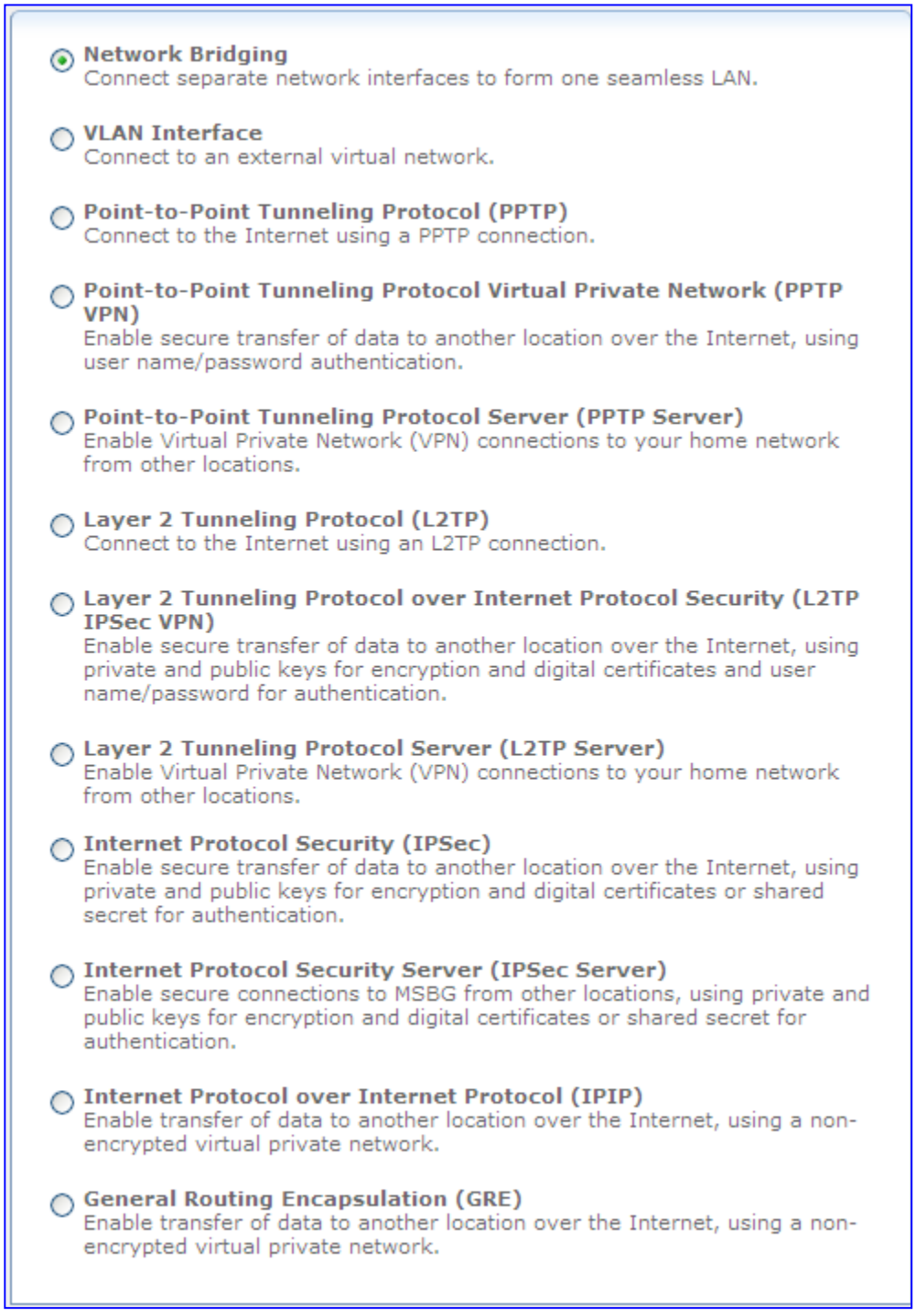
b. Select the VPN connection type, click **Next**, and then follow the instructions provided by the wizard. You can connect the device to a business network using a VPN so you can work from home, your workplace or another location. The device can either act as a VPN server (accepting VPN client connection requests) or a VPN client. The VPN over Internet options are shown below:

Figure 31-6: VPN Connection Types



- **For manually configuring a new connection:**

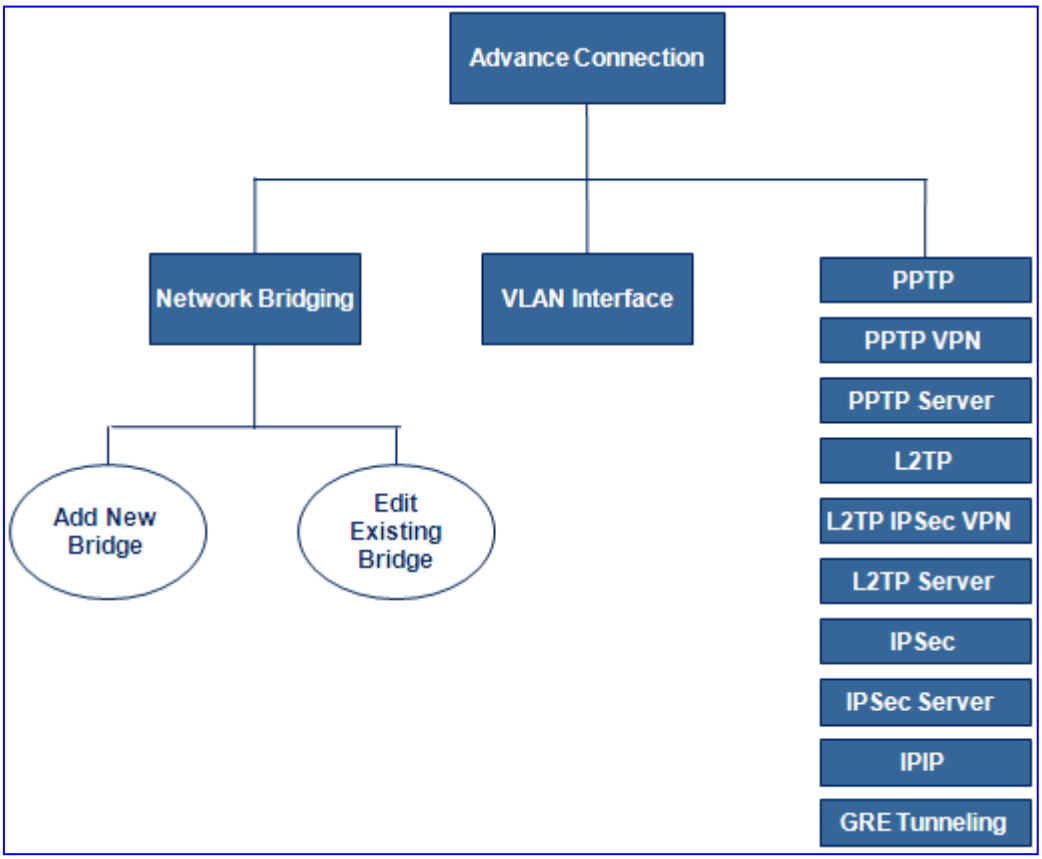
  a. Select the **Advanced Connection** option, and then click **Next**; the following wizard page appears:



b. Select the required connection type, click **Next**, and then follow the instructions provided by the wizard.

The Advanced Connection wizard tree is illustrated below:

Figure 31-7: Advanced Connection Wizard Tree



3. When the wizard completes the initial configuration (by clicking **Finish**), the new connection type appears listed in the Network Connections page.

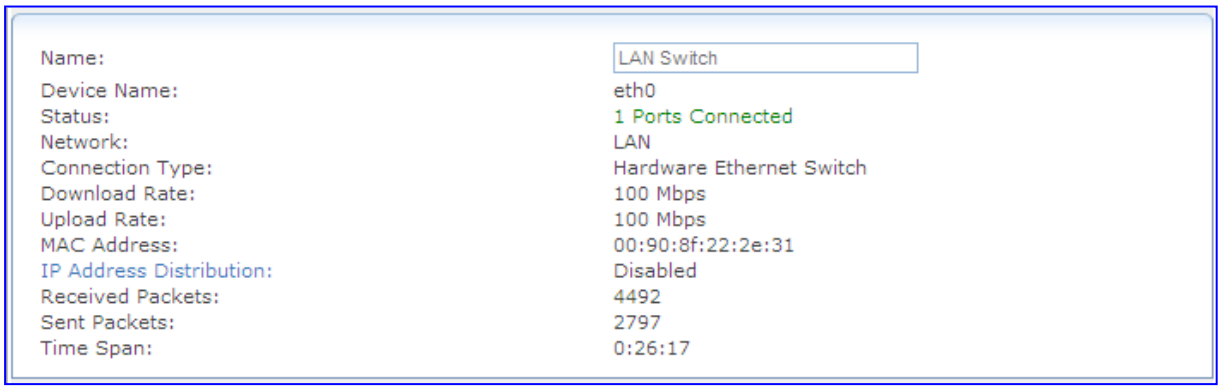
31.2 LAN Switch

The LAN Switch interface represents all the device's ports. The switch ports are physical sockets on the device to which different cables connect. You can assign VLAN's to each port.

➢ **To view and edit LAN switch ports:**

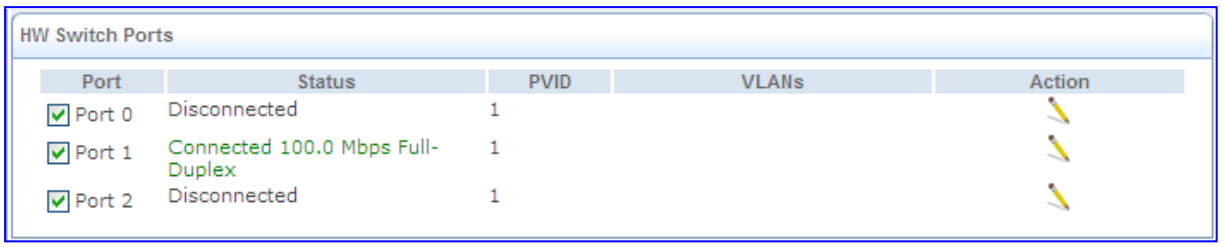
1. In the Connections page, click the **Edit** icon corresponding to the **LAN switch** connection; the **General** tab page is displayed.

Figure 31-8: General Tab - LAN Switch



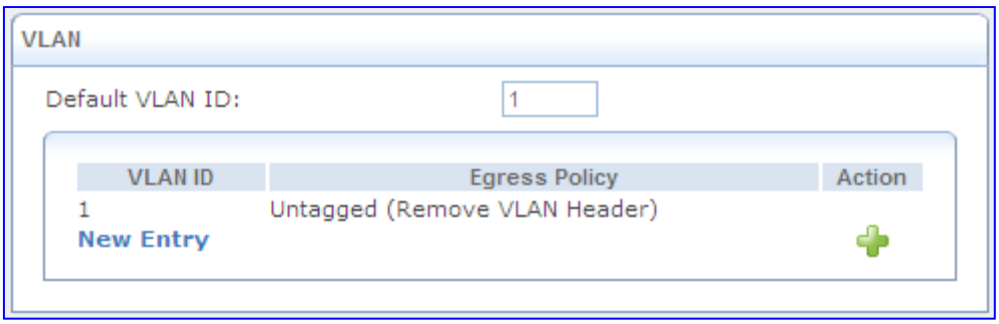
2. Select the **Switch** tab; the displayed table lists all available ports, their status, and the VLANs of which they are members. Untagged packets (packets with no VLAN tag) that arrive at a port are tagged with the VLAN number that appears under the PVID (Port VLAN Identifier) column.

Figure 31-9: Switch Tab



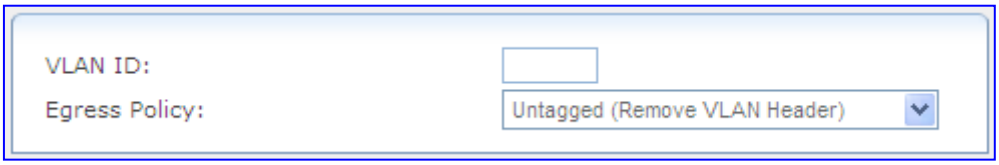
You can edit the configuration of each port by clicking the corresponding **Edit** icon.

Figure 31-10: Assigning VLAN to Port



a. In the 'Default VLAN ID' field, enter the port's VLAN identifier. You may add additional identifiers to the VLAN by clicking the **New** icon.

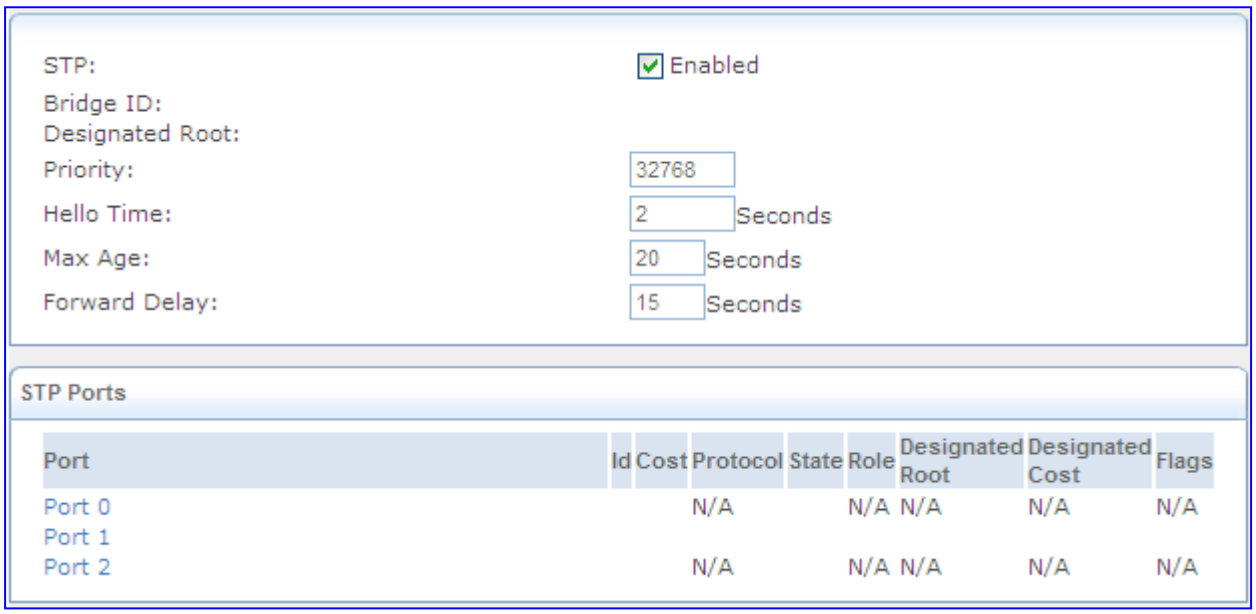
Figure 31-11: Defining VLANs



b. In the 'VLAN ID' field, enter the new VLAN ID.
c. From the 'Egress Policy' drop-down list, select whether or not to tag incoming packets with the port's VLAN header.
d. Click **OK**.

3. Select the **STP** tab.

Figure 31-12: STP Tab



4. Select the 'STP' check box to enable the Spanning Tree Protocol on the device. You should use this to ensure that there are no loops in your network configuration, and apply these settings in case your network consists of multiple switches, or other bridges apart from those created by the device.
   - Bridge ID: Identifies the bridge priority and MAC address.
   - Designated Root: Indicates the ID of the bridge with the lowest path cost to the instance ID.
   - Priority: Specifies the bridge priority value. When switches or bridges are running STP, each is assigned a priority. After exchanging BPDUs, the device with the lowest priority value becomes the root bridge. The default value is 32768. The port priority value is provided in increments of 4096. For example, 4096, 8192, 12288, and so on. The range is 0 to 65535.
   - Hello Time: Specifies the device Hello Time. The Hello Time indicates the amount of time (in seconds) a root bridge waits between configuration messages. The default is 2 seconds. The range is 1 to 10 seconds.
   - Max Age: Specifies the device Maximum Age Time. The Maximum Age Time indicates the amount of time (in seconds) a bridge waits before sending configuration messages. The default is 20 seconds. The range is 6 to 40 seconds.

- Forward Delay: Specifies the device forward delay time. The Forward Delay Time indicates the amount of time (in seconds) a bridge remains in a listening and learning state before forwarding packets. The default is 15 seconds. The range is 4 to 30 seconds.
- STP Ports:
  - Priority: Priority value of the port. The priority value influences the port choice when a bridge has two ports connected in a loop. The priority value is between 0 and 240. The priority value is in increments of 16.
  - Cost: Indicates the cost of the port participating in the STP topology. Ports with a lower cost are less likely to be blocked if STP detects loops.
  - Point-to-Point: Specifies if a point-to-point links is established, or permits the device to establish a point-to-point link. The possible field values are Enable, Disable, or Auto.
  - Edge: Specifies if a edge links is established, or permits the device to establish a point-to-point link. The possible field values are Enable, Disable, or Auto.

31.3 WAN Ethernet

The WAN Ethernet connection can connect the device to another network directly or via an external modem. The Connection Wizard provides three methods to configure this connection:

- Dynamic Host Configuration Protocol (see 'Dynamic Host Control Protocol' on page 641)
- Manual IP Address Configuration (see 'Manual WAN IP Address' on page 642)
- Point-to-Point Protocol over Ethernet (see 'Point-to-Point Protocol over Ethernet (PPPoE)' on page 643)

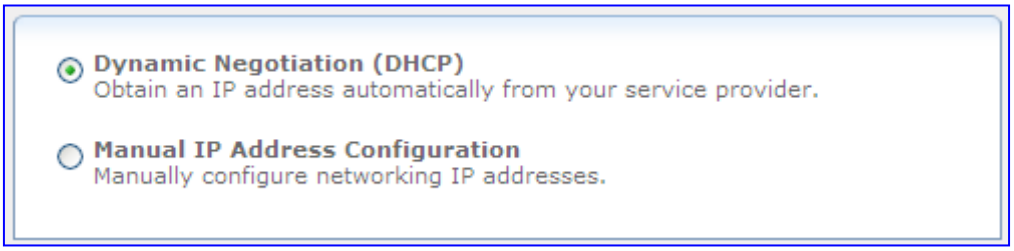
31.3.1 Dynamic Host Control Protocol

The Dynamic Host Configuration Protocol (DHCP) connection wizard utility is one of the three methods used to configure the physical WAN Ethernet connection. It is a dynamic negotiation method, where the client obtains an IP address automatically from the service provider when connecting to the Internet.

➢ **To configure a DHCP connection:**

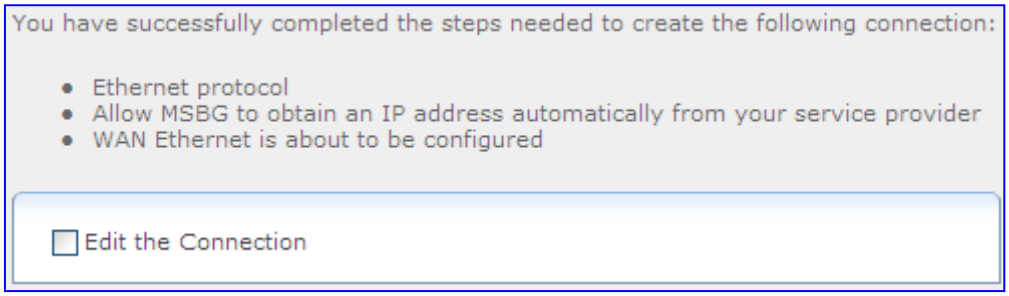
1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Internet Connection** option, and then click **Next**; the Internet connection options are displayed.
3. Select the **Ethernet Connection** option, and then click **Next**; the following page appears.

Figure 31-13: Selecting Internet Ethernet Connection



4. Select the **Dynamic Negotiation (DHCP)** option, and then click **Next**; a summary of the new connection is shown.

Figure 31-14: Internet Connection for External Cable Modem Added



5. Select the 'Edit the Connection' check box if you want to edit the new connection after clicking **Finish**.
6. Click **Finish** to save the settings; the WAN Ethernet connection is added and an IP address is obtained from a DHCP server.

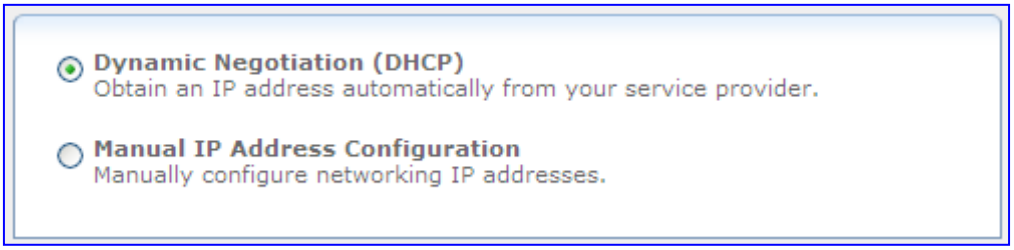
31.3.2 Manual WAN IP Address

The Manual IP Address Configuration connection wizard utility is one of the three methods used to configure the physical WAN Ethernet connection. It is used to manually configure the networking IP addresses when connecting to the Internet.

➢ **To manually configure the IP address:**

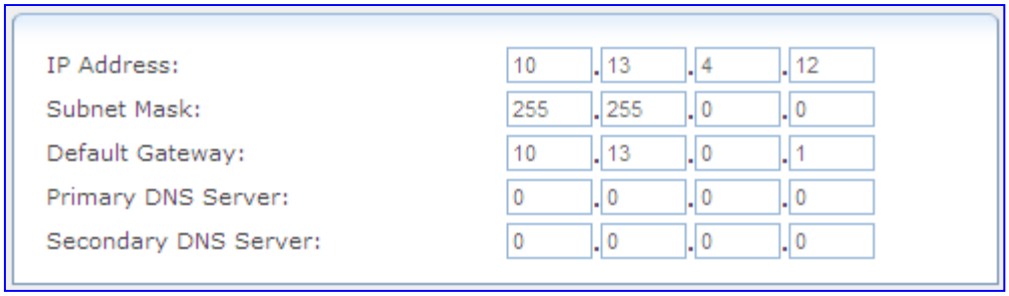
1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Internet Connection** option, and then click **Next**; the Internet connection options are displayed.
3. Select the **Ethernet Connection** option, and then click **Next**; the following page appears.

Figure 31-15: Selecting Internet Ethernet Connection



4. Select the **Manual IP Address Configuration** option, and then click **Next**; a summary of the new connection is shown.

Figure 31-16: Manual IP Address Configuration

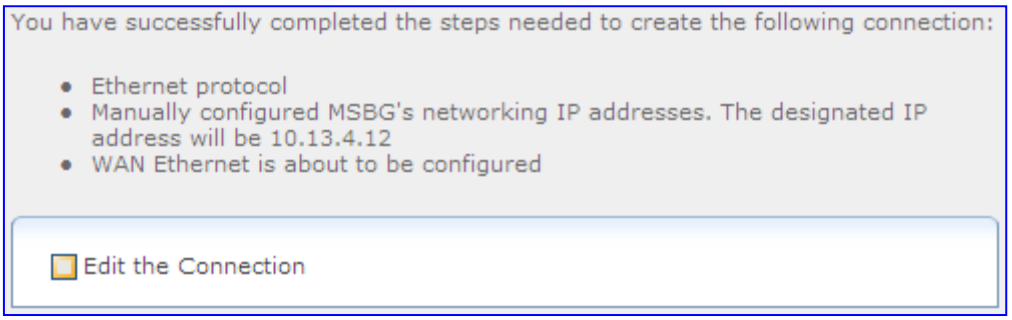


5. Enter the IP address, subnet mask, default gateway, and DNS server addresses in their respective fields. These values should either be provided to you by your ISP or

configured by your system administrator.

6. Click **Next**; a summary of the new connection is shown.

Figure 31-17: Manual WAN Ethernet Added Successfully



7. Select the 'Edit the Connection' check box if you want to edit the new connection after clicking **Finish**.
8. Click **Finish** to save the settings; the WAN Ethernet connection is added.

31.3.3 Point-to-Point Protocol over Ethernet (PPPoE)

Point-to-Point Protocol over Ethernet (PPPoE) relies on two widely accepted standards, PPP and Ethernet. PPPoE enables your home network PCs that communicate on an Ethernet network to exchange information with PCs on the Internet. PPPoE supports the protocol layers and authentication widely used in PPP and enables a point-to-point connection to be established in the normally multi-point architecture of Ethernet. A discovery process in PPPoE determines the Ethernet MAC address of the remote device in order to establish a session.

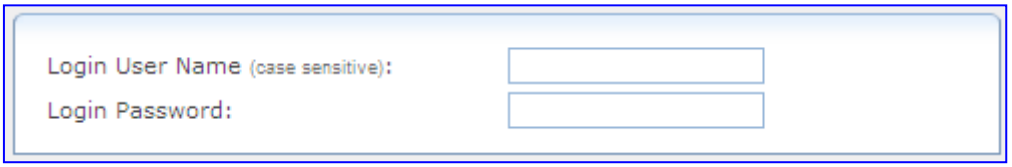
➢ **To create a PPPoE connection:**

1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Internet Connection** option, and then click **Next**.
3. Select the **Point-to-Point Protocol over Ethernet (PPPoE)** option, and then click **Next**.

Figure 31-18: Selecting Underlying Device

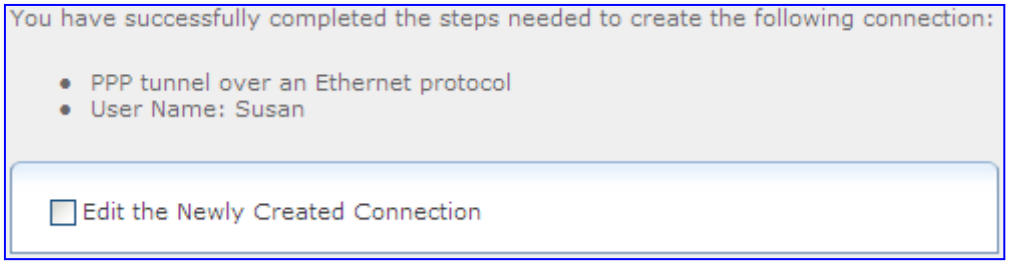
4. Select the underlying device for which you want to create the connection, and then click **Next**.

Figure 31-19: Defining PPPoE Properties



5. In the 'Login User Name' and 'Login Password' fields, enter the PPPoE username and password provided by your ISP, and then click **Next**; the following is displayed if successfully configured:

Figure 31-20: PPPoE Connection Added Successfully



6. Select the 'Edit the Newly Created Connection' check box if you want to edit the new connection after clicking **Finish**.
7. Click **Finish** to save the settings; the new PPPoE connection is added to the network connections list.

For editing the PPPoE connection, follow the procedure below:

➢ **To edit the PPPoE connection:**

1. In the Connections page, click the **WAN PPPoE** link; the **General** tab appears displaying general properties.
2. Select the **Settings** tab to edit various settings (see 'Editing Existing Connections' on page 671).
3. Select the **Routing** tab to edit the routing parameters (see 'Editing Existing Connections' on page 671).
4. Select the **PPP** tab.
   - Service Name: Specify the networking peer's service name, if provided by your ISP.
   - On Demand: Select this check box to initiate the PPP session only when packets are sent over the Internet.
   - Time Between Reconnect Attempts: Specify the duration between PPP reconnected attempts, as provided by your ISP.
   - PPP Authentication: PPP currently supports four authentication protocols: Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and Microsoft CHAP version 1 and 2. Select the authentication protocols that the device may use when negotiating with a PPTP server. Select all the protocols if no information is available about the server's authentication protocols. Note that encryption is performed only if 'Support Microsoft CHAP', 'Support Microsoft CHAP Version 2', or both are selected.
     - Login User Name: As agreed with ISP.
     - Login Password: As agreed with ISP.
     - Support Unencrypted Password (PAP): PAP is a simple, plain-text authentication scheme. The user name and password are requested by your networking peer in plain-text. PAP, however, is not a secure authentication protocol. Man-in-the-middle attacks can easily determine the remote access client's password. PAP offers no protection against replay attacks, remote client impersonation, or remote server impersonation.
     - Support Challenge Handshake Authentication (CHAP): CHAP is a challenge-response authentication protocol that uses MD5 to hash the response to a challenge. CHAP protects against replay attacks by using an arbitrary challenge string per authentication attempt.
     - Support Microsoft CHAP: Select this check box if you are communicating with a peer that uses Microsoft CHAP authentication protocol.

- Support Microsoft CHAP Version 2: Select this check box if you are communicating with a peer that uses Microsoft CHAP Version 2 authentication protocol.

- PPP Encryption: PPP supports encryption facilities to secure the data across the network connection. A wide variety of encryption methods may be negotiated, although typically only one method is used in each direction of the link. Select the encryption methods that the device may use when negotiating with a PPTP server. Select all the methods if no information is available about the server's encryption methods. Note that PPP encryption can only be used with MS-CHAP or MS-CHAP-V2 authentication protocols.
  - Require Encryption: Select this check box to ensure that the PPP connection is encrypted.
  - Support Encryption (40 Bit Keys): Select this check box if your peer supports 40 bit encryption keys.
  - Support Maximum Strength Encryption (128 Bit Keys): Select this check box if your peer supports 128 bit encryption keys.
  - MPPE Encryption Mode: Select the Microsoft Point to Point Encryption (MPPE) mode. This is a means of representing PPP packets in an encrypted form.
  - PPP Compression: The PPP Compression Control Protocol (CCP) is responsible for configuring, enabling, and disabling data compression algorithms on both ends of the point-to-point link. It is also used to signal a failure of the compression/ decompression mechanism in a reliable manner. For each compression algorithm, select one of the following from the drop down menu:
    - ✓ Reject: Reject PPP connections with peers that use the compression algorithm.
    - ✓ Allow: Allow PPP connections with peers that use the compression algorithm.
    - ✓ Require: Ensure a connection with a peer is using the compression algorithm.

5. Select the **Advanced** tab to enable the firewall for this network connection (see 'Editing Existing Connections' on page 671).

31.4 LAN Bridge

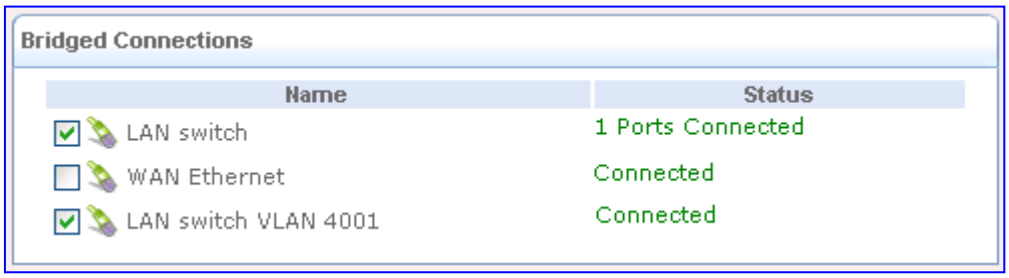
The LAN bridge connection is used to combine several LAN devices under one virtual network. Note that when a bridge is removed, its formerly underlying devices inherit the bridge's DHCP settings. For example, the removal of a bridge that is configured as DHCP client automatically configures the LAN devices formerly constituting the bridge as DHCP clients with the exact DHCP client configuration.

➢ **To create a LAN bridge:**

1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Advanced Connection** option, and then click **Next**; the Advanced Connection page appears.

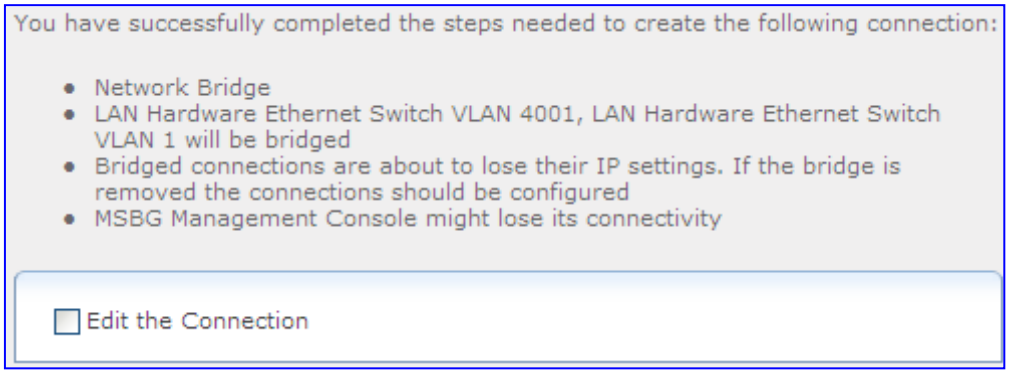
3. Select the **Network Bridging** option, and then click **Next**; the following page appears.

Figure 31-21: Selecting LAN Interfaces for Bridge Connection



4. Add new connections or remove existing ones, by selecting or clearing their respective check boxes.
5. Click **Next**; the LAN bridge is successfully added.

Figure 31-22: LAN Bridge Successfully Added



6. Select the 'Edit the Connection' check box if you want to edit the new connection after clicking **Finish**.
7. Click **Finish** to save the settings; the new bridge is added to the network connections list.

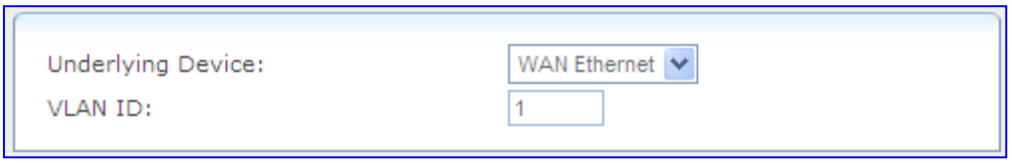
31.5 Virtual LAN Interface (VLAN)

A virtual LAN interface enables you to group workstations together into one broadcast domain, even if they are not located on the same LAN segment. The device allows you to create virtual Ethernet-based networks according to the IEEE 802.1Q standard.

➢ **To create a VLAN interface:**

1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Advanced Connection** option, and then click **Next**; the Advanced Connection page appears.
3. Select the **VLAN Interface** option, and then click **Next**; the following page appears.

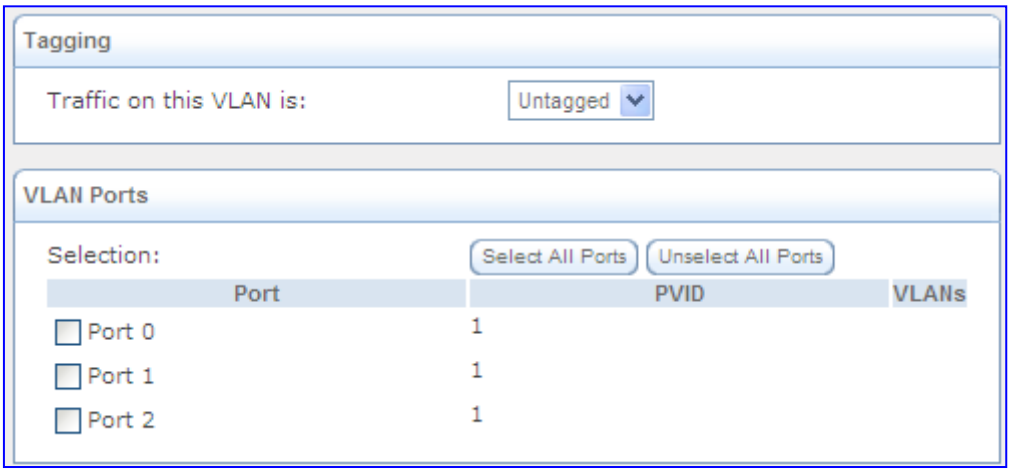
Figure 31-23: Adding a VLAN Interface



Note: By default, all the device's physical LAN devices are enslaved by the device's LAN bridge. A VLAN cannot be created over an enslaved network device. Therefore, remove a device from the bridge before creating a VLAN over it.

4. From the 'Underlying Device' drop-down list, select the underlying device for this interface. The list displays the device's Ethernet connections.
5. In the 'VLAN ID' field, enter a value for the VLAN ID, and then click **Next**. If you chose to create the VLAN over the WAN, skip to Step 9. If you chose to create the VLAN over the LAN bridge, the following page appears.

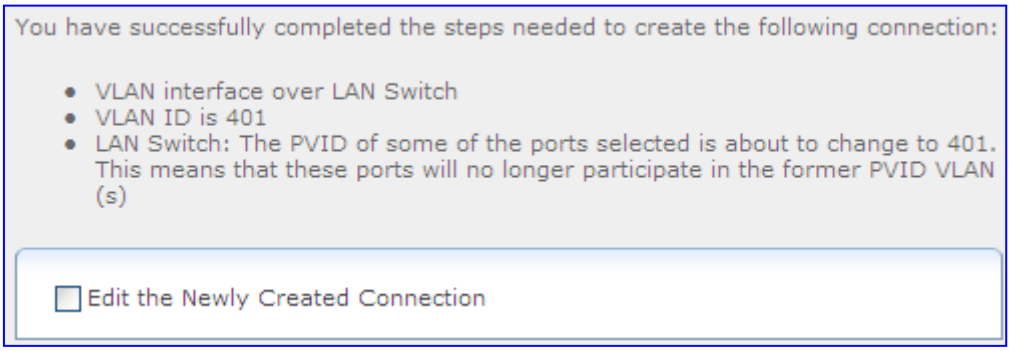
Figure 31-24: Assigning VLAN to LAN Ports



6. From the 'Traffic on this VLAN is' drop-down list, select one of the following:
   - 'Untagged': the VLAN is determined based on information such as the ID of a port on which the data arrives (PVID).
   - 'Tagged': adds a tag header (a 32-bit label serving as a VLAN ID) to the frames transferred over the VLAN.

   Note: If the created virtual network is intended for VLAN-unaware hosts, it is recommended that you select the 'Untagged' option.
7. In the 'VLAN Ports' group, select the LAN bridge ports on which you want to enable the VLAN.
8. Click **Next**; a summary of the VLAN configuration is displayed.

Figure 31-25: VLAN Added Successfully



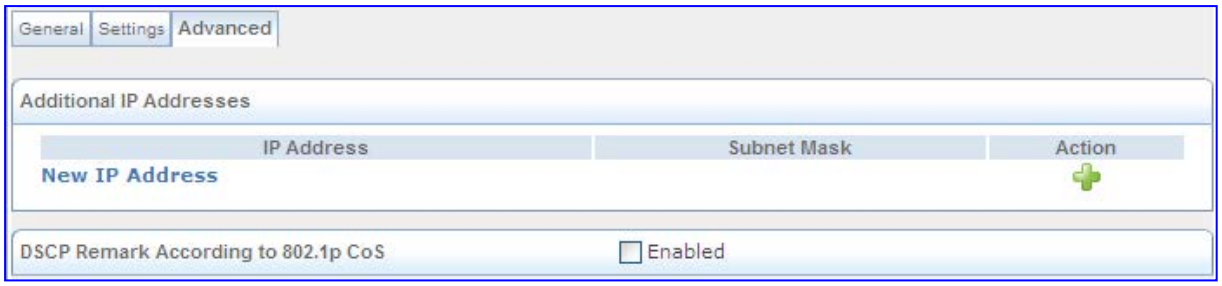
9. Select the 'Edit the Newly Created Connection' check box if you want to edit the new connection after clicking **Finish**.

10. Click **Finish** to save the settings; the new VLAN interface is added to the network connections list.

➢ **To edit the VLAN interface connection:**

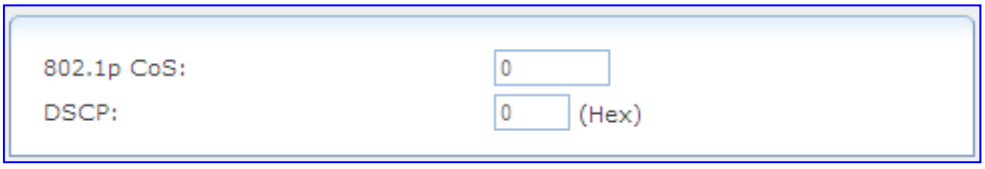
1. In the Connections page, click the VLAN link (e.g., "LAN Switch VLAN 401"); the **General** tab appears displaying general properties.
2. Select the **Settings** tab to edit various settings (see 'Editing Existing Connections' on page 671).
3. Select the **Advanced** tab.

Figure 31-26: VLAN Interface Advanced Tab



a. If your VLAN interface is over WAN, then you can enable the firewall for this network connection by selecting the 'Internet Connection Firewall' check box.
b. You can add alias names (additional IP addresses) to the device by clicking under the 'Additional IP Addresses' group the **New IP Address** link. This enables you to access the device using these aliases in addition to the device's defined IP address.
c. When creating a VLAN interface over a LAN connection, it is possible to determine the IP header's Differentiated Services Code Point (DSCP) priority value according to the VLAN header's 802.1p Class of Service (CoS) tag. The DSCP value can then be used for Quality of Service (Qos) traffic prioritization.
   a. Select the 'Enabled' check box.
   b. Click the **New DSCP Remark** link; the following page appears:

Figure 31-27: Defining DSCP Remarkings



   c. Map the required 802.1p CoS value to a DSCP value, and then click **OK**; the new value pair appears in the table.
4. Click **OK** to save the settings.

31.6 Point-to-Point Tunneling Protocol (PPTP)

Point-to-Point Tunneling Protocol (PPTP) is a protocol developed by Microsoft targeted at creating VPN connections over the Internet. This enables remote users to access the device via any ISP that supports PPTP on its servers. PPTP encapsulates network traffic, encrypts content using Microsoft's Point-to-Point Encryption (MPPE) protocol that is based on RC4, and routes using the generic routing encapsulation (GRE) protocol. PPTP is targeted at serving two purposes:

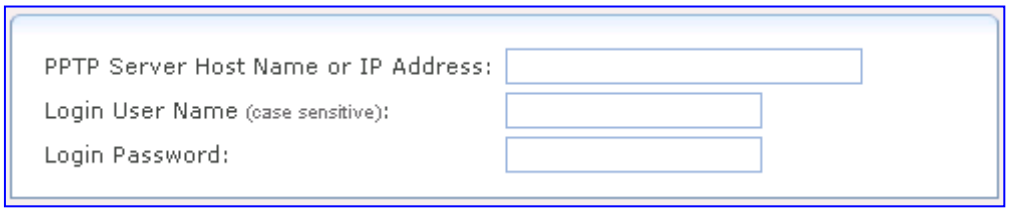
- Connecting the device to the Internet when it is used as a cable modem, or when using an external cable modem. Such a connection is established using user name and password authentication.

- Connecting the device to a remote network using a Virtual Private Network (VPN) tunnel over the Internet. This enables secure transfer of data to another location over the Internet, using user name and password authentication.

➢ **To create a PPTP connection:**

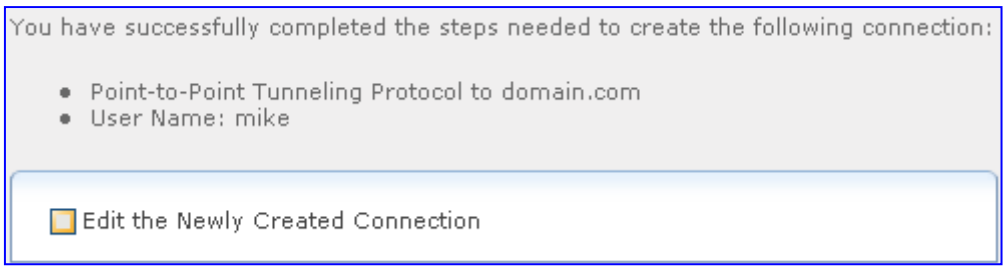
1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Advanced Connection** option, and then click **Next**; the Advanced Connection page appears.
3. Select the **Point-to-Point Tunneling Protocol (PPTP)** option, and then click **Next**; the following page appears.

Figure 31-28: Defining PPTP Properties



4. In the 'PPTP Server Host Name or IP Address' field, enter the PPTP server host name or IP address provided by your ISP.
5. In the 'Login User Name' and 'Login Password' fields, enter the username and password provided by your ISP.
6. Click **Next**; the following is displayed if successfully configured:

Figure 31-29: PPTP Connection Added Successfully



7. Select the 'Edit the Newly Created Connection' check box if you want to edit the new connection after clicking **Finish**.
8. Click **Finish** to save the settings; the new PPTP connection is added to the network connections list.

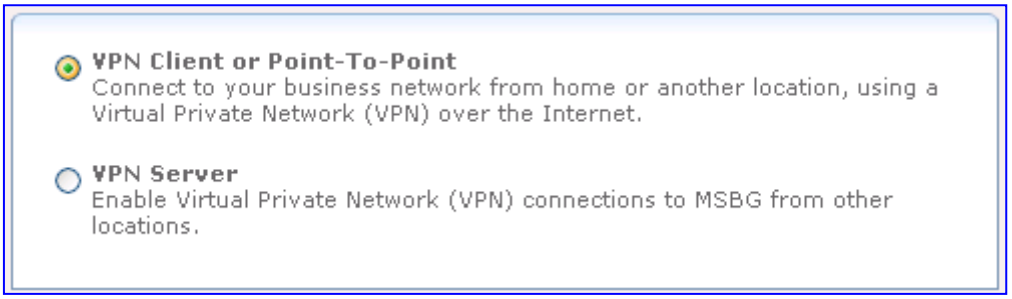
The following procedure describes how to create a PPTP VPN connection.

➢ **To create a PPTP VPN connection:**

1. In the Connections page, click the **New** icon; the Connection Wizard opens.

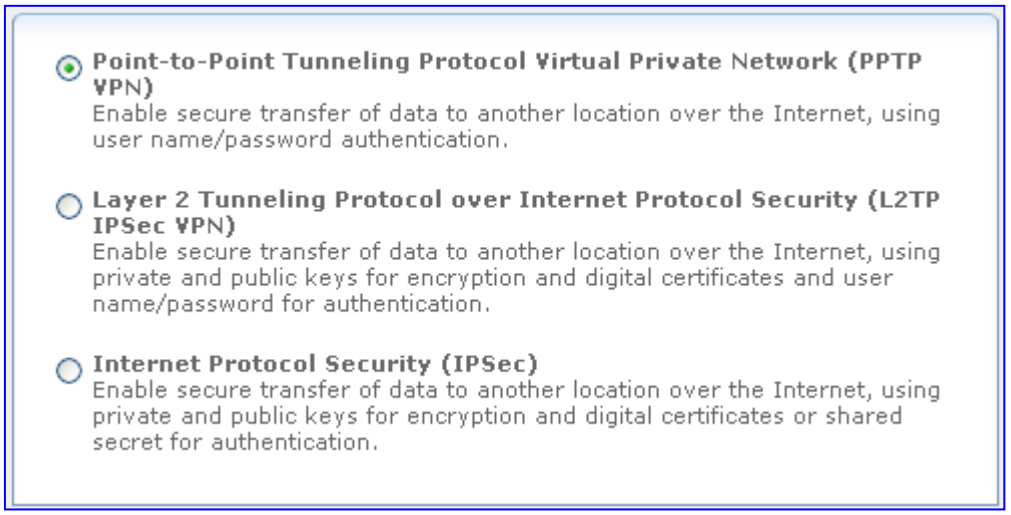
2. Select the **Connect to a Virtual Private Network over the Internet** option, and then click **Next**.

Figure 31-30: Selecting VPN Type for IPSec



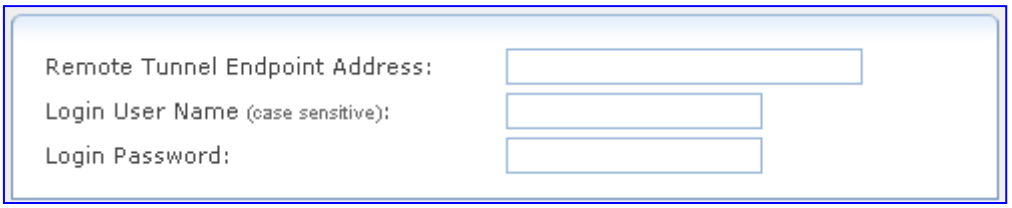
3. Select the **VPN Client or Point-To-Point** option, and then click **Next**.

Figure 31-31: Selecting Protocol to Connect to Remote VPN Server



4. Select the **Point-to-Point Tunneling Protocol Virtual Private Network (PPTP VPN)** option, and then click **Next**.

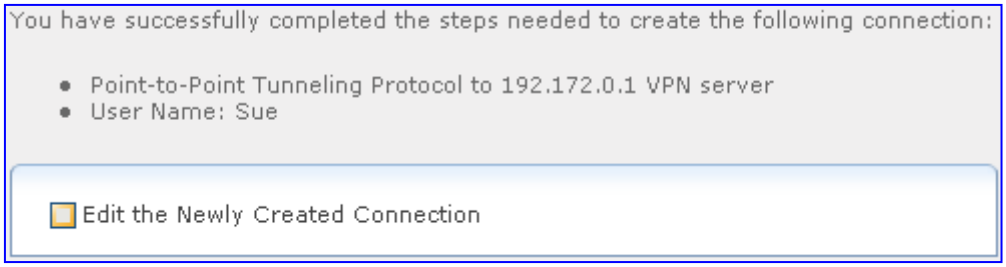
Figure 31-32: Defining PPTP VPN Properties



5. In the 'Remote Tunnel Endpoint Address' field, enter the remote tunnel endpoint address. This is the IP address or domain name of the remote network computer, which serves as the tunnel's endpoint.
6. In the 'Login User Name' and Login Password' fields, enter the username and password provided by the administrator of the network you are trying to access.

7. Click **Next**; the following is displayed if successfully configured:

Figure 31-33: PPTP VPN Successfully Added



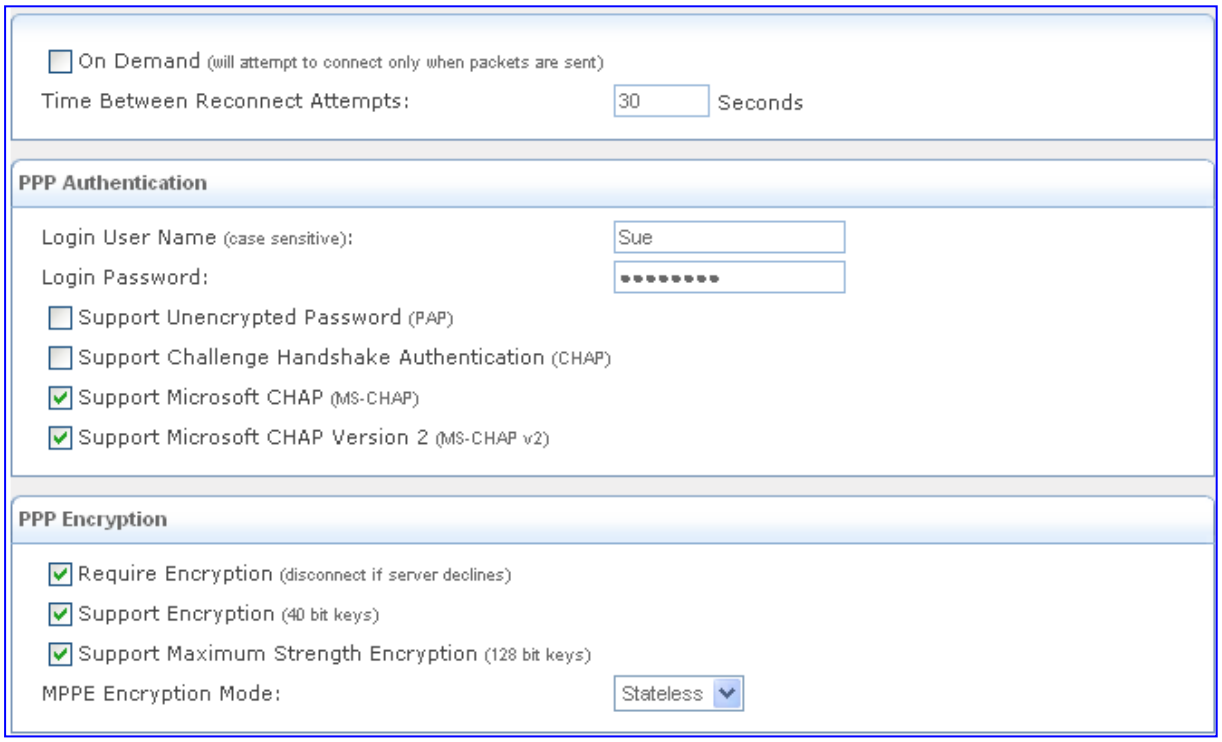
8. Select the 'Edit the Newly Created Connection' check box if you want to edit the new connection after clicking **Finish**.
9. Click **Finish** to save the settings; the new PPTP VPN connection is added to the network connections list.

To view and edit the PPTP connection, follow the procedure below:

➢ **To edit the PPTP connection:**

1. In the Connections page, click the **PPTP VPN** link; the **General** tab appears displaying general properties.
2. Select the **Settings** tab to edit various settings (see 'Editing Existing Connections' on page 671).
3. Select the **Routing** tab to edit the routing parameters (see 'Editing Existing Connections' on page 671).
4. Select the **PPP** tab.

Figure 31-34: PPP Tab

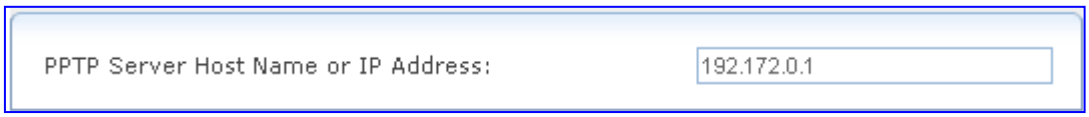


- On Demand: Select this check box to initiate the PPP session only when packets are sent over the Internet.
- Time Between Reconnect Attempts: Specify the duration between PPP reconnected attempts, as provided by your ISP.

- PPP Authentication: PPP currently supports four authentication protocols: Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and Microsoft CHAP version 1 and 2. Select the authentication protocols that the device may use when negotiating with a PPTP server. Select all the protocols if no information is available about the server's authentication protocols. Note that encryption is performed only if 'Support Microsoft CHAP', 'Support Microsoft CHAP Version 2', or both are selected.
  - Login User Name: As agreed with ISP.
  - Login Password: As agreed with ISP.
  - Support Unencrypted Password (PAP): PAP is a simple, plain-text authentication scheme. The user name and password are requested by your networking peer in plain-text. PAP, however, is not a secure authentication protocol. Man-in-the-middle attacks can easily determine the remote access client's password. PAP offers no protection against replay attacks, remote client impersonation, or remote server impersonation.
  - Support Challenge Handshake Authentication (CHAP): CHAP is a challenge-response authentication protocol that uses MD5 to hash the response to a challenge. CHAP protects against replay attacks by using an arbitrary challenge string per authentication attempt.
  - Support Microsoft CHAP: Select this check box if you are communicating with a peer that uses Microsoft CHAP authentication protocol.
  - Support Microsoft CHAP Version 2: Select this check box if you are communicating with a peer that uses Microsoft CHAP Version 2 authentication protocol.
- PPP Encryption: PPP supports encryption facilities to secure the data across the network connection. A wide variety of encryption methods may be negotiated, although typically only one method is used in each direction of the link. Select the encryption methods that the device may use when negotiating with a PPTP server. Select all the methods if no information is available about the server's encryption methods. Note that PPP encryption can only be used with MS-CHAP or MS-CHAP-V2 authentication protocols.
  - Require Encryption: Select this check box to ensure that the PPP connection is encrypted.
  - Support Encryption (40 Bit Keys): Select this check box if your peer supports 40 bit encryption keys.
  - Support Maximum Strength Encryption (128 Bit Keys): Select this check box if your peer supports 128 bit encryption keys.
  - MPPE Encryption Mode: Select the Microsoft Point to Point Encryption (MPPE) mode. This is a means of representing PPP packets in an encrypted form.

5. Select the **PPTP** tab.

Figure 31-35: PPTP Tab



- In the 'PPTP Server Host Name or IP Address' field, enter the connection's host name or IP address obtained from your ISP.

6. Select the **Advanced** tab to enable the firewall for this network connection (see 'Editing Existing Connections' on page 671).

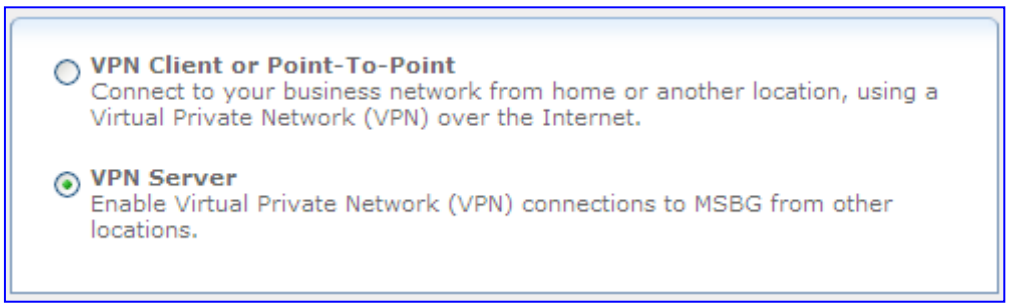
31.7 Point-to-Point Tunneling Protocol Server (PPTP Server)

The device can act as a Point-to-Point Tunneling Protocol Server (PPTP Server), accepting PPTP client connection requests.

➢ **To create a PPTP server:**

1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Connect to a Virtual Private Network over the Internet** option, and then click **Next**.

Figure 31-36: VPN Connection Type

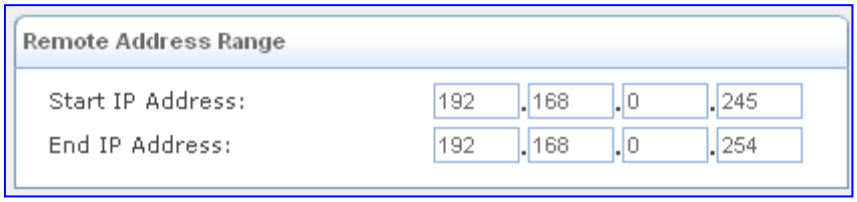


3. Select the **VPN Server** option, and then click **Next**.

Figure 31-37: Selecting the VPN Protocol - PPTP Server

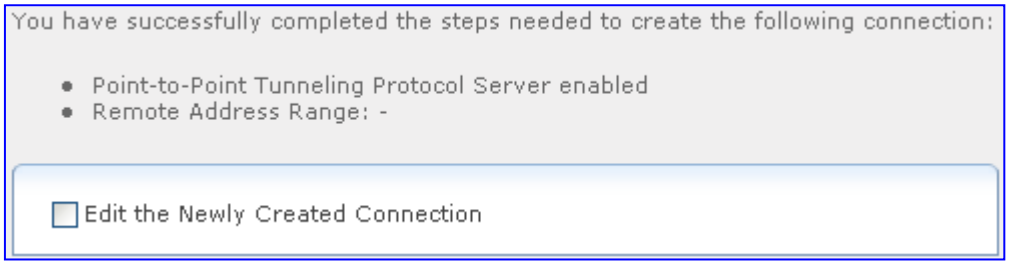
4. Select the **Point-to-Point Tunneling Protocol Server (PPTP Server)** option, and then click **Next**.

Figure 31-38: Defining Remote Client Address Range



5. Specify the IP address range that the device reserves for remote users, and then click **Next**; the following is displayed if successfully configured:

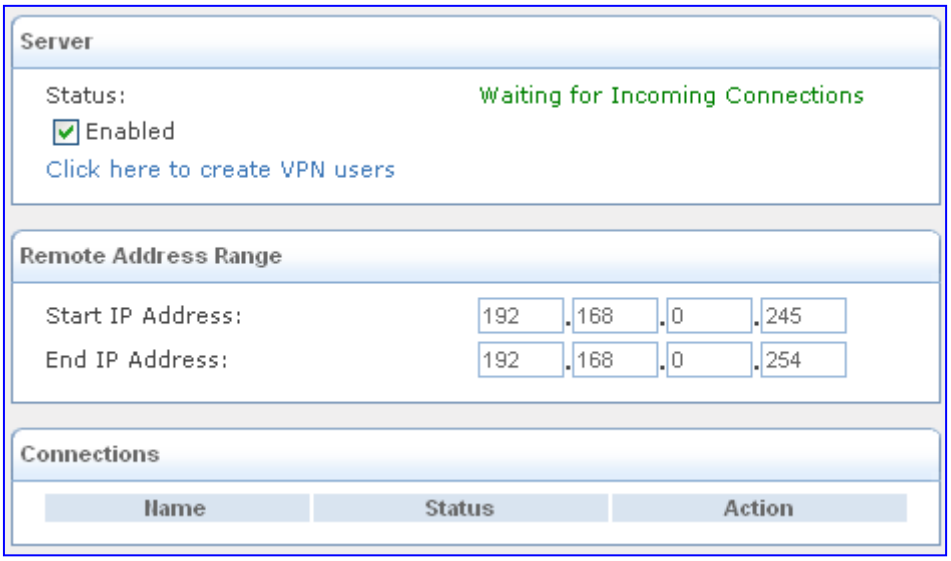
Figure 31-39: PPTP Server Added Successfully



Note that the attention message alerting that there are no users with VPN permissions.

6. Select the 'Edit the Newly Created Connection' check box, and then click **Finish**.

Figure 31-40: Editing VPN Server



7. Click the **Click here to create VPN users** link to define remote users that will be granted access to your home network.
8. Click **OK** to save settings; the new PPTP server connection is added to the Network Connection list.

31.8 Layer 2 Tunneling Protocol (L2TP)

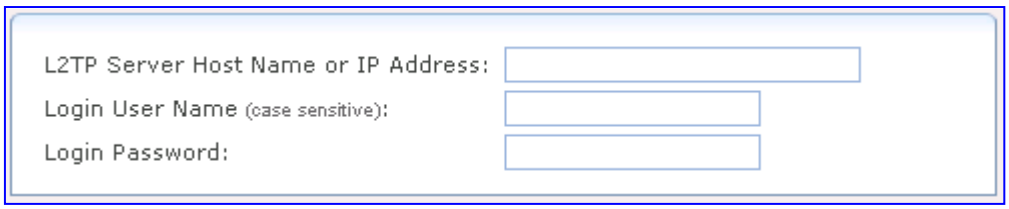
Layer 2 Tunneling Protocol (L2TP) is an extension to the PPP protocol, enabling your device to create VPN connections. Derived from Microsoft's Point-to-Point Tunneling Protocol (PPTP) and Cisco's Layer 2 Forwarding (L2F) technology, L2TP encapsulates PPP frames into IP packets either at the remote user's PC or at an ISP that has an L2TP Remote Access Concentrator (LAC). The LAC transmits the L2TP packets over the network to the L2TP Network Server (LNS) at the corporate side. L2TP is targeted at serving two purposes:

- Connecting the device to the Internet when it is used as a cable modem, or when using an external cable modem. Such a connection is established using user name and password authentication.
- Connecting the device to a remote network using a Virtual Private Network (VPN) tunnel over the Internet. This enables secure transfer of data to another location over the Internet, using private and public keys for encryption and digital certificates, and user name and password for authentication.

➢ **To create a L2TP connection:**

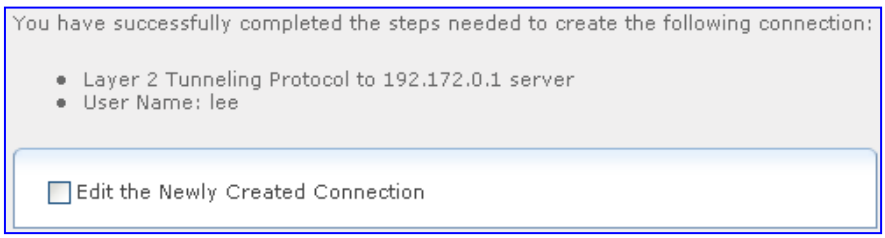
1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Advanced Connection** option, and then click **Next**; the Advanced Connection page appears.
3. Select the **Layer 2 Tunneling Protocol (L2TP)** option, and then click **Next**.

Figure 31-41: Defining L2TP Properties



4. In the 'L2TP Server Host Name or IP Address' field, enter the L2TP server host name or IP address provided by your ISP.
5. In the 'Login User Name' and 'Login Password' fields, enter the username and password provided by your ISP.
6. Click **Next**; the following is displayed if successfully configured:

Figure 31-42: L2TP Connection Added Successfully



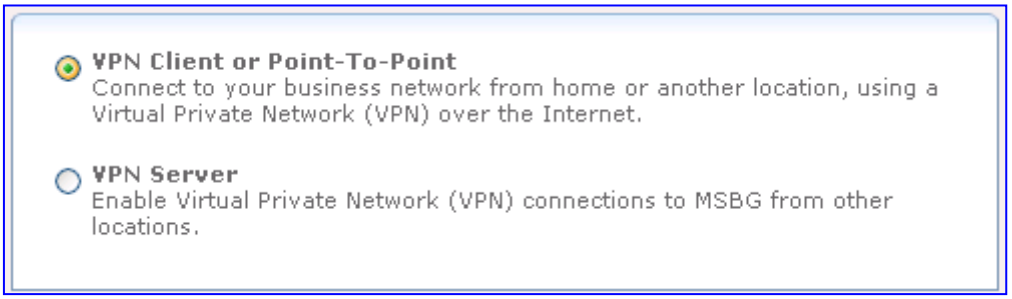
7. Select the 'Edit the Newly Created Connection' check box if you want to edit the new connection after clicking **Finish**.
8. Click **Finish** to save the settings; the new L2TP connection is added to the network connections list.

The following procedure describes how to create a L2TP VPN connection.

➢ **To create a L2TP VPN connection:**

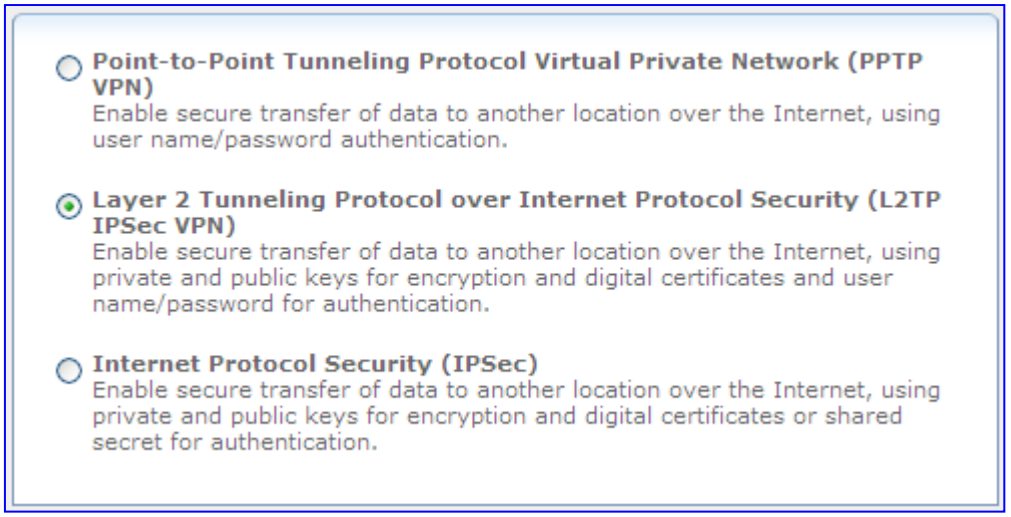
1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Connect to a Virtual Private Network over the Internet** option, and then click **Next**.

Figure 31-43: Selecting VPN Type for IPSec



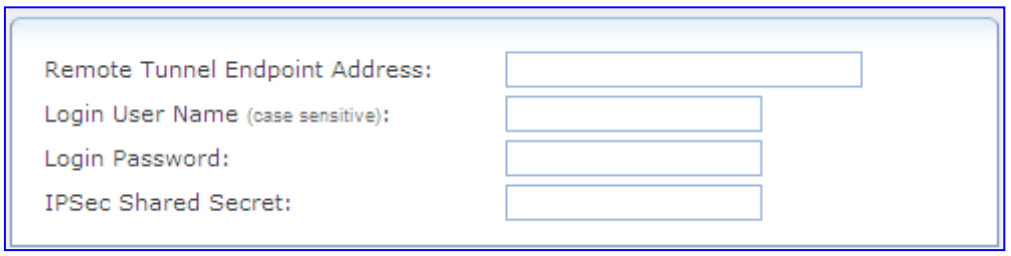
3. Select the **VPN Client or Point-To-Point** option, and then click **Next**.

Figure 31-44: Selecting L2TP to Connect to Remote VPN Server



4. Select the **Layer 2 Tunneling Protocol over Internet Protocol Security (L2TP IPSec VPN)** option, and then click **Next**.

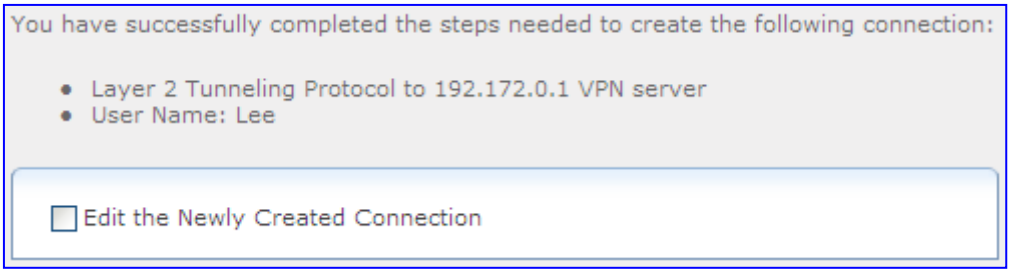
Figure 31-45: Defining L2TP Properties



5. In the 'Remote Tunnel Endpoint Address' field, enter the remote tunnel endpoint address. This is the IP address or domain name of the remote network computer, which serves as the tunnel's endpoint.
6. In the 'Login User Name' and 'Login Password' fields, enter the username and password provided by the administrator of the network you are trying to access.
7. In the 'IPSec Shared Secret' field, enter the IPSec shared secret, which is the encryption key jointly decided upon with the network you are trying to access.

8. Click **Next**; the following is displayed if successfully configured:

Figure 31-46: L2TP Successfully Added



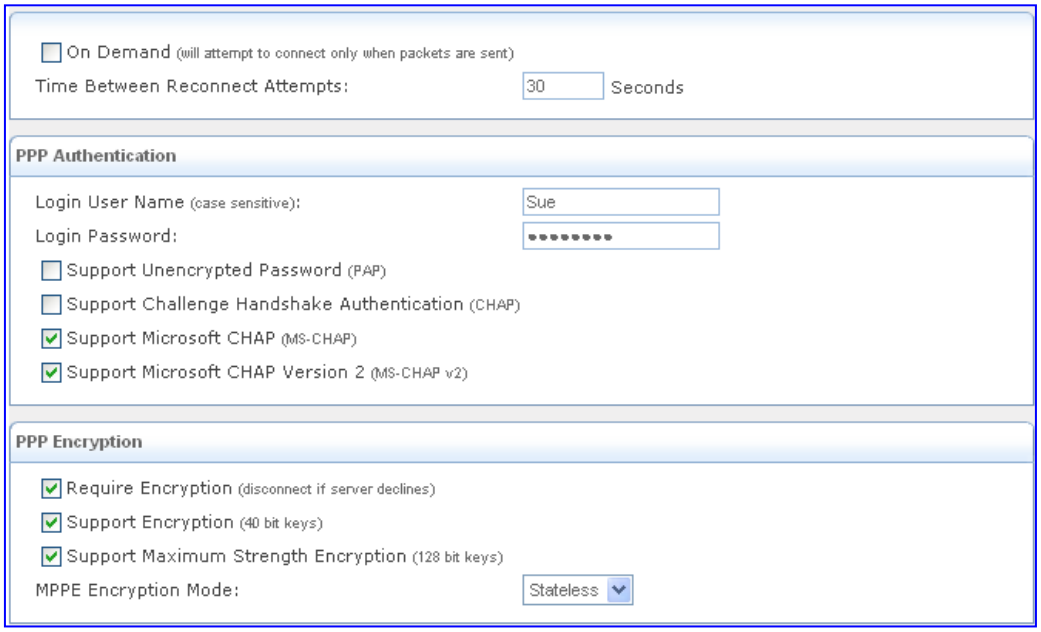
9. Select the 'Edit the Newly Created Connection' check box if you want to edit the new connection after clicking **Finish**.
10. Click **Finish** to save the settings; the new L2TP IPSec VPN connection is added to the network connections list.

To view and edit the L2TP connection, follow the procedure below:

➢ **To edit the L2TP connection:**

1. In the Connections page, click the **L2TP VPN** link; the **General** tab appears displaying general properties.
2. Select the **Settings** tab to edit various settings (see 'Editing Existing Connections' on page 671).
3. Select the **Routing** tab to edit the routing parameters (see 'Editing Existing Connections' on page 671).
4. Select the **PPP** tab.

Figure 31-47: PPP Tab

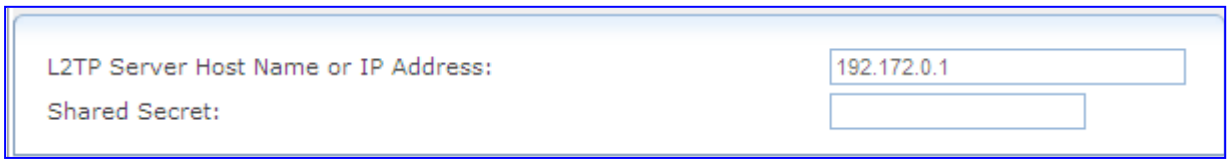


- On Demand: Select this check box to initiate the PPP session only when packets are sent over the Internet.
- Time Between Reconnect Attempts: Specify the duration between PPP reconnected attempts, as provided by your ISP.

- PPP Authentication: PPP currently supports four authentication protocols: Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and Microsoft CHAP version 1 and 2. Select the authentication protocols that the device may use when negotiating with a PPTP server. Select all the protocols if no information is available about the server's authentication protocols. Note that encryption is performed only if 'Support Microsoft CHAP', 'Support Microsoft CHAP Version 2', or both are selected.
  - Login User Name: As agreed with ISP.
  - Login Password: As agreed with ISP.
  - Support Unencrypted Password (PAP): PAP is a simple, plain-text authentication scheme. The user name and password are requested by your networking peer in plain-text. PAP, however, is not a secure authentication protocol. Man-in-the-middle attacks can easily determine the remote access client's password. PAP offers no protection against replay attacks, remote client impersonation, or remote server impersonation.
  - Support Challenge Handshake Authentication (CHAP): CHAP is a challenge-response authentication protocol that uses MD5 to hash the response to a challenge. CHAP protects against replay attacks by using an arbitrary challenge string per authentication attempt.
  - Support Microsoft CHAP: Select this check box if you are communicating with a peer that uses Microsoft CHAP authentication protocol.
  - Support Microsoft CHAP Version 2: Select this check box if you are communicating with a peer that uses Microsoft CHAP Version 2 authentication protocol.
- PPP Encryption: PPP supports encryption facilities to secure the data across the network connection. A wide variety of encryption methods may be negotiated, although typically only one method is used in each direction of the link. Select the encryption methods that the device may use when negotiating with a PPTP server. Select all the methods if no information is available about the server's encryption methods. Note that PPP encryption can only be used with MS-CHAP or MS-CHAP-V2 authentication protocols.
  - Require Encryption: Select this check box to ensure that the PPP connection is encrypted.
  - Support Encryption (40 Bit Keys): Select this check box if your peer supports 40 bit encryption keys.
  - Support Maximum Strength Encryption (128 Bit Keys): Select this check box if your peer supports 128 bit encryption keys.
  - MPPE Encryption Mode: Select the Microsoft Point to Point Encryption (MPPE) mode. This is a means of representing PPP packets in an encrypted form.

5. Select the **L2TP** tab.

Figure 31-48: L2TP Tab



- In the 'L2TP Server Host Name or IP Address' field, enter the connection's host name or IP address obtained from your ISP.
- In the 'Shared Secret' field, enter the shared secret value obtained from your ISP.

6. Select the **Advanced** tab to enable the firewall for this network connection (see 'Editing Existing Connections' on page 671).

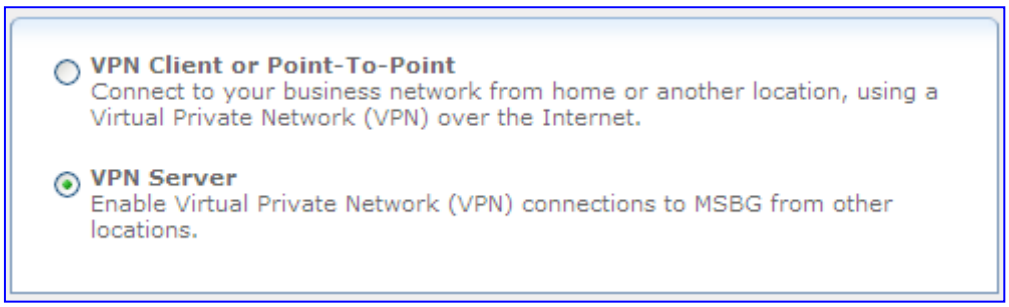
31.9 Layer 2 Tunneling Protocol Server (L2TP Server)

The device can act as a Layer 2 Tunneling Protocol Server (L2TP Server), accepting L2TP client connection requests.

➢ **To create an L2TP server:**

1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Connect to a Virtual Private Network over the Internet** option, and then click **Next**.

Figure 31-49: VPN Connection Type

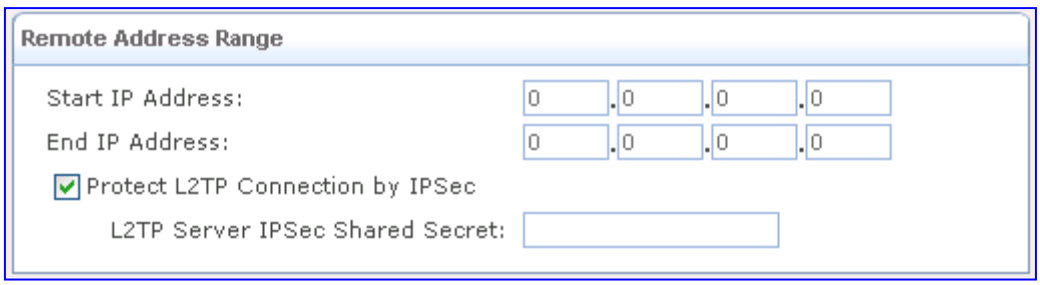


3. Select the **VPN Server** option, and then click **Next**.

Figure 31-50: Selecting L2TP Server VPN Protocol

4. Select the **Layer 2 Tunneling Protocol Server (L2TP Server)** option, and then click **Next**.

Figure 31-51: Defining L2TP Properties

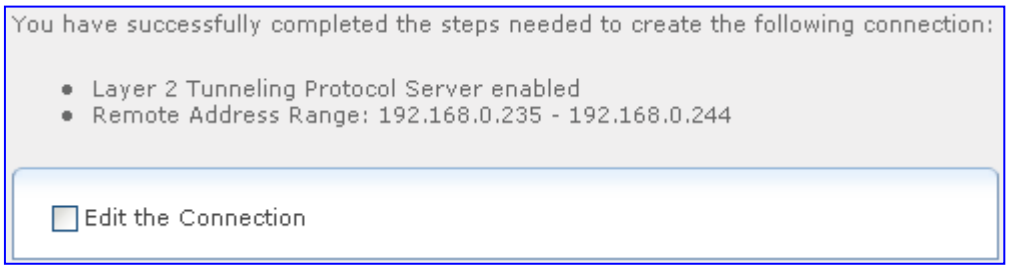


5. In the 'Start IP Address' and 'End IP Address' fields, specify the address range that the device reserves for remote users.
6. By default, the L2TP connection is protected by the IP Security (IPSec) protocol (the option is selected). However, if you wish to keep this setting, you must provide a string

that will serve as the 'L2TP Server IPSec Shared Secret'. Alternatively, clear this option to disable L2TP protection by IPSec.

7. Click **Next**; the following is displayed if successfully configured:

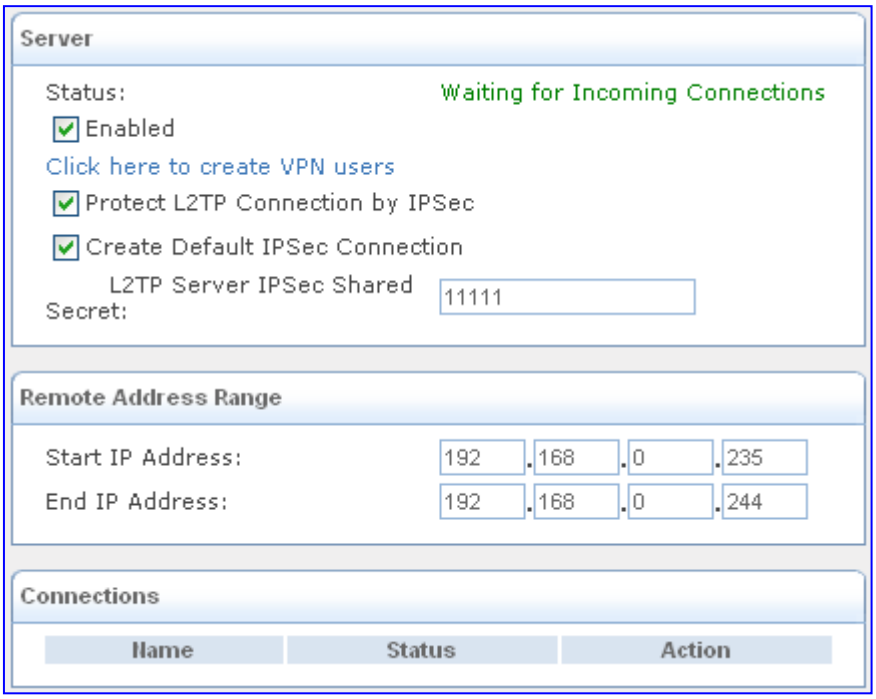
Figure 31-52: L2TP Server Added Successfully



Note that the attention message alerting that there are no users with VPN permissions.

8. Select the 'Edit the Newly Created Connection' check box, and then click **Finish**.

Figure 31-53: Defining Advanced L2TP Properties



9. Click the **Click here to create VPN users** link to define remote users that will be granted access to your home network.
10. Click **OK** to save settings; the new L2TP server connection is added to the Network Connection list.

31.10 Internet Protocol Security (IPSec)

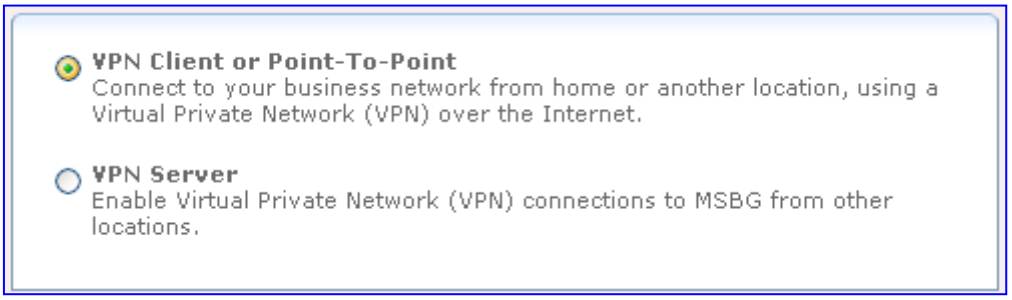
Internet Protocol Security (IPSec) is a series of guidelines for the protection of Internet Protocol (IP) communications. It specifies procedures for securing private information transmitted over public networks.

➢ **To create an IPSec connection:**

1. In the Connections page, click the **New** icon; the Connection Wizard opens.

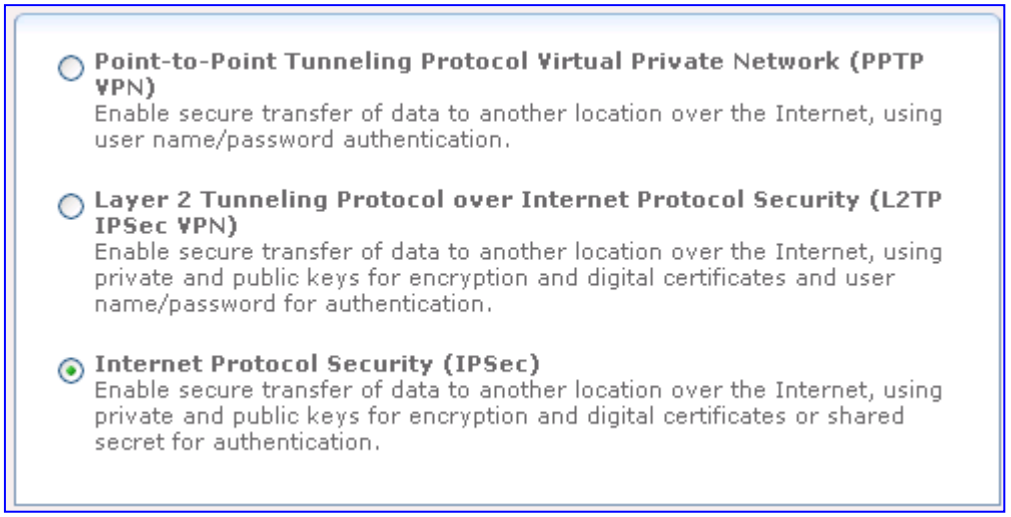
2. Select the **Connect to a Virtual Private Network over the Internet** option, and then click **Next**.

Figure 31-54: Selecting VPN Type for IPSec



3. Select the **VPN Client or Point-To-Point** option, and then click **Next**.

Figure 31-55: Selecting IPSec



4. Select the **Internet Protocol Security Server (IPSec)** option, and then click **Next**.

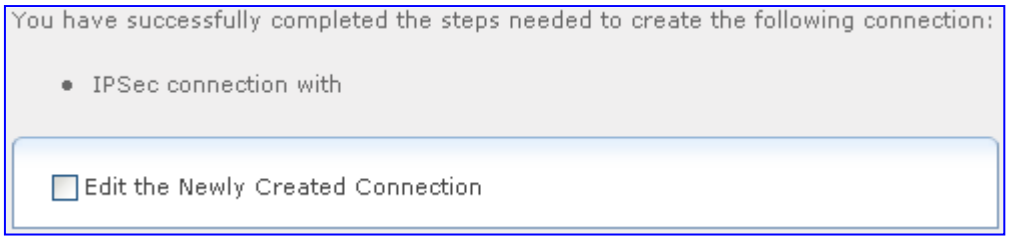
Figure 31-56: Defining IPSec Properties

5. In the 'Host Name or IP Address of Destination Gateway' field, enter the host or IP address of the destination gateway.
6. From the 'Remote IP' drop-down list, select the method for specifying the remote IP address, which serves as the tunnel's endpoint. Use "Same as Gateway" when connecting your LAN to a remote gateway. When connecting your LAN to a remote network (a group of computers beyond a gateway), use one of the remaining three options. Also, use the transport encapsulation type in a gateway-to-gateway scenario only. Upon selection of an option, the screen refreshes providing you with the appropriate fields for entering the data.
   - Same as Gateway – the default option that uses the gateway IP entered above. When selecting this option, you must also select the encapsulation type, tunnel or transport, from its drop-down list.

- IP Address – a 'Remote IP Address' field appears. Specify the IP address.
- IP Subnet – 'Remote Subnet IP Address' and 'Remote Subnet Mask' fields appear. Specify these parameters.
- IP Range – 'From IP Address' and 'To IP Address' fields appear. Specify the IP range.

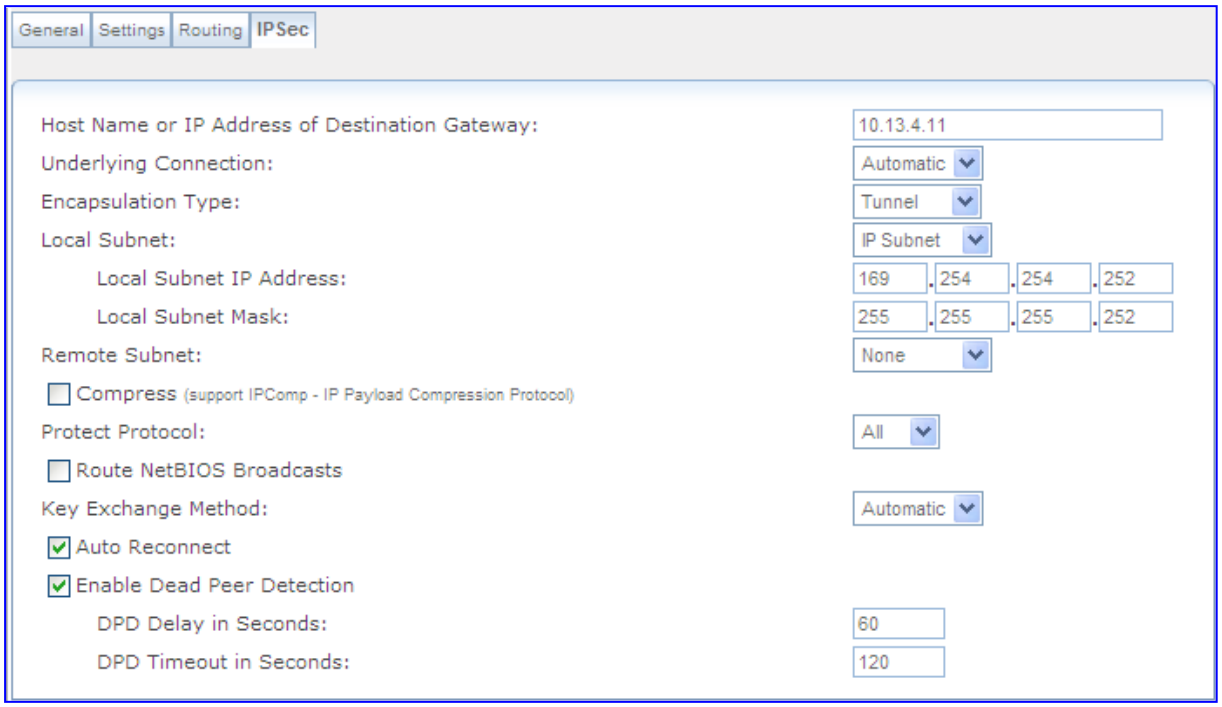
7. In the 'Shared Secret' field, enter the IPSec shared secret, which is the encryption key jointly decided upon with the network you are trying to access.
8. Click **Next**; the following is displayed if successfully configured:

Figure 31-57: IPSec Added Successfully



9. Select the 'Edit the Newly Created Connection' check box if you want to edit the new connection after clicking **Finish**.
10. Click **Finish** to save the settings; the new IPSec connection is added to the network connections list.
11. To define (edit) additional properties, click the **Edit** icon corresponding to the **VPN IPSec** connection in the connection list; the **General**, **Settings**, **Routing**, and **IPSec** tabs appear. For descriptions of the parameters in the **General**, **Settings**, and **Routing** tabs, see 'Editing Existing Connections' on page 671. Click the **IPSec** tab; the following apears (only the first part of the page is displayed due to page size):

Figure 31-58: IPSec Tab

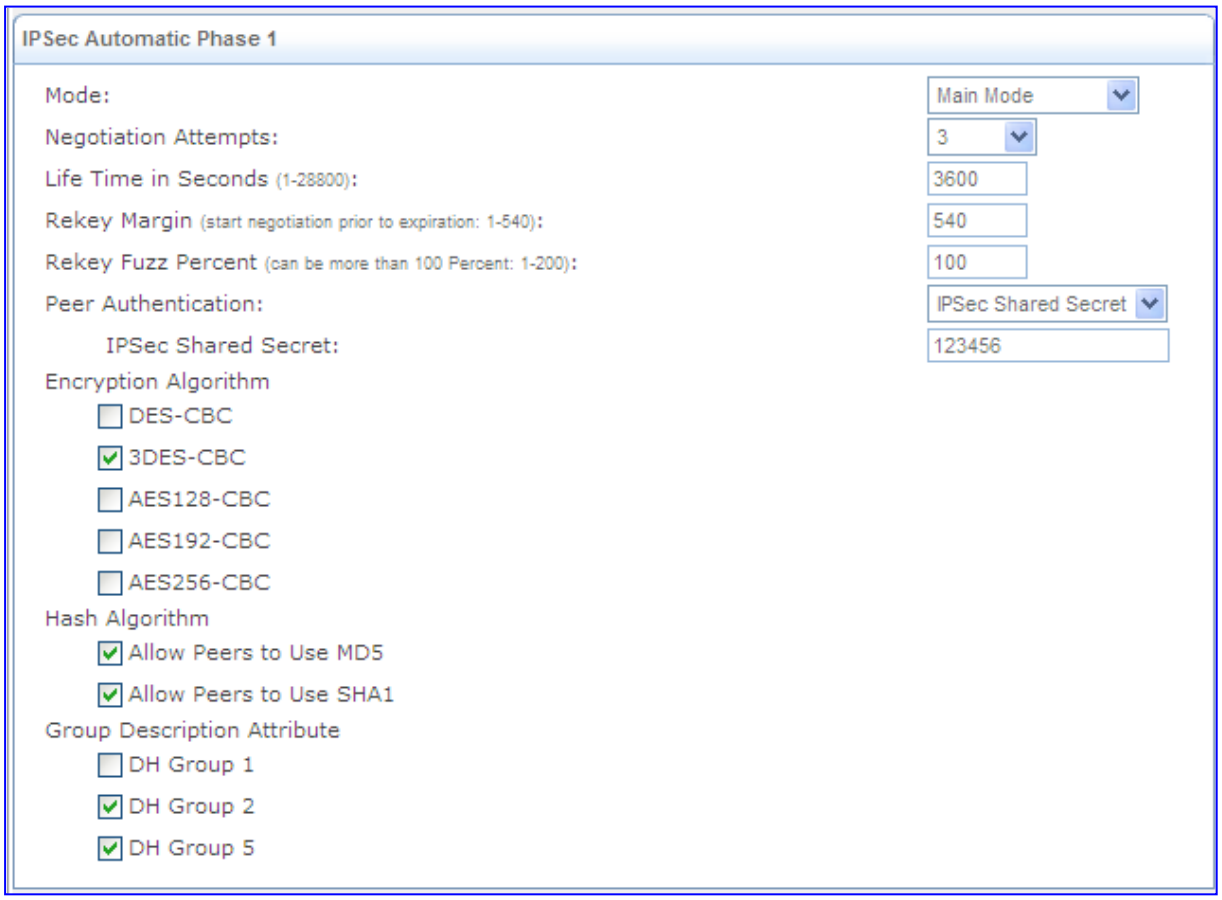


- **Host Name or IP Address of Destination Gateway:** The IP address of your IPSec peer. If your connection is an IPSec Server, this field displays "Any Remote Gateway".

- **Underlying Connection:** In a single WAN scenario, the underlying connection parameter is set to "Automatic" (non-configurable). However, if you have multiple WAN devices, a drop-down list appears, enabling you to choose the underlying WAN device. The IPSec connection only uses your chosen device, unless failover is enabled. In this case, the failed-to device is used instead (assuming its route rules consent), until the chosen device is up again. Note that if you select "Automatic", no attempt is made to return to the original device from the failed-to-device.
- **Encapsulation Type:** Select between 'Tunneling' or 'Transport' encapsulation. 'Transport' encapsulation is performed between two gateways (no subnets), and therefore needs no explicit configuration. 'Tunneling' requires that you configure the following parameters:
  - **Local Subnet:** Define your local endpoint, by selecting one of the following options:
    - ✓ **IP Subnet (default):** Enter the device's Local Subnet IP Address and Local Subnet Mask.
    - ✓ **IP Range:** Enter the 'From' and 'To' IP addresses, forming the endpoints range of the local subnet(s).
    - ✓ **IP Address:** Enter the Local IP Address to define the endpoint as a single host.
    - ✓ **None:** Select this option if you do not want to define a local endpoint. The endpoint is set to the gateway.
  - **Remote Subnet:** This section is identical to the 'Local Subnet' section above, but is for defining the remote endpoint.
    - ✓ **Compress (Support IPComp protocol):** Select this check box to compress packets during encapsulation with the IP Payload Compression protocol. Note that this reduces performance (and is therefore unchecked by default).
- **Protect Protocol:** Select the protocols to protect with IPSec: All, TCP, UDP, ICMP or GRE. When selecting TCP or UDP, additional source port and destination port drop-down lists appear, enabling you to select **All** or to specify 'Single' ports in order to define the protection of specific packets. For example, to protect L2TP packets, select UDP and specify 1701 as both single source and single destination ports.
- **Route NetBIOS Broadcasts:** Select this option to allow NetBIOS packets through the IPSec tunnel, which otherwise would not meet the routing conditions specified.
- **Key Exchange Method:** The IPSec key exchange method can be 'Automatic' (default) or 'Manual'. Selecting one of these options alters the rest of the page.
  - **Automatic key exchange settings:**
    - ✓ **Auto Reconnect:** The IPSec connection reconnects automatically if disconnected for any reason.
    - ✓ **Enable Dead Peer Detection:** The device detects whether the tunnel endpoint has ceased to operate, in which case it terminates the connection. Note that this feature is functional only if the other tunnel endpoint supports it. This is determined during the negotiation phase of the two endpoints.
    - ✓ **DPD Delay in Seconds:** The timeframe in which no traffic has passed through the tunnel. After this timeframe, the device sends a packet to test the tunnel endpoint, expecting a reply.
    - ✓ **DPD Timeout in Seconds:** The timeframe the device waits for the test reply, after which it terminates the connection.

IPSec Automatic Phase 1 – Peer Authentication:

Figure 31-59: IPSec Tab - IPSec Automatic Phase 1

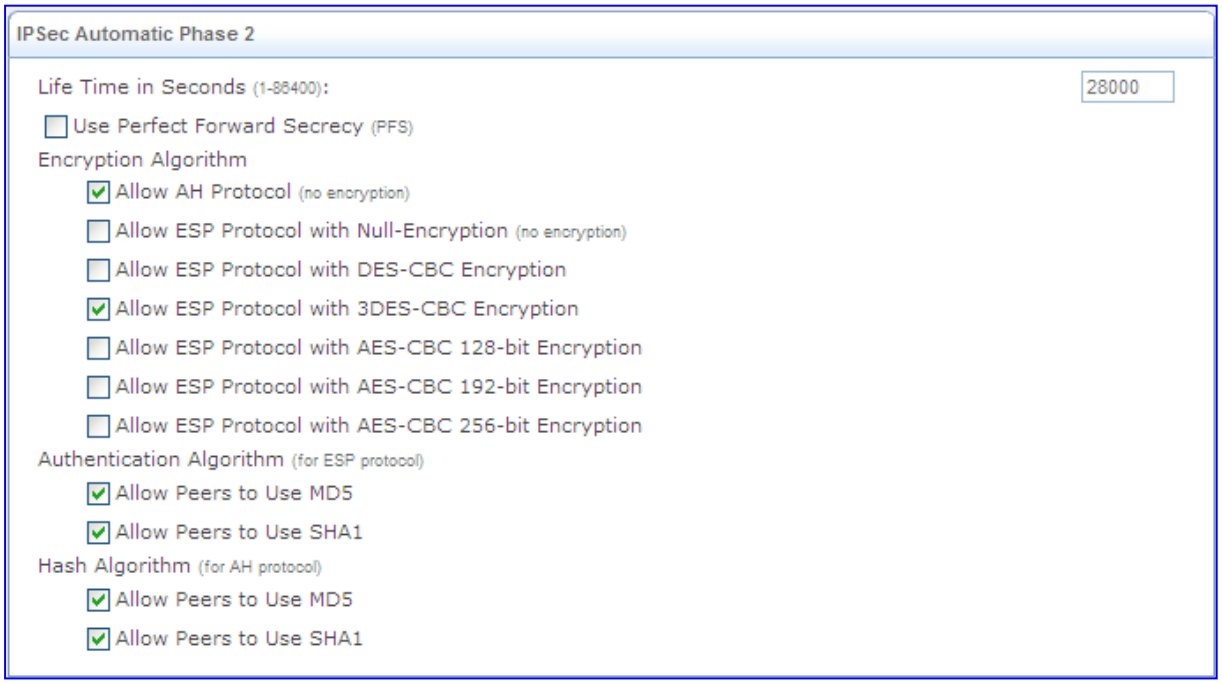


- ✓ **Mode:** Select the IPSec mode – either 'Main Mode' or 'Aggressive Mode'. Main mode is a secured but slower mode, which presents negotiable propositions according to the authentication algorithms that you select in the check boxes. Aggressive Mode is faster but less secured. When selecting this mode, the algorithm check boxes are replaced by options, presenting strict propositions according to your selections.
- ✓ **Negotiation Attempts:** Select the number of negotiation attempts to be performed in the automatic key exchange method. If all attempts fail, the device waits for a negotiation request.
- ✓ **Life Time in Seconds:** The timeframe in which the peer authentication is valid.
- ✓ **Rekey Margin:** Specifies how long before connection expiry should attempts to negotiate a replacement begin. It is similar to that of the key life time and is given as an integer denoting seconds.
- ✓ **Rekey Fuzz Percent:** Specifies the maximum percentage by which Rekey Margin should be randomly increased to randomize re-keying intervals.
- ✓ **Peer Authentication:** Select the method by which the device authenticates your IPSec peer.

- ✓ **IPSec Shared Secret:** Enter the IPSec shared secret.
  - **RSA Signature** – Enter the peer's RSA signature (based on the device's public key).
  - **Certificate** – If a certificate exists on the device, it appears when you select this option. Enter the certificate's local ID and peer ID.
- ✓ **Encryption Algorithm:** Select the encryption algorithms that the device attempts to use when negotiating with the IPSec peer.
- ✓ **Hash Algorithm:** Select the hash algorithms that the device attempts to use when negotiating with the IPSec peer.
- ✓ **Group Description Attribute:** Select the Diffie-Hellman (DH) group description(s). Diffie-Hellman is a public-key cryptography scheme that allows two parties to establish a shared secret over an insecure communications channel.

IPSec Automatic Phase 2 – Key Definition:

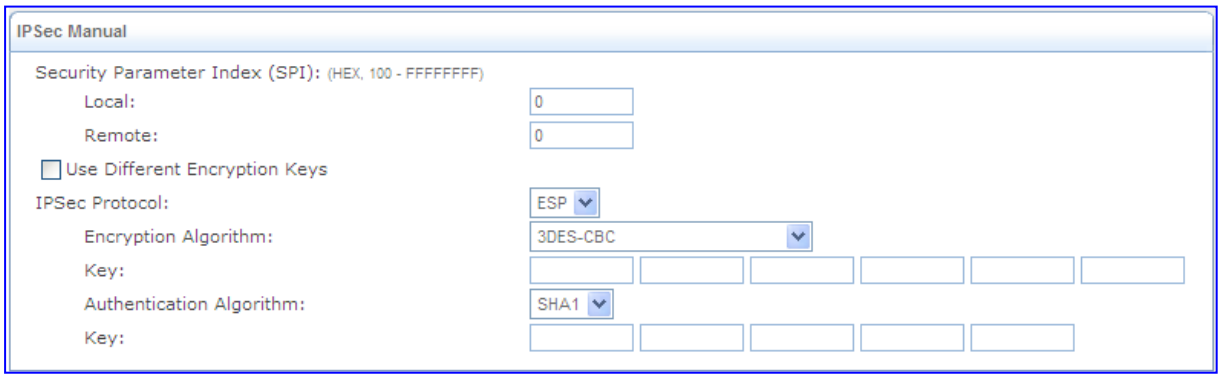
Figure 31-60: IPSec Tab - IPSec Automatic Phase 2



- ✓ **Life Time in Seconds:** The length of time before a security association automatically performs renegotiation.
- ✓ **Use Perfect Forward Secrecy (PFS):** Select whether Perfect Forward Secrecy of keys is required on the connection's keying channel (with PFS, penetration of the key-exchange protocol does not compromise keys negotiated earlier). Deselecting this option hides the next parameter.
  - **Group Description Attribute:** Select whether to use the same group chosen in phase 1, or reselect specific groups.
- ✓ **Encryption Algorithm:** Select the encryption algorithms that the device attempts to use when negotiating with the IPSec peer.
- ✓ **Authentication Algorithm (for ESP protocol):** Select the authentication algorithms that the device attempts to use when negotiating with the IPSec peer.
- ✓ **Hash Algorithm (for AH protocol):** Select the hash algorithms that the device attempts to use when negotiating with the IPSec peer.

- **Manual key definition:**

Figure 31-61: IPSec Tab - IPSec Manual



- ✓ **Security Parameter Index (SPI):** A 32 bit value that together with an IP address and a security protocol, uniquely identifies a particular security association. The local and remote values must be coordinated with their respective values on the IPSec peer.
- ✓ **Use Different Encryption Keys:** Selecting this option allows you to define both local and remote algorithm keys when defining the IPSec protocol.
- ✓ **IPSec Protocol:** Select between the ESP and AH IPSec protocols. The page refreshes accordingly:
  - **ESP** – Select the encryption and authentication algorithms, and enter the algorithm keys in hexadecimal representation.
  - **AH** – Select the hash algorithm, and enter the algorithm key in hexadecimal representation.

12. Click **OK** to save the settings.

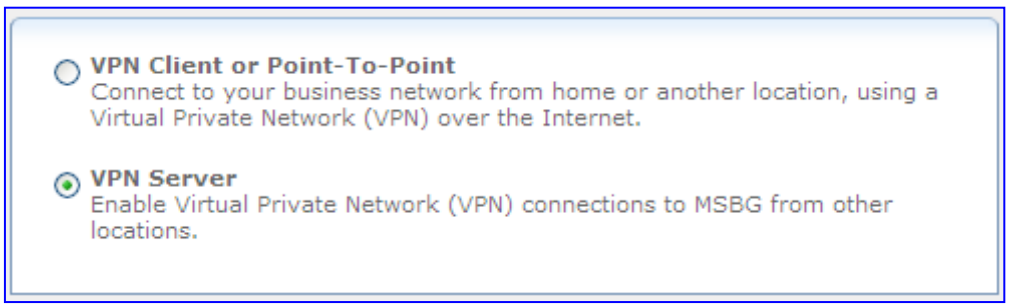
31.11 Internet Protocol Security Server (IPSec Server)

The below procedure describes how to create an IPSec server.

➢ **To create an IPSec server connection:**

1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Connect to a Virtual Private Network over the Internet** option, and then click **Next**.

Figure 31-62: VPN Connection Type

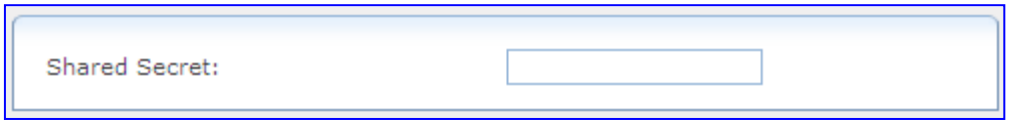


3. Select the **VPN Server** option, and then click **Next**.

Figure 31-63: VPN Protocols

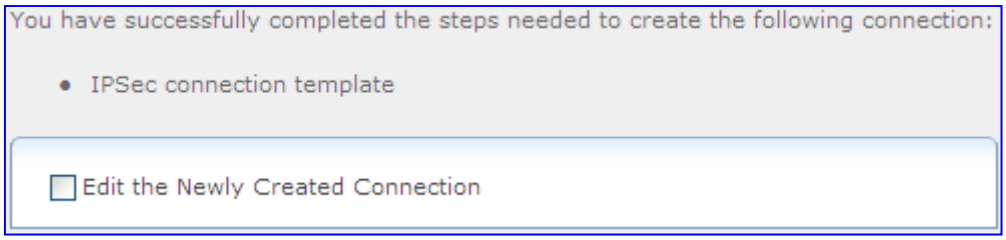
4. Select the **Internet Protocol Security Server (IPSec Server)** option, and then click **Next**.

Figure 31-64: IPSec Shared Secret Key



5. Enter the IPSec shared secret, which is the encryption key jointly decided upon with the network you are trying to access, and then click **Next**; the following is displayed if successfully configured:

Figure 31-65: IPSec Connection Added Successfully



6. Select the 'Edit the Newly Created Connection' check box if you want to edit the new connection after clicking **Finish**.
7. Click **Finish** to save the settings; the new IPSec Server is added to the network connections list.

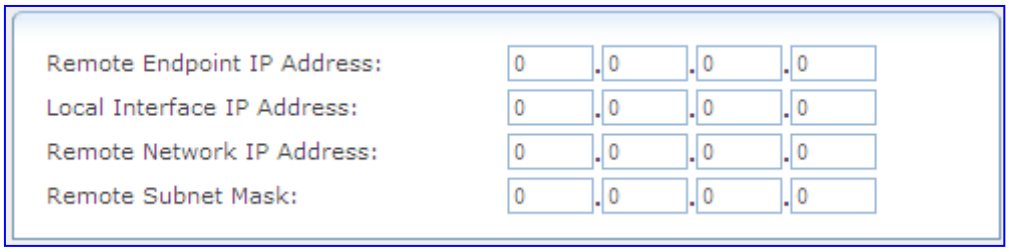
31.12 General Routing Encapsulation (GRE)

The device allows you to create a GRE tunnel to transport multicast traffic and IPv6, in addition to other existing tunneling capabilities (e.g. L2TP and PPTP).

➢ **To create a new GRE tunnel:**

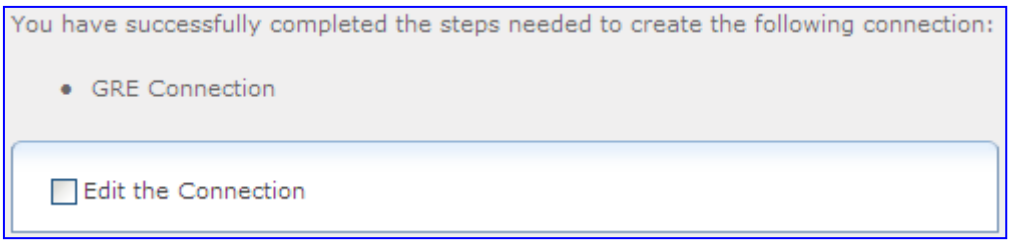
1. In the Connections page, click the **New** icon; the Connection Wizard opens.
2. Select the **Advanced Connection** option, and then click **Next**; the Advanced Connection page appears.
3. Select the **General Routing Encapsulation (GRE)** option, and then click **Next**; the following page appears.

Figure 31-66: Configuring General IPIP Parameters



4. In the 'Remote Endpoint IP Address' field, enter the tunnel's remote endpoint IP address.
5. In the 'Local Interface IP Address' field, enter the local IP address for the interface.
6. In the 'Remote Network IP Address' and 'Remote Subnet Address fields, enter the IP address and subnet mask (respectively) of the remote network that is to be accessed via the tunnel, and then click **Next**; the following GRE connection message is displayed if successfully configured:

Figure 31-67: GRE Connection Successfully Added



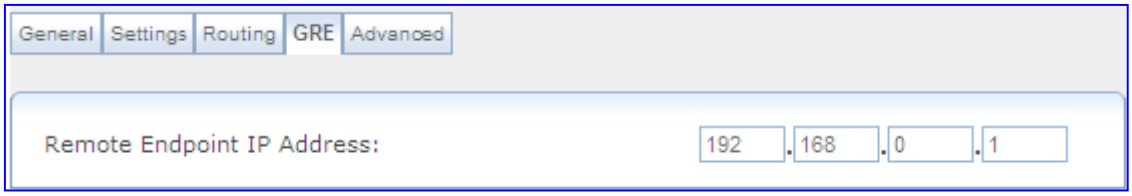
7. Select the 'Edit the Connection' check box if you want to edit the new connection after clicking **Finish**.
8. Click **Finish** to save the settings; the new GRE tunnel is added to the network connections list.

➢ **To edit the GRE tunnel connection:**

1. In the Connections page, click the "WAN GRE" link; the **General** tab appears displaying general properties.
2. Select the **Settings** tab to edit various settings (see 'Editing Existing Connections' on page 671).
3. Select the **Routing** tab to edit the routing parameters (see 'Editing Existing Connections' on page 671).

4. Select the **GRE** tab to define the tunnel's remote endpoint IP address.

Figure 31-68: Editing GRE Remote Endpoint IP Address



5. Select the **Advanced** tab to enable the firewall for this network connection (see 'Editing Existing Connections' on page 671).

31.12.1 GRE Example Scenario

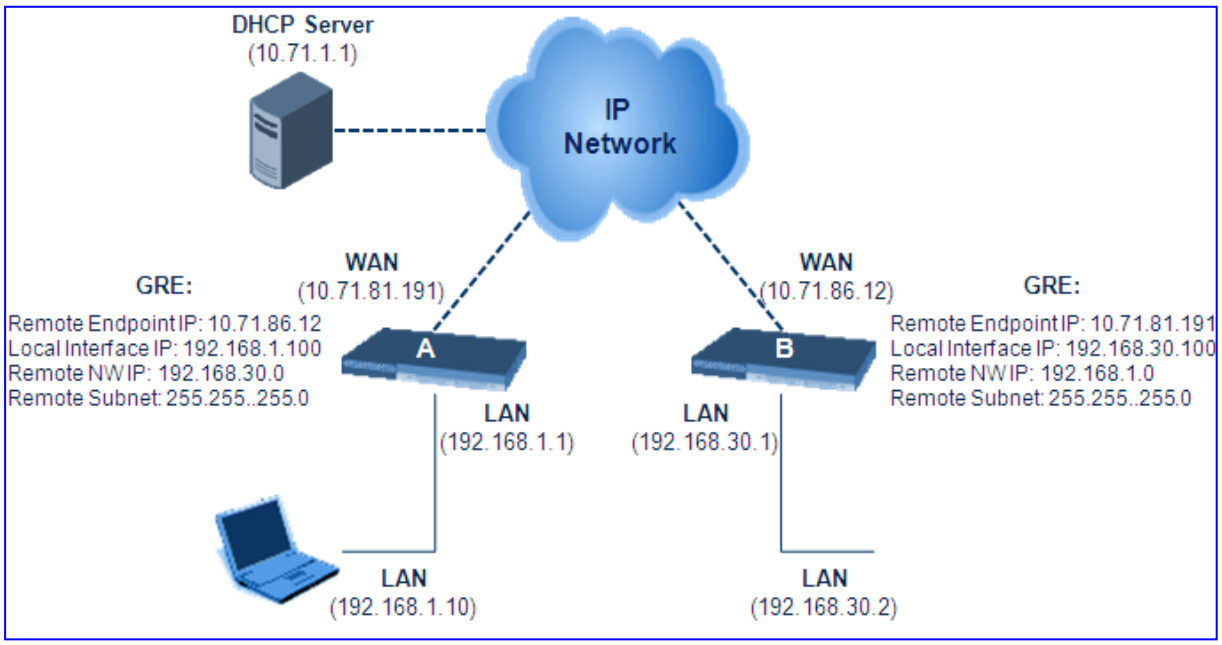
The following example demonstrates the usage of a GRE interface to communicate between two hosts located on different LANs, behind separate MSBG devices.

Note: A GRE tunnel is an unsecured (unencrypted) tunnel. Safety measures must be taken when setting up such a tunnel.

This example includes the following setup:

- Two devices:
  - "A" - WAN IP address is 10.71.81.191; LAN IP address is 192.168.1.1
  - "B" - WAN IP address is 10.71.86.12; LAN IP address is 192.168.30.1
- Two LAN hosts
- A WAN host serving as a DHCP server
- Each LAN host is connected to a LAN port on a different device
- The devices' WAN ports are connected to the WAN (where the DHCP server is available)

Figure 31-69: Example Scenario Setup

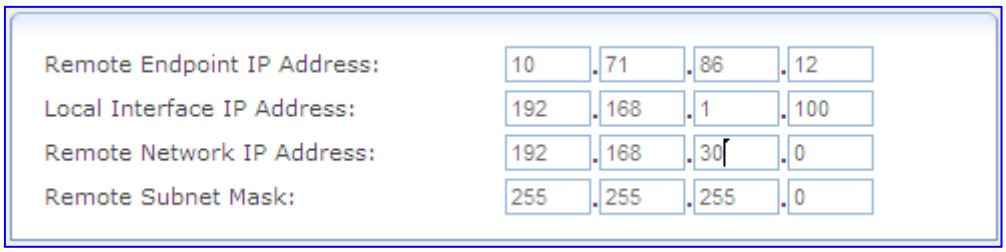


To create a tunnel, each MSBG device must be made aware of the other's WAN IP address (the information must be exchanged).

➢ **To configure a GRE tunnel:**

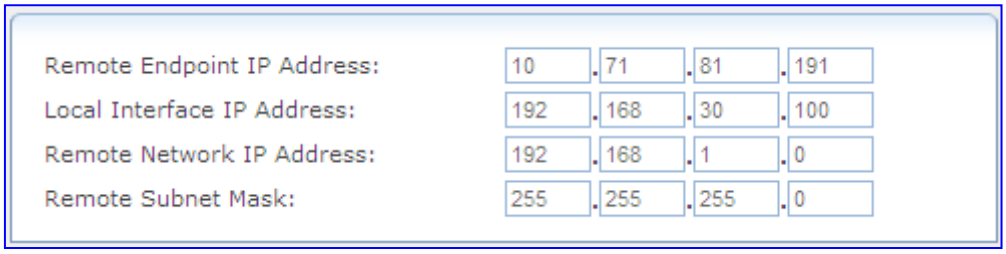
1. Create a GRE tunnel for device "A":
   a. In the Connections page, click the **New** icon.
   b. Select the **Advanced Connection** option, and then click **Next**.
   c. Select the **General Routing Encapsulation (GRE)** option, and then click **Next**.

Figure 31-70: Defining GRE Tunnel for Device A



   d. Enter 10.71.86.12 as the tunnel's remote endpoint IP address.
   e. Enter 192.168.1.100 as the local interface IP address.
   f. Enter 192.168.30.0 as the IP address of the remote network that will be accessed via the tunnel, and 255.255.255.0 as the subnet mask.
   g. Click **Next**, and then click **Finish**.
2. Create a GRE tunnel for device "B":
   a. In the Connections page, click the **New** icon.
   b. Select the **Advanced Connection** option, and then click **Next**.
   c. Select the **General Routing Encapsulation (GRE)** option, and then click **Next**.

Figure 31-71: Defining GRE Tunnel for Device B



   d. Enter 10.71.81.191 as the tunnel's remote endpoint IP address.
   e. Enter 192.168.30.100 as the local interface IP address.
   f. Enter 192.168.1.0 as the IP address of the remote network that will be accessed via the tunnel, and 255.255.255.0 as the subnet mask.
   g. Click **Next**, and then click **Finish**.
3. Running the Scenario: After verifying that each host has properly received an IP address in the subnet of its respective device, send a ping from host "A" (192.168.1.10) to host "B" (192.168.30.2). If the GRE connection is successful, host "B" should reply.

31.13 Editing Existing Connections

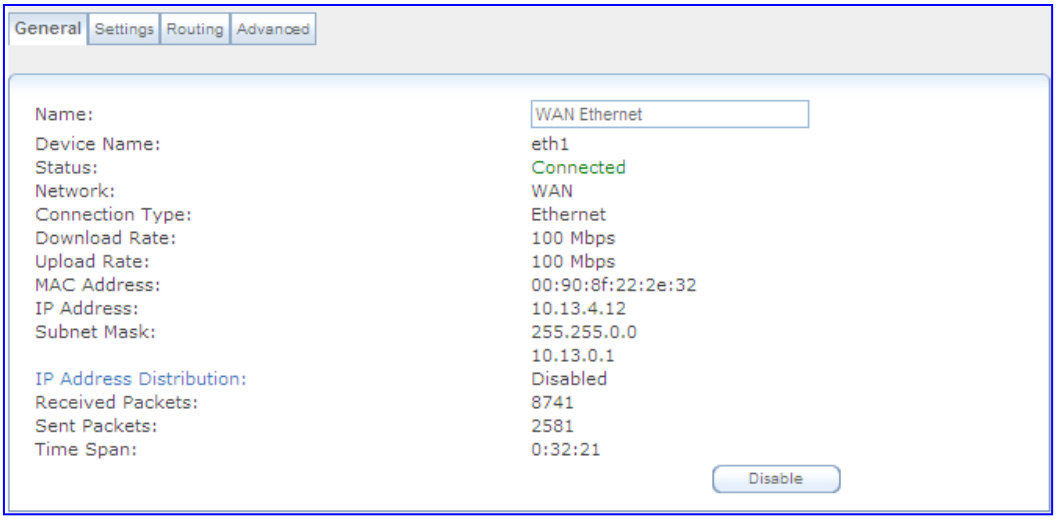
As many of the configuration parameters for the different connection types are similar, the basic procedure for editing the connections is described in summary below. Editing connections allows you to configure additional parameters that do not appear when initially creating new connections.

Note: Tabs specific to the connection type are described in later sections relevant to that connection type.

➢ **To edit connections:**

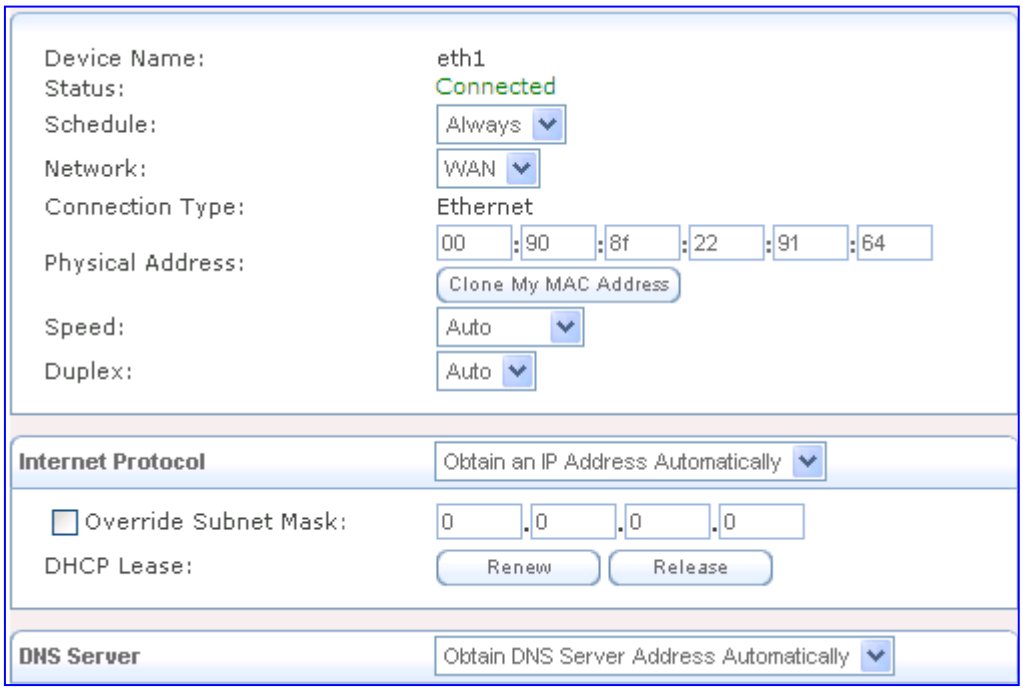
1. Access the configuration tabs:
   - In the Connections page, click the **Edit** icon corresponding to the network connection that you want to edit; the **General** tab is displayed, showing general properties of the connection type (e.g., WAN Ethernet connection).
   - From the WAN Access page (see 'WAN Access Settings' on page 567), click the **Click here for Advanced Settings** link.

Figure 31-72: Editing Network Connection - General Tab



2. Select the **Settings** tab:

Figure 31-73: Editing Network Connection - Settings Tab

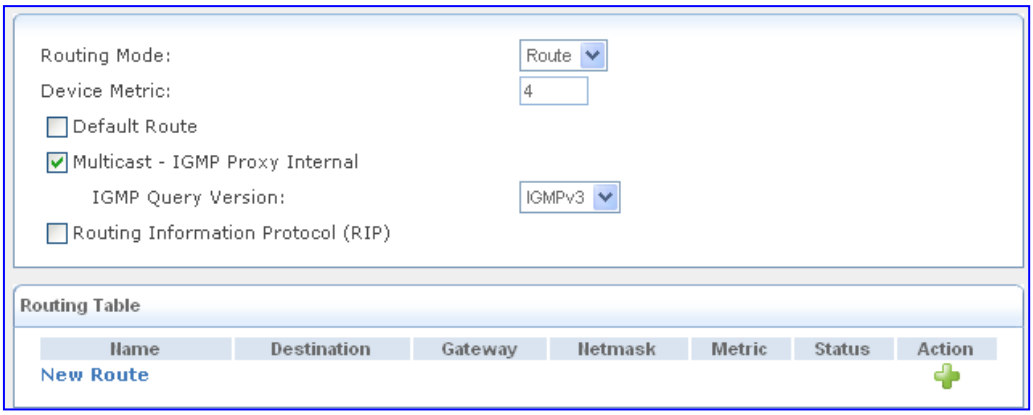


- **Network:** select whether the parameters you are configuring relate to a WAN, LAN or DMZ connection.
- **Physical Address:** physical address of the network card used for your network. Some cards allow you to change this address.
- **Speed:** select the transmission speed of the network interface (10Base-T, 100Base-T, 1000Base-T, or autonegotiation).
  Note: This is applicable only to the WAN Ethernet port.
- **Duplex:** select the duplex mode (half-duplex, full-duplex, or autonegotiation).
  Note: This is applicable only to the WAN Ethernet port.
- **MTU:** Maximum Transmission Unit - specifies the largest packet size permitted for Internet transmission. By default ('Automatic'), the device selects the best MTU for your Internet connection. Select **Automatic by DHCP** to have the DHCP determine the MTU. If you select **Manual**, it is recommended to enter a value in the 1200 to 1500 range.
- **Internet Protocol:** select one of the following Internet protocol options from the 'Internet
  - No IP Address - select this option if you require that your device have no IP address. This can be useful if you are working in an environment where you are not connected to other networks, such as the Internet.
  - Obtain an IP Address Automatically - your connection is configured by default to act as a DHCP client. You should keep this configuration in case your service provider supports DHCP, or if you are connecting using a dynamic IP address. The server that assigns the device with an IP address, also assigns a subnet mask. You can override the dynamically assigned subnet mask by selecting the 'Override Subnet Mask' check box and specifying your own mask instead. You can click the **Release** button to release the current leased IP address. Once the address has been released, the button text changes to **Renew**. Use this button to renew the leased IP address.
  - Use the Following IP Address - your connection can be configured using a permanent (static) IP address. Your service provider should provide you with such an IP address and subnet mask.

- **DNS Server:** Domain Name System (DNS) is the method by which Web site domain names are translated into IP addresses:
  - Obtain DNS Server Address Automatically - configure the connection to automatically obtain a DNS server address, or specify such an address manually, according to the information provided by your ISP.
  - Use the Following DNS Server Addresses - manually configure DNS server addresses. You can specify up to two different DNS server addresses, one primary, another secondary.

3. Click **OK** to save the settings.
4. Select the **Routing** tab:

Figure 31-74: Editing Network Connection - Routing Tab



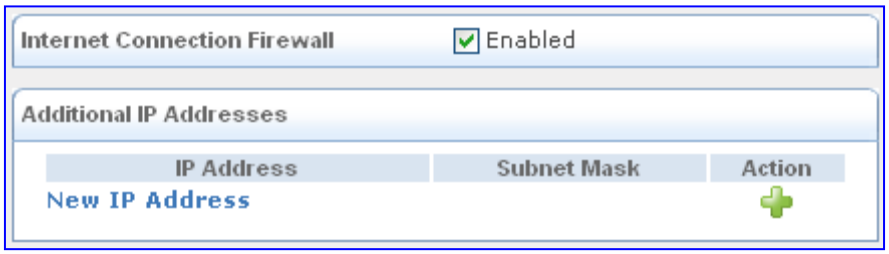
You can choose to setup your device to use static or dynamic routing. Dynamic routing automatically adjusts how packets travel on the network, whereas static routing specifies a fixed routing path to destinations.

- **Routing Mode:** select one of the following routing modes:
  - Route: the device functions as a router between two networks.
  - NAPT: Network Address and Port Translation (NAPT) refers to network address translation involving the mapping of port numbers, allowing multiple hosts to share a single IP address. Use NAPT if your LAN encompasses multiple devices, a topology that necessitates port translation in addition to address translation.
- **Device Metric:** a value used by the device to determine whether one route is superior to another, considering parameters such as bandwidth and delay.
- **Default Route:** defines this device as a the default route.
- **Multicast – IGMP Proxy Internal:** the device serves as an IGMP proxy, issuing IGMP host messages on behalf of its LAN hosts. This check box is enabled on LAN connections by default, meaning that if a LAN multicast server is available, other LAN hosts asking to join multicast groups (by sending IGMP requests) are able to join its multicast group. However, this check box is disabled on the WAN connection by default, meaning that LAN hosts are unable to join multicast groups of WAN multicast servers.
  - IGMP Query Version - the device supports all three versions of IGMP. Select the version you would like to use. Note that this drop-down list appears for LAN connections only.
- **Routing Information Protocol (RIP):** enables the Routing Information Protocol (RIP). RIP determines a route based on the smallest hop count between source and destination. When RIP is enabled, select the following:
  - Listen to RIP Messages - select **None**, **RIPv1**, **RIPv2** or **RIPv1/2**.
  - Send RIP Messages - select **None**, **RIPv1**, **RIPv2 - Broadcast** or **RIPv2 - Multicast**.

- **Routing Table:** allows you to add or modify routes when this host is active. Use the **New Route** link to add a route or edit existing routes. To learn more about this feature, see 'Routing Settings' on page 623.

5. Click **OK** to save the settings.
6. Select the **Advanced** tab:

Figure 31-75: Editing Network Connection - Advanced Tab



- **Internet Connection Firewall:** Your device's firewall helps protect your computer by preventing unauthorized users from gaining access to it through a network such as the Internet. The firewall can be activated per network connection. To enable the firewall on this network connection, select the 'Enabled' check box.
- **Additional IP Addresses:** You can add alias names (additional IP addresses) to the device by clicking the **New IP Address** link. This enables you to access the device using these aliases (in addition to the default 192.168.1.1).

7. Click **OK** to save the settings.

Part VI

Maintenance

This part describes the maintenance procedures.

Reader's Notes

32 Basic Maintenance

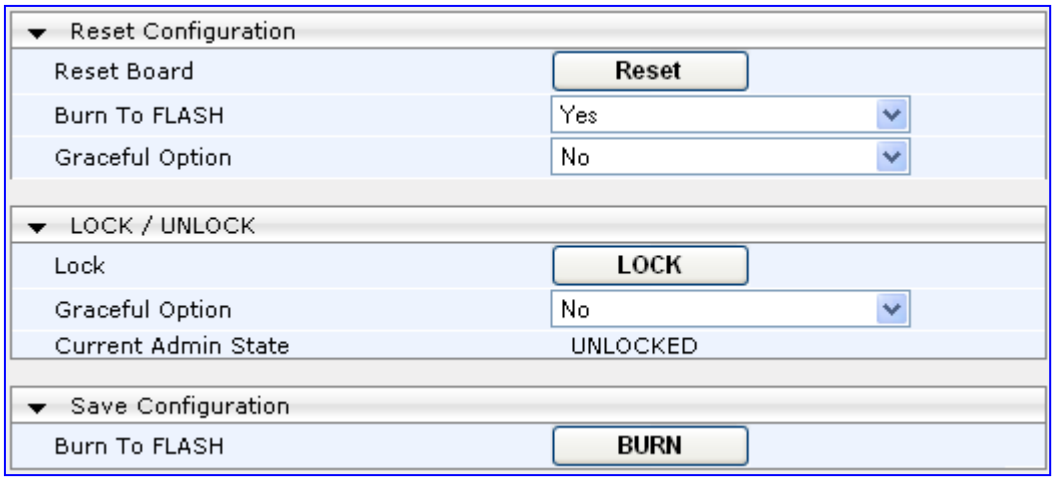
The Maintenance Actions page allows you to perform the following:

- Reset the device - see 'Resetting the Device' on page 677
- Lock and unlock the device - see 'Locking and Unlocking the Device' on page 679
- Save configuration to the device's flash memory - see 'Saving Configuration' on page 680

➢ **To access the Maintenance Actions page, do one of the following:**

- On the toolbar, click the **Device Actions** button, and then from the drop-down menu, choose **Reset**.
- On the Navigation bar, click the **Maintenance** tab, and then in the Navigation tree, select the **Maintenance** menu and choose **Maintenance Actions**.

Figure 32-1: Maintenance Actions Page



32.1 Resetting the Device

The Maintenance Actions page allows you to remotely reset the device. In addition, before resetting the device, you can choose the following options:

- Save the device's current configuration to the device's flash memory (non-volatile).
- Perform a graceful shutdown, i.e., device reset starts only after a user-defined time (i.e., timeout) or after no more active traffic exists (the earliest thereof).

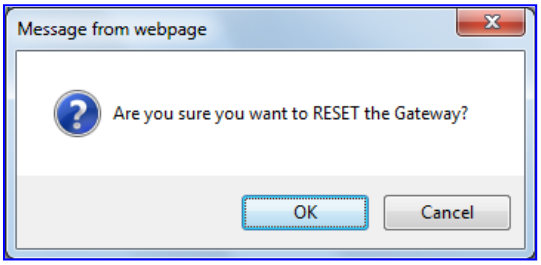
Notes:

- Throughout the Web interface, parameters preceded by the lightning symbol are not applied on-the-fly and require that you reset the device for them to take effect.
- When you modify parameters that require a device reset, once you click the **Submit** button in the relevant page, the toolbar displays "Reset" (see 'Toolbar' on page 54) to indicate that a device reset is required.
- After you reset the device, the Web GUI is displayed in Basic view (see 'Displaying Navigation Tree in Basic and Full View' on page 56).
- Upon reboot, the device restores the settings from its configuration file. However, if reboot attempts fail three times consecutively, the device resets the configuration file by restoring factory defaults before attempting to reboot.

➢ **To reset the device:**

1. Open the Maintenance Actions page (see 'Basic Maintenance' on page 677).
2. Under the 'Reset Configuration' group, from the 'Burn To FLASH' drop-down list, select one of the following options:
   - **Yes:** The device's current configuration is saved (*burned*) to the flash memory prior to reset (default).
   - **No:** Resets the device without saving the current configuration to flash (discards all unsaved modifications).
3. Under the 'Reset Configuration' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - **Yes:** Reset starts only after the user-defined time in the 'Shutdown Timeout' field (see Step 4) expires or after no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - **No:** Reset starts regardless of traffic, and any existing traffic is terminated at once.
4. In the 'Shutdown Timeout' field (relevant only if the 'Graceful Option' in the previous step is set to **Yes**), enter the time after which the device resets. Note that if no traffic exists and the time has not yet expired, the device resets.
5. Click the **Reset** button; a confirmation message box appears, requesting you to confirm.

Figure 32-2: Reset Confirmation Message Box



6. Click **OK** to confirm device reset; if the parameter 'Graceful Option' is set to **Yes** (in Step 3), the reset is delayed and a screen displaying the number of remaining calls and time is displayed. When the device begins to reset, a message appears notifying you of this.

32.2 Locking and Unlocking the Device

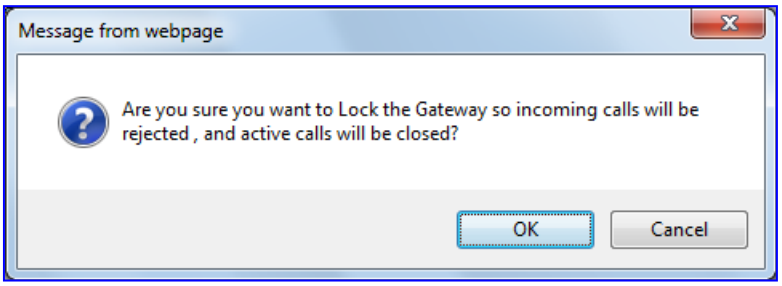
The Lock and Unlock options allow you to lock the device so that it doesn't accept any new calls. This is useful when, for example, you are uploading new software files to the device and you don't want any traffic to interfere with the process.

➢ **To lock the device:**

1. Open the Maintenance Actions page (see 'Basic Maintenance' on page 677).
2. Under the 'LOCK / UNLOCK' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - **Yes:** The device is 'locked' only after the user-defined time in the 'Lock Timeout' field (see Step 3) expires or no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - **No:** The device is 'locked' regardless of traffic. Any existing traffic is terminated immediately.

   Note: These options are only available if the current status of the device is in the Unlock state.
3. In the 'Lock Timeout' field (relevant only if the parameter 'Graceful Option' in the previous step is set to **Yes**), enter the time (in seconds) after which the device locks. Note that if no traffic exists and the time has not yet expired, the device locks.
4. Click the **LOCK** button; a confirmation message box appears requesting you to confirm device Lock.

Figure 32-3: Device Lock Confirmation Message Box



5. Click **OK** to confirm device Lock; if 'Graceful Option' is set to **Yes**, the lock is delayed and a screen displaying the number of remaining calls and time is displayed. Otherwise, the lock process begins immediately. The Current Admin State' field displays the current state - "LOCKED" or "UNLOCKED".

➢ **To unlock the device:**

1. Open the Maintenance Actions page (see 'Maintenance Actions' on page 677).
2. Under the 'LOCK / UNLOCK' group, click the **UNLOCK** button. Unlock starts immediately and the device accepts new incoming calls.

Note: The Home page's General Information pane displays whether the device is locked or unlocked (see 'Using the Home Page' on page 70).

32.3 Saving Configuration

The Maintenance Actions page allows you to save (*burn*) the current parameter configuration (including loaded auxiliary files) to the device's *non-volatile* memory (i.e., flash). The parameter modifications that you make throughout the Web interface's pages are temporarily saved (to the *volatile* memory - RAM) when you click the **Submit** button on these pages. Parameter settings that are saved only to the device's RAM revert to their previous settings after a hardware/software reset (or power failure). Therefore, to ensure that your configuration changes are retained, you must save them to the device's flash memory using the burn option described below.

➢ **To save the changes to the non-volatile flash memory :**

1. Open the Maintenance Actions page (see 'Basic Maintenance' on page 677).
2. Under the 'Save Configuration' group, click the **BURN** button; a confirmation message appears when the configuration successfully saves.

Notes:

- Saving configuration to the *non-volatile* memory may disrupt current traffic on the device. To avoid this, disable all new traffic before saving, by performing a graceful lock (see 'Locking and Unlocking the Device' on page 679).
- Throughout the Web interface, parameters preceded by the lightning symbol are not applied on-the-fly and require that you reset the device for them to take effect (see 'Resetting the Device' on page 677).
- The Home page's General Information pane displays whether the device is currently "burning" the configuration (see 'Using the Home Page' on page 70).

33 Software Upgrade

The **Software Update** menu allows you to upgrade the device's software, install Software Upgrade Key, and load/save configuration file. This menu includes the following page items:

- Load Auxiliary Files (see 'Loading Auxiliary Files' on page 681)
- Software Upgrade Key (see 'Loading Software Upgrade Key' on page 696)
- Software Upgrade Wizard (see 'Software Upgrade Wizard' on page 699)
- Configuration File (see 'Backing Up and Loading Configuration File' on page 702)

33.1 Loading Auxiliary Files

The Load Auxiliary Files page allows you to load various auxiliary files to the device. These auxiliary files are briefly described in the table below:

Table 33-1: Auxiliary Files Descriptions

File	Description
INI	Provisions the device's parameters. The Web interface enables practically full device provisioning, but customers may occasionally require new feature configuration parameters in which case this file is loaded. **Note:** Loading this file only provisions those parameters that are included in the *ini* file. For more information on the *ini* file, see 'INI File-Based Management' on page 97.
CAS	CAS auxiliary files containing the CAS Protocol definitions that are used for CAS-terminated trunks (for various types of CAS signaling). You can use the supplied files or construct your own files. Up to eight different CAS files can be loaded to the device. For more information on CAS files, see CAS Files on page 690.
Voice Prompts	Voice announcement file containing a set of Voice Prompts (VP) that are played by the device during operation. For more information on VP files, see Voice Prompts File on page 689.
Call Progress Tones	This is a region-specific, telephone exchange-dependent file that contains the Call Progress Tones (CPT) levels and frequencies for the device. The default CPT file is U.S.A. For more information on the CPT file, see 'Call Progress Tones File' on page 684.
Prerecorded Tones	The Prerecorded Tones (PRT) file enhances the device's capabilities of playing a wide range of telephone exchange tones that cannot be defined in the CPT file. For more information on the PRT file, see 'Prerecorded Tones File' on page 688.
Dial Plan	This file contains dialing plans, used by the device, for example, to know when to stop collecting the dialed digits and start sending them on. For more information on the Dial Plan file, see Dial Plan File on page 690.
VXML	Voice Extensible Markup Language (VXML) script file. For more information on VXML, see Voice XML Interpreter on page 540.

File	Description
User Info	The User Information file maps PBX extensions to IP numbers. This file can be used to represent PBX extensions as IP phones in the global 'IP world'. For more information on the User Info file, see 'User Information File' on page 691.
AMD Sensitivity	Answer Machine Detector (AMD) Sensitivity file containing the AMD Sensitivity suites. For more information on the AMD file, see AMD Sensitivity File on page 694.

Notes:

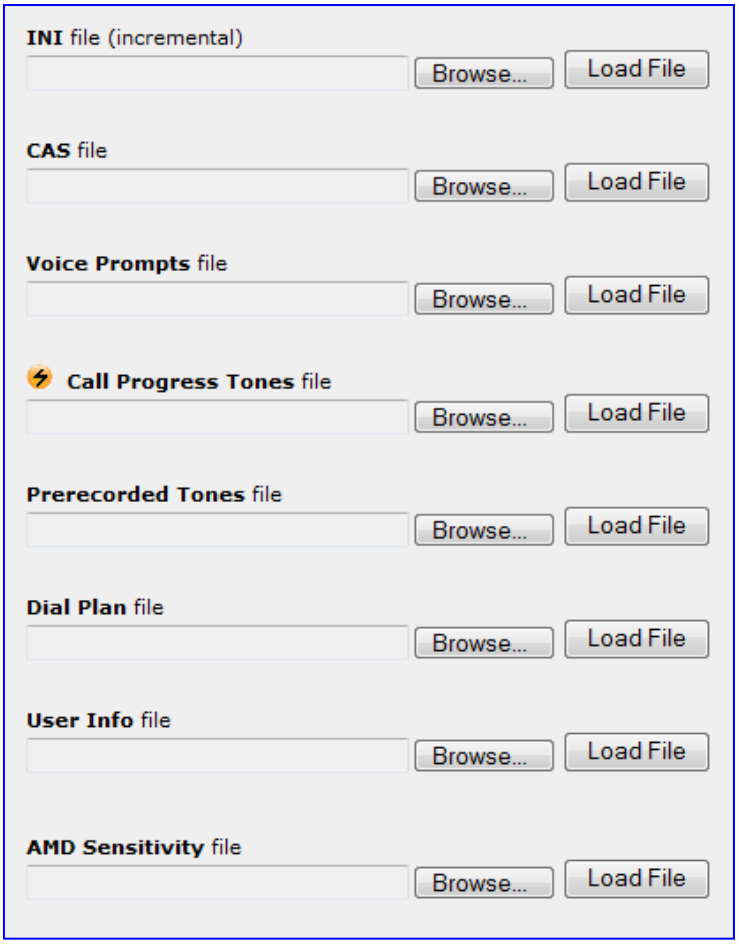
- You can schedule automatic loading of updated auxiliary files using HTTP/HTTPS (for more information, refer to the *Product Reference Manual*).
- For more information on auxiliary files, see 'Auxiliary Configuration Files' on page 684.
- When loading an *ini* file using this Web page, parameters that are excluded from the loaded *ini* file retain their current settings (*incremental*).
- Saving an auxiliary file to flash memory may disrupt traffic on the device. To avoid this, disable all traffic on the device, by performing a graceful lock (see 'Locking and Unlocking the Device' on page 679).
- For deleting auxiliary files, see 'Viewing Device Information' on page 709.

The auxiliary files can be loaded to the device using the Web interface's 'Load Auxiliary Files page, as described in the procedure below.

➢ **To load an auxiliary file to the device using the Web interface:**

1. Open the Load Auxiliary Files page (**Maintenance** tab > **Software Update** menu > **Load Auxiliary Files**).

Figure 33-1: Load Auxiliary Files Page



Note: The appearance of certain file load fields depends on the installed Software Upgrade Key.

2. Click the **Browse** button corresponding to the file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name and path of the file appear in the field next to the **Browse** button.
3. Click the **Load File** button corresponding to the file you want to load.
4. Repeat steps 2 through 3 for each file you want to load.
5. Save the loaded auxiliary files to flash memory, see 'Saving Configuration' on page 680 and reset the device (if you have loaded a Call Progress Tones file), see 'Resetting the Device' on page 677.

33.1.1 Auxiliary Configuration Files

This section describes the auxiliary files that can be loaded to the device:

- Call Progress Tones (see 'Call Progress Tones File' on page 684
- Distinctive Ringing in the ini file (see Distinctive Ringing on page 687)
- Prerecorded Tones (see 'Prerecorded Tones File' on page 688
- Voice Prompts (see Voice Prompts File on page 689)
- CAS (see CAS Files on page 690)
- Dial Plan (see Dial Plan File on page 690)
- User Information (see 'User Information File' on page 691)
- AMD Sensitivity (see AMD Sensitivity File on page 694)

You can load these auxiliary files to the device using one of the following methods:

- Loading the files directly to the device using the device's Web interface (see 'Loading Auxiliary Files' on page 681).
- Specifying the auxiliary file name in the *ini* file (see 'Auxiliary and Configuration Files Parameters' on page 1011) and then loading the *ini* file to the device. The Auxiliary files listed in the *ini* file are then uploaded to the device through TFTP during device startup. If the *ini* file does not contain a specific auxiliary file type, the device uses the last auxiliary file of that type that was stored on its non-volatile memory.

33.1.1.1 Call Progress Tones File

The Call Progress Tones (CPT) and Distinctive Ringing (applicable to analog interfaces) auxiliary file is comprised of two sections:

- The first section contains the definitions of the Call Progress Tones (levels and frequencies) that are detected/generated by the device.
- The second section contains the characteristics of the Distinctive Ringing signals that are generated by the device (see Distinctive Ringing on page 687).

You can use one of the supplied auxiliary files (.dat file format) or create your own file. To create your own file, it's recommended to modify the supplied *usa_tone.ini* file (in any standard text editor) to suit your specific requirements and then convert the modified *ini* file into binary format using the TrunkPack Downloadable Conversion Utility (DConvert). For a description on converting a CPT *ini* file into a binary *dat* file, refer to the *Product Reference Manual*.

Note: Only the *dat* file format can be loaded to the device.

You can create up to 32 different Call Progress Tones, each with frequency and format attributes. The frequency attribute can be single or dual-frequency (in the range of 300 to 1980 Hz) or an Amplitude Modulated (AM). Up to 64 different frequencies are supported. Only eight AM tones, in the range of 1 to 128 kHz, can be configured (the detection range is limited to 1 to 50 kHz). Note that when a tone is composed of a single frequency, the second frequency field must be set to zero.

The format attribute can be one of the following:

- **Continuous:** A steady non-interrupted sound (e.g., a dial tone). Only the 'First Signal On time' should be specified. All other on and off periods must be set to zero. In this case, the parameter specifies the detection period. For example, if it equals 300, the

tone is detected after 3 seconds (300 x 10 msec). The minimum detection time is 100 msec.

- **Cadence:** A repeating sequence of on and off sounds. Up to four different sets of on/off periods can be specified.
- **Burst:** A single sound followed by silence. Only the 'First Signal On time' and 'First Signal Off time' should be specified. All other on and off periods must be set to zero. The burst tone is detected after the off time is completed.

You can specify several tones of the same type. These additional tones are used only for tone detection. Generation of a specific tone conforms to the first definition of the specific tone. For example, you can define an additional dial tone by appending the second dial tone's definition lines to the first tone definition in the *ini* file. The device reports dial tone detection if either of the two tones is detected.

The Call Progress Tones section of the *ini* file comprises the following segments:

- **[NUMBER OF CALL PROGRESS TONES]:** Contains the following key: 'Number of Call Progress Tones' defining the number of Call Progress Tones that are defined in the file.
- **[CALL PROGRESS TONE #X]:** containing the Xth tone definition, starting from 0 and not exceeding the number of Call Progress Tones less 1 defined in the first section (e.g., if 10 tones, then it is 0 to 9), using the following keys:
  - **Tone Type:** Call Progress Tone types:
    - **[1]** Dial Tone
    - **[2]** Ringback Tone
    - **[3]** Busy Tone
    - **[7]** Reorder Tone
    - **[8]** Confirmation Tone
    - **[9]** Call Waiting Tone - heard by the called party
    - **[15]** Stutter Dial Tone
    - **[16]** Off Hook Warning Tone
    - **[17]** Call Waiting Ringback Tone - heard by the calling party
    - **[18]** Comfort Tone
    - **[23]** Hold Tone
    - **[46]** Beep Tone
  - **Tone Modulation Type:** Amplitude Modulated (1) or regular (0)
  - **Tone Form:** The tone's format can be one of the following:
    - Continuous (1)
    - Cadence (2)
    - Burst (3)
  - **Low Freq [Hz]:** Frequency (in Hz) of the lower tone component in case of dual frequency tone, or the frequency of the tone in case of single tone. This is not relevant to AM tones.
  - **High Freq [Hz:** Frequency (in Hz) of the higher tone component in case of dual frequency tone, or zero (0) in case of single tone (not relevant to AM tones).
  - **Low Freq Level [-dBm]:** Generation level 0 dBm to -31 dBm in dBm (not relevant to AM tones).
  - **High Freq Level:** Generation level of 0 to -31 dBm. The value should be set to 32 in the case of a single tone (not relevant to AM tones).
  - **First Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the first cadence on-off cycle. For continuous tones, this parameter defines the detection period. For burst tones, it defines the tone's duration.

- **First Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, this parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, this parameter is ignored.
- **Second Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Second Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Third Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Third Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Fourth Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Fourth Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Carrier Freq [Hz]:** Frequency of the carrier signal for AM tones.
- **Modulation Freq [Hz]:** Frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).
- **Signal Level [-dBm]:** Level of the tone for AM tones.
- **AM Factor [steps of 0.02]:** Amplitude modulation factor (valid range from 1 to 50). Recommended values from 10 to 25.

Notes:

- When the same frequency is used for a continuous tone and a cadence tone, the 'Signal On Time' parameter of the continuous tone must have a value that is greater than the 'Signal On Time' parameter of the cadence tone. Otherwise, the continuous tone is detected instead of the cadence tone.
- The tones frequency must differ by at least 40 Hz between defined tones.

For example, to configure the dial tone to 440 Hz only, enter the following text:

```
[NUMBER OF CALL PROGRESS TONES]
Number of Call Progress Tones=1
#Dial Tone
[CALL PROGRESS TONE #0]
Tone Type=1
Tone Form =1 (continuous)
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10 dBm)
High Freq Level [-dBm]=32 (use 32 only if a single tone is
required)
First Signal On Time [10msec]=300; the dial tone is detected after
3 sec
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

33.1.1.2 Distinctive Ringing

Distinctive Ringing is applicable only to FXS interfaces. Using the Distinctive Ringing section of the Call Progress Tones auxiliary file, you can create up to 16 Distinctive Ringing patterns. Each ringing pattern configures the ringing tone frequency and up to four ringing cadences. The same ringing frequency is used for all the ringing pattern cadences. The ringing frequency can be configured in the range of 10 to 200 Hz with a 5 Hz resolution.

Each of the ringing pattern cadences is specified by the following parameters:

- **Burst Ring On Time:** Configures the cadence to be a burst cadence in the entire ringing pattern. The burst relates to On time and the Off time of the same cadence. It must appear between 'First/Second/Third/Fourth' string and the 'Ring On/Off Time' This cadence rings once during the ringing pattern. Otherwise, the cadence is interpreted as cyclic: it repeats for every ringing cycle.
- **Ring On Time:** Specifies the duration of the ringing signal.
- **Ring Off Time:** Specifies the silence period of the cadence.

The Distinctive Ringing section of the *ini* file format contains the following strings:

- **[NUMBER OF DISTINCTIVE RINGING PATTERNS]:** Contains the following key:
  - 'Number of Distinctive Ringing Patterns' defining the number of Distinctive Ringing signals that are defined in the file.
- **[Ringing Pattern #X]:** Contains the Xth ringing pattern definition (starting from 0 and not exceeding the number of Distinctive Ringing patterns defined in the first section minus 1) using the following keys:
  - **Ring Type:** Must be equal to the Ringing Pattern number.
  - **Freq [Hz]:** Frequency in hertz of the ringing tone.
  - **First (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the first cadence on-off cycle.
  - **First (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the first cadence on-off cycle.
  - **Second (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the second cadence on-off cycle.
  - **Second (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the second cadence on-off cycle.
  - **Third (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the third cadence on-off cycle.
  - **Third (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the third cadence on-off cycle.
  - **Fourth (Burst) Ring On Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.
  - **Fourth (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.

Note: In SIP, the Distinctive Ringing pattern is selected according to the Alert-Info header in the INVITE message. For example:
Alert-Info:<Bellcore-dr2>, or Alert-Info:<http://…/Bellcore-dr2>
'dr2' defines ringing pattern #2. If the Alert-Info header is missing, the default ringing tone (0) is played.

An example of a **ringing burst** definition is shown below:

```
#Three ringing bursts followed by repeated ringing of 1 sec on and
3 sec off.
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=1
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=25
First Burst Ring On Time [10msec]=30
First Burst Ring Off Time [10msec]=30
Second Burst Ring On Time [10msec]=30
Second Burst Ring Off Time [10msec]=30
Third Burst Ring On Time [10msec]=30
Third Burst Ring Off Time [10msec]=30
Fourth Ring On Time [10msec]=100
Fourth Ring Off Time [10msec]=300
```

An example of **various ringing signals** definition is shown below:

```
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=3
#Regular North American Ringing Pattern
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400

#GR-506-CORE Ringing Pattern 1
[Ringing Pattern #1]
Ring Type=1
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400

#GR-506-CORE Ringing Pattern 2
[Ringing Pattern #2]
Ring Type=2
Freq [Hz]=20
First Ring On Time [10msec]=80
First Ring Off Time [10msec]=40
Second Ring On Time [10msec]=80
Second Ring Off Time [10msec]=400
```

33.1.1.3 Prerecorded Tones File

The CPT file mechanism has several limitations such as a limited number of predefined tones and a limited number of frequency integrations in one tone. To overcome these limitations and provide tone generation capability that is more flexible, the Prerecorded Tones (PRT) file can be used. If a specific prerecorded tone exists in the PRT file, it takes precedence over the same tone that exists in the CPT file and is played instead of it.

Note: The PRT are used only for generation of tones. Detection of tones is performed according to the CPT file.

The PRT is a *.dat* file containing a set of prerecorded tones that can be played by the device. Up to 40 tones (totaling approximately 10 minutes) can be stored in a single PRT file on the device's flash memory. The prerecorded tones are prepared offline using standard recording utilities (such as CoolEdit™) and combined into a single file using the DConvert utility (refer to the *Product Reference Manual*).

The raw data files must be recorded with the following characteristics:

- **Coders:** G.711 A-law or G.711 µ-law
- **Rate:** 8 kHz
- **Resolution:** 8-bit
- **Channels:** mono

Once created, the PRT file can then be loaded to the device using the Web interface (see 'Loading Auxiliary Files' on page 681).

The prerecorded tones are played repeatedly. This allows you to record only part of the tone and then play the tone for the full duration. For example, if a tone has a cadence of 2 seconds on and 4 seconds off, the recorded file should contain only these 6 seconds. The PRT module repeatedly plays this cadence for the configured duration. Similarly, a continuous tone can be played by repeating only part of it.

33.1.1.4 Voice Prompts File

The Voice Prompts (VP) file contains a set of voice prompts (or announcements) that can be played by the device during operation. The voice announcements are prepared offline using standard recording utilities and then combined into a single file using the DConvert utility. The VP file can then be loaded to the device using the Web interface.

The VP file is a collection of raw voice recordings and/or *wav* files. These recordings can be prepared using standard utilities such as CoolEdit and Goldwave™.

The raw data files must be recorded with the following characteristics:

- **Coders:** Linear G.711 A-law or G.711 µ-law
- **Rate:** 8000 kHz
- **Resolution:** 8-bit
- **Channels:** mono

When the list of recorded files is converted to a single *voiceprompts.dat* file, every Voice Prompt is tagged with an ID number, starting with '1'. This ID is later used by the device to play the correct announcement. Up to 1,000 Voice Prompts can be defined. If the size of the combined VP file is less than 1 MB, it can be permanently stored on flash memory. Larger files (up to 10 MB) are stored in RAM, and should be loaded again after the device is reset.

The device can be provided with a professionally recorded English (U.S.) VP file.

➢ **To generate and load the VP file:**

1. Prepare one or more voice files using standard utilities.
2. Use the DConvert utility to generate the *voiceprompts.dat* file from the pre-recorded voice messages (refer to the *Product Reference Manual*).
3. Load the *voiceprompts.dat* file to the device using the Web interface (see 'Loading Auxiliary Files' on page 681).

33.1.1.5 CAS Files

The CAS auxiliary files contain the CAS Protocol definitions that are used for CAS-terminated trunks. You can use the supplied files or construct your own files. Up to eight files can be loaded to the device. Different files can be assigned to different trunks (CASTableIndex_x) and different CAS tables can be assigned to different B-channels (CASChannelIndex).

The CAS files can be loaded to the device using the Web interface or *ini* file (see 'Loading Auxiliary Files' on page 681).

Note: All CAS files loaded together must belong to the same Trunk Type (i.e., either E1 or T1).

33.1.1.6 Dial Plan File

The Dial Plan file contains a list of up to eight dial plans, supporting a total of up to 8,000 user-defined, distinct prefixes (e.g. area codes, international telephone number patterns) for the PSTN to which the device is connected. The Dial Plan is used for the following:

- ISDN Overlap Dialing, FXS, and FXO collecting digit mode (Tel-to-IP calls): The file includes up to eight patterns (i.e., eight dial plans). These allow the device to know when digit collection ends, after which it starts sending all the collected (or dialed) digits (in the INVITE message). This also provides enhanced digit mapping.
- CAS E1 MF-CR2 (Tel-to-IP calls): Useful for E1 MF-CR2 variants that do not support I-15 terminating digits (e.g., in Brazil and Mexico). The Dial Plan file allows the device to detect end-of-dialing in such cases. The CasTrunkDialPlanName_x ini file parameter determines which dial plan (in the Dial Plan file) to use for a specific trunk.

Note: To use this Dial Plan, you must also use a special CAS .dat file that supports this feature (contact your AudioCodes sales representative).

- Prefix tags (for IP-to-Tel routing): Provides enhanced routing rules based on Dial Plan prefix tags. For more information, see Dial Plan Prefix Tags for IP-to-Tel Routing on page 379.

The Dial Plan file is first created using a text-based editor (such as Notepad) and saved with the file extension .ini. This *ini* file is then converted to a binary file (.dat) using the DConvert utility (refer to the *Product Reference Manual*). Once converted, it can then be loaded to the device using the Web interface (see 'Loading Auxiliary Files' on page 681).

The Dial Plan file must be prepared in a textual *ini* file with the following syntax:

- Every line in the file defines a known dialing prefix and the number of digits expected to follow that prefix. The prefix must be separated from the number of additional digits by a comma (',').
- Empty lines are ignored.
- Lines beginning with a semicolon (';') are ignored.
- Multiple dial plans may be specified in one file; a name in square brackets on a separate line indicates the beginning of a new dial plan. Up to eight dial plans can be defined.
- Asterisks ('*') and number-signs ('#') can be specified as part of the prefix.

- Numeric ranges are allowed in the prefix.
- A numeric range is allowed in the number of additional digits.

Notes:

- The prefixes must not overlap. Attempting to process an overlapping configuration by the DConvert utility results in an error message specifying the problematic line.
- For more information on working with Dial Plan files, see 'External Dial Plan File' on page 377.

An example of a Dial Plan file in *ini*-file format (i.e., before converted to .dat) that contains two dial plans is shown below:

```
; Example of dial-plan configuration.
; This file contains two dial plans:
[ PLAN1 ]
; Defines cellular/VoIP area codes 052, 054, and 050.
; In these area codes, phone numbers have 8 digits.
052,8
054,8
050,8
; Defines International prefixes 00, 012, 014.
; The number following these prefixes may
; be 7 to 14 digits in length.
00,7-14
012,7-14
014,7-14
; Defines emergency number 911.
; No additional digits are expected.
911,0
[ PLAN2 ]
; Defines area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
0[2-4],7
; Operator services starting with a star: *41, *42, *43.
; No additional digits are expected.
*4[1-3],0
```

33.1.1.7 User Information File

The User Information file is a text-based file that can be used for mapping PBX extensions connected to the device to "global" IP numbers or for creating an SBC users database.

The User Information file can be loaded to the device by using one of the following methods:

- *ini* file, using the parameter UserInfoFileName (described in 'Auxiliary and Configuration Files Parameters' on page 1011)
- Web interface (see 'Loading Auxiliary Files' on page 681)
- Automatic update mechanism, using the parameter UserInfoFileURL (refer to the *Product Reference Manual*)

33.1.1.8 User Information File for PBX Extensions and "Global" Numbers

The User Information file can be used to map PBX extensions, connected to the device, to global IP numbers. In this context, a global phone number (alphanumerical) serves as a routing identifier for calls in the 'IP world'. The PBX extension uses this mapping to emulate the behavior of an IP phone.

Note: By default, the mapping mechanism is disabled and must be activated using the parameter EnableUserInfoUsage.

The maximum size of the file is 10,800 bytes (for analog modules) and 108,000 bytes for digital modules. Each line in the file represents a mapping rule of a single PBX extension. Up to 1,000 rules can be configured. Each line includes five items separated with commas. The items are described in the table below:

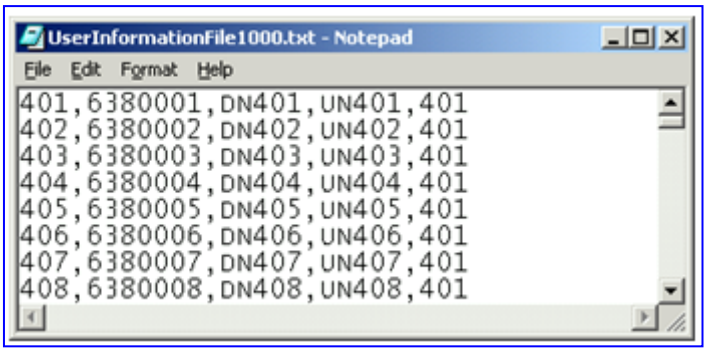
Table 33-2: User Information Items

Item	Description	Maximum Size (Characters)
PBX extension #	The relevant PBX extension number.	10
Global phone #	The relevant global phone number.	20
Display name	A string that represents the PBX extensions for the Caller ID.	30
Username	A string that represents the user name for SIP registration.	40
Password	A string that represents the password for SIP registration.	20

Note: For FXS ports, when the device is required to send a new request with the 'Authorization' header (for example, after receiving a SIP 401 reply), it uses the user name and password from the Authentication table. To use the username and password from the User Info file, change the parameter 'Password' from its default value.

An example of a User Information file is shown in the figure below:

Figure 33-2: Example of a User Information File



Note: The last line in the User Information file must end with a carriage return (i.e., by pressing the <Enter> key).

Each PBX extension registers separately (a REGISTER message is sent for each entry only if AuthenticationMode is set to Per Endpoint) using the"Global phone number" in the From/To headers. The REGISTER messages are sent gradually. Initially, the device sends requests according to the maximum number of allowed SIP dialogs (configured by the parameter NumberOfActiveDialogs). After each received response, the subsequent request is sent. Therefore, no more than NumberOfActiveDialogs dialogs are active simultaneously. The user name and password are used for SIP Authentication when required.

The calling number of outgoing Tel-to-IP calls is translated to a "Global phone number" only after Tel-to-IP manipulation rules (if defined) are performed. The Display Name is used in the From header in addition to the "Global phone number". The called number of incoming IP-to-Tel calls is translated to a PBX extension only after IP-to-Tel manipulation rules (if defined) are performed.

33.1.1.8.1.1 User Information File for SBC Users Database

You can create an SBC users database from a loaded User Information file. The same text-based User Information file for the GW / IP2IP application is used for this feature, enabled by the EnableUserInfoUsage parameter. Up to 600 SBC users can be defined in the User Information file.

The User Information file lists the SBC users under the "**[SBC]"** section and the Gateway PBX users under the "**[GW]"** section, as shown below:

```
[ GW ]
FORMAT
PBXExtensionNum,GlobalPhoneNum,DisplayName,UserName,Password
4000,039764000,lee,lee_user,lee_pass
4001,0547771234,mike,mike_user,mike_pass
[ SBC ]
FORMAT LocalUser,UserName,Password,IPGroupID
john,john_user,john_pass,2
sue,sue_user,sue_pass,1
```

where:

- LocalUser identifies the user and is used as the URI user part for the AOR in the database
- UserName is the user's authentication username.
- Password is the user's authentication password.
- IPGroupID is the IP Group ID to which the user belongs and is used as the URI source host part for the AOR in the database

The SBC users database can be used for the following:

- Register to an external registrar server on behalf of a specific user.
- Authenticate (for any SIP request and as a client) on behalf of a specific user if challenged by an external server.
- Authenticate (as a server) incoming user requests (for SBC security).

If the SBC registers on behalf of users and the users do not perform registration at all, any SIP request destined to the user is routed to the Proxy Set associated with the user's IP Group.

33.1.1.9 AMD Sensitivity File

The AMD Sensitivity file allows you to configure the device with different AMD Sensitivity suites. You can load the device with up to four AMD Sensitivity suites. Each suite can be configured to a different language, country or region, thereby fine tuning the detection algorithm of the DSP according to requirements.

The structure of the file can be viewed in the example below. Each file consists of at least one parameter suite with its suite ID. Each parameter suite consists of up to 16 sensitivity levels, where each level possessing 3 coefficients A, B and C. When loading a new parameter suite, the existing parameter suite with the same ID is overwritten.

The file is created in .xml format and installed on the device as a binary file (with a .dat extension). The XML to binary file format is processed by the DConvert utility (refer to the *Product Reference Manual*).

The file can be installed on the board in the following ways:

- TFTP at initialization time, by setting the *ini* file parameter AMDSensitivityFileName with the .dat file name, and adding the file to the TFTP directory.
- Auxiliary files Web page (see 'Loading Auxiliary Files' on page 681).
- Using the AutoUpdate mechanism (refer to the *Product Reference Manual*). In this case the AMDSensitivityFileUrl parameter must be set using SNMP or *ini* file.

The following example shows an xml file with two parameter suites:

- Parameter Suite 0 with 6 sensitivity levels,
- Parameter Suite 2 with 3 sensitivity levels.

```
<AMDSENSITIVITY>
<PARAMETERSUIT>
    <PARAMETERSUITID>0</PARAMETERSUITID>
    <!-- First  language/country -->
    <NUMBEROFLEVELS>8</NUMBEROFLEVELS>
          <AMDSENSITIVITYLEVEL>
          <!-- Level 0 -->
                <AMDCOEFFICIENTA>15729</AMDCOEFFICIENTA>
                <AMDCOEFFICIENTB>58163</AMDCOEFFICIENTB>
                <AMDCOEFFICIENTC>32742</AMDCOEFFICIENTC>
          </AMDSENSITIVITYLEVEL>
          <AMDSENSITIVITYLEVEL>
          <!-- Level 1 -->
                <AMDCOEFFICIENTA>19923</AMDCOEFFICIENTA>
                <AMDCOEFFICIENTB>50790</AMDCOEFFICIENTB>
                <AMDCOEFFICIENTC>30720</AMDCOEFFICIENTC>
          </AMDSENSITIVITYLEVEL>
          <AMDSENSITIVITYLEVEL>
          <!-- Level 2 -->
                <AMDCOEFFICIENTA>10486</AMDCOEFFICIENTA>
                <AMDCOEFFICIENTB>57344</AMDCOEFFICIENTB>
                <AMDCOEFFICIENTC>25600</AMDCOEFFICIENTC>
          </AMDSENSITIVITYLEVEL>
          <AMDSENSITIVITYLEVEL>
          <!-- Level 3 -->
                <AMDCOEFFICIENTA>8389</AMDCOEFFICIENTA>
                <AMDCOEFFICIENTB>62259</AMDCOEFFICIENTB>
                <AMDCOEFFICIENTC>23040</AMDCOEFFICIENTC>
```

```
            </AMDSENSITIVITYLEVEL>
            <AMDSENSITIVITYLEVEL>
            <!-- Level 4 -->
                  <AMDCOEFFICIENTA>10486</AMDCOEFFICIENTA>
                  <AMDCOEFFICIENTB>50790</AMDCOEFFICIENTB>
                  <AMDCOEFFICIENTC>28160</AMDCOEFFICIENTC>
            </AMDSENSITIVITYLEVEL>
            <AMDSENSITIVITYLEVEL>
            <!-- Level 5 -->
                  <AMDCOEFFICIENTA>6291</AMDCOEFFICIENTA>
                  <AMDCOEFFICIENTB>58982</AMDCOEFFICIENTB>
                  <AMDCOEFFICIENTC>23040</AMDCOEFFICIENTC>
            </AMDSENSITIVITYLEVEL>
            <AMDSENSITIVITYLEVEL>
            <!-- Level 6 -->
                  <AMDCOEFFICIENTA>7864</AMDCOEFFICIENTA>
                  <AMDCOEFFICIENTB>58982</AMDCOEFFICIENTB>
                  <AMDCOEFFICIENTC>12800</AMDCOEFFICIENTC>
            </AMDSENSITIVITYLEVEL>
            <AMDSENSITIVITYLEVEL>
            <!-- Level 7 -->
                  <AMDCOEFFICIENTA>7340</AMDCOEFFICIENTA>
                  <AMDCOEFFICIENTB>64717</AMDCOEFFICIENTB>
                  <AMDCOEFFICIENTC>3840</AMDCOEFFICIENTC>
            </AMDSENSITIVITYLEVEL>
 </PARAMETERSUIT>
<PARAMETERSUIT>
    <PARAMETERSUITID>2</PARAMETERSUITID>
    <!-- Second  language/country -->
    <NUMBEROFLEVELS>3</NUMBEROFLEVELS>
            <AMDSENSITIVITYLEVEL>
            <!-- Level 0 -->
                  <AMDCOEFFICIENTA>15729</AMDCOEFFICIENTA>
                  <AMDCOEFFICIENTB>58163</AMDCOEFFICIENTB>
                  <AMDCOEFFICIENTC>32742</AMDCOEFFICIENTC>
            </AMDSENSITIVITYLEVEL>
            <AMDSENSITIVITYLEVEL>
            <!-- Level 1 -->
                  <AMDCOEFFICIENTA>5243</AMDCOEFFICIENTA>
                  <AMDCOEFFICIENTB>9830</AMDCOEFFICIENTB>
                  <AMDCOEFFICIENTC>24320</AMDCOEFFICIENTC>
            </AMDSENSITIVITYLEVEL>
            <AMDSENSITIVITYLEVEL>
            <!-- Level 2 -->
                  <AMDCOEFFICIENTA>13107</AMDCOEFFICIENTA>
                  <AMDCOEFFICIENTB>61440</AMDCOEFFICIENTB>
                  <AMDCOEFFICIENTC>26880</AMDCOEFFICIENTC>
            </AMDSENSITIVITYLEVEL>
</PARAMETERSUIT>
</AMDSENSITIVITY>
```

33.2 Loading Software Upgrade Key

The Software Upgrade Key Status page allows you to load a new Software Upgrade Key to the device. The device is supplied with a Software Upgrade Key, which determines the device's supported features, capabilities, and available resources. The availability of certain Web pages depends on the loaded Software Upgrade Key. You can upgrade or change your device's supported features by purchasing a new Software Upgrade Key to match your requirements.

The Software Upgrade Key is provided in string format in a text-based file (.out). When you load a Software Upgrade Key, it is loaded to the device's non-volatile flash memory and overwrites the previously installed key.

You can load a Software Upgrade Key using one of the following management tools:

- Web interface
- AudioCodes' EMS (refer to *EMS User's Manual* or *EMS Product Description*)

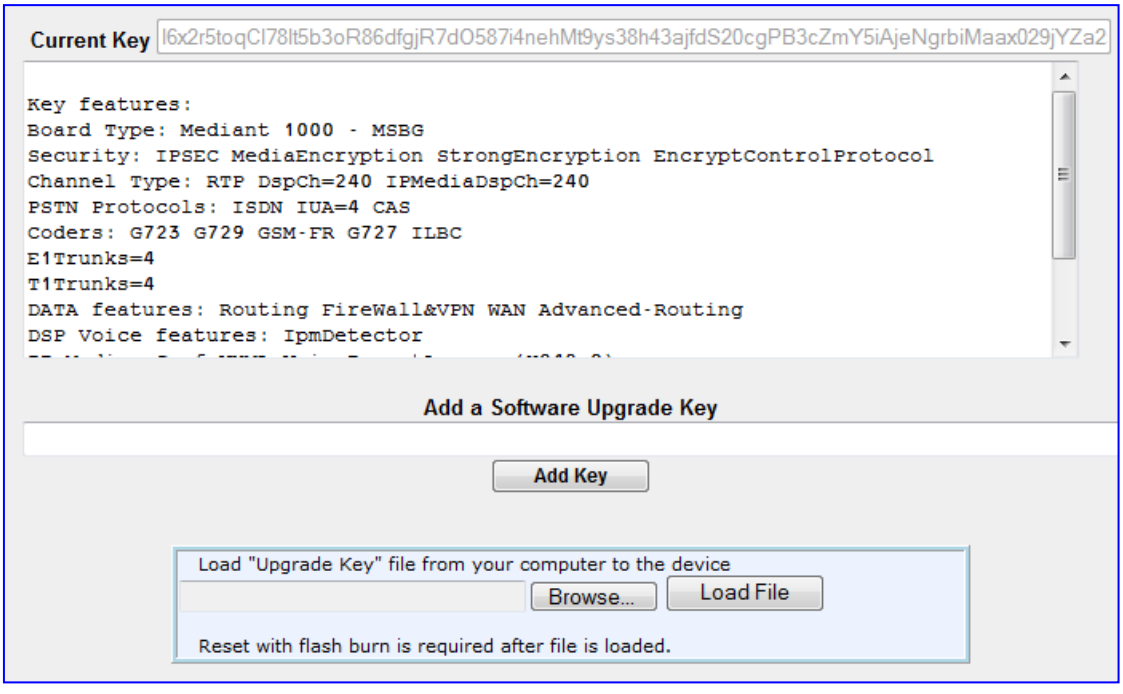
Warning: Do not modify the contents of the Software Upgrade Key file.

Note: The Software Upgrade Key is an encrypted key.

➢ To load a Software Upgrade Key:

1. Open the Software Upgrade Key Status page (**Maintenance** tab > **Software Update** menu > **Software Upgrade Key**).

Figure 33-3: Software Upgrade Key Status Page



2. Backup your current Software Upgrade Key as a precaution so that you can re-load this backup key to restore the device's original capabilities if the new key doesn't comply with your requirements:
   a. In the 'Current Key' field, copy the string of text and paste it into any standard text file.
   b. Save the text file to a folder on your PC with a name of your choosing and file extension .out.
3. Open the new Software Upgrade Key file and ensure that the first line displays '**[LicenseKeys]**' and that it contains one or more lines in the following format: S/N<serial number> = <long Software Upgrade Key string>

   For example: S/N370604 = jCx6r5tovCIKaBBbhPtT53Yj...

   One S/N must match the serial number of your device. The device's serial number can be viewed in the 'Device Information page (see 'Viewing Device Information' on page 709).
4. Follow one of the following procedures, depending on whether you are loading a single or multiple key S/N lines:
   - **Single key S/N line:**
     a. Open the Software Upgrade Key text file (using, for example, Microsoft® Notepad).
     b. Select and copy the key string and paste it into the field 'Add a Software Upgrade Key'.
     c. Click the **Add Key** button.
   - **Multiple S/N lines (as shown below):**

Figure 33-4: Software Upgrade Key with Multiple S/N Lines

   a. In the 'Load Upgrade Key file' field, click the **Browse** button and navigate to the folder in which the Software Upgrade Key text file is located on your PC.
   b. Click the **Load File** button; the new key is loaded to the device and validated. If the key is valid, it is burned to memory and displayed in the 'Current Key' field.

5. Verify that the Software Upgrade Key file was successfully loaded to the device, by using one of the following methods:
   - In the 'Key features' group, ensure that the features and capabilities activated by the installed string match those that were ordered.
   - Access the Syslog server (refer to the *Product Reference Manual*) and ensure that the following message appears in the Syslog server:
     "S/N___ Key Was Updated. The Board Needs to be Reloaded with ini file\n".
6. Reset the device; the new capabilities and resources are active.

Note: If the Syslog server indicates that the Software Upgrade Key file was unsuccessfully loaded (i.e., the 'SN_' line is blank), do the following preliminary troubleshooting procedures:

1. Open the Software Upgrade Key file and check that the S/N line appears. If it does not appear, contact AudioCodes.
2. Verify that you've loaded the correct file. Open the file and ensure that the first line displays **[LicenseKeys].**
3. Verify that the content of the file has not been altered.

33.3 Software Upgrade Wizard

The Software Upgrade Wizard allows you to upgrade the device's firmware (compressed .cmp file) as well as load an *ini* file and/or auxiliary files (typically loaded using the Load Auxiliary File page described in 'Loading Auxiliary Files' on page 681). However, it is mandatory when using the wizard to first load a .cmp file to the device. You can then choose to also load an *ini* file and/or auxiliary files, but this cannot be done without first loading a .cmp file. For the *ini* and each auxiliary file type, you can choose to load a new file or not load a file but use the existing file (i.e., maintain existing configuration) running on the device.

Warning: The Software Upgrade Wizard requires the device to be reset at the end of the process, which may disrupt traffic. To avoid this, disable all traffic on the device before initiating the wizard, by performing a graceful lock (see 'Basic Maintenance' on page 677).

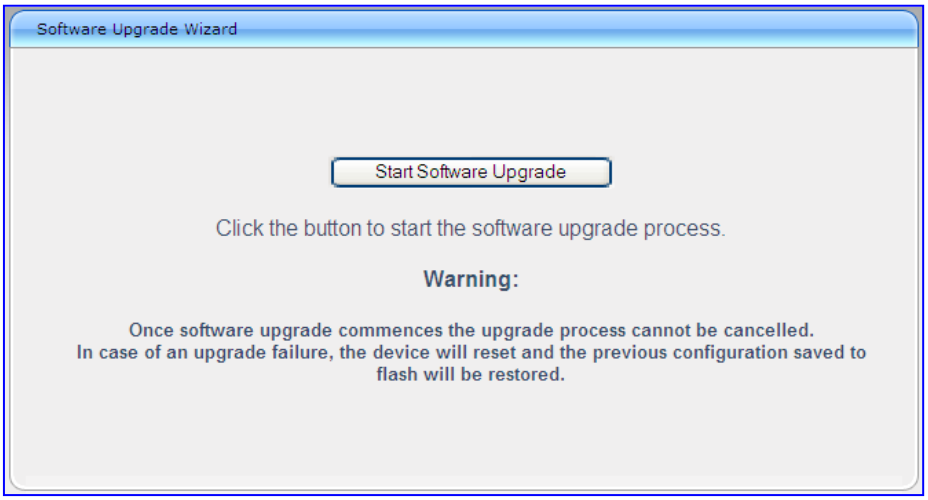
Notes:

- You can get the latest software files from AudioCodes Web site at http://www.audiocodes.com/downloads.
- Before upgrading the device, it is recommended that you save a copy of the device's configuration settings (i.e., *ini* file and data file) to your PC. If an upgrade failure occurs, you can then restore your configuration settings by uploading the backup file to the device. For saving and restoring configuration, see 'Backing Up and Loading Configuration File' on page 702.
- Before you can load an *ini* or auxiliary file, you must first load a .cmp file.
- When you activate the wizard, the rest of the Web interface is unavailable. After the files are successfully loaded, access to the full Web interface is restored.
- If you upgraded your .cmp and the "SW version mismatch" message appears in the Syslog or Web interface, then your Software Upgrade Key does not support the new .cmp file version. Contact AudioCodes support for assistance.
- If you use the wizard to load an *ini* file, parameters excluded from the *ini* file are assigned default values (according to the .cmp file running on the device), thereby, overriding values previously defined for these parameters.
- You can schedule automatic loading of these files using HTTP/HTTPS (refer to the *Product Reference Manual*).

➢ **To load files using the Software Upgrade Wizard:**

1. Stop all traffic on the device using the Graceful Lock feature (refer to the warning bulletin above).
2. Open the Software Upgrade wizard, by performing one of the following:
   - Select the **Maintenance** tab, click the **Software Update** menu, and then click **Software Upgrade Wizard**.
   - On the toolbar, click **Device Actions**, and then choose **Software Upgrade Wizard**.

Figure 33-5: Start Software Upgrade Wizard Screen



3. Click the **Start Software Upgrade** button; the wizard starts, requesting you to browses to a .cmp file for uploading.

Note: At this stage, you can quit the Software Update Wizard, by clicking **Cancel**, without requiring a device reset. However, once you start uploading a cmp file, the process must be completed with a device reset. If you choose to quit the process in any of the subsequent pages, the device resets.

4. Click the **Browse** button, navigate to the .cmp file, and then click **Load File**; a progress bar appears displaying the status of the loading process. When the .cmp file is successfully loaded to the device, a message appears notifying you of this.
5. If you want to load **only** a .cmp file, then click the **Reset** button to reset the device with the newly loaded .cmp file, utilizing the existing configuration (*ini*) and auxiliary files. To load additional files, skip to Step 7.

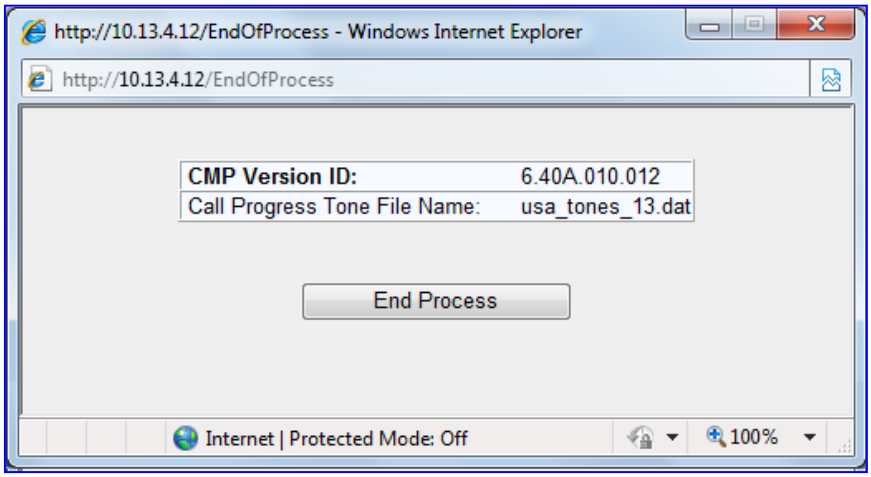
Note: Device reset may take a few minutes depending on cmp file version (this may even take up to 10 minutes).

6. Click the **Next** button; the wizard page for loading an *ini* file appears. You can now perform one of the following:
   - Load a new *ini* file: Click **Browse**, navigate to the *ini* file, and then click **Send File**; the *ini* file is loaded to the device and you're notified as to a successful loading.
   - Retain the existing configuration (*ini* file): Do not select an *ini* file, and ensure that the 'Use existing configuration' check box is selected (default).
   - Return the device's configuration settings to factory defaults: Do not select an *ini* file, and clear the 'Use existing configuration' check box.
7. Click the **Next** button to progress to the relevant wizard pages for loading the desired auxiliary files. To return to the previous wizard page, click the **Back** button. As you navigate between wizard pages, the relevant file type corresponding to the Wizard page is highlighted in the left pane.
8. When you have completed loading all the desired files, click the **Next** button until the last wizard page appears ("FINISH" is highlighted in the left pane).
9. Click the **Reset** button to complete the upgrade process; the device 'burns' the newly loaded files to flash memory and then resets the device.

Note: Device reset may take a few minutes (depending on .cmp file version, this may even take up to 30 minutes).

After the device resets, the End of Process wizard page appears displaying the new .cmp and auxiliary files loaded to the device.

Figure 33-6: End Process Wizard Page



10. Click **End Process** to close the wizard; the Web Login dialog box appears.
11. Enter your login user name and password, and then click **OK**; a message box appears informing you of the new .cmp file.
12. Click **OK**; the Web interface becomes active, reflecting the upgraded device.

33.4 Backing Up and Loading Configuration File

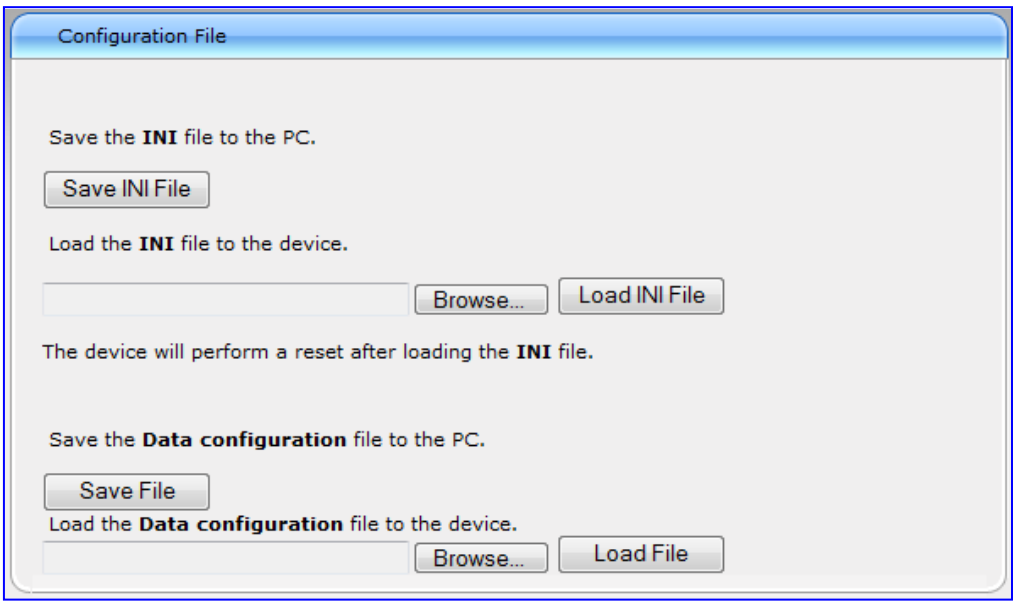
You can save a copy/backup of the device's current configuration settings as an *ini* file to a folder on your PC, using the 'Configuration File page. The saved *ini* file includes only parameters that were modified and parameters with other than default values. The Configuration File page also allows you to load an *ini* file to the device. If the device has "lost" its configuration, you can restore the device's configuration by loading the previously saved *ini* file or by simply loading a newly created *ini* file.

Note: When loading an *ini* file using this Web page, parameters not included in the *ini* file are reset to default settings.

➢ **To save the ini / data file:**

1. Open the Configuration File page (**Maintenance** tab > **Software Update** menu > **Configuration File**). You can also access this page from the toolbar, by clicking **Device Actions**, and then choosing **Load Configuration File** or **Save Configuration File**.

Figure 33-7: Configuration File Page



2. To save the Voice *ini* file to a folder on your PC, perform the following:
   a. Click the **Save INI File** button; the 'File Download' dialog box appears.
   b. Click the **Save** button, navigate to the folder in which you want to save the *ini* file on your PC, and then click **Save**; the device copies the *ini* file to the selected folder.
3. To save the Data configuration ini file to a folder on your PC, perform the following:
   a. Under the 'Save the Data configuration file to the PC' group, click the **Save File** button; the 'File Download' dialog box appears.
   b. Click the **Save** button, navigate to the folder in which you want to save the file on your PC, and then click Save; the device saves the Data ini file to the selected folder.

➢ **To load the ini / data file:**

1. To load the Voice *ini* file to the device, perform the following:
   a. Click the **Browse** button, navigate to the folder in which the *ini* file is located, select the file, and then click **Open**; the name and path of the file appear in the field beside the **Browse** button.
   b. Click the **Load INI File** button, and then at the prompt, click **OK**; the device uploads the *ini* file and then resets (from the *cmp* version stored on the flash memory). Once complete, the Login screen appears, requesting you to enter your user name and password.
2. To load the Data configuration ini file to the device, perform the following:
   a. Under the 'Send the Data Configuration file to the device' group, click the **Browse** button, navigate to the folder in which the file is located, select the file, and then click **Open**; the name and path of the file appear in the field beside the **Browse** button.
   b. Click the **Load File** button, and then at the prompt, click **OK**; the device uploads the file and then resets (from the cmp version stored on the flash memory). Once complete, the Login screen appears, requesting you to enter your user name and password.

Reader's Notes

34 Restoring Factory Default Settings

You can restore the device's configuration to factory defaults using one of the following methods:

- Using the CLI (see 'Restoring Defaults using CLI' on page 705)
- Using the hardware Reset button (see Restoring Defaults using Hardware Reset Button on page 706)
- Loading an empty *ini* file (see 'Restoring Defaults using an ini File' on page 706)

34.1 Restoring Defaults using CLI

The device can be restored to factory defaults using CLI, as described in the procedure below.

➢ **To restore factory defaults using CLI:**

1. Access the CLI:
   a. Connect the RS-232 serial port of the device to the communication port on your PC. For cabling the device, refer to the *Hardware Installation Manual*.
   b. Establish serial communication with the device using a serial communication program (such as HyperTerminal™) with the following communication port settings:
      - **Baud Rate:** 115,200 bps
      - **Data Bits:** 8
      - **Parity:** None
      - **Stop Bits:** 1
      - **Flow Control:** None
2. At the CLI prompt, type the username (default is "Admin" - case sensitive), and then press Enter:

   ```
   # Username: Admin
   ```

3. At the prompt, type the password (default is "Admin" - case sensitive), and then press Enter:

   ```
   # Password: Admin
   ```

4. At the prompt, type the following, and then press Enter:

   ```
   # enable
   ```

5. At the prompt, type the password again, and then press Enter:

   ```
   # Password: Admin
   ```

6. At the prompt, type the following to reset the device to default settings, and then press Enter:

   ```
   # write factory
   ```

34.2 Restoring Defaults using Hardware Reset Button

The device's hardware Reset pinhole button can be used to reset the device to default settings.

➢ **To restore default settings using the hardware Reset button:**

- With a paper clip or any other similar pointed object, press and hold down the Reset button (located on the CRMX module) for at least 12 seconds (but **no more** than 25 seconds).

34.3 Restoring Defaults using an ini File

You can restore the device to factory default settings by loading an empty *ini* file to the device, using the Web interface's Configuration File page (see 'Backing Up and Loading Configuration File' on page 702). The only settings that are not restored to default are the management (OAMP) LAN IP address and the Web interface's login user name and password. The loaded *ini* file must be empty (i.e., contain no parameters), or include only comment signs (i.e., semicolons ";") preceding lines (parameters). The default values assigned to the parameters are according to the *cmp* file running on the device.

Part VII

Status, Performance Monitoring & Reporting

This part describes how to view the status of the <device>, monitor its performance, and report performance information to external application servers.

Reader's Notes

35 System Status

This section describes how to view system status.

- Device information - see 'Viewing Device Information' on page 709
- Ethernet port onformation - see 'Viewing Ethernet Port Information' on page 710

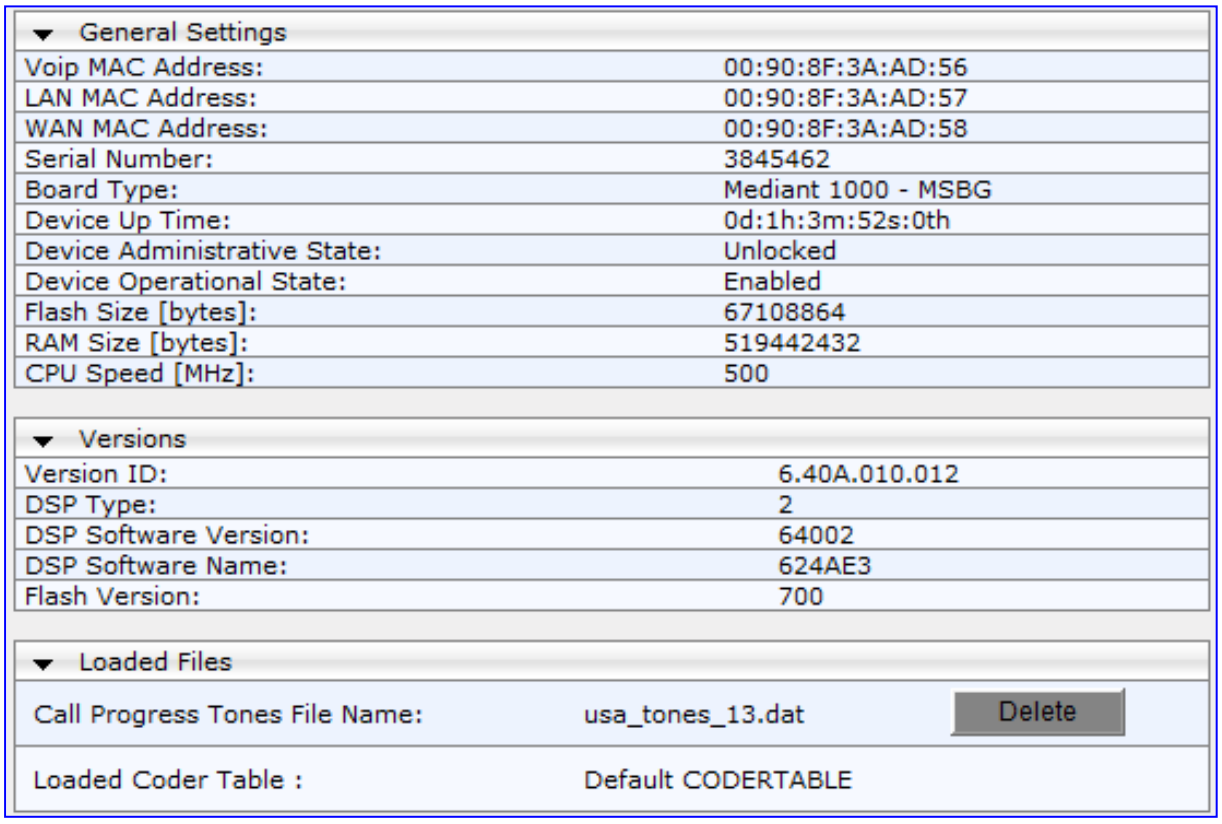
35.1 Viewing Device Information

The Device Information page displays the device's specific hardware and software product information. This information can help you expedite troubleshooting. Capture the page and e-mail it to AudioCodes Technical Support personnel to ensure quick diagnosis and effective corrective action. This page also displays any loaded files used by the device (stored in the RAM) and allows you to remove them.

➢ **To access the Device Information page:**

- Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**).

Figure 35-1: Device Information Page



➢ **To delete a loaded file:**

- Click the **Delete** button corresponding to the file that you want to delete. Deleting a file takes effect only after device reset (see 'Resetting the Device' on page 677).

35.2 Viewing Ethernet Port Information

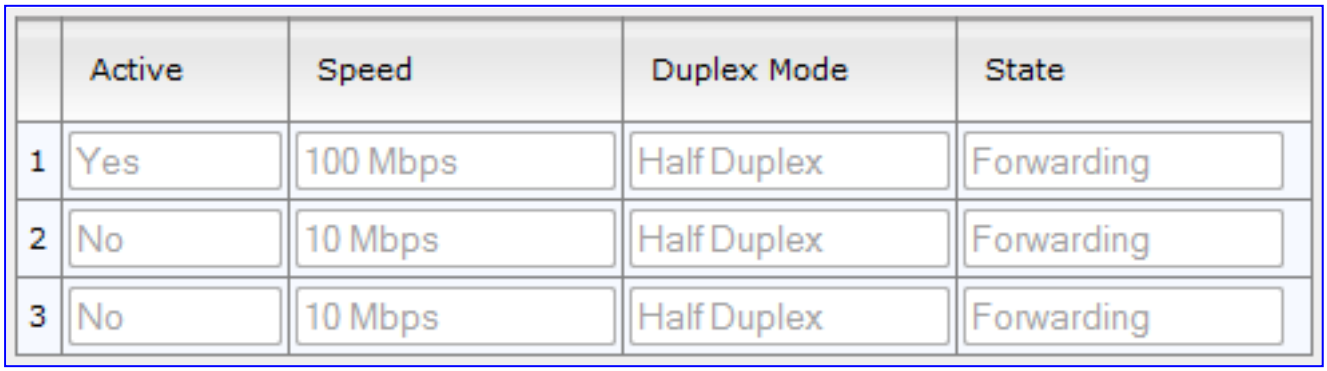
The Ethernet Port Information page displays read-only information on the Ethernet port connections. This includes information such as activity status, duplex mode, and speed.

Note: The Ethernet Port Information page can also be accessed from the Home page (see 'Using the Home Page' on page 70).

➢ **To view Ethernet port information:**

- Open the Ethernet Port Information page (**Status & Diagnostics** tab > **System Status** menu > **Ethernet Port Information**).

Figure 35-2: Ethernet Port Information Page



	Active	Speed	Duplex Mode	State
1	Yes	100 Mbps	Half Duplex	Forwarding
2	No	10 Mbps	Half Duplex	Forwarding
3	No	10 Mbps	Half Duplex	Forwarding

Table 35-1: Ethernet Port Information Parameters

Parameter	Description
Active	Displays whether the port is active or not.
Speed	Displays the speed (in Mbps) of the Ethernet port.
Duplex Mode	Displays whether the port is half- or full-duplex mode.
State	Displays one of the following: ▪ Forwarding - port is receiving and sending data. ▪ Disabled - port is disabled

36 Carrier-Grade Alarms

This section describes how to view the following types of alarms:

- Active alarms - see 'Viewing Active Alarms' on page 711
- Alarm history - see 'Viewing Alarm History' on page 712

36.1 Viewing Active Alarms

The Active Alarms page displays a list of currently active alarms. You can also access this page from the Home page (see 'Using the Home Page' on page 70).

➢ **To view the list of active alarms:**

- Open the Active Alarms page (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Active Alarms**).

Figure 36-1: Active Alarms Page

Sequential number	Severity	Source	Description	Date
1	Major	Board#1/WanLink#1	WAN link alarm. FE interface 1 is down.	25.8.2011 , 16:28:47
2	Minor	Board#1/EthernetLink#2	Ethernet link alarm. LAN port number 2 is down.	25.8.2011 , 16:28:47
3	Minor	Board#1/EthernetLink#3	Ethernet link alarm. LAN port number 3 is down.	25.8.2011 , 16:28:47

For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical - alarm displayed in red
  - Major - alarm displayed in orange
  - Minor - alarm displayed in yellow
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

You can view the next 20 alarms (if exist), by clicking the **Go to page** button.

36.2 Viewing Alarm History

The Alarms History page displays a list of alarms that have been raised and traps that have been cleared.

➢ **To view the list of history alarms:**

- Open the Alarms History page (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Alarms History**).

Figure 36-2: Alarms History Page

Sequential number	Severity	Source	Description	Date
1	Major	Board#1	Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy	6.1.2010 , 14:1:26
2	cleared	Board#1	Alarm cleared: Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy	6.1.2010 , 14:1:26
3	Major	Board#1	Controller failure alarm Proxy Set ID 0	6.1.2010 , 14:1:26
4	Major	Board#1/WanLink#1	WAN link alarm. FE interface 1 is down.	6.1.2010 , 14:1:29
5	Minor	Board#1/EthernetLink#2	Ethernet link alarm. LAN port number 2 is down.	6.1.2010 , 14:1:29
6	Major	Board#1	NTP server alarm. No connection to NTP server.	6.1.2010 , 14:11:14

For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical - alarm displayed in red
  - Major - alarm displayed in orange
  - Minor - alarm displayed in yellow
  - Cleared - alarm displayed in green
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

You can view the next 20 alarms (if exist), by clicking the **Go to page** button.

37 Performance Monitoring

This section describes how to view the following performance monitoring graphs:

- Trunk Utilization - see 'Viewing Trunk Utilization' on page 713
- MOS per Media Realm - see 'Viewing MOS per Media Realm' on page 715
- Quality of Experience - see Viewing Quality of Experience on page 716
- Average Call Duration - see Viewing Average Call Duration on page 718

37.1 Viewing Trunk Utilization

The Trunk Utilization page provides an X-Y graph that displays the number of active channels per trunk over time. The x-axis indicates the time; the y-axis indicates the number of active trunk channels.

Notes:

- This page is available only if you have trunks and the SBC application is disabled.
- If you navigate to a different page, the data displayed in the graph and all its settings are cleared.

➢ **To view the number of active trunk channels**

1. Open the Trunk Utilization page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **Trunk Utilization**).

Figure 37-1: Trunk Utilization Page

2. From the 'Trunk' drop-down list, select the trunk for which you want to view active channels.

For more graph functionality, see the following table:

Table 37-1: Additional Graph Functionality for Trunk Utilization

Button	Description
Add button	Displays additional trunks in the graph. Up to five trunks can be displayed simultaneously in the graph. To view another trunk, click this button and then from the new 'Trunk' drop-down list, select the required trunk. Each trunk is displayed in a different color, according to the legend shown in the top-left corner of the graph.
Remove button	Removes the selected trunk display from the graph.
Disable check box	Hides or shows an already selected trunk. Select this check box to temporarily hide the trunk display; clear this check box to show the trunk. This is useful if you do not want to remove the trunk entirely (using the **Remove** button).
Get Most Active button	Displays only the trunk with the most active channels (i.e., trunk with the most calls).
Pause button	Pauses the display in the graph.
Play button	Resumes the display in the graph.
Zoom slide ruler and buttons	Increases or reduces the trunk utilization display resolution concerning time. The **Zoom In** button increases the time resolution; the **Zoom Out** button decreases it. Instead of using the buttons, you can use the slide ruler. As you increase the resolution, more data is displayed on the graph. The minimum resolution is about 30 seconds; the maximum resolution is about an hour.

37.2 Viewing MOS per Media Realm

The MOS Per Media Realm page displays statistics on Media Realms (configured in 'Configuring Media Realms' on page 181). This page provides two graphs:

- Upper graph: displays the Mean Opinion Score (MOS) quality in RTCP data per selected Media Realm.
- Lower graph: displays the bandwidth of transmitted media (in Kbps) in RTCP data per Media Realm.

➢ **To view the MOS per Media Realm graph:**

1. Open the MOS Per Media Realm page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **MOS Per Media Realm**).

Figure 37-2: MOS Per Media Realm Graph

2. From the 'Media Realm' drop-down list, select the Media Realm for which you want to view.

Use the **Zoom In** button to increase the displayed time resolution or the **Zoom Out** button to decrease it. Instead of using these zoom buttons, you can use the slide ruler. As you increase the resolution, more data is displayed on the graph. The minimum resolution is about 30 seconds; the maximum resolution is about an hour.

To pause the graph, click the **Pause** button; click **Play** to resume.

37.3 Viewing Quality of Experience

The Quality Of Experience page provides statistical information on calls per SRD or IP Group. The statistics can be further filtered to display incoming and/or outgoing call direction, and type of SIP dialog (INVITE, SUBSCRIBE, or all).

Note: This page is available only if the SBC application has been enabled.

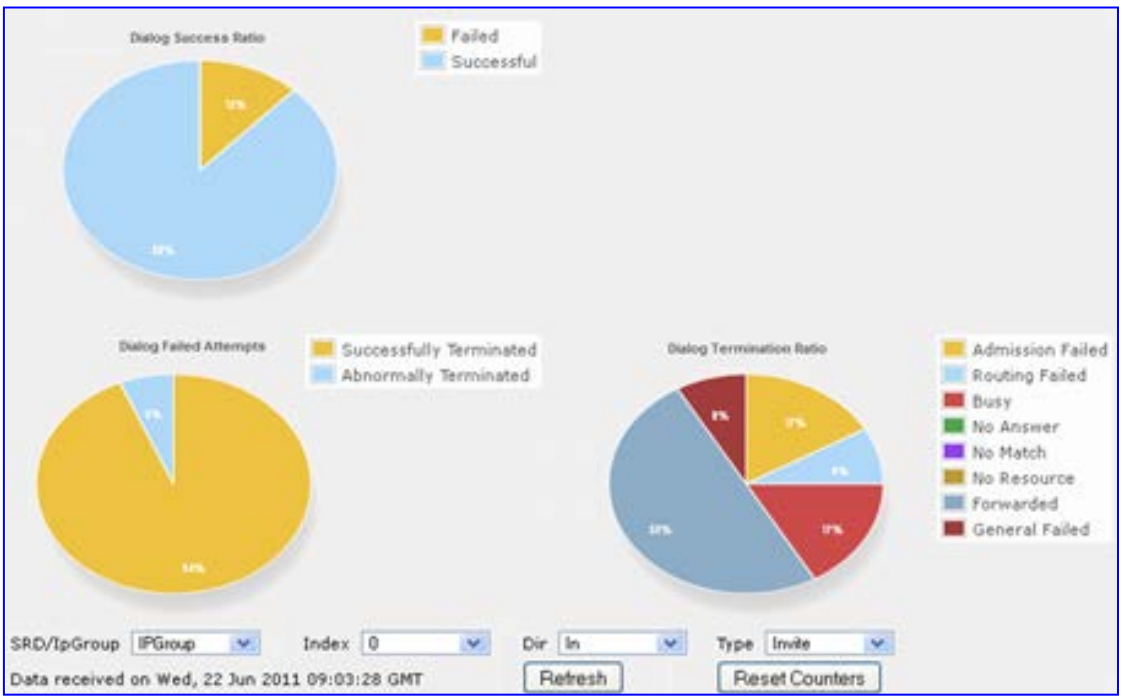
This page provides three pie charts:

- Dialog Success Ratio: displays the SIP call and subscribe (SUBSCRIBE) dialog success-failed ratio.
- Dialog Failed Attempts: displays the failed call attempts. This includes the number of calls and subscribes which were successfully and abnormally terminated.
- Dialog Termination Ratio: displays call termination by reason (e.g., due to no answer).

➢ **To view Quality of Experience:**

1. Open the Quality Of Experience page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **Quality Of Experience**).

Figure 37-3: Quality Of Experience Graph



2. From the 'SRD/IpGroup' drop-down list, select whether you want to view QoE for an SRD or IP Group.
3. From the 'Index' drop-down list, select the SRD or IP Group index.

4. From the 'Dir' drop-down list, select the call direction:
   - **In** - incoming calls
   - **Out** - outgoing calls
   - **Both** - incoming and outgoing calls
5. From the 'Type' drop-down list, select the SIP message type:
   - **Invite** - INVITE
   - **Subscribe** - SUBSCRIBE
   - **Other** - all SIP messages

To refresh the charts, click **Refresh**. To reset the counters, click **Reset Counters**.

37.4 Viewing Average Call Duration

The Average Call Duration page displays information about a specific SRD or IP Group. This page includes two graphs:

- Upper graph: displays the number of calls (INVITEs).
- Lower graph: displays the average call duration.

Note: This page is available only if the SBC application has been enabled.

➢ **To view average call duration:**

1. Open the Average Call Duration page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **Average Call Duration**).

Figure 37-4: Average Call Duration Graph

2. From the 'SRD/IpGroup' drop-down list, select whether you want to view information for an SRD or IP Group.
3. From the 'Index' drop-down list, select the SRD or IP Group index.

Use the **Zoom In** button to increase the displayed time resolution or the **Zoom Out** button to decrease it. Instead of using these zoom buttons, you can use the slide ruler. As you increase the resolution, more data is displayed on the graph. The minimum resolution is about 30 seconds; the maximum resolution is about an hour.

To pause the graph, click the **Pause** button; click **Play** to resume.

38 VoIP Status

This section describes how to view the following VoIP status and statistics:

- IP network interface - see 'Viewing Active IP Interfaces' on page 719
- Performance - see 'Viewing Performance Statistics' on page 719
- IP-to-Tel calls - see 'Viewing Call Counters' on page 720
- Tel-to-IP calls - see 'Viewing Call Counters' on page 720
- SAS/SBC registered users - see Viewing SAS/SBC Registered Users on page 722
- Call routing - see 'Viewing Call Routing Status' on page 723
- Registration - see Viewing Registration Status on page 724
- IP connectivity - see 'Viewing IP Connectivity' on page 725

38.1 Viewing Active IP Interfaces

The IP Interface Status page displays the device's active IP interfaces, which are configured in the 'Multiple Interface Table page (see 'Configuring IP Interface Settings' on page 117).

➢ **To view the 'Active IP Interfaces page:**

- Open the IP Interface Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **IP Interface Status**).

Figure 38-1: IP Interface Status Page

Index	Application Type	Address Type	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	O+M+C	IPv4	IPv4 Manual	10.13.4.12	16	10.13.0.1	1	Voice
NA	Internal	IPv4	IPv4 Manual	169.254.254.254	30	169.254.254.253	4001	InternalIF

38.2 Viewing Performance Statistics

The Basic Statistics page provides read-only, device performance statistics. This page is refreshed every 60 seconds. The duration that the currently displayed statistics has been collected is displayed above the statistics table.

➢ **To view performance statistics:**

- Open the Basic Statistics page (**Status & Diagnostics** tab > **VoIP Status** menu > **Performance Statistics**).

Figure 38-2: Basic Statistics Page

(Statistics for 251040 seconds)	
Active TDM channels	0
Active DSP resources	0
Active analog channels	0
Active G.711 channels	0
Average voice delay (ms)	0
Average voice jitter (ms)	0
Total RTP packets TX	0
Total RTP packets RX	0
Total call attempts	0

To reset the performance statistics to zero, click the **Reset Statistics** button.

38.3 Viewing Call Counters

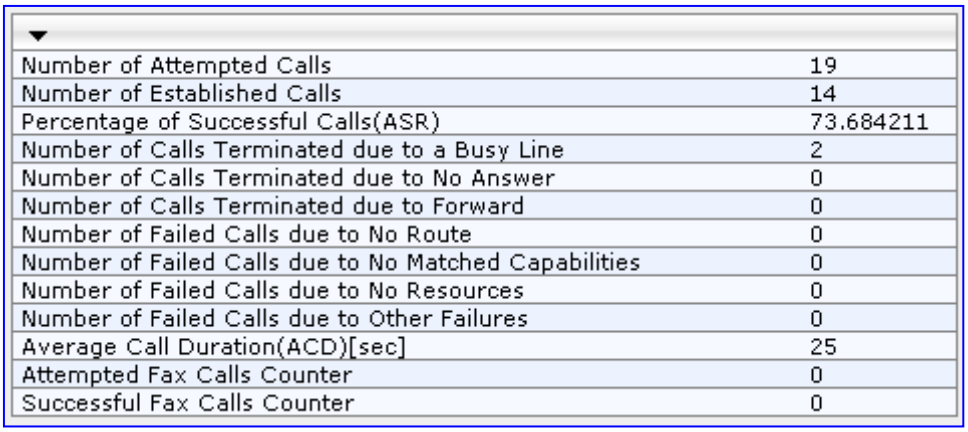
The IP to Tel Calls Count page and Tel to IP Calls Count page provide you with statistical information on incoming (IP-to-Tel) and outgoing (Tel-to-IP) calls. The statistical information is updated according to the release reason that is received after a call is terminated (during the same time as the end-of-call Call Detail Record or CDR message is sent). The release reason can be viewed in the 'Termination Reason' field in the CDR message.

You can reset the statistical data displayed on the page (i.e., refresh the display), by clicking the **Reset Counters** button located on the page.

➢ **To view the IP-to-Tel and Tel-to-IP Call Counters pages:**

- Open the Call Counters page that you want to view (**Status & Diagnostics** tab > **VoIP Status** menu > **IP to Tel Calls Count** or **Tel to IP Calls Count**); the figure below shows the IP to Tel Calls Count page.

Figure 38-3: Calls Count Page



Number of Attempted Calls	19
Number of Established Calls	14
Percentage of Successful Calls(ASR)	73.684211
Number of Calls Terminated due to a Busy Line	2
Number of Calls Terminated due to No Answer	0
Number of Calls Terminated due to Forward	0
Number of Failed Calls due to No Route	0
Number of Failed Calls due to No Matched Capabilities	0
Number of Failed Calls due to No Resources	0
Number of Failed Calls due to Other Failures	0
Average Call Duration(ACD)[sec]	25
Attempted Fax Calls Counter	0
Successful Fax Calls Counter	0

The fields in this page are described in the following table:

Table 38-1: Call Counters Description

Counter	Description
Number of Attempted Calls	Indicates the number of attempted calls. It is composed of established and failed calls. The number of established calls is represented by the 'Number of Established Calls' counter. The number of failed calls is represented by the failed-call counters. Only one of the established / failed call counters is incremented every time.
Number of Established Calls	Indicates the number of established calls. It is incremented as a result of one of the following release reasons if the duration of the call is greater than zero: ▪ GWAPP_REASON_NOT_RELEVANT (0) ▪ GWAPP_NORMAL_CALL_CLEAR (16) ▪ GWAPP_NORMAL_UNSPECIFIED (31) And the internal reasons: ▪ RELEASE_BECAUSE_UNKNOWN_REASON ▪ RELEASE_BECAUSE_REMOTE_CANCEL_CALL ▪ RELEASE_BECAUSE_MANUAL_DISC ▪ RELEASE_BECAUSE_SILENCE_DISC ▪ RELEASE_BECAUSE_DISCONNECT_CODE

Counter	Description
	Note: When the duration of the call is zero, the release reason GWAPP_NORMAL_CALL_CLEAR increments the 'Number of Failed Calls due to No Answer' counter. The rest of the release reasons increment the 'Number of Failed Calls due to Other Failures' counter.
Percentage of Successful Calls (ASR)	The percentage of established calls from attempted calls.
Number of Calls Terminated due to a Busy Line	Indicates the number of calls that failed as a result of a busy line. It is incremented as a result of the following release reason: GWAPP_USER_BUSY (17)
Number of Calls Terminated due to No Answer	Indicates the number of calls that weren't answered. It's incremented as a result of one of the following release reasons: ▪ GWAPP_NO_USER_RESPONDING (18) ▪ GWAPP_NO_ANSWER_FROM_USER_ALERTED (19) ▪ GWAPP_NORMAL_CALL_CLEAR (16) (when the call duration is zero)
Number of Calls Terminated due to Forward	Indicates the number of calls that were terminated due to a call forward. The counter is incremented as a result of the following release reason: RELEASE_BECAUSE_FORWARD
Number of Failed Calls due to No Route	Indicates the number of calls whose destinations weren't found. It is incremented as a result of one of the following release reasons: ▪ GWAPP_UNASSIGNED_NUMBER (1) ▪ GWAPP_NO_ROUTE_TO_DESTINATION (3)
Number of Failed Calls due to No Matched Capabilities	Indicates the number of calls that failed due to mismatched device capabilities. It is incremented as a result of an internal identification of capability mismatch. This mismatch is reflected to CDR via the value of the parameter DefaultReleaseReason (default is GWAPP_NO_ROUTE_TO_DESTINATION (3)) or by the GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED (79) reason.
Number of Failed Calls due to No Resources	Indicates the number of calls that failed due to unavailable resources or a device lock. The counter is incremented as a result of one of the following release reasons: ▪ GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED ▪ RELEASE_BECAUSE_GW_LOCKED
Number of Failed Calls due to Other Failures	This counter is incremented as a result of calls that failed due to reasons not covered by the other counters.
Average Call Duration (ACD) [sec]	The average call duration (ACD) in seconds of established calls. The ACD value is refreshed every 15 minutes and therefore, this value reflects the average duration of all established calls made within a 15 minute period.
Attempted Fax Calls Counter	Indicates the number of attempted fax calls.
Successful Fax Calls Counter	Indicates the number of successful fax calls.

38.4 Viewing SAS/SBC Registered Users

The SAS/SBC Registered Users page displays a list of registered SAS/SBC users recorded in the device's database.

➢ **To view registered users:**

- Open the SAS/SBC Registered Users page (**Status & Diagnostics** tab > **VoIP Status** menu > **SAS/SBC Registered Users**).

Figure 38-4: SAS/SBC Registered Users Page

Address Of Record	Contact
1000@10.8.5.71	<sip:1000@10.8.5.71:5060>;expires=180; Active status: 1
1001@10.8.5.71	<sip:1001@10.8.5.71:5060>;expires=180; Active status: 1
1100@10.8.5.71	<sip:1100@10.8.5.71:5060>;expires=180; Active status: 1
1101@10.8.5.71	<sip:1101@10.8.5.71:5060>;expires=180; Active status: 1
2000@10.8.5.72	<sip:2000@10.8.5.72:5060>;expires=180; Active status: 1

Table 38-2: SAS/SBC Registered Users Parameters

Column Name	Description
Address of Record	An address-of-record (AOR) is a SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI (Contact) where the user might be available.
Contact	SIP URI that can be used to contact that specific instance of the User Agent for subsequent requests.

38.5 Viewing Call Routing Status

The Call Routing Status page provides you with information on the current routing method used by the device. This information includes the IP address and FQDN (if used) of the Proxy server with which the device currently operates.

➢ **To view the call routing status:**

- Open the Call Routing Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Call Routing Status**).

Figure 38-5: Call Routing Status Page

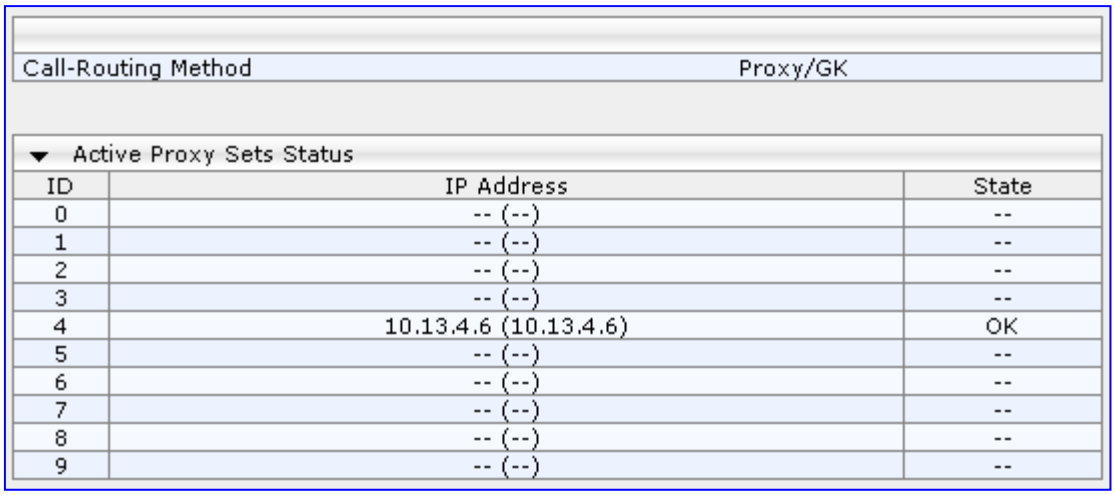


Table 38-3: Call Routing Status Parameters

Parameter	Description
Call-Routing Method	▪ Proxy/GK = Proxy server is used to route calls. ▪ Routing Table = The Outbound IP Routing Table is used to route calls.
IP Address	▪ Not Used = Proxy server isn't defined. ▪ IP address and FQDN (if exists) of the Proxy server with which the device currently operates.
State	▪ N/A = Proxy server isn't defined. ▪ OK = Communication with the Proxy server is in order. ▪ Fail = No response from any of the defined Proxies.

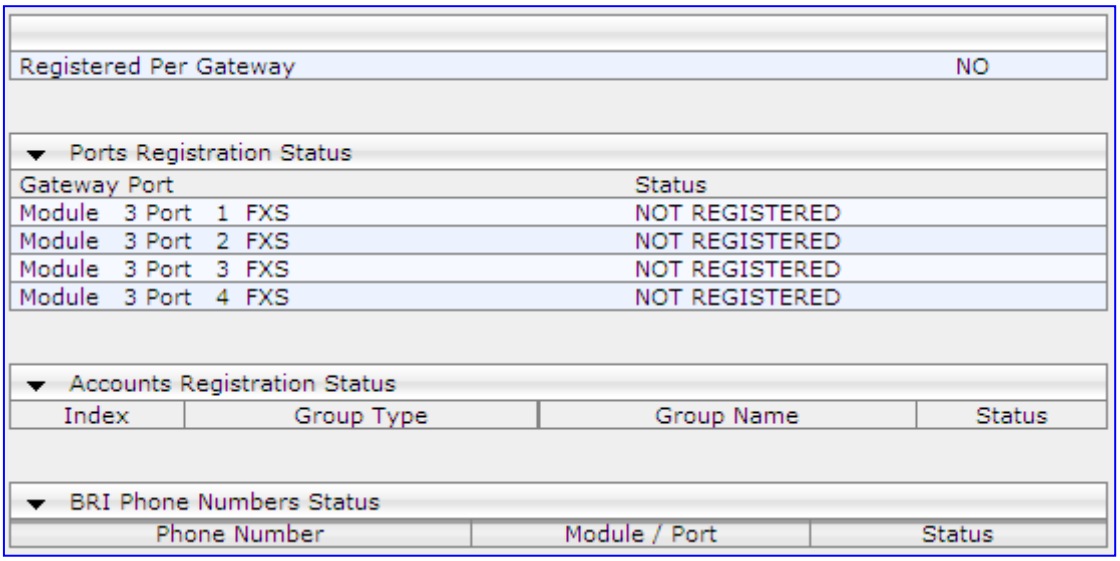
38.6 Viewing Registration Status

The Registration Status page displays whether the device, its endpoints, SIP Accounts, and BRI endpoints are registered to a SIP Registrar/Proxy server.

➢ **To view Registration status:**

- Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registration Status**).

Figure 38-6: Registration Status Page



- **Registered Per Gateway:**
  - "YES" = registration is per device
  - "NO"= registration is not per device
- **Ports Registration Status:**
  - "REGISTERED" = channel is registered
  - "NOT REGISTERED" = channel not registered
- **Accounts Registration Status:** registration status based on the Accounts table (configured in 'Configuring Account Table' on page 237):
  - **Group Type:** type of served group - Trunk Group or IP Group
  - **Group Name:** name of the served group, if applicable
  - **Status:** indicates whether or not the group is registered ("Registered" or "Unregistered")
- BRI Phone Number Status:
  - **Phone Number:** phone number of BRI endpoint
  - **Module/Port:** module/port number of BRI endpoint
  - **Status:** indicates whether or not the BRI endpoint is registered ("Registered" or "Unregistered")

Note: The registration mode (i.e., per device, endpoint, account. or no registration) is configured in the Trunk Group Settings table (see 'Configuring Trunk Group Settings' on page 273) or using the TrunkGroupSettings *ini* file parameter.

38.7 Viewing IP Connectivity

The IP Connectivity page displays online, read-only network diagnostic connectivity information on all destination IP addresses configured in the Outbound IP Routing Table page (see 'Configuring Outbound IP Routing Table' on page 291).

Notes:

- This information is available only if the parameter 'Enable Alt Routing Tel to IP'/AltRoutingTel2IPMode (see 'Configuring General Routing Parameters' on page 291) is set to 1 (Enable) or 2 (Status Only).
- The information in columns 'Quality Status' and 'Quality Info' (per IP address) is reset if two minutes elapse without a call to that destination.

➢ **To view IP connectivity information:**

1. In the Routing General Parameters page, set the 'Enable Alt Routing Tel to IP' parameter (AltRoutingTel2IPEnable) to **Enable** or **Status Only**.
2. Open the IP Connectivity page (**Status & Diagnostics** tab > **VoIP Status** menu > **IP Connectivity**).

Figure 38-7: IP Connectivity Page

	IP Address	Host Name	Connectivity Method	Connectivity Status	Quality Status	Quality Info	DNS Status
1	Unused	---	---	---	---	---	---
2	Unused	---	---	---	---	---	---
3	Unused	---	---	---	---	---	---
4	Unused	---	---	---	---	---	---
5	Unused	---	---	---	---	---	---
6	Unused	---	---	---	---	---	---
7	Unused	---	---	---	---	---	---
8	Unused	---	---	---	---	---	---
9	Unused	---	---	---	---	---	---
10	Unused	---	---	---	---	---	---

Table 38-4: IP Connectivity Parameters

Column Name	Description
IP Address	The IP address can be one of the following: ▪ IP address defined as the destination IP address in the Outbound IP Routing Table'. ▪ IP address resolved from the host name defined as the destination IP address in the Outbound IP Routing Table'.
Host Name	Host name (or IP address) as defined in the Outbound IP Routing Table'.
Connectivity Method	The method according to which the destination IP address is queried periodically (ICMP ping or SIP OPTIONS request).
Connectivity Status	The status of the IP address' connectivity according to the method in the 'Connectivity Method' field. ▪ OK = Remote side responds to periodic connectivity queries. ▪ Lost = Remote side didn't respond for a short period. ▪ Fail = Remote side doesn't respond.

Column Name	Description
	▪ Init = Connectivity queries not started (e.g., IP address not resolved). ▪ Disable = The connectivity option is disabled, i.e., parameter 'Alt Routing Tel to IP Mode' (AltRoutingTel2IPMode *ini*) is set to 'None' or 'QoS'.
Quality Status	Determines the QoS (according to packet loss and delay) of the IP address. ▪ Unknown = Recent quality information isn't available. ▪ OK ▪ Poor **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
Quality Info.	Displays QoS information: delay and packet loss, calculated according to previous calls. **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
DNS Status	DNS status can be one of the following: ▪ DNS Disable ▪ DNS Resolved ▪ DNS Unresolved

39 Data Status

This section describes how to view the following data-touter status and statistics:

- WAN - see 'Viewing WAN Status' on page 727
- Connection - see 'Viewing Network Connection Statistics' on page 727
- Security log - see 'Viewing Logged Security Events' on page 728
- QoS queues - see 'Viewing QoS Queues Statistics' on page 730
- Data log - see 'Viewing Logged Data Events' on page 731

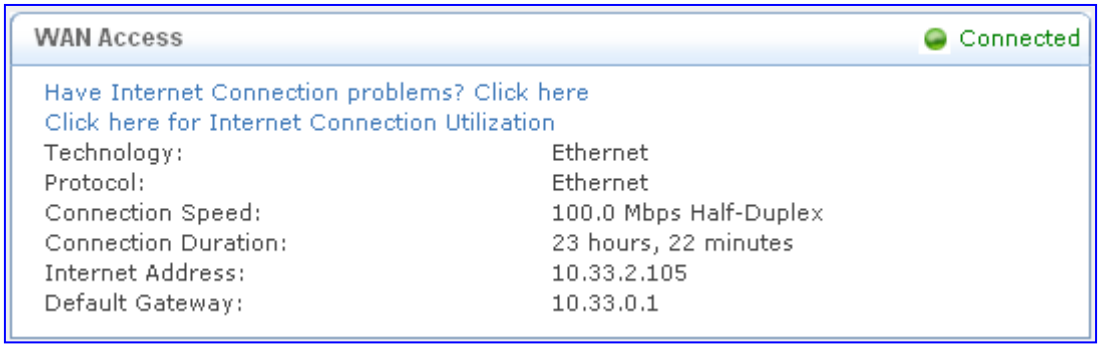
39.1 Viewing WAN Status

The **WAN Status** item allows you to view the WAN access status and provides a series of diagnostic tests to validate your device's Internet connection. For the diagnostic tests, see 'Running Internet Connectivity Diagnostic Tests' on page 727.

➢ **To view the status of the WAN connection:**

- Click the **WAN Status** item (**Status & Diagnostics** tab > **Data Status** menu > **WAN Status**); the following page appears:

Figure 39-1: WAN Status



The status of the WAN interface is depicted by the ball-shaped icon located in the top-right corner:

- Green - "Connected": indicates a valid connection to the WAN network.
- Red - "Cable Disconnected": Green: indicates that there is no connection to the WAN network (cable disconnected).

39.2 Viewing Network Connection Statistics

The Network Connections page displays a table summarizing the monitored connection data. The device constantly monitors traffic within the local network and between the local network and the Internet. You can view statistical information about data received from and transmitted to the Internet (WAN) and to computers in the local network (LAN).

➢ **To view data on network connections:**

- Click the **Connection Statistics** item (**Status & Diagnostics** tab > **Data Status** menu > **Connection Statistics**); the following page appears:

Figure 39-2: Connection Statistics Page

Name	LAN switch	WAN Ethernet	LAN switch VLAN 4001	LAN switch VLAN 1
Device Name	eth0	eth1	eth0.4001	eth0.1
Status	1 Ports Connected	Cable Disconnected	Connected	Connected
Network	LAN	WAN	LAN	LAN
Underlying Device/s			LAN switch	LAN switch
Connection Type	Hardware Ethernet Switch	Ethernet	Ethernet	Ethernet
Download Rate	100 Mbps		100 Mbps	100 Mbps
Upload Rate	100 Mbps		100 Mbps	100 Mbps
MAC Address	00:90:8f:3a:ad:57	00:90:8f:3a:ad:58	00:90:8f:3a:ad:57	00:90:8f:3a:ad:57
IP Address			169.254.254.253	192.168.0.1
Subnet Mask			255.255.255.252	255.255.255.0
IP Address Distribution	Disabled	Disabled	Disabled	DHCP Server
Received Packets	714299		286963	384551
Sent Packets	292106		291720	237
Received Bytes	67260084		19684130	32315822
Sent Bytes	29085632		29027537	45129
Receive Errors	0		0	0
Receive Drops	0		0	0
Time Span	22:08:37		22:08:20	22:08:19

To update the display, click the **Refresh** button, or click the **Automatic Refresh On** button to constantly update the displayed parameters.

39.3 Viewing Logged Security Events

The **Security Log** item displays a list of firewall-related events, including attempts to establish inbound and outbound connections, attempts to authenticate through an administrative interface (Web or Telnet terminal), firewall configuration and system start-up.

➢ **To view logs of firewall-related events:**

1. Click the **Security Log** item (**Status & Diagnostics** tab > **Data Status** menu > **Security Log**); the following page appears:

Figure 39-3: Firewall - Log Page

Time	Event	Event-Type	Details
Jan 1 17:25:25 2003	Firewall Setup	Firewall internal	Firewall configuration succeeded
Jan 1 17:25:25 2003	Firewall Setup	Firewall internal	Starting firewall configuration
Jan 1 17:25:25 2003	Firewall Setup	Firewall internal	Firewall configuration succeeded
Jan 1 17:25:25 2003	Firewall Setup	Firewall internal	Starting firewall configuration
Jan 1 00:00:07 2003	Firewall Setup	Firewall internal	Firewall configuration succeeded

The log table displays the following details:

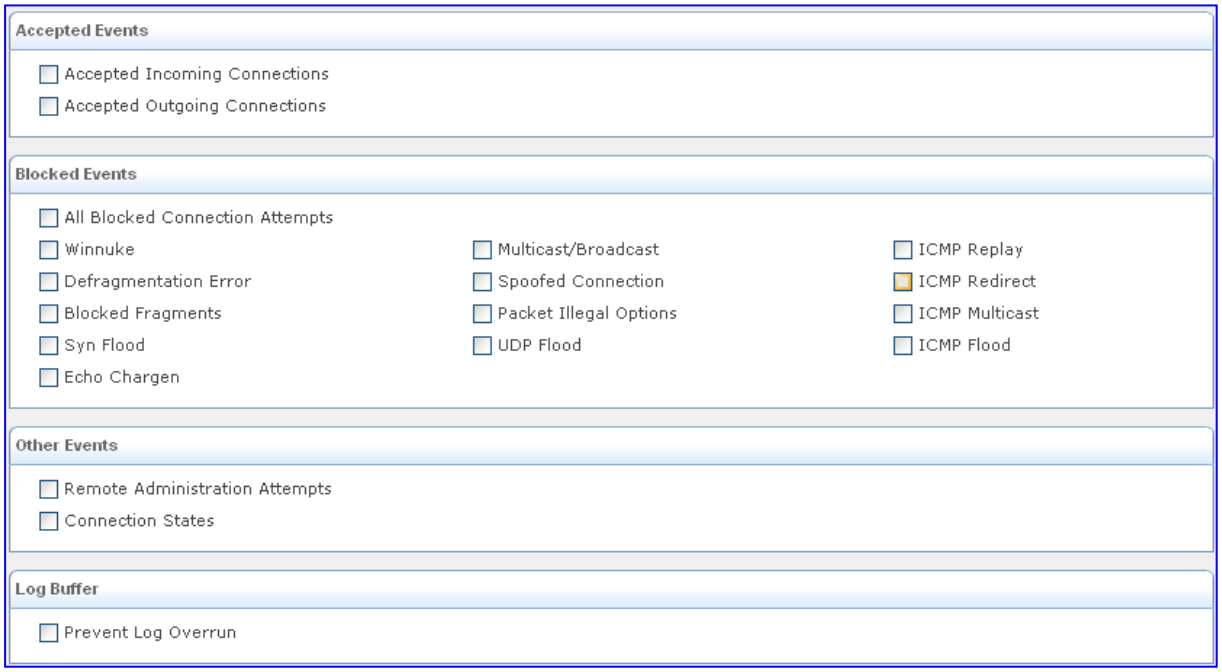
- **Time:** time the event occurred.

- **Event:** there are five kinds of events:
  - **Inbound Traffic:** event is a result of an incoming packet.
  - **Outbound Traffic:** event is a result of outgoing packet.
  - **Firewall Setup:** configuration message
  - **WBM Login:** indicates that a user has logged in to the Web interface.
  - **CLI Login:** indicates that a user has logged in to CLI (via Telnet).
- **Event-Type:** textual description of the event:
  - **Blocked:** packet was blocked (message is colored red).
  - **Accepted:** packet was accepted (message is colored green).
- **Details:** additional details about the packet or the event such as protocol, IP addresses, ports, etc.

The page also provides you with the following buttons:

- **Clear Log**: clears currently displayed log messages from the table.
- **Refresh:** updates the log display with the latest log messages.
- **Settings:** allows you to select the types of activities for which you want to have a log message generated, as shown below:

Figure 39-4: Log Settings Page



- Accepted Events group:
  - **Accepted Incoming Connections:** generates a log message for each successful attempt to establish an inbound connection to the home network.
  - **Accepted Outgoing Connections:** generates a log message for each successful attempt to establish an outgoing connection to the public network.
- Blocked Events group:
  - **All Blocked Connection Attempts:** generates a log message for each blocked attempt to establish an inbound connection to the home network or vice versa. You can enable logging of blocked packets of specific types by disabling this option, and enabling some of the more specific options listed below it.
  - Generate a log message for specific events that are blocked such as SynFlood. A log message is generated if either the corresponding check box

is checked, or the 'All Blocked Connection Attempts' check box is selected.

- Other Events group:
  - **Remote Administration Attempts:** generates a log message for each remote administration connection attempt, whether successful or not.
  - **Connection States:** provide additional information about every change in a connection opened by the firewall. Use this option to track connection handling by the firewall and Application Level Gateways (ALGs).
- Log Buffer group:
  - **Prevent Log Overrun:** stops logging firewall activities when the memory allocated for the log fills up.

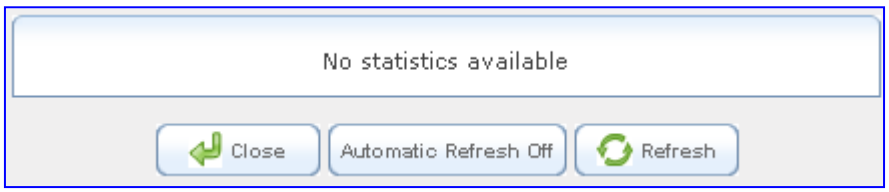
39.4 Viewing QoS Queues Statistics

You can view an accurate, real-time information on the traffic moving through your defined device classes. For example, the amount of packets sent, dropped or delayed are just a few of the parameters that you can monitor per shaping class.

➢ **To view your class statistics:**

- Click the **QoS Class Statistics** item (**Status & Diagnostics** tab > **Data Status** menu > **QoS Queues Statistics**); the following page appears:

Figure 39-5: QoS Queues Statistics Page



Note: Class statistics are only available after defining at least one class; otherwise, the page does not display any information.

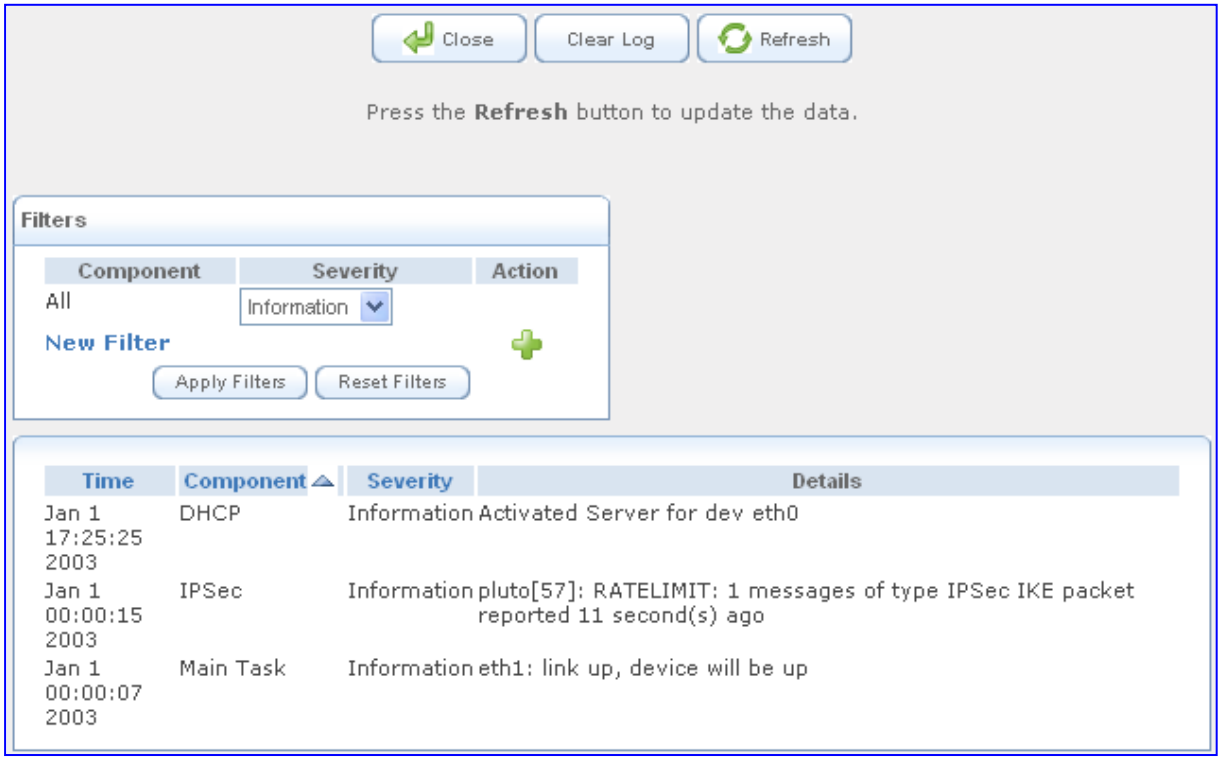
39.5 Viewing Logged Data Events

The **Data Log** item displays a list of recent events occurred on the device.

➢ **To view logged messages:**

- Click the **Data Log** item (**Status & Diagnostics** tab > **Data Status** menu > **Data Log**); the following page appears:

Figure 39-6: System Log Page

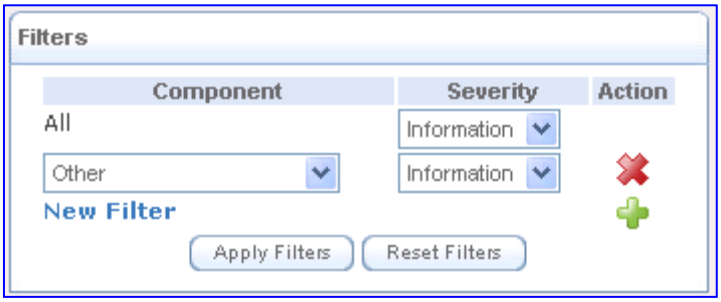


By default, all log messages are displayed one after another, sorted by their order of posting by the device (latest on top). You can sort the messages according to the column titles 'Time', 'Component', or 'Severity', by clicking the column header. You can also filter the log messages by the component that generated them or by their severity, providing a more refined list. By default, the page displays log messages with 'debug' severity level and higher, for all components. You may change the severity level for this filter.

➢ **To add a new log display filter:**

1. In the 'Filters' group, click the **New Filter** link; the 'Filters' group displays a new Component entry.

Figure 39-7: Adding a New Filter



2. Using the drop-down lists, select the component and severity level by which to sort the log messages.
3. Click **Apply Filters** to display the messages in your specified criteria.

You can also delete filters using their respective action icons. Clicking **Reset Filters** deletes all the defined filters. Defined filters override the default filter that displays all messages.

You can use the buttons located at the top of the page to perform the following:

- **Close:** closes the 'Log page and returns to the device's Home page.
- **Clear Log:** clears all currently displayed log messages.
- **Refresh:** refreshes the page to display the latest log messages.

40 Reporting Information to External Party

40.1 Generating Call Detail Records

The Call Detail Record (CDR) contains vital statistic information on calls made from the device. CDRs are generated at the end and optionally, at the beginning of each call (defined by the CDRReportLevel parameter). Once generated, they are sent to a Syslog server. The destination IP address for CDR logs is defined by the CDRSyslogServerIP parameter. For CDR in RADIUS format, see 'Supported RADIUS Attributes' on page 739.

40.1.1 CDR Fields for Gateway Application

The CDR fields for the Gateway (and IP-to-IP) applications are listed in the table below.

Table 40-1: CDR Fields for Gateway/IP2IP Application

Field Name	Description
ReportType	Report type (call started, call connected, or call released)
Cid	Port number
SessionId	SIP session identifier
Trunk	Physical trunk number
BChan	Selected B-channel
ConId	SIP conference ID
TG	Trunk Group ID
EPTyp	Endpoint type (FXS or FXO)
Orig	Call originator ("LCL" for Tel side; "RMT" for IP side)
SourceIp	Source IP address
DestIp	Destination IP address
TON	Source phone number type
NPI	Source phone number plan
SrcPhoneNum	Source phone number
SrcNumBeforeMap	Source number before manipulation
TON	Destination phone number type
NPI	Destination phone number plan
DstPhoneNum	Destination phone number
DstNumBeforeMap	Destination number before manipulation
Durat	Call duration
Coder	Selected coder
Intrv	Packet interval
RtpIp	RTP IP address
Port	Remote RTP port

Field Name	Description
TrmSd	Initiator of call release (IP, Tel, or Unknown)
TrmReason	Termination reason (see 'Release Reasons in CDR' on page 735)
Fax	Fax transaction during call
InPackets	Number of incoming packets
OutPackets	Number of outgoing packets
PackLoss	Local packet loss
RemotePackLoss	Number of outgoing lost packets
SIPCallId	Unique SIP call ID
SetupTime	Call setup time
ConnectTime	Call connect time
ReleaseTime	Call release time
RTPdelay	RTP delay
RTPjitter	RTP jitter
RTPssrc	Local RTP SSRC
RemoteRTPssrc	Remote RTP SSRC
RedirectReason	Redirect reason
TON	Redirection phone number type
NPI	Redirection phone number plan
RedirectPhonNum	Redirection phone number
MeteringPulses	Number of generated metering pulses
SrcHost	Source host name
SrcHostBeforeMap	Source host name before manipulation
DstHost	Destination host name
DstHostBeforeMap	Destination host name before manipulation
IPG	IP Group description
LocalRtpIp	Remote RTP IP address
LocalRtpPort	Local RTP port
TrmReasonCategory	Termination reason category
RedirectNumBeforeMap	Redirect number before manipulation
SrdId	SRD ID
SIPInterfaceId	SIP interface ID
TransportType	SIP transport type (UDP, TCP, or TLS)
TxRTPIPDiffServ	Media IP DiffServ
TxSigIPDiffServ	Signaling IP DiffServ
LocalRFactor	Local R-factor

Field Name	Description
RemoteRFactor	Remote R-factor
LocalMosCQ	Local MOS for conversation quality
RemoteMosCQ	Remote MOS for conversation quality
SourcePort	Source RTP port
DestPort	Destination RTP port

40.1.2 Release Reasons in CDR

The possible reasons for call termination which is represented in the CDR field **TrmReason** are listed below:

- "REASON N/A"
- "RELEASE_BECAUSE_NORMAL_CALL_DROP"
- "RELEASE_BECAUSE_DESTINATION_UNREACHABLE"
- "RELEASE_BECAUSE_DESTINATION_BUSY"
- "RELEASE_BECAUSE_NOANSWER"
- "RELEASE_BECAUSE_UNKNOWN_REASON"
- "RELEASE_BECAUSE_REMOTE_CANCEL_CALL"
- "RELEASE_BECAUSE_UNMATCHED_CAPABILITIES"
- "RELEASE_BECAUSE_UNMATCHED_CREDENTIALS"
- "RELEASE_BECAUSE_UNABLE_TO_HANDLE_REMOTE_REQUEST"
- "RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT"
- "RELEASE_BECAUSE_CONFERENCE_FULL"
- "RELEASE_BECAUSE_VOICE_PROMPT_PLAY_ENDED"
- "RELEASE_BECAUSE_VOICE_PROMPT_NOT_FOUND"
- "RELEASE_BECAUSE_TRUNK_DISCONNECTED"
- "RELEASE_BECAUSE_RSRC_PROBLEM"
- "RELEASE_BECAUSE_MANUAL_DISC"
- "RELEASE_BECAUSE_SILENCE_DISC"
- "RELEASE_BECAUSE_RTP_CONN_BROKEN"
- "RELEASE_BECAUSE_DISCONNECT_CODE"
- "RELEASE_BECAUSE_GW_LOCKED"
- "RELEASE_BECAUSE_NORTEL_XFER_SUCCESS"
- "RELEASE_BECAUSE_FAIL"
- "RELEASE_BECAUSE_FORWARD"
- "RELEASE_BECAUSE_ANONYMOUS_SOURCE"
- "RELEASE_BECAUSE_IP_PROFILE_CALL_LIMIT"
- "GWAPP_UNASSIGNED_NUMBER"
- "GWAPP_NO_ROUTE_TO_TRANSIT_NET"
- "GWAPP_NO_ROUTE_TO_DESTINATION"
- "GWAPP_CHANNEL_UNACCEPTABLE"
- "GWAPP_CALL_AWARDED_AND "

- "GWAPP_PREEMPTION"
- "PREEMPTION_CIRCUIT_RESERVED_FOR_REUSE"
- "GWAPP_NORMAL_CALL_CLEAR"
- "GWAPP_USER_BUSY"
- "GWAPP_NO_USER_RESPONDING"
- "GWAPP_NO_ANSWER_FROM_USER_ALERTED"
- "MFCR2_ACCEPT_CALL"
- "GWAPP_CALL_REJECTED"
- "GWAPP_NUMBER_CHANGED"
- "GWAPP_NON_SELECTED_USER_CLEARING"
- "GWAPP_INVALID_NUMBER_FORMAT"
- "GWAPP_FACILITY_REJECT"
- "GWAPP_RESPONSE_TO_STATUS_ENQUIRY"
- "GWAPP_NORMAL_UNSPECIFIED"
- "GWAPP_CIRCUIT_CONGESTION"
- "GWAPP_USER_CONGESTION"
- "GWAPP_NO_CIRCUIT_AVAILABLE"
- "GWAPP_NETWORK_OUT_OF_ORDER"
- "GWAPP_NETWORK_TEMPORARY_FAILURE"
- "GWAPP_NETWORK_CONGESTION"
- "GWAPP_ACCESS_INFORMATION_DISCARDED"
- "GWAPP_REQUESTED_CIRCUIT_NOT_AVAILABLE"
- "GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED"
- "GWAPP_PERM_FR_MODE_CONN_OUT_OF_S"
- "GWAPP_PERM_FR_MODE_CONN_OPERATIONAL"
- "GWAPP_PRECEDENCE_CALL_BLOCKED"
  - "RELEASE_BECAUSE_PREEMPTION_ANALOG_CIRCUIT_RESERVED_FOR_REUSE"
  - "RELEASE_BECAUSE_PRECEDENCE_CALL_BLOCKED"
- "GWAPP_QUALITY_OF_SERVICE_UNAVAILABLE"
- "GWAPP_REQUESTED_FAC_NOT_SUBSCRIBED"
- "GWAPP_BC_NOT_AUTHORIZED"
- "GWAPP_BC_NOT_PRESENTLY_AVAILABLE"
- "GWAPP_SERVICE_NOT_AVAILABLE"
- "GWAPP_CUG_OUT_CALLS_BARRED"
- "GWAPP_CUG_INC_CALLS_BARRED"
- "GWAPP_ACCES_INFO_SUBS_CLASS_INCONS"
- "GWAPP_BC_NOT_IMPLEMENTED"
- "GWAPP_CHANNEL_TYPE_NOT_IMPLEMENTED"
- "GWAPP_REQUESTED_FAC_NOT_IMPLEMENTED"
- "GWAPP_ONLY_RESTRICTED_INFO_BEARER"
- "GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED"
- "GWAPP_INVALID_CALL_REF"
- "GWAPP_IDENTIFIED_CHANNEL_NOT_EXIST"

- "GWAPP_SUSPENDED_CALL_BUT_CALL_ID_NOT_EXIST"
- "GWAPP_CALL_ID_IN_USE"
- "GWAPP_NO_CALL_SUSPENDED"
- "GWAPP_CALL_HAVING_CALL_ID_CLEARED"
- "GWAPP_INCOMPATIBLE_DESTINATION"
- "GWAPP_INVALID_TRANSIT_NETWORK_SELECTION"
- "GWAPP_INVALID_MESSAGE_UNSPECIFIED"
- "GWAPP_NOT_CUG_MEMBER"
- "GWAPP_CUG_NON_EXISTENT"
- "GWAPP_MANDATORY_IE_MISSING"
- "GWAPP_MESSAGE_TYPE_NON_EXISTENT"
- "GWAPP_MESSAGE_STATE_INCONSISTENCY"
- "GWAPP_NON_EXISTENT_IE"
- "GWAPP_INVALID_IE_CONTENT"
- "GWAPP_MESSAGE_NOT_COMPATIBLE"
- "GWAPP_RECOVERY_ON_TIMER_EXPIRY"
- "GWAPP_PROTOCOL_ERROR_UNSPECIFIED"
- "GWAPP_INTERWORKING_UNSPECIFIED"
- "GWAPP_UKNOWN_ERROR"
- "RELEASE_BECAUSE_HELD_TIMEOUT"

40.1.3 CDR Fields for SBC

The device provides CDRs for the SBC application. This includes CDR fields for signaling and media.

40.1.3.1 CDR Fields for SBC Signaling

The CDR fields for SBC signaling are listed in the table below. The signaling CDRs are published for each SBC leg.

Table 40-2: CDR Fields for SBC Signaling

CDR Field Name	Description
SBCReportType	Report Type (call start, connect, or end)
EPTyp	Endpoint type
SIPCallId	Unique ID of call
SessionId	Unique Session ID
Orig	Call originator ("LCL" for local; "RMT" for remote)
SourceIp	Source IP address
SourcePort	Source UDP port
DestIp	Destination IP address
DestPort	Destination UDP port
TransportType	Transport type (UDP, TCP, or TLS)

CDR Field Name	Description
SrcURI	Source URI
SrcURIBeforeMap	Source URI before manipulation
DstURI	Destination URI
DstURIBeforeMap	Destination URI before manipulation
Durat	Call duration
TrmSd	Termination side (local or remote)
TrmReason	Termination reason
TrmReasonCategory	Termination reason category
SetupTime	Call setup time
ConnectTime	Call connect time
ReleaseTime	Call release time
RedirectReason	Redirect reason
RedirectURINum	Redirection URI
RedirectURINumBeforeMap	Redirect URI number before manipulation
TxSigIPDiffServ	Signaling IP DiffServ
IPGroup	IP Group description
SrdId	SRD name
SIPInterfaceId	SIP Interface ID
ProxySetId	Proxy Set ID
IpProfileId	IP Profile ID
MediaRealm	Media Realm name
DirectMedia	Direct media or traversing SBC (yes or no)

40.1.3.2 CDR Fields for SBC Media

The CDR fields for SBC media are listed in the table below. The media CDRs are published for each active media stream, thereby allowing multiple media CDRs, where each media CDR has a unique call ID corresponding to the signaling CDR.

Table 40-3: CDR Fields for SBC Media

CDR Field Name	Description
MediaReportType	Report type (media start, update, or end)
SIPCallId	Unique call ID
Cid	Channel CID
MediaType	Media type (audio, video, or text)
Coder	Coder name
PacketInterval	Coder packet interval

CDR Field Name	Description
LocalRtpIp	Local RTP IP address
LocalRtpPort	Local RTP port
RemoteRtpIp	Remote RTP IP address
RemoteRtpPort	Remote RTP port
InPackets	Number of received packets
OutPackets	Number of sent packets
LocalPackLoss	Local packet loss
RemotePackLoss	Remote packet loss
RTPdelay	RTP delay
RTPjitter	RTP jitter
TxRTPssrc	Tx RTP SSRC
RxRTPssrc	Local RTP SSRC
LocalRFactor	Local conversation quality
RemoteRFactor	Remote conversation quality
LocalMosCQ	Local MOS for conversation
RemoteMosCQ	Remote MOS for conversation
TxRTPIPDiffServ	Media IP DiffServ

40.1.4 Supported RADIUS Attributes

The following table provides descriptions on the RADIUS attributes included in the communication packets transmitted between the device and a RADIUS Server.

Table 40-4: Supported RADIUS Attributes

Attribute Number	Attribute Name	VSA No.	Purpose	Value Format	Example	AAA[1]
Request Attributes						
1	User-Name		Account number or calling party number or blank	String up to 15 digits long	5421385747	Start Acc Stop Acc
4	NAS-IP-Address		IP address of the requesting device	Numeric	192.168.14.43	Start Acc Stop Acc
6	Service-Type		Type of service requested	Numeric	1: login	Start Acc Stop Acc
26	H323-Incoming-Conf-Id	1	SIP call identifier	Up to 32 octets		Start Acc Stop Acc
26	H323-Remote-	23	IP address of the remote gateway	Numeric		Stop Acc

Attribute Number	Attribute Name	VSA No.	Purpose	Value Format	Example	AAA[1]
	Address					
26	H323-Conf-ID	24	H.323/SIP call identifier	Up to 32 octets		Start Acc Stop Acc
26	H323-Setup-Time	25	Setup time in NTP format 1	String		Start Acc Stop Acc
26	H323-Call-Origin	26	The call's originator: Answering (IP) or Originator (PSTN)	String	Answer, Originate etc	Start Acc Stop Acc
26	H323-Call-Type	27	Protocol type or family used on this leg of the call	String	VoIP	Start Acc Stop Acc
26	H323-Connect-Time	28	Connect time in NTP format	String		Stop Acc
26	H323-Disconnect-Time	29	Disconnect time in NTP format	String		Stop Acc
26	H323-Disconnect-Cause	30	Q.931 disconnect cause code	Numeric		Stop Acc
26	H323-Gw-ID	33	Name of the gateway	String	SIPIDString	Start Acc Stop Acc
26	SIP-Call-ID	34	SIP Call ID	String	abcde@ac.com	Start Acc Stop Acc
26	Call-Terminator	35	The call's terminator: PSTN-terminated call (Yes); IP-terminated call (No).	String	Yes, No	Stop Acc
30	Called-Station-ID			String	8004567145	Start Acc
			Destination phone number	String	2427456425	Stop Acc
			Calling Party Number (ANI)	String	5135672127	Start Acc Stop Acc
			Account Request Type (start or stop) **Note:** 'start' isn't supported on the Calling Card application.	Numeric	1: start, 2: stop	Start Acc Stop Acc
			No. of seconds tried in sending a particular record	Numeric	5	Start Acc Stop Acc
			Number of octets received for that call duration	Numeric		Stop Acc
			Number of octets sent for that call duration	Numeric		Stop Acc
			A unique accounting identifier - match start &	String	34832	Start Acc Stop Acc

Attribute Number	Attribute Name	VSA No.	Purpose	Value Format	Example	AAA[1]
			stop			
			For how many seconds the user received the service	Numeric		Stop Acc
			Number of packets received during the call	Numeric		Stop Acc
			Number of packets sent during the call	Numeric		Stop Acc
			Physical port type of device on which the call is active	String	0: Asynchronous	Start Acc Stop Acc
Response Attributes						
26	H323-Return-Code	103	The reason for failing authentication (0 = ok, other number failed)	Numeric	0 Request accepted	Stop Acc
44	Acct-Session-ID		A unique accounting identifier – match start & stop	String		Stop Acc

Below is an example of RADIUS Accounting, where the non-standard parameters are preceded with brackets.

```
Accounting-Request (361)
user-name = 111
acct-session-id = 1
nas-ip-address = 212.179.22.213
nas-port-type = 0
acct-status-type = 2
acct-input-octets = 4841
acct-output-octets = 8800
acct-session-time = 1
acct-input-packets = 122
acct-output-packets = 220
called-station-id = 201
calling-station-id = 202
// Accounting non-standard parameters:
(4923 33) h323-gw-id =
(4923 23) h323-remote-address = 212.179.22.214
(4923 1) h323-ivr-out = h323-incoming-conf-id:02102944 600a1899
3fd61009 0e2f3cc5
(4923 30) h323-disconnect-cause = 22 (0x16)
(4923 27) h323-call-type = VOIP
(4923 26) h323-call-origin = Originate
(4923 24) h323-conf-id = 02102944 600a1899 3fd61009 0e2f3cc5
```

40.2 Event Notification using X-Detect Header

The device supports the sending of notifications to a remote party notifying the occurrence (or detection) of certain events on the media stream. Event detection and notifications is performed using the SIP X-Detect message header and only when establishing a SIP dialog.

For supporting some events, certain device configurations need to be performed. The table below lists the supported event types (and subtypes) and the corresponding device configurations, if required:

Table 40-5: Supported X-Detect Event Types

Events Type	Subtype	Required Configuration
AMD	voice automatic silence unknown beep	EnableDSPIPMDetectors = 1 AMDTimeout = 2000 (msec) For AMD beep detection, AMDBeepDetectionMode = 1 or 2
CPT	SIT-NC SIT-IC SIT-VC SIT-RO Busy Reorder Ringtone beep	SITDetectorEnable = 1 UserDefinedToneDetectorEnable = 1 **Notes:** ▪ Ensure that the CPT file is configured with the required tone type. ▪ On beep detection, a SIP INFO message is sent with type AMD/CPT and subtype beep. ▪ The beep detection must be started using regular X-detect extension, with AMD or CPT request.
FAX	CED	(IsFaxUsed ≠ 0) or (IsFaxUsed = 0, and FaxTransportMode ≠ 0)
	modem	VxxModemTransportType = 3
PTT	voice-start voice-end	EnableDSPIPMDetectors = 1

The device can detect and report the following Special Information Tones (SIT) types from the PSTN:

- SIT-NC (No Circuit found)
- SIT-IC (Operator Intercept)
- SIT-VC (Vacant Circuit - non-registered number)
- SIT-RO (Reorder - System Busy)

There are additional three SIT tones that are detected as one of the above SIT tones:

- The NC* SIT tone is detected as NC
- The RO* SIT tone is detected as RO
- The IO* SIT tone is detected as VC

The device can map these SIT tones to a Q.850 cause and then map them to SIP 5xx/4xx responses, using the parameters SITQ850Cause, SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO.

Table 40-6: Special Information Tones (SITs) Reported by the device

Special Information Tones (SITs) Name	Description	First Tone Frequency Duration		Second Tone Frequency Duration		Third Tone Frequency Duration	
		(Hz)	(ms)	(Hz)	(ms)	(Hz)	(ms)
NC1	No circuit found	985.2	380	1428.5	380	1776.7	380
IC	Operator intercept	913.8	274	1370.6	274	1776.7	380
VC	Vacant circuit (non registered number)	985.2	380	1370.6	274	1776.7	380
RO1	Reorder (system busy)	913.8	274	1428.5	380	1776.7	380
NC*	-	913.8	380	1370.6	380	1776.7	380
RO*	-	985.2	274	1370.6	380	1776.7	380
IO*	-	913.8	380	1428.5	274	1776.7	380

For example:

```
INFO sip:5001@10.33.2.36 SIP/2.0
Via: SIP/2.0/UDP 10.33.45.65;branch=z9hG4bKac2042168670
Max-Forwards: 70
From: <sip:5000@10.33.45.65;user=phone>;tag=1c1915542705
To: <sip:5001@10.33.2.36;user=phone>;tag=WQJNIDDPCOKAPIDSCOTG
Call-ID: AIFHPETLLMVVFWPDXUHD@10.33.2.36
CSeq: 1 INFO
Contact: <sip:2206@10.33.45.65>
Supported: em,timer,replaces,path,resource-priority
Content-Type: application/x-detect
Content-Length: 28
Type= CPT
SubType= SIT-IC
```

The X-Detect event notification process is as follows:

1. For IP-to-Tel or Tel-to-IP calls, the device receives a SIP request message (using the X-Detect header) that the remote party wishes to detect events on the media stream. For incoming (IP-to-Tel) calls, the request must be indicated in the initial INVITE and responded to either in the 183 response (for early dialogs) or in the 200 OK response (for confirmed dialogs).
2. Once the device receives such a request, it sends a SIP response message (using the X-Detect header) to the remote party, listing all supported events that can be detected. The absence of the X-Detect header indicates that no detections are available.
3. Each time the device detects a supported event, the event is notified to the remote party by sending an INFO message with the following message body:
   - Content-Type: application/X-DETECT
   - Type = **[AMD |** CPT | FAX | PTT…]
   - Subtype = xxx (according to the defined subtypes of each type)

Below is an example of SIP messages using the X-Detect header:

```
INVITE sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X- Detect: Request=CPT,FAX
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>;tag=1c19282
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:101@10.33.2.53>
X- Detect: Response=CPT,FAX
INFO sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X- Detect: Response=CPT,FAX
Content-Type: Application/X-Detect
Content-Length: xxx
Type = CPT
Subtype = SIT
```

40.3 Querying Device Channel Resources using SIP OPTIONS

The device reports its maximum and available channel resources in SIP 200 OK responses upon receipt of SIP OPTIONS messages. The device sends this information in the SIP X-Resources header with the following parameters:

- **telchs:** specifies the total telephone channels as well as the number of free (available) telephone channels
- **mediachs:** specifies the total and the free number of channels associated with media services (e.g., announcements and conferencing)

Below is an example of the X-Resources:

```
X-Resources: telchs= 12/4;mediachs=0/0
```

In the example above, "telchs" specifies the number of available channels and the number of occupied channels (4 channels are occupied and 12 channels are available).

Part VIII

Diagnostics

This part describes the diagnostics procedures.

Reader's Notes

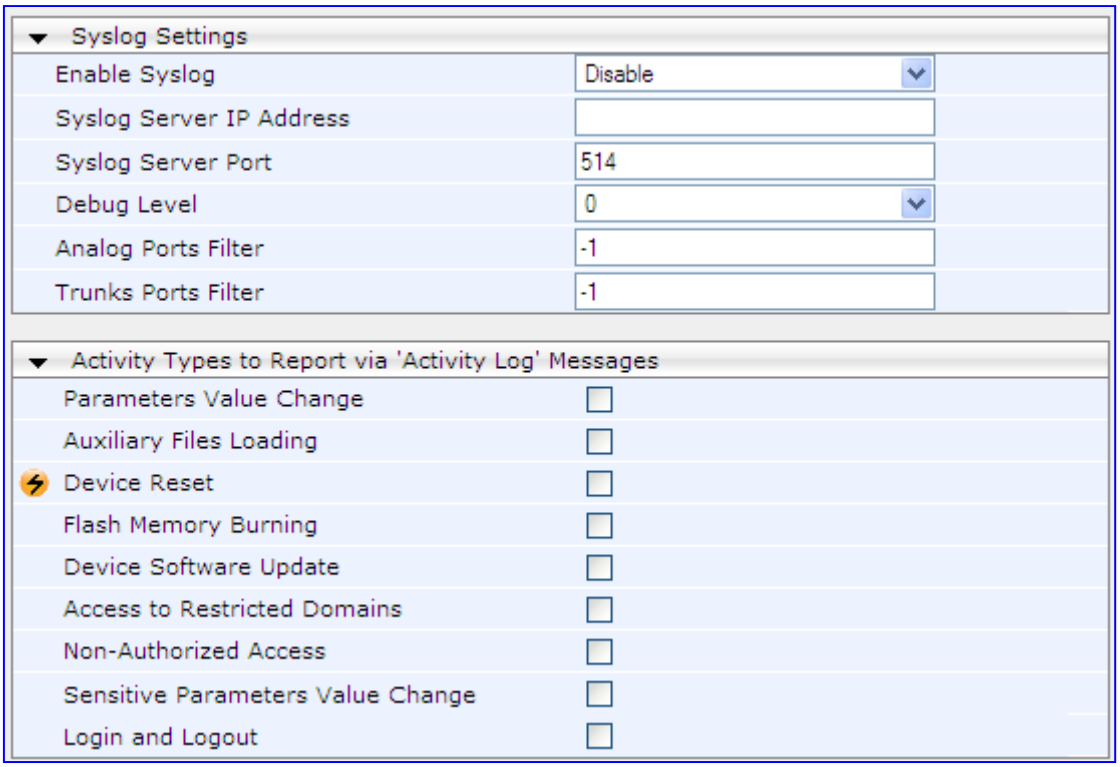
41 Configuring Syslog Settings

The Syslog Settings page allows you to configure the device's embedded Syslog client. For a detailed description on the Syslog parameters, see 'Syslog, CDR and Debug Parameters' on page 775. For more information on Syslog messages and using third-party Syslog servers, refer to the *Product Reference Manual*.

➢ **To configure the Syslog client:**

1. Open the Syslog Settings page (**Configuration** tab > **System** menu > **Syslog Settings**).

Figure 41-1: Syslog Settings Page



2. Configure the parameters as required, and then click Submit to apply your changes.
3. To save the changes to flash memory, see 'Saving Configuration' on page 680.

Reader's Notes

42 Capturing VoIP and Data-Router Network Traffic

You can capture VoIP and data-router network traffic by sending traces to the CLI or to a file that is later sent to a TFTP server. The traffic can be captured using the following CLI commands to start the debug capture process:

```
debug capture voip
debug capture data
```

Reader's Notes

43 Running Data-Router Diagnostic Tests

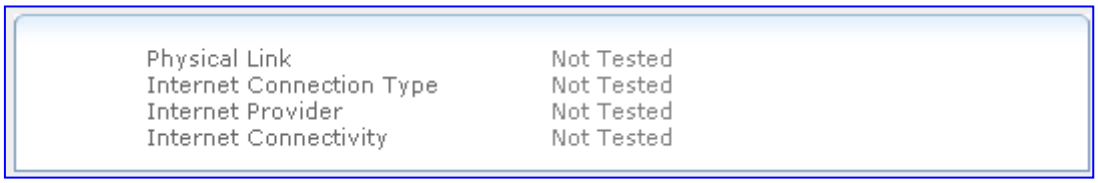
43.1 Internet Connectivity Diagnostic Tests

The procedure below describes how to run a series of tests to validate your device's Internet connection. These tests diagnose and resolve Internet connectivity problems.

➢ **To run Internet connectivity tests:**

1. Click the **WAN Status** item (**Status & Diagnostics** tab > **Data Status** menu > **WAN Status**), and then click the **Have Internet Connection problems? Click here** link; the following page appears:

Figure 43-1: Running Internet Connectivity Diagnostics Tests



2. Click the **Run** button. While testing is in progress, you may abort the diagnostics process by clicking the **Abort** button.
3. Should a failure message appear, click **Repair** to initiate the Diagnostics Wizard procedure. The device performs the following consecutive diagnostic tests:
   a. **Test Ethernet Link** - tests the physical integrity of the WAN connection (e.g., Ethernet cable is plugged).
   b. **Analyze Internet Connection Type** - checks whether the Internet connection type is correctly configured (if not, change the connection type, e.g., to 'Automatic IP Address Ethernet Connection').
   c. **Setup Internet Connection** - performed if your Internet connection requires login details provided by your ISP, e.g. when using PPPoE (if the test fails, re-enter the login user name and password).
   d. **Test Service Provider Connection** - tests the connectivity to your ISP.
   e. **Test Internet Connection** - tests the connectivity to the Internet.
4. When the Diagnostics Wizard completes all the tests, it provides a summary of all the above test results:
5. Click **Finish** to end the Wizard Diagnostics.

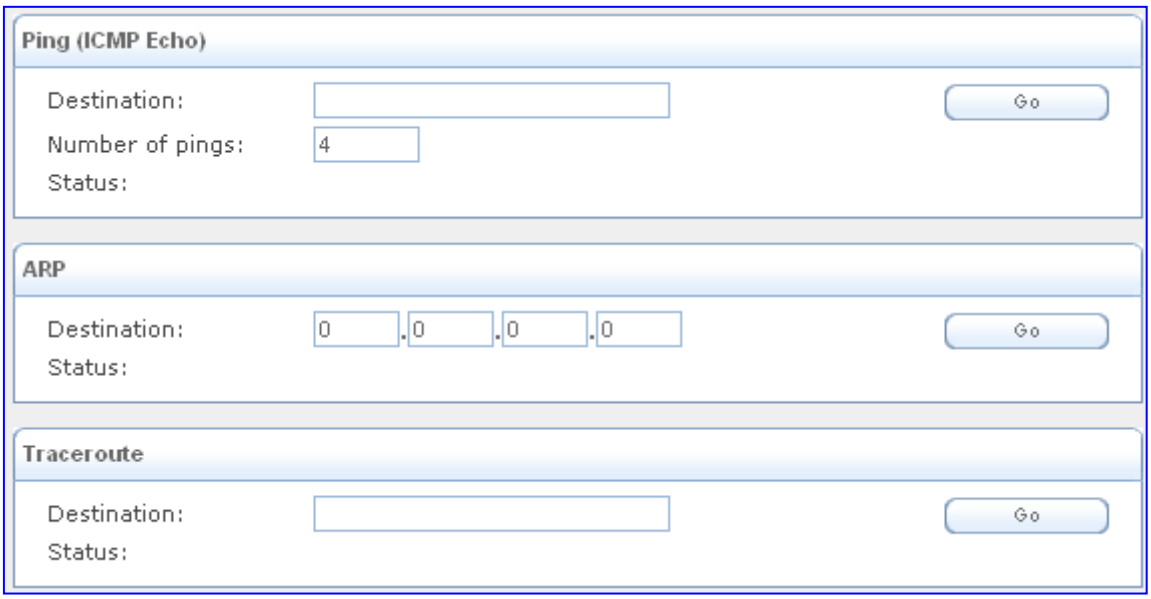
43.2 Running Diagnostic Tests

The **Diagnostics** item can assist you in testing network connectivity and viewing statistics such as the number of packets transmitted and received, round-trip time and success status. This page allows you to run network connectivity tests (ping), query the physical address (MAC) of a host, and run a trace route test.

➢ **To run diagnostic tests:**

1. Click the **Diagnostics** item (**Status & Diagnostics** tab > **Data Status** menu > **Diagnostics**); the following page appears:

Figure 43-2: System - Diagnostics Page



2. To diagnose network connectivity, under the 'Ping (ICMP Echo)' group, perform the following:
   a. In the 'Destination' field, enter the IP address or URL to be tested.
   b. In the 'Number of Pings', enter the number of pings you would like to run.
   c. Click **Go**; in a few moments, diagnostic statistics are displayed. If no new information appears, click **Refresh**.
3. To query the physical address (MAC) of a host, under the 'ARP' group, perform the following:
   a. In the 'Destination' field, enter an IP address of the target host.
   b. Click **Go**; in a few moments, diagnostic statistics are displayed. If no new information is displayed, click **Refresh**.
4. To run a traceroute test, under the Traceroute group, perform the following:
   a. In the 'Destination' field, enter the IP address or URL to be tested.
   b. Click **Go**; the traceroute test commences, constantly refreshing the page.
   c. To stop the test and view the results, click **Cancel**.

Part IX

Appendices

This part includes appendices.

Reader's Notes

A Configuration Parameters Reference

The device's VoIP functionality (not data-routing functionality) configuration parameters, default values, and their descriptions are documented in this section.

Parameters and values enclosed in square brackets (**[...]) represent** the *ini* file parameters and their enumeration values; parameters not enclosed in square brackets represent their corresponding Web interface and/or EMS parameters.

Note: Some parameters are configurable only through the *ini* file.

A.1 Networking Parameters

This subsection describes the device's networking parameters.

A.1.1 VoIP Multiple Network Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

Table A-1: IP Network Interfaces and VLAN Parameters

Parameter	Description
Multiple Interface Table	
Web: Multiple Interface Table **EMS: IP Interface Settings** CLI: configure voip > interface network-if display **[InterfaceTable]**	This *para*meter table configures the Multiple Interface table for configuring the IP addresses of the voice and/or data functionalities and logical IP addresses. The format of this parameter is as follows: **[**InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName, InterfaceTable_PrimaryDNSServerIPAddress, InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingInterface; [\InterfaceTable] For example: InterfaceTable 0 = 0, 0, 192.168.85.14, 16, 0.0.0.0, 1, Management; InterfaceTable 1 = 2, 0, 200.200.85.14, 24, 0.0.0.0, 200, Control; InterfaceTable 2 = 1, 0, 211.211.85.14, 24, 211.211.85.1, 211, Media; **Notes:** ▪ For this *para*meter table to take effect, a device reset is required. ▪ Up to 12 logical IP addresses with associated VLANs can be defined (indices 0-11). ▪ Each interface index must be unique. ▪ Each interface must have a unique VLAN ID. ▪ Each interface must have a unique subnet. ▪ Subnets in different interfaces must not overlap (e.g., defining two interfaces with 10.0.0.1/8 and 10.50.10.1/24 is invalid). Each interface must have its own address space.

Parameter	Description
	▪ Upon device start up, this table is parsed and passes comprehensive validation tests. If any errors occur during this validation phase, the device sends an error message to the Syslog server and falls back to a “safe mode”, using a single IPv4 interface and without VLANs. Therefore, check the Syslog for any error messages. ▪ To configure multiple VoIP IP interfaces in the Web interface and for a detailed description of the table's parameters, see 'Configuring IP Interface Settings' on page 117). ▪ For a description of configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
[EnableNTPasOAM]	Defines the application type for NTP services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control. **Note:** For this parameter to take effect, a device reset is required.

A.1.2 VoIP Static Routing Parameters

The static routing parameters are described in the table below.

Table A-2: Static Routing Parameters

Parameter	Description
Static IP Routing Table	
Web/EMS: IP Routing Table CLI: configure voip > static **[StaticRouteTable]**	Defines up to 30 static VoIP IP routing rules for the device. These rules can be associated with IP interfaces defined in the Multiple Interface table (InterfaceTable parameter). The routing decision for sending the outgoing IP packet is based on the source subnet/VLAN. If not associated with an IP interface, the static IP rule is based on destination IP address. When the destination of an outgoing IP packet does not match one of the subnets defined in the Multiple Interface table, the device searches this table for an entry that matches the requested destination host/network. If such an entry is found, the device sends the packet to the indicated router (i.e., next hop). If no explicit entry is found, the packet is sent to the default gateway according to the source interface of the packet (if defined). The format of this parameter is as follows: [StaticRouteTable] FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description; [\StaticRouteTable] **Notes:** ▪ The Gateway address must be in the same subnet as configured in the Multiple Interface table for VoIP network interfaces (refer to 'Configuring IP Interface Settings' on page 117). ▪ The StaticRouteTable_Description parameter is a string value of up to 30 characters. ▪ The metric value (next hop) is automatically set to 1.

A.1.3 Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below.

The device allows you to specify DiffServ (Differentiated Services) values for four predifined service classes:

- Premium Media service class – used for RTP Media traffic
- Premium Control Service class – used for Call Control traffic
- Gold Service class – used for streaming applications
- Bronze Service class – used for OAMP applications

The Layer-3 QoS parameters enables setting the values of the DiffServ field in the IP Header of the frames related to a specific service class. The Layer-2 QoS parameters enable setting the values for the 3 priority bits in the VLAN tag (IEEE 802.1p standard) according to the value of the DiffServ field found in the packet IP header.

Table A-3: QoS Parameters

Parameter	Description
Layer-2 Class Of Service (CoS) Parameters (VLAN Tag Priority Field)	
Web: DiffServ Table EMS: QoS Settings – DSCP to QoS Mapping CLI: configure voip > vlan-mapping **[DiffServToVlanPriority]**	This parameter table allows you to configure DiffServ-to-VLAN Priority mapping. For each packet sent to the LAN, the VLAN Priority of the packet is set according to the DiffServ value in the IP header of the packet. The format of this ini file is as follows: [DiffServToVlanPriority] FORMAT DiffServToVlanPriority_Index = DiffServToVlanPriority_DiffServ, DiffServToVlanPriority_VlanPriority; [\DiffServToVlanPriority] For example: DiffServToVlanPriority 0 = 46, 6; DiffServToVlanPriority 1 = 40, 6; DiffServToVlanPriority 2 = 26, 4; DiffServToVlanPriority 3 = 10, 2; **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ You can configure up to 64 VLAN tag priorities (i.e., indices 0-63). ▪ The valid range of the parameter DiffServ is 0-63. ▪ The valid range of the parameter VlanPriority is 0-7. ▪ To set a default VLAN Priority (used for ARPs and automated ICMP packets, as well as for IP Packets without DiffServ value), set a VLAN Priority value for DiffServ 0.
Layer-3 Class of Service (TOS/DiffServ) Parameters	
Web: Media Premium QoS EMS: Premium Service Class Media Diff Serv CLI: media-qos **[PremiumServiceClassMediaDiffServ]**	Defines the DiffServ value for Premium Media CoS content. The valid range is 0 to 63. The default value is 46. **Notes:**

Parameter	Description
	▪ For this parameter to take effect, a device reset is required. ▪ The value for the Premium Control DiffServ is determined by the following (according to priority): ✓ IPDiffServ value in the selected IP Profile (IPProfile parameter). ✓ PremiumServiceClassMediaDiffServ.
Web: Control Premium QoS EMS: Premium Service Class Control Diff Serv CLI: control-qos **[PremiumServiceClassControlDiffServ]**	Defines the DiffServ value for Premium Control CoS content (Call Control applications). The valid range is 0 to 63. The default value is 40. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The value for the Premium Control DiffServ is determined by the following (according to priority): ✓ SigIPDiffserv value in the selected IP Profile (IPProfile parameter). ✓ PremiumServiceClassControlDiffServ.
Web: Gold QoS EMS: Gold Service Class Diff Serv CLI: gold-qos **[GoldServiceClassDiffServ]**	Defines the DiffServ value for the Gold CoS content (Streaming applications). The valid range is 0 to 63. The default value is 26. **Note:** For this parameter to take effect, a device reset is required.
Web: Bronze QoS EMS: Bronze Service Class Diff Serv CLI: bronze-qos **[BronzeServiceClassDiffServ]**	Defines the DiffServ value for the Bronze CoS content (OAMP applications). The valid range is 0 to 63. The default value is 10. **Note:** For this parameter to take effect, a device reset is required.

A.1.4 NAT and STUN Parameters

The Network Address Translation (NAT) parameters are described in the table below.

Table A-4: NAT Parameters

Parameter	Description
NAT Parameters	
Web/EMS: NAT Traversal CLI: disable-NAT-traversal **[DisableNAT]**	Enables the NAT mechanism. ▪ **[0]** Enable ▪ **[1]** Disable (default) **Note:** The compare operation that is performed on the IP address is enabled by default and is configured by the parameter EnableIPAddrTranslation. The compare operation that is performed on the UDP port is disabled by default and is configured by the parameter EnableUDPPortTranslation.
Web: NAT IP Address EMS: Static NAT IP Address CLI: nat-ip-addr **[StaticNatIP]**	Defines the global (public) IP address of the device to enable static NAT between the device and the Internet. **Note:** For this parameter to take effect, a device reset is required.
[EnableIPAddrTranslation]	Enables IP address translation for RTP, RTCP, and T.38 packets. ▪ **[0]** = Disable IP address translation. ▪ **[1]** = Enable IP address translation (default). When enabled, the device compares the source IP address of the first incoming packet to the remote IP address stated in the opening of the channel. If the two IP addresses don't match, the NAT mechanism is activated. Consequently, the remote IP address of the outgoing stream is replaced by the source IP address of the first incoming packet. **Note:** The NAT mechanism must be enabled for this parameter to take effect (i.e., the parameter DisableNAT is set to 0).
[EnableUDPPortTranslation]	Enables UDP port translation. ▪ **[0]** = Disables UDP port translation (default). ▪ **[1]** = Enables UDP port translation. The device compares the source UDP port of the first incoming packet to the remote UDP port stated in the opening of the channel. If the two UDP ports don't match, the NAT mechanism is activated. Consequently, the remote UDP port of the outgoing stream is replaced by the source UDP port of the first incoming packet. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The NAT mechanism and the IP address translation must be enabled for this parameter to take effect (i.e., set the parameter DisableNAT to 0 and the parameter EnableIpAddrTranslation to 1).

A.1.5 NFS Parameters

The Network File Systems (NFS) configuration parameters are described in the table below.

Table A-5: NFS Parameters

Parameter	Description
CLI: base-port **[NFSBasePort]**	Defines the start of the range of numbers used for local UDP ports used by the NFS client. The maximum number of local ports is maximum channels plus maximum NFS servers. The valid range is 0 to 65535. The default is 47000.
Web: NFS Table **EMS: NFS Settings**	
[NFSServers]	This *para*meter table defines up to 16 NFS file systems so that the device can access a remote server's shared files and directories for loading cmp, ini, and auxiliary files (using the Automatic Update mechanism). As a file system, the NFS is independent of machine types, OSs, and network architectures. Note that an NFS file server can share multiple file systems. There must be a separate row for each remote file system shared by the NFS file server that needs to be accessed by the device. The format of this ini file table parameter is as follows: [NFSServers] FORMAT NFSServers_Index = NFSServers_HostOrIP, NFSServers_RootPath, NFSServers_NfsVersion, NFSServers_AuthType, NFSServers_UID, NFSServers_GID, NFSServers_VlanType; [\NFSServers] For example: NFSServers 1 = 101.1.13, /audio1, 3, 1, 0, 1, 1; **Notes:** ▪ You can configure up to 16 NFS file systems (where the first index is 0). ▪ To avoid terminating current calls, a row must not be deleted or modified while the device is currently accessing files on the remote NFS file system. ▪ The combination of host/IP and Root Path must be unique for each index in the table. For example, the table must include only one index entry with a Host/IP of '192.168.1.1' and Root Path of '/audio'. ▪ This parameter is applicable only if VLANs are enabled or Multiple IPs is configured. ▪ For a detailed description of the table's parameters and to configure NFS using the Web interface, see 'Configuring NFS Settings' on page 141. ▪ For a description of configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.1.6 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

Table A-6: DNS Parameters

Parameter	Description
Internal DNS Table	
Web: Internal DNS Table EMS: DNS Information CLI: configure voip > control-network dns Dns2Ip **[DNS2IP]**	This *para*meter table defines the internal DNS table for resolving host names into IP addresses. Up to four different IP addresses (in dotted-decimal notation) can be assigned to a host name. The format of this parameter is as follows: [Dns2Ip] FORMAT Dns2Ip_Index = Dns2Ip_DomainName, Dns2Ip_FirstIpAddress, Dns2Ip_SecondIpAddress, Dns2Ip_ThirdIpAddress, Dns2Ip_FourthIpAddress; [\Dns2Ip] For example: Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2, 3.3.3.3, 4.4.4.4; **Notes:** ▪ This parameter can include up to 20 indices. ▪ If the internal DNS table is used, the device first attempts to resolve a domain name using this table. If the domain name isn't found, the device performs a DNS resolution using an external DNS server. ▪ To configure the internal DNS table using the Web interface and for a description of the parameters in this *ini* file table parameter, see 'Configuring the Internal DNS Table' on page 137. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Internal SRV Table	
Web: Internal SRV Table EMS: DNS Information CLI: configure voip > control-network dns Srv2Ip **[SRV2IP]**	This parameter table defines the internal SRV table for resolving host names into DNS A-Records. Three different A-Records can be assigned to a host name. Each A-Record contains the host name, priority, weight, and port. The format of this parameter is as follows: [SRV2IP] FORMAT SRV2IP_Index = SRV2IP_InternalDomain, SRV2IP_TransportType, SRV2IP_Dns1, SRV2IP_Priority1, SRV2IP_Weight1, SRV2IP_Port1, SRV2IP_Dns2, SRV2IP_Priority2, SRV2IP_Weight2, SRV2IP_Port2, SRV2IP_Dns3, SRV2IP_Priority3, SRV2IP_Weight3, SRV2IP_Port3; [\SRV2IP] For example: SRV2IP 0 = SrvDomain,0,Dnsname1,1,1,500,Dnsname2,2,2,501,$$,0,0,0; **Notes**: ▪ This parameter can include up to 10 indices. ▪ If the Internal SRV table is used, the device first attempts to resolve a domain name using this table. If the domain name isn't located, the device performs an SRV resolution using an external DNS server. ▪ To configure the Internal SRV table using the Web interface and for a description of the parameters in this *ini* file table parameter, see 'Configuring the Internal SRV Table' on page 138.

Parameter	Description
	▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.1.7 DHCP Parameters

The Dynamic Host Control Protocol (DHCP) parameters are described in the table below.

Table A-7: DHCP Parameters

Parameter	Description
Web: Enable DHCP EMS: DHCP Enable **[DHCPEnable]**	Enables Dynamic Host Control Protocol (DHCP) functionality. ▪ **[0]** Disable = Disable DHCP support on the device (default). ▪ **[1]** Enable = Enable DHCP support on the device. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ After you enable the DHCP server, perform the following procedure: **a.** Enable DHCP and save the configuration. **b.** Perform a cold reset using the device's hardware reset button (soft reset using the Web interface doesn't trigger the DHCP procedure and this parameter reverts to 'Disable'). ▪ For more information on DHCP, refer to the *Product Reference Manual*. ▪ This parameter is a special 'Hidden' parameter. Once defined and saved in flash memory, its assigned value doesn't revert to its default even if the parameter doesn't appear in the *ini* file.
EMS: DHCP Speed Factor **[DHCPSpeedFactor]**	Defines the DHCP renewal speed. ▪ **[0]** = Disable ▪ **[1]** = Normal (default) ▪ **[2]** to **[10]** = Fast When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is divided by this factor. Some DHCP-enabled routers perform better when set to 4. **Note:** For this parameter to take effect, a device reset is required.

A.1.8 NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

Table A-8: NTP and Daylight Saving Time Parameters

Parameter	Description
NTP Parameters **Note:** For more information on Network Time Protocol (NTP), see 'Simple Network Time Protocol Support' on page 112.	
Web: NTP Server IP Address EMS: Server IP Address CLI: primary-server **[NTPServerIP]**	Defines the IP address (in dotted-decimal notation) of the NTP server. The default IP address is 0.0.0.0 (i.e., internal NTP client is disabled).
Web: NTP UTC Offset EMS: UTC Offset CLI: utc-offset **[NTPServerUTCOffset]**	Defines the Universal Time Coordinate (UTC) offset (in seconds) from the NTP server. The default offset is 0. The offset range is -43200 to 43200.
Web: NTP Update Interval EMS: Update Interval CLI: update-interval **[NTPUpdateInterval]**	Defines the time interval (in seconds) that the NTP client requests for a time update. The default interval is 86400 (i.e., 24 hours). The range is 0 to 214783647. **Note:** It is not recommend to set this parameter to beyond one month (i.e., 2592000 seconds).
Daylight Saving Time Parameters	
Web: Day Light Saving Time EMS: Mode CLI: summer-time **[DayLightSavingTimeEnable]**	Enables daylight saving time. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Start Time EMS: Start CLI: start **[DayLightSavingTimeStart]**	Defines the date and time when daylight saving begins. The format of the value is mo:dd:hh:mm (month, day, hour, and minutes).
Web: End Time EMS: End CLI: end **[DayLightSavingTimeEnd]**	Defines the date and time when daylight saving ends. The format of the value is mo:dd:hh:mm (month, day, hour, and minutes).
Web/EMS: Offset CLI: offset **[DayLightSavingTimeOffset]**	Defines the daylight saving time offset (in minutes). The valid range is 0 to 120. The default is 60.

A.2 Management Parameters

This subsection describes the device's Web and Telnet parameters.

A.2.1 General Parameters

The general management parameters are described in the table below.

Table A-9: General Management Parameters

Parameter	Description
WAN OAMP Interface CLI: bind GigabitEthernet <slot/port.vlanId> oamp **[OAMPWanInterfaceName]**	Binds the OAMP interface to a WAN interface, which can later be associated with a Virtual Routing and Forwarding (VRF).
Web: Allow WAN access to HTTP CLI: wan-http-allow **[AllowWanHTTP]**	Enables WAN access to the management interface through HTTP. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to HTTPS CLI: wan-https-allow **[AllowWanHTTPS]**	Enables WAN access to the management interface through HTTPS. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to SNMP CLI: wan-snmp-allow **[AllowWanSNMP]**	Enables WAN access to the management interface through SNMP. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to Telnet **CLI: wan-telnet-allow** **[AllowWanTelnet**	Enables WAN access to the management interface through Telnet. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to SSH CLI: wan-ssh-allow **[AllowWanSSH]**	Enables WAN access to the management interface through SSH. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Web and Telnet Access List Table EMS: Web Access Addresses **[WebAccessList_x]**	Defines up to ten IP addresses that are permitted to access the device's Web interface and Telnet interfaces. Access from an undefined IP address is denied. When no IP addresses are defined in this table, this security feature is inactive (i.e., the device can be accessed from any IP address). The default value is 0.0.0.0 (i.e., the device can be accessed from any IP address). For example: WebAccessList_0 = 10.13.2.66 WebAccessList_1 = 10.13.77.7 For defining the Web and Telnet Access list using the Web interface, see 'Configuring Web and Telnet Access List' on page 82.

Parameter	Description
Web: Use RADIUS for Web/Telnet Login EMS: Web Use Radius Login CLI: enable-mgmt-login **[WebRADIUSLogin]**	Enables RADIUS queries for Web and Telnet authentication. ▪ **[0]** Disable (default). ▪ **[1]** Enable = Logging into the device's Web and Telnet embedded servers is done through a RADIUS server. The device contacts a user-defined server and verifies the given user name and password against a remote database, in a secure manner. **Notes:** ▪ The parameter EnableRADIUS must be set to 1. ▪ RADIUS authentication requires HTTP basic authentication, meaning the user name and password are transmitted in clear text over the network. Therefore, it's recommended to set the parameter HTTPSOnly to 1 to force the use of HTTPS, since the transport is encrypted. ▪ If using RADIUS authentication when logging in to the CLI, only the primary Web User Account (which has Security Administration access level) can access the device's CLI (see 'Configuring Web User Accounts' on page 76).

A.2.2 Web Parameters

The Web parameters are described in the table below.

Table A-10: Web Parameters

Parameter	Description
Web: Deny Access On Fail Count **[DenyAccessOnFailCount]**	Defines the maximum number of login attempts after which the requesting IP address is blocked. The valid value range is 0 to 32768. The values 0 and 1 mean immediate block. The default is 3.
Web: Deny Authentication Timer **[DenyAuthenticationTimer]**	Defines the time (in seconds) that login to the Web interface is denied for a user that has reached maximum login attempts as defined by the DenyAccessOnFailCount parameter. Only after this time expires can the user attempt to login from the same IP address. The default is 0.
Web: Display Login Information **[DisplayLoginInformation]**	Enables display of user's login information on each successful login attempt. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
[EnableMgmtTwoFactorAuthentication]	Enables Web login authentication using a third-party, smart card. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable When enabled, the device retrieves the Web user's login username from the smart card, which is automatically displayed (read-only) in the Web Login screen; the user is then required to provide only the login password. Typically, a TLS connection is established between the smart card and the device's Web interface, and a RADIUS server is implemented to authenticate the password with the username.

Parameter	Description
	Thus, this feature implements a two-factor authentication - what the user has (the physical card) and what the user knows (i.e., the login password).
[DisableWebTask]	Enables device management through the Web interface. ▪ **[0]** = Enable Web management (default). ▪ **[1]** = Disable Web management. **Note:** For this parameter to take effect, a device reset is required.
CLI: http-port **[HTTPport]**	Defines the LAN HTTP port for Web management (default is 80). To enable Web management from the LAN, configure the desired port. **Note:** For this parameter to take effect, a device reset is required.
EMS: Disable WEB Config **[DisableWebConfig]**	Determines whether the entire Web interface is read-only. ▪ **[0]** = Enables modifications of parameters (default). ▪ **[1]** = Web interface is read-only. When in read-only mode, parameters can't be modified. In addition, the following pages can't be accessed: 'Web User Accounts', 'Certificates', 'Regional Settings', 'Maintenance Actions' and all file-loading pages ('Load Auxiliary Files', 'Software Upgrade Wizard', and 'Configuration File'). **Note:** For this parameter to take effect, a device reset is required.
[ResetWebPassword]	Determines whether the device resets the username and password of the primary and secondary accounts to their default settings. ▪ **[0]** = Password and username retain their values (default). ▪ **[1]** = Password and username are reset. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The username and password cannot be reset from the Web interface (i.e., via AdminPage or by loading an *ini* file).
[WelcomeMessage]	This *para*meter table defines the Welcome message that appears after a Web interface login. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text [\WelcomeMessage] For Example: FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "**********" ; WelcomeMessage 2 = "********* This is a Welcome message ***" ; WelcomeMessage 3 = "**********" ; **Notes:** ▪ Each index represents a line of text in the Welcome message box. Up to 20 indices can be defined. ▪ The configured text message must be enclosed in double quotation marks (i.e., "..."). ▪ If this parameter is not configured, no Welcome message is displayed. ▪ For a description on using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.2.3 Telnet Parameters

The Telnet parameters are described in the table below. **Note:** Telnet is currently supported only for debugging from the LAN interface.

Table A-11: Telnet Parameters

Parameter	Description
Web: Embedded Telnet Server EMS: Server Enable CLI: telnet **[TelnetServerEnable]**	Enables the device's embedded Telnet server. Telnet is disabled by default for security. ▪ **[0]** Disable (default) ▪ **[1]** Enable Unsecured ▪ **[2]** Enable Secured (SSL) **Note:** Only the primary Web User Account (which has Security Administration access level) can access the device using Telnet (see 'Configuring Web User Accounts' on page 76).
Web: Telnet Server TCP Port EMS: Server Port CLI: telnet-port **[TelnetServerPort]**	Defines the port number for the embedded Telnet server. The valid range is all valid port numbers. The default port is 23.
Web: Telnet Server Idle Timeout EMS: Server Idle Disconnect CLI: idle-timeout **[TelnetServerIdleDisconnect]**	Defines the timeout (in minutes) for disconnection of an idle Telnet session. When set to zero, idle sessions are not disconnected. The valid range is any value. The default value is 0. **Note:** For this parameter to take effect, a device reset is required.

A.2.4 SNMP Parameters

The SNMP parameters are described in the table below.

Table A-12: SNMP Parameters

Parameter	Description
Web: Enable SNMP CLI: disable **[DisableSNMP]**	Enables SNMP. ▪ **[0]** Enable = SNMP is enabled (default). ▪ **[1]** Disable = SNMP is disabled and no traps are sent.
CLI: port **[SNMPPort]**	Defines the device's local (LAN) UDP port used for SNMP Get/Set commands. The range is 100 to 3999. The default port is 161. **Note:** For this parameter to take effect, a device reset is required.
CLI: trusted-managers **[SNMPTrustedMGR_x]**	Defines up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes SNMP Get and Set requests. **Notes:** ▪ By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests.

Parameter	Description
	▪ If no values are assigned to these parameters any manager can access the device. ▪ Trusted managers can work with all community strings.
[ChassisPhysicalAlias]	Defines the 'alias' name object for the physical entity as specified by a network manager, and provides a non-volatile 'handle' for the physical entity. The valid range is a string of up to 255 characters.
[ChassisPhysicalAssetID]	Defines the user-assigned asset tracking identifier object for the device's chassis as specified by an EMS, and provides non-volatile storage of this information. The valid range is a string of up to 255 characters.
[ifAlias]	Defines the textual name of the interface. The value is equal to the ifAlias SNMP MIB object. The valid range is a string of up to 64 characters.
EMS: Keep Alive Trap Port **[KeepAliveTrapPort]**	Defines the port to which keep-alive traps are sent. The valid range is 0 - 65534. The default is port 162.
[SendKeepAliveTrap]	Enables keep-alive traps and sends them every 9/10 of the time as defined by the NATBindingDefaultTimeout parameter. ▪ **[0]** = Disable ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
CLI: sys-oid **[SNMPSysOid]**	Defines the base product system OID. The default is eSNMP_AC_PRODUCT_BASE_OID_D. **Note:** For this parameter to take effect, a device reset is required.
[SNMPTrapEnterpriseOid]	Defines the Trap Enterprise OID. The default is eSNMP_AC_ENTERPRISE_OID. The inner shift of the trap in the AcTrap subtree is added to the end of the OID in this parameter. **Note:** For this parameter to take effect, a device reset is required.
[acUserInputAlarmDescription]	Defines the description of the input alarm.
[acUserInputAlarmSeverity]	Defines the severity of the input alarm.
[AlarmHistoryTableMaxSize]	Defines the maximum number of rows in the Alarm History table. This parameter can be controlled by the Config Global Entry Limit MIB (located in the Notification Log MIB). The valid range is 50 to 1000. The default value is 500. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
CLI: engine-id **[SNMPEngineIDString]**	Defines the SNMP engine ID for SNMPv2/SNMPv3 agents. This is used for authenticating a user attempting to access the SNMP agent on the device. The ID can be a string of up to 36 characters. The default value is 00:00:00:00:00:00:00:00:00:00:00:00 (12 Hex octets characters). The provided key must be set with 12 Hex values delimited by a colon (":") in the format xx:xx:...:xx. For example, 00:11:22:33:44:55:66:77:88:99:aa:bb **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Before setting this parameter, all SNMPv3 users must be deleted; otherwise, the parameter setting is ignored. ▪ If the supplied key does not pass validation of the 12 Hex values input or it is set with the default value, the engine ID is generated according to RFC 3411.
Web: SNMP Trap Destination Parameters **EMS: Network > SNMP Managers Table** **Note:** Up to five SNMP trap managers can be defined.	
SNMP Manager CLI: is-used **[SNMPManagerIsUsed_x]**	Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used to receive SNMP traps. ▪ **[0]** (Check box cleared) = Disabled (default) ▪ **[1]** (Check box selected) = Enabled
Web: IP Address EMS: Address CLI: ip-address **[SNMPManagerTableIP_x]**	Defines the IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to this IP address. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255.
Web: Trap Port EMS: Port CLI: port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to this port. The valid SNMP trap port range is 100 to 4000. The default port is 162.
Web: Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Enables the sending of traps to the corresponding SNMP manager. ▪ **[0]** Disable = Sending is disabled. ▪ **[1]** Enable = Sending is enabled (default).
CLI: send-trap **[SNMPManagerTrapUser_x]**	This parameter can be set to the name of any configured SNMPV3 user to associate with this trap destination. This determines the trap format, authentication level, and encryption level. By default, the trap is associated with the SNMP trap community string.
Web: Trap Manager Host Name CLI: manager-host-name **[SNMPTrapManagerHostName]**	Defines an FQDN of a remote host that is used as an SNMP manager. The resolved IP address replaces the last entry in the Trap Manager table (defined by the parameter SNMPManagerTableIP_x) and the last trap manager entry of snmpTargetAddrTable in the snmpTargetMIB. For example: 'mngr.corp.mycompany.com'. The valid range is a 99-character string.
SNMP Community String Parameters	

Parameter	Description
Community String CLI: ro-community-string **[SNMPReadOnlyCommunityString_x]**	Defines up to five read-only SNMP community strings (up to 19 characters each). The default string is 'public'.
Community String CLI: rw-community-string **[SNMPReadWriteCommunityString_x]**	Defines up to five read/write SNMP community strings (up to 19 characters each). The default string is 'private'.
Trap Community String CLI: community-string **[SNMPTrapCommunityString]**	Defines the Community string used in traps (up to 19 characters). The default string is 'trapuser'.
Web: SNMP V3 Table **EMS: SNMP V3 Users**	
[SNMPUsers]	This *para*meter table defines SNMP v3 users. The format of this parameter is as follows: **[**SNMPUsers] FORMAT SNMPUsers_Index = SNMPUsers_Username, SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group; [\SNMPUsers] For example: SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1; The example above configures user 'v3admin1' with security level authNoPriv(2), authentication protocol MD5, authentication text password 'myauthkey', and ReadWriteGroup2. **Notes:** ▪ This parameter can include up to 10 indices. ▪ For a description of this table's individual parameters and for configuring the table using the Web interface, see 'Configuring SNMP V3 Users' on page 92. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98

A.2.5 CLI Parameters

The command-line interface (CLI) parameters are described in the table below.

Table A-13: CLI Parameters

Parameter	Description
CLI: aaa authentication login tacacs+ **[TacPlusEnable]**	Enables the Terminal Access Controller Access-Control System (TACACS+) remote authentication protocol and user authentication for CLI login. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled **Note:** For this parameter to take effect, a device reset is required.
CLI: tacacs-server host <host-ip> **[TacPlusServerIP]**	Defines the IP address (in dotted-decimal notation) of the TACACS+ primary authentication server.
CLI: tacacs-server host <host-ip> **[TacPlusSecondaryServerIP]**	Defines the IP address (in dotted-decimal notation) of the TACACS+ secondary authentication server.
CLI: tacacs-server port <port-num> **[TacPlusPort]**	Defines the TACACS+ authentication port (UDP) for authenticating with the RADIUS server. The valid value range is 1 to 15. The default is 49.
CLI: tacacs-server timeout <seconds> **[TacPlusTimeout]**	Defines the TACACS+ response timeout (in seconds). If no response is received within this period, retransmission is required. The valid value range is 1 to 15. The default is 5.
CLI: tacacs-server key <password> **[TacPlusSharedSecretand]**	Defines the TACACS+ shared secret between client and server. The valid value can be a string of up to 64 characters. The default value is “msbg”.

A.2.6 Serial Parameters

The RS-232 serial parameters are described in the table below.

Table A-14: Serial Parameters

Parameter	Description
[DisableRS232]	Enables the device's RS-232 (serial) port. ▪ **[0]** = Enabled ▪ **[1]** = Disabled (default) The RS-232 serial port can be used to change the networking parameters and view error/notification messages. For how to establish a serial communication with the device, refer to the *Installation Manual*. **Note:** For this parameter to take effect, a device reset is required.
EMS: Baud Rate **[SerialBaudRate]**	Defines the RS-232 baud rate. The valid values include the following: 1200, 2400, 9600, 14400, 19200, 38400, 57600, or 115200 (default). **Note:** For this parameter to take effect, a device reset is required.
EMS: Data **[SerialData]**	Defines the RS-232 data bit. ▪ **[7]** = 7-bit. ▪ **[8]** = 8-bit (default). **Note:** For this parameter to take effect, a device reset is required.
EMS: Parity **[SerialParity]**	Defines the RS-232 polarity. ▪ **[0]** = None (default). ▪ **[1]** = Odd. ▪ **[2]** = Even. **Note:** For this parameter to take effect, a device reset is required.
EMS: Stop **[SerialStop]**	Defines the RS-232 stop bit. ▪ **[1]** = 1-bit (default). ▪ **[2]** = 2-bit. **Note:** For this parameter to take effect, a device reset is required.
EMS: Flow Control **[SerialFlowControl]**	Defines the RS-232 flow control. ▪ **[0]** = None (default). ▪ **[1]** = Hardware. **Note:** For this parameter to take effect, a device reset is required.

A.3 Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

A.3.1 General Parameters

The general debugging and diagnostic parameters are described in the table below.

Table A-15: General Debugging and Diagnostic Parameters

Parameter	Description
CLI: enablesecsyslog **[EnableSecSyslog]**	Enables the reporting of security-related events for the data-router networking. When enabled, the data-router access list rules, configured using the access-list CLI command, which are set to "log", send Syslog messages whenever traffic matching the access list is encountered. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled
[EnableAutoRAITransmitBER]	Enables the device to send RAI when the bit error rate (BER) is above 0.001. ▪ **[0]** Disable (default) ▪ **[1]** Enable
EMS: Enable Diagnostics **[EnableDiagnostics]**	Determines the method for verifying correct functioning of the different hardware components on the device. On completion of the check and if the test fails, the device sends information on the test results of each hardware component to the Syslog server. ▪ **[0]** = Rapid and Enhanced self-test mode (default). ▪ **[1]** = Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY and Flash). ▪ **[2]** = A quicker version of the Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY, but partial test of Flash). For more information, refer to the *Product Reference Manual*. **Note:** For this parameter to take effect, a device reset is required.
Web: Enable LAN Watchdog **[EnableLanWatchDog]**	Enables the LAN watchdog feature**.** ▪ **[0]** Disable (default). ▪ **[1]** Enable. When LAN watchdog is enabled, the device's overall communication integrity is checked periodically. If no communication is detected for about three minutes, the device performs a self test: ▪ If the self-test succeeds, the problem is a logical link down (i.e., Ethernet cable disconnected on the switch side) and the Busy Out mechanism is activated if enabled (i.e., the parameter EnableBusyOut is set to 1). ▪ If the self-test fails, the device restarts to overcome internal fatal communication error. **Notes:**

Parameter	Description
	▪ For this parameter to take effect, a device reset is required. ▪ Enable LAN watchdog is relevant only if the Ethernet connection is full duplex.
[LifeLineType]	Defines the scenario upon which the Lifeline analog (FXS) feature is activated. The Lifeline feature can be activated upon a power outage or network failure (i.e., loss of IP connectivity). Upon any of these scenarios, the Lifeline feature provides PSTN connectivity (and call continuity) for the FXS phone users. The Lifeline (FXS) phone is connected to the following port: ▪ FXS Port 1 of each FXS module FXS Port 1 connects to the POTS (Lifeline) phone as well as to the PSTN / PBX, using a splitter cable. ▪ **[0]** = Lifeline is activated upon power outage (default). ▪ **[1]** = Lifeline is activated upon power outage. ▪ **[2]** = Lifeline is activated upon a power outage or upon network failure (logical link disconnected). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces. ▪ To enable Lifeline upon a network failure, the LAN watch dog must be activated (i.e., set the parameter EnableLANWatchDog to 1). ▪ A Lifeline phone connection can be setup for each FXS module (using Port I) housed in the chassis. ▪ For information on Lifeline cabling, refer to the Installation Manual.
Web: Delay After Reset **[sec]** CLI: delay-after-reset **[GWAppDelayTime]**	Defines the time interval (in seconds) that the device's operation is delayed after a reset. The valid range is 0 to 45. The default value is 7 seconds. **Note:** This feature helps overcome connection problems caused by some LAN routers or IP configuration parameters' modifications by a DHCP server.
[Mediant1000DualPowerSupplySupported]	Determines whether the device sends raised alarms (to the SNMP client and/or Web interface) concerned with the Power Supply modules. ▪ **[1]** (default) = No alarms are sent. ▪ **[2]** = The device sends alarms if one of the Power Supply modules is removed from the chassis. These alarms are reflected in the SNMP and Web interface. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If this parameter is set to 2 and for this feature to be functional, both Main and Redundant Power Supply modules must be present in the chassis.

A.3.2 Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

Table A-16: Syslog, CDR and Debug Parameters

Parameter	Description
Web: Enable Syslog EMS: Syslog enable CLI: syslog **[EnableSyslog]**	Determines whether the device sends logs and error messages generated by the device to a Syslog server. ▪ **[0]** Disable = Logs and errors are not sent to the Syslog server (default). ▪ **[1]** Enable = Enables the Syslog server. **Notes:** ▪ If you enable Syslog, you must enter an IP address of the Syslog server (using the SyslogServerIP parameter). ▪ Syslog messages may increase the network traffic. ▪ To configure Syslog SIP message logging levels, use the GwDebugLevel parameter. ▪ For more information on Syslog, refer to the *Product Reference Manual*. ▪ By default, logs are also sent to the RS-232 serial port. For how to establish serial communication with the device, refer to the Installation Manual.
Web/EMS: Syslog Server IP Address CLI: syslog-ip **[SyslogServerIP]**	Defines the IP address (in dotted-decimal notation) of the computer on which the Syslog server is running. The Syslog server is an application designed to collect the logs and error messages generated by the device. Default IP address is 0.0.0.0. For information on Syslog, refer to the *Product Reference Manual*.
Web: Syslog Server Port EMS: Syslog Server Port Number CLI: syslog-port **[SyslogServerPort]**	Defines the UDP port of the Syslog server. The valid range is 0 to 65,535. The default port is 514. For information on Syslog, refer to the *Product Reference Manual*.
CLI: mx-syslog-lgth **[MaxBundleSyslogLength]**	Defines the maximum size (in bytes) threshold of logged Syslog messages bundled into a single UDP packet, after which they are sent to a Syslog server. The valid value range is 0 to 1220 (where 0 indicates that no bundling occurs). The default is 1220. **Note:** This parameter is applicable only if the GWDebugLevel parameter is set to 7.
Web: CDR Server IP Address EMS: IP Address of CDR Server CLI: cdr-srvr-ip-adrr **[CDRSyslogServerIP]**	Defines the destination IP address to where CDR logs are sent. The default value is a null string, which causes CDR messages to be sent with all Syslog messages to the Syslog server. **Notes:** ▪ The CDR messages are sent to UDP port 514 (default Syslog port). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web/EMS: CDR Report Level CLI: cdr-report-level	Determines whether Call Detail Records (CDR) are sent to the Syslog server and when they are sent. ▪ **[0]** None = CDRs are not used (default).

Parameter	Description
[CDRReportLevel]	▪ **[1]** End Call = CDR is sent to the Syslog server at the end of each call. ▪ **[2]** Start & End Call = CDR report is sent to Syslog at the start and end of each call. ▪ **[3]** Connect & End Call = CDR report is sent to Syslog at connection and at the end of each call. ▪ **[4]** Start & End & Connect Call = CDR report is sent to Syslog at the start, at connection, and at the end of each call. **Notes:** ▪ The CDR Syslog message complies with RFC 3161 and is identified by: Facility = 17 (local1) and Severity = 6 (Informational). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web/EMS: Debug Level CLI: debug-level **[GwDebugLevel]**	Defines the Syslog debug logging level. ▪ **[0]** 0 (default) = Debug is disabled. ▪ **[1]** 1 = Flow debugging is enabled. ▪ **[5]** 5 = Flow, device interface, stack interface, session manager, and device interface expanded debugging are enabled. ▪ **[7]** 7 = This option is recommended when the device is running under "heavy" traffic. In this mode: ✓ The Syslog debug level automatically changes between level 5, level 1, and level 0, depending on the device's CPU consumption so that VoIP traffic isn't affected. ✓ Syslog messages are bundled into a single UDP packet, after which they are sent to a Syslog server (bundling size is determined by the MaxBundleSyslogLength parameter). Bundling reduces the number of UDP Syslog packets, thereby improving CPU utilization. Note that when this option is used, in order to read Syslog messages with Wireshark, a special plug-in (i.e., acsyslog.dll) must be used. Once the plug-in is installed, the Syslog messages are decoded as "AC SYSLOG" and are dispalyed using the 'acsyslog' filter instead of the regular 'syslog' filter. **Notes:** ▪ This parameter is typically set to 5 if debug traces are required. However, in cases of heavy traffic, option 7 is recommended. ▪ Options 2, 3, 4, and 6 are not recommended.
Syslog Facility Number **[SyslogFacility]**	Defines the Facility level (0 through 7) of the device's Syslog messages, according to RFC 3164. This allows you to identify Syslog messages generated by the device. This is useful, for example, if you collect the device's and other equipments' Syslog messages, at one single server. The device's Syslog messages can easily be identified and distinguished from other Syslog messages by its Facility level. Therefore, in addition to filtering Syslog messages according to IP address, the messages can be filtered according to Facility level. ▪ **[16]** = local use 0 (local0) - default ▪ **[17]** = local use 1 (local1) ▪ **[18]** = local use 2 (local2) ▪ **[19]** = local use 3 (local3) ▪ **[20]** = local use 4 (local4) ▪ **[21]** = local use 5 (local5) ▪ **[22]** = local use 6 (local6)

Parameter	Description
	▪ **[23]** = local use 7 (local7)
Web: Activity Types to Report via Activity Log Messages **[ActivityListToLog]**	Defines the Activity Log mechanism of the device, which sends log messages (to a Syslog server) for reporting certain types of Web operations according to the below user-defined filters. ▪ **[pvc]** Parameters Value Change = Changes made on-the-fly to parameters. ▪ **[afl]** Auxiliary Files Loading = Loading of auxiliary files. ▪ **[dr]** Device Reset = Reset of device via the 'Maintenance Actions page. **Note:** For this option to take effect, a device reset is required. ▪ **[fb]** Flash Memory Burning = Burning of files or parameters to flash (in 'Maintenance Actions page). ▪ **[swu]** Device Software Update = cmp file loading via the Software Upgrade Wizard. ▪ **[ard]** Access to Restricted Domains = Access to restricted domains, which include the following Web pages: ✓ (1) ini parameters (AdminPage) ✓ (2) General Security Settings ✓ (3) Configuration File ✓ ✓ (5) Software Upgrade Key Status ✓ ✓ (7) Web & Telnet Access List ✓ (8) WEB User Accounts ▪ **[naa]** Non-Authorized Access = Attempt to access the Web interface with a false or empty user name or password. ▪ **[spc]** Sensitive Parameters Value Change = Changes made to sensitive parameters: ✓ (1) IP Address ✓ (2) Subnet Mask ✓ (3) Default Gateway IP Address ✓ (4) ActivityListToLog ▪ **[ll]** Login and Logout = Every login and logout attempt. For example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc' **Note:** For the *ini* file, values must be enclosed in single quotation marks.
CLI: isdn-facility-trace **[FacilityTrace]**	Enables ISDN traces of Facility Information Elements (IE) for ISDN call diagnostics. This allows you to trace all the parameters contained in the Facility IE and view them in the Syslog. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this feature to be functional, the GWDebugLevel parameter must be enabled (i.e., set to at least level 1).

A.3.3 Resource Allocation Indication Parameters

The Resource Allocation Indication (RAI) parameters are described in the table below.

Table A-17: RAI Parameters

Parameter	Description
[EnableRAI]	Enables RAI alarm generation if the device's busy endpoints exceed a user-defined threshold. ▪ **[0]** = Disable RAI (Resource Available Indication) service (default). ▪ **[1]** = RAI service enabled and an SNMP 'acBoardCallResourcesAlarm' Alarm Trap is sent.
[RAIHighThreshold]	Defines the high threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints exceeds this high threshold, the device sends the SNMP acBoardCallResourcesAlarm alarm trap with a 'major' alarm status. The range is 0 to 100. The default value is 90. **Note:** The percentage of busy endpoints is calculated by dividing the number of busy endpoints by the total number of “enabled” endpoints (trunks are physically connected and synchronized with no alarms and endpoints are defined in the Trunk Group Table).
[RAILowThreshold]	Defines the low threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints falls below this low threshold, the device sends an SNMP acBoardCallResourcesAlarm alarm trap with a 'cleared' alarm status. The range is 0 to 100%. The default value is 90%.
[RAILoopTime]	Defines the time interval (in seconds) that the device periodically checks call resource availability. The valid range is 1 to 200. The default is 10.

A.4 Security Parameters

This subsection describes the device's security parameters.

A.4.1 General Parameters

The general security parameters are described in the table below.

Table A-18: General Security Parameters

Parameter	Description
Web: Internal Firewall Parameters **EMS: Firewall Settings** CLI: configure voip > access-list	
[AccessList]	This *para*meter table defines the device's access list (firewall), which defines network traffic filtering rules. For each packet received on the network interface, the table is scanned from the top down until a matching rule is found. This rule can either deny (block) or permit (allow) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted. The format of this parameter is as follows: [AccessList] FORMAT AccessList_Index = AccessList_Source_IP, AccessList_Source_Port, AccessList_PrefixLen, AccessList_Source_Port, AccessList_Start_Port, AccessList_End_Port, AccessList_Protocol, AccessList_Use_Specific_Interface, AccessList_Interface_ID, AccessList_Packet_Size, AccessList_Byte_Rate, AccessList_Byte_Burst, AccessList_Allow_Type; [\AccessList] For example: AccessList 10 = mgmt.customer.com, , , 32, 0, 80, tcp, 1, OAMP, 0, 0, 0, allow; AccessList 22 = 10.4.0.0, , , 16, 4000, 9000, any, 0, , 0, 0, 0, block; In the example above, Rule #10 allows traffic from the host 'mgmt.customer.com' destined to TCP ports 0 to 80 on interface OAMP (OAMP). Rule #22 blocks traffic from the subnet 10.4.xxx.yyy destined to ports 4000 to 9000. **Notes:** ▪ This parameter can include up to 50 indices. ▪ To configure the firewall using the Web interface and for a description of the parameters of this *ini* file table parameter, see 'Configuring Firewall Settings' on page 145. ▪ For a description of configuring with *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.4.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

Table A-19: HTTPS Parameters

Parameter	Description
Web: Secured Web Connection (HTTPS) EMS: HTTPS Only CLI: secured-connection **[HTTPSOnly]**	Determines the protocol used to access the Web interface. ▪ **[0]** HTTP and HTTPS (default). ▪ **[1]** HTTPs Only = Unencrypted HTTP packets are blocked. **Note:** For this parameter to take effect, a device reset is required.
EMS: HTTPS Port CLI: https-port **[HTTPSPort]**	Defines the local Secured HTTPS port of the device. This parameter allows secure remote device Web management from the LAN. To enable secure Web management from the LAN, configure the desired port. The valid range is 1 to 65535 (other restrictions may apply within this range). The default port is 443. **Note:** For this parameter to take effect, a device reset is required.
EMS: HTTPS Cipher String CLI: https-cipher-string **[HTTPSCipherString]**	Defines the Cipher string for HTTPS (in OpenSSL cipher list format). For the valid range values, refer to URL http://www.openssl.org/docs/apps/ciphers.html. The default value is 'EXP' (Export encryption algorithms). For example, use 'ALL' for all ciphers suites (e.g., for ARIA encryption for TLS). The only ciphers available are RC4 and DES, and the cipher bit strength is limited to 56 bits. **Notes:** ▪ If the "Strong Encryption" Software Upgrade Key is enabled, the default of the HTTPSCipherString parameter is changed to 'RC4:EXP', enabling RC-128bit encryption. ▪ The value 'ALL' can be configured only if the "Strong Encryption" Software Upgrade Key is enabled.
Web: HTTP Authentication Mode EMS: Web Authentication Mode CLI: http-auth-mode **[WebAuthMode]**	Determines the authentication mode used for the Web interface. ▪ **[0]** Basic Mode = Basic authentication (clear text) is used (default). ▪ **[1]** Digest When Possible = Digest authentication (MD5) is used. ▪ **[2]** Basic if HTTPS, Digest if HTTP = Digest authentication (MD5) is used for HTTP, and basic authentication is used for HTTPS. **Note:** When RADIUS login is enabled (i.e., the parameter WebRADIUSLogin is set to 1), basic authentication is forced.
CLI: req-client-cert **[HTTPSRequireClientCertificate]**	Determines whether client certificates are required for HTTPS connection. ▪ **[0]** = Client certificates are not required (default). ▪ **[1]** = Client certificates are required. The client certificate must be preloaded to the device and its matching private key must be installed on the managing PC. Time and date must be correctly set on the device for the client certificate to be

Parameter	Description
	verified. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a description on implementing client certificates, see 'Client Certificates' on page 109.

A.4.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

Table A-20: SRTP Parameters

Parameter	Description
Web: Media Security EMS: Enable Media Security CLI: media-security-enable **[EnableMediaSecurity]**	Enables Secure Real-Time Transport Protocol (SRTP). ▪ **[0]** Disable = SRTP is disabled (default). ▪ **[1]** Enable = SRTP is enabled. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Media Security Behavior **[MediaSecurityBehaviour]**	Determines the device's mode of operation when SRTP is used (i.e., when the parameter EnableMediaSecurity is set to 1). ▪ **[0]** Preferable = The device initiates encrypted calls. However, if negotiation of the cipher suite fails, an unencrypted call is established. Incoming calls that don't include encryption information are accepted. (default) ▪ **[1]** Mandatory = The device initiates encrypted calls, but if negotiation of the cipher suite fails, the call is terminated. Incoming calls that don't include encryption information are rejected. ▪ **[2]** Disable = The IP Profile for which this parameter is set does not support encrypted calls (i.e., SRTP). ▪ **[3]** Preferable - Single Media = The device sends SDP with a single media ('m=') line only (e.g., m=audio 6000 RTP/AVP 4 0 70 96) with RTP/AVP and crypto keys. The remote UA can respond with SRTP or RTP parameters: ✓ If the remote SIP UA does not support SRTP, it uses RTP and ignores the crypto lines. ✓ In the opposite direction, if the device receives an SDP offer with a single media (as shown above), it responds with SRTP (RTP/SAVP) if the EnableMediaSecurity parameter is set to 1. If SRTP is not supported (i.e., EnableMediaSecurity is set to 0), it responds with RTP. **Notes:** ▪ Before configuring this parameter, set the EnableMediaSecurity parameter to 1. ▪ Option **[2]** Disable is applicable only to IP Profiles. ▪ If this parameter is set to Preferable [3] and two 'm=' lines are received in the SDP offer, the device prefers the SAVP (secure audio video profile) regardless of the order in the SDP. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: Master Key Identifier	Defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP

Parameter	Description
(MKI) Size EMS: Packet MKI Size CLI: SRTP-tx-packet-MKI-size **[SRTPTxPacketMKISize]**	Tx packets. The range is 0 to 4. The default value is 0.

CLI: symmetric-mki
[EnableSymmetricMKI]
	Enables symmetric MKI negotiation.
	▪ **[0]** = Disabled (default) - the device includes the MKI in its 200 OK response according to the SRTPTxPacketMKISize parameter (if set to 0, then it is not included; if set to any other value, it is included with this value).
	▪ **[1]** = Enabled - the answer crypto line contains (or excludes) an MKI value according to the selected crypto line in the offer. For example, assume that the device receives an INVITE containing the following two crypto lines in SDP:
	`a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:TAaxNnQt8/qLQMnDuG4vxYfWl6K7eBK/ufk04pR4|2^31|1:1`
	`a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:bnuYZnMxSfUiGitviWJZmzr7OF3AiRO0l5Vnh0kH|2^31`
	The first crypto line includes the MKI parameter "1:1". In the 200 OK response, the device selects one of the crypto lines (i.e., '2' or '3'). If it selects crypto line '2', it includes the MKI parameter in its answer SDP, for example:
	`a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:R1VyA1xV/qwBjkEklu4kSJyl3wCtYeZLq1/QFuxw|2^31|1:1`
	If the device selects a crypto line that does not contain the MKI parameter, then the MKI parameter is not included in the crypto line in the SDP answer (even if the SRTPTxPacketMKISize parameter is set to any value other than 0).
	Note: To enable symmetric MKI, the SRTPTxPacketMKISize parameter must be set to any value other than 0.

Web/EMS: SRTP offered Suites CLI: offer-srtp-cipher **[SRTPofferedSuites]**	Defines the offered crypto suites (cipher encryption algorithms) for SRTP. ▪ **[0]** = All available crypto suites (default) ▪ **[1]** CIPHER SUITES AES CM 128 HMAC SHA1 80 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 80-bit tag. ▪ **[2]** CIPHER SUITES AES CM 128 HMAC SHA1 32 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag. **Note:** This parameter also affects the selection of the crypto in the device's answer. For example, if the device receives an offer with two crypto lines containing HMAC_SHA1_80 and HMAC_SHA_32, it uses the HMAC_SHA_32 key in its SIP 200 OK response if the parameter is set to 2.
Web: Disable Authentication On Transmitted RTP Packets EMS: RTP AuthenticationDisable Tx CLI: RTP-authentication-	Enables authentication on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable

Parameter	Description
disable-tx **[RTPAuthenticationDisableTx]**	
Web: Disable Encryption On Transmitted RTP Packets EMS: RTP EncryptionDisable Tx CLI: RTP-encryption-disable-tx **[RTPEncryptionDisableTx]**	Enables encryption on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTCP Packets EMS: RTCP EncryptionDisable Tx CLI: RTCP-encryption-disable-tx **[RTCPEncryptionDisableTx]**	Enables encryption on transmitted RTCP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable

A.4.4 TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

Table A-21: TLS Parameters

Parameter	Description
Web/EMS: TLS Version CLI: version **[TLSVersion]**	Determines the supported versions of SSL/TLS (Secure Socket Layer/Transport Layer Security. ▪ **[0]** SSL 2.0-3.0 and TLS 1.0 = SSL 2.0, SSL 3.0, and TLS 1.0 are supported (default). ▪ **[1]** TLS 1.0 Only = only TLS 1.0 is used. When set to 0, SSL/TLS handshakes always start with SSL 2.0 and switch to TLS 1.0 if both peers support it. When set to 1, TLS 1.0 is the only version supported; clients attempting to contact the device using SSL 2.0 are rejected. **Note:** For this parameter to take effect, a device reset is required.
Web: TLS Client Re-Handshake Interval EMS: TLS Re Handshake Interval CLI: tls-re-hndshk-int **[TLSReHandshakeInterval]**	Defines the time interval (in minutes) between TLS Re-Handshakes initiated by the device. The interval range is 0 to 1,500 minutes. The default is 0 (i.e., no TLS Re-Handshake).
Web: TLS Mutual Authentication EMS: SIPS Require Client Certificate **[SIPSRequireClientCertificate]**	Determines the device's behavior when acting as a server for TLS connections. ▪ **[0]** Disable = The device does not request the client certificate (default). ▪ **[1]** Enable = The device requires receipt and verification of the client certificate to establish the TLS connection. **Notes:**

Parameter	Description
	▪ For this parameter to take effect, a device reset is required. ▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName.
Web/EMS: Peer Host Name Verification Mode **[PeerHostNameVerificationMode]**	Determines whether the device verifies the Subject Name of a remote certificate when establishing TLS connections. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Server Only = Verify Subject Name only when acting as a server for the TLS connection. ▪ **[2]** Server & Client = Verify Subject Name when acting as a server or client for the TLS connection. When a remote certificate is received and this parameter is not disabled, the value of SubjectAltName is compared with the list of available Proxies. If a match is found for any of the configured Proxies, the TLS connection is established. The comparison is performed if the SubjectAltName is either a DNS name (DNSName) or an IP address. If no match is found and the SubjectAltName is marked as 'critical', the TLS connection is not established. If DNSName is used, the certificate can also use wildcards ('*') to replace parts of the domain name. If the SubjectAltName is not marked as 'critical' and there is no match, the CN value of the SubjectName field is compared with the parameter TLSRemoteSubjectName. If a match is found, the connection is established. Otherwise, the connection is terminated.
Web: TLS Client Verify Server Certificate EMS: Verify Server Certificate CLI: tls-vrfy-srvr-cert **[VerifyServerCertificate]**	Determines whether the device, when acting as a client for TLS connections, verifies the Server certificate. The certificate is verified with the Root CA information. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** If Subject Name verification is necessary, the parameter PeerHostNameVerificationMode must be used as well.
Web/EMS: TLS Remote Subject Name CLI: tls-rmt-subs-name **[TLSRemoteSubjectName]**	Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections. If the SubjectAltName of the received certificate is not equal to any of the defined Proxies Host names/IP addresses and is not marked as 'critical', the Common Name (CN) of the Subject field is compared with this value. If not equal, the TLS connection is not established. If the CN uses a domain name, the certificate can also use wildcards ('*') to replace parts of the domain name. The valid range is a string of up to 49 characters. **Note:** This parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2.

Parameter	Description
Web: Client Cipher String CLI: client-cipher-string **[TLSClientCipherString]**	Defines the cipher-suite string for TLS clients. The valid value is up to 255 strings. The default is "ALL:!ADH". For example: TLSClientCipherString = 'EXP' This parameter complements the HTTPSCipherString parameter (which affects TLS servers). For possible values and additional details, refer to: http://www.openssl.org/docs/apps/ciphers.html
CLI: pkey-size **[TLSPkeySize]**	Defines the key size (in bits) for RSA public-key encryption for newly self-signed generated keys for SSH. ▪ **[512]** ▪ **[768]** ▪ **[1024]** (default) ▪ **[2048]**

A.4.5 SSH Parameters

Secure Shell (SSH) parameters are described in the table below.

Table A-22: SSH Parameters

Parameter	Description
Web/EMS: SSH Server Enable CLI: ssh **[SSHServerEnable]**	Enables the device's embedded SSH server. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: SSH Server Port cli: ssh-port **[SSHServerPort]**	Defines the port number for the embedded SSH server. Range is any valid port number. The default port is 22.
CLI: ssh-admin-key **[SSHAdminKey]**	Defines the RSA public key for strong authentication for logging in to the SSH interface (if enabled). The value should be a base64-encoded string. The value can be a maximum length of 511 characters. For more information, refer to the *Product Reference Manual*.
CLI: ssh-max-login-attempts **[SSHMaxLoginAttempts]**	Defines the maximum SSH login attempts allowed for entering an incorrect password by an administrator before the SSH session is rejected. The valid range is 1 to 3. the default is 3.
CLI: ssh-last-login-message **[SSHEnableLastLoginMessage]**	Enables message display in SSH sessions of the time and date of the last SSH login. The SSH login message displays the number of unsuccessful login attempts since the last successful login. ▪ **[0]** Disable ▪ **[1]** Enable (default) **Note:** The last SSH login information is cleared when the device is reset.
CLI: ssh-max-sessions **[SSHMaxSessions]**	Defines the maximum number of simultaneous SSH sessions. The valid range is 1 to 2. The default is 2 sessions.
CLI: ssh-require-public-key **[SSHRequirePublicKey]**	Enables RSA public keys for SSH. ▪ **[0]** = RSA public keys are optional if a value is configured for

Parameter	Description
	the parameter SSHAdminKey (default). ▪ **[1]** = RSA public keys are mandatory. **Note:** To define the key size, use the TLSPkeySize parameter.

A.4.6 OCSP Parameters

The Online Certificate Status Protocol (OCSP) parameters are described in the table below.

Table A-23: OCSP Parameters

Parameter	Description
Web: Enable OCSP Server EMS: OCSP Enable CLI: enable **[OCSPEnable]**	Enables or disables certificate checking using OCSP. ▪ **[0]** Disable (default). ▪ **[1]** Enable. For a description of OCSP, refer to the *Product Reference Manual*.
Web: Primary Server IP EMS: OCSP Server IP CLI: server-ip **[OCSPServerIP]**	Defines the IP address of the OCSP server. The default IP address is 0.0.0.0.
Web: Secondary Server IP CLI: secondary-server-ip **[OCSPSecondaryServerIP]**	Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default IP address is 0.0.0.0.
Web: Server Port EMS: OCSP Server Port CLI: server-port **[OCSPServerPort]**	Defines the OCSP server's TCP port number. The default port number is 2560.
Web: Default Response When Server Unreachable EMS: OCSP Default Response CLI: default-response **[OCSPDefaultResponse]**	Determines the default OCSP behavior when the server cannot be contacted. ▪ **[0]** Disable = Rejects peer certificate (default). ▪ **[1]** Enable = Allows peer certificate.

A.5 RADIUS Parameters

The RADIUS parameters are described in the table below. For supported RADIUS attributes, see 'Supported RADIUS Attributes' on page 739.

Table A-24: RADIUS Parameters

Parameter	Description
Web: Enable RADIUS Access Control CLI: enable **[EnableRADIUS]**	Enables the RADIUS application. ▪ **[0]** Disable = RADIUS application is disabled (default). ▪ **[1]** Enable = RADIUS application is enabled. **Note:** For this parameter to take effect, a device reset is required.
Web: Accounting Server IP	Defines the IP address of the RADIUS accounting server.

Parameter	Description
Address CLI: accounting-server-ip **[RADIUSAccServerIP]**	
Web: Accounting Port CLI: accounting-port **[RADIUSAccPort]**	Defines the port of the RADIUS accounting server. The default value is 1646.
Web/EMS: RADIUS Accounting Type CLI: radius-accounting **[RADIUSAccountingType]**	Determines when the RADIUS accounting messages are sent to the RADIUS accounting server. ▪ **[0]** At Call Release = Sent at call release only (default). ▪ **[1]** At Connect & Release = Sent at call connect and release. ▪ **[2]** At Setup & Release = Sent at call setup and release.
Web: AAA Indications EMS: Indications CLI: aaa-indications **[AAAIndications]**	Determines the Authentication, Authorization and Accounting (AAA) indications. ▪ **[0]** None = No indications (default). ▪ **[3]** Accounting Only = Only accounting indications are used.
Web: Device Behavior Upon RADIUS Timeout CLI: timeout-behavior **[BehaviorUponRadiusTimeout]**	Defines the device's response upon a RADIUS timeout. ▪ **[0]** Deny Access = Denies access. ▪ **[1]** Verify Access Locally = Checks password locally (default).
[MaxRADIUSSessions]	Defines the number of concurrent calls that can communicate with the RADIUS server (optional). The valid range is 0 to 240. The default value is 240.
[RADIUSRetransmission]	Defines the number of retransmission retries. The valid range is 1 to 10. The default value is 3.
[RadiusTO]	Defines the time interval (measured in seconds) that the device waits for a response before a RADIUS retransmission is issued. The valid range is 1 to 30. The default value is 10.
Web: RADIUS Authentication Server IP Address CLI: auth-server-ip **[RADIUSAuthServerIP]**	Defines the IP address of the RADIUS authentication server. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Authentication Server Port CLI: auth-server-port **[RADIUSAuthPort]**	Defines the port of the RADIUS Authentication Server. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Shared Secret CLI: shared-secret **[SharedSecret]**	Defines the 'Secret' used to authenticate the device to the RADIUS server. This should be a cryptically strong password.
Web: Default Access Level CLI: default-access-level **[DefaultAccessLevel]**	Defines the default access level for the device when the RADIUS (authentication) response doesn't include an access level attribute. The valid range is 0 to 255. The default value is 200 (Security Administrator').
Web: Local RADIUS Password Cache Mode CLI: local-cache-mode **[RadiusLocalCacheMode]**	Determines the device's mode of operation regarding the timer (configured by the parameter RadiusLocalCacheTimeout) that determines the validity of the user name and password (verified by the RADIUS server).

Parameter	Description
	▪ **[0]** Absolute Expiry Timer = when you access a Web page, the timeout doesn't reset, instead it continues decreasing. ▪ **[1]** Reset Timer Upon Access = upon each access to a Web page, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).
Web: Local RADIUS Password Cache Timeout CLI: local-cache-timeout **[RadiusLocalCacheTimeout]**	Defines the time (in seconds) the locally stored user name and password (verified by the RADIUS server) are valid. When this time expires, the user name and password become invalid and a must be re-verified with the RADIUS server. The valid range is 1 to 0xFFFFFF. The default value is 300 (5 minutes). ▪ **[-1]** = Never expires. ▪ **[0]** = Each request requires RADIUS authentication.
Web: RADIUS VSA Vendor ID CLI: vsa-vendor-id **[RadiusVSAVendorID]**	Defines the vendor ID that the device accepts when parsing a RADIUS response packet. The valid range is 0 to 0xFFFFFFFF. The default value is 5003.
Web: RADIUS VSA Access Level Attribute CLI: vsa-access-level **[RadiusVSAAccessAttribute]**	Defines the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet. The valid range is 0 to 255. The default value is 35.

A.6 SIP Media Realm Parameters

The Media Realm parameters are described in the table below.

Table A-25: Media Realm Parameters

Parameter	Description
Media Realm Table	
Web: Media Realm Table **EMS: Protocol Definition > Media Realm** CLI: configure voip > media realm **[CpMediaRealm]**	This *para*meter table defines the Media Realm table. The Media Realm table allows you to divide a Media-type interface (defined in the Multiple Interface table) into several realms, where each realm is specified by a UDP port range. The format of this parameter is as follows: [CpMediaRealm] FORMAT CpMediaRealm_Index = CpMediaRealm_MediaRealmName, CpMediaRealm_IPv4IF, CpMediaRealm_IPv6IF, CpMediaRealm_PortRangeStart, CpMediaRealm_MediaSessionLeg, CpMediaRealm_PortRangeEnd, CpMediaRealm_TransRateRatio, CpMediaRealm_IsDefault; [\CpMediaRealm] For example, CpMediaRealm 1 = Mrealm1, Voice, , 6600, 20, 6790, , 1; CpMediaRealm 2 = Mrealm2, Voice, , 6800, 10, 6890; , 0; **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This table can include up to 64 indices (where 0 is the first index). ▪ Each table index must be unique. ▪ A Media Realm can be assigned to an IP Group (in the IP Group

Parameter	Description
	table) or an SRD (in the SRD table). If different Media Realms are assigned to both an IP Group and SRD, the IP Group's Media Realm takes precedence. ▪ The parameter IPv6IF is not applicable. ▪ For a detailed description of all the parameters included in this *ini* file table parameter and for configuring Media Realms using the Web interface, see 'Configuring Media Realms' on page 181. ▪ For a description on configuring ini file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.7 Quality of Experience Reporting

The Quality of Experience parameters are described in the table below.

Table A-26: Quality of Experience Parameters

Parameter	Description
CLI: server-ip **[QOEServerIP]**	Defines the IP address of the Session Experience Manager (SEM) server. **Note:** For this parameter to take effect, a device reset is required.
[QOEPort]	Defines the port of the SEM server. The valid value range is 0 to 65534. The default is 5000.
[QOEInterfaceName]	Defines the IP network interface on which the quality experience reports are sent. The default is "DEFAULT". **Note:** For this parameter to take effect, a device reset is required
CLI: use-mos-lq **[QOEUseMosLQ]**	Enables the reporting of the MOS-LQ (listening quality). If disabled, the MOS-CQ (conversational quality) is reported. MOS-LQ measures the quality of audio for listening purposes only. MOS-LQ does not take into account bi-directional effects such as delay and echo. MOS-CQ takes into account listening quality in both directions, as well as the bi-directional effects. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Media Realm > Quality of Experience Table	
Web: Media Realm > Quality Of Experience **[QOERules]**	This table configures Quality of Experience parameters per Media Realm. [QOERules] FORMAT QOERules_Index = QOERules_MediaRealmIndex, QOERules_RuleIndex, QOERules_MonitoredParam, QOERules_Profile, QOERules_GreenYellowThreshold, QOERules_GreenYellowHysteresis, QOERules_YellowRedThreshold, QOERules_YellowRedHystersis; [\QOERules]

A.8 Control Network Parameters

A.8.1 IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

Table A-27: Proxy, Registration and Authentication SIP Parameters

Parameter	Description
IP Group Table	
Web: IP Group Table EMS: Endpoints > IP Group CLI: configure voip > control-network ip-group **[IPGroup]**	This parameter table configures the IP Group table. The format of this parameter is as follows: [IPGroup] FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Description, IPGroup_ProxySetId, IPGroup_SIPGroupName, IPGroup_ContactUser, IPGroup_EnableSurvivability, IPGroup_ServingIPGroup, IPGroup_SipReRoutingMode, IPGroup_AlwaysUseRouteTable, IPGroup_RoutingMode, IPGroup_SRD, IPGroup_MediaRealm, IPGroup_ClassifyByProxySet, IPGroup_ProfileId, IPGroup_MaxNumOfRegUsers, IPGroup_InboundManSet, IPGroup_OutboundManSet, IPGroup_RegistrationMode, IPGroup_AuthenticationMode, IPGroup_MethodList, IPGroup_EnableSBCClientForking, IPGroup_ContactName; [/IPGroup] For example: IPGroup 1 = 0, "dol gateway", 1, firstIPgroup, , 0, -1, 0, 0, -1, 0, mrealm1, 1, 1, ; IPGroup 2 = 0, "abc server", 2, secondIPgroup, , 0, -1, 0, 0, -1, 0, mrealm2, 1, 2, ; IPGroup 3 = 1, "IP phones", 1, thirdIPGroup, , 0, -1, 0, 0, -1, 0, mrealm3, 1, 2, ; **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This table parameter can include up to 32 indices (where 1 is the first index). ▪ For a detailed description of the *ini* file table's parameters and for configuring this table using the Web interface, see 'Configuring IP Groups' on page 212. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Web: Authentication Table EMS: SIP Endpoints > Authentication	
[Authentication]	This parameter table defines a user name and password for authenticating each device port. The format of this parameter is as follows: [Authentication] FORMAT Authentication_Index = Authentication_UserId, Authentication_UserPassword, Authentication_Module, Authentication_Port;

Parameter	Description
	[\Authentication] Where, ▪ UserId = User name ▪ UserPassword = Password ▪ Module = Module number (where 1 denotes the module in Slot 1) ▪ Port = Port number (where 1 denotes the Port 1 of the module) For example: Authentication 0 = john,1325,1,1; (user name "john" with password 1325 for authenticating Port 1 of Module 1) Authentication 1 = lee,1552,1,2; (user name "lee" with password 1552 for authenticating Port 2 of Module 1) **Notes:** ▪ The indexing of this parameter starts at 0. ▪ The parameter AuthenticationMode determines whether authentication is performed per port or for the entire device. If authentication is performed for the entire device, the configuration in this table parameter is ignored. ▪ If the user name or password are not configured, the port's phone number (configured using the parameter TrunkGroup - Trunk Group Table) and global password (using the individual parameter Password) are used for authentication. ▪ Authentication is typically used for FXS interfaces, but can also be used for FXO interfaces. ▪ For configuring the Authentication table using the Web interface, see Configuring Authentication on page 355. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

Parameter	Description
Account Table	
Web: Account Table **EMS: SIP Endpoints > Account** CLI: configure voip > sip-definition account **[Account]**	This *para*meter table configures the Account table for registering and/or authenticating (digest) Trunk Groups or IP Groups (e.g., an IP-PBX) to a Serving IP Group (e.g., an Internet Telephony Service Provider - ITSP). The format of this parameter is as follows: [Account] FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroup, Account_ServingIPGroup, Account_Username, Account_Password, Account_HostName, Account_Register, Account_ContactUser, Account_ApplicationType; [\Account] For example: Account 1 = 1, -1, 1, user, 1234, acl, 1, ITSP1, 0; **Notes:** ▪ This table can include up to 32 indices (where 1 is the first index). ▪ You can define multiple table indices with the same ServedTrunkGroup but different ServingIPGroups, username, password, HostName, and ContactUser. This provides the capability for registering the same Trunk Group or IP Group to several ITSP's (i.e., Serving IP Groups). ▪ For a detailed description of this table's parameters and for configuring this table using the Web interface, see 'Configuring Account Table' on page 237. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Proxy Registration Parameters	
Web: Use Default Proxy EMS: Proxy Used CLI: enable-proxy **[IsProxyUsed]**	Enables the use of a SIP proxy server. ▪ **[0]** No = Proxy isn't used and instead, the internal routing table is used (default). ▪ **[1]** Yes = Proxy server is used. Define the IP address of the proxy server in the Proxy Sets table (see 'Configuring Proxy Sets Table' on page 219). **Note:** If you are not using a proxy server, you must define outbound IP call routing rules in the Outbound IP Routing Table (described in 'Configuring Outbound IP Routing Table' on page 291).
Web/EMS: Proxy Name CLI: proxy-name **[ProxyName]**	Defines the Home Proxy domain name. If specified, this name is used as the Request-URI in REGISTER, INVITE, and other SIP messages, and as the host part of the To header in INVITE messages. If not specified, the Proxy IP address is used instead. The value must be string of up to 49 characters.
Web: Redundancy Mode EMS: Proxy Redundancy Mode CLI: redundancy-mode **[ProxyRedundancyMode]**	Determines whether the device switches back to the primary Proxy after using a redundant Proxy. ▪ **[0]** Parking = device continues working with a redundant (now active) Proxy until the next failure, after which it works with the next redundant Proxy (default). ▪ **[1]** Homing = device always tries to work with the primary

Parameter	Description
	Proxy server (i.e., switches back to the primary Proxy whenever it's available). **Note:** To use this Proxy Redundancy mechanism, you need to enable the keep-alive with Proxy option, by setting the parameter EnableProxyKeepAlive to 1 or 2.
Web: Proxy IP List Refresh Time EMS: IP List Refresh Time CLI: proxy-ip-lst-rfrsh-time **[ProxyIPListRefreshTime]**	Defines the time interval (in seconds) between each Proxy IP list refresh. The range is 5 to 2,000,000. The default interval is 60.
Web: Enable Fallback to Routing Table EMS: Fallback Used CLI: fallback-to-routing **[IsFallbackUsed]**	Determines whether the device falls back to the Outbound IP Routing Table for call routing when Proxy servers are unavailable. ▪ **[0]** Disable = Fallback is not used (default). ▪ **[1]** Enable = The Outbound IP Routing Table is used when Proxy servers are unavailable. When the device falls back to the Outbound IP Routing Table', it continues scanning for a Proxy. When the device locates an active Proxy, it switches from internal routing back to Proxy routing. **Note:** To enable the redundant Proxies mechanism, set the parameter EnableProxyKeepAlive to 1 or 2.
Web/EMS: Prefer Routing Table CLI: prefer-routing-table **[PreferRouteTable]**	Determines whether the device's internal routing table takes precedence over a Proxy for routing calls. ▪ **[0]** No = Only a Proxy server is used to route calls (default). ▪ **[1]** Yes = The device checks the routing rules in the Outbound IP Routing Table for a match with the Tel-to-IP call. Only if a match is not found is a Proxy used.
Web/EMS: Always Use Proxy CLI: always-use-proxy **[AlwaysSendToProxy]**	Determines whether the device sends SIP messages and responses through a Proxy server. ▪ **[0]** Disable = Use standard SIP routing rules (default). ▪ **[1]** Enable = All SIP messages and responses are sent to the Proxy server. **Note:** This parameter is applicable only if a Proxy server is used (i.e., the parameter IsProxyUsed is set to 1).
Web: SIP ReRouting Mode EMS: SIP Re-Routing Mode CLI: sip-rerouting-mode **[SIPReroutingMode]**	Determines the routing mode after a call redirection (i.e., a 3xx SIP response is received) or transfer (i.e., a SIP REFER request is received). ▪ **[0]** Standard = INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI, according to the Refer-To header in the REFER message, or Contact header in the 3xx response (default). ▪ **[1]** Proxy = Sends a new INVITE to the Proxy. **Note:** This option is applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0. ▪ **[2]** Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination. **Notes:** ▪ When this parameter is set to **[1]** and the INVITE sent to the Proxy fails, the device re-routes the call according to the

Parameter	Description
	Standard mode **[0].** ▪ When this parameter is set to **[2]** and the INVITE fails, the device re-routes the call according to the Standard mode **[0].** If DNS resolution fails, the device attempts to route the call to the Proxy. If routing to the Proxy also fails, the Redirect/Transfer request is rejected. ▪ When this parameter is set to **[2],** the XferPrefix parameter can be used to define different routing rules for redirect calls. ▪ This parameter is disregarded if the parameter AlwaysSendToProxy is set to 1.
Web/EMS: DNS Query Type CLI: dns-query **[DNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to resolve Proxy and Registrar servers and to resolve all domain names that appear in the SIP Contact and Record-Route headers. ▪ **[0]** A-Record (default) ▪ **[1]** SRV ▪ **[2]** NAPTR If set to A-Record **[0],** no NAPTR or SRV queries are performed. If set to SRV **[1]** and the Proxy/Registrar IP address parameter, Contact/Record-Route headers, or IP address defined in the Routing tables contain a domain name, an SRV query is performed. The device uses the first host name received from the SRV query. The device then performs a DNS A-record query for the host name to locate an IP address. If set to NAPTR **[2],** an NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type. If the Proxy/Registrar IP address parameter, the domain name in the Contact/Record-Route headers, or the IP address defined in the Routing tables contain a domain name with port definition, the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Note:** To enable NAPTR/SRV queries for Proxy servers only, use the parameter ProxyDNSQueryType.
Web: Proxy DNS Query Type CLI: proxy-dns-query **[ProxyDNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to discover Proxy servers. ▪ **[0]** A-Record (default) ▪ **[1]** SRV ▪ **[2]** NAPTR If set to A-Record **[0],** no NAPTR or SRV queries are performed. If set to SRV **[1]** and the Proxy IP address parameter contains a domain name without port definition (e.g., ProxyIP = domain.com), an SRV query is performed. The SRV query returns up to four Proxy host names and their weights. The device then performs DNS A-record queries for each Proxy host name (according to the received weights) to locate up to four Proxy IP addresses. Therefore, if the first SRV query

Parameter	Description
	returns two domain names and the A-record queries return two IP addresses each, no additional searches are performed. If set to NAPTR **[2],** an NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type. If the Proxy IP address parameter contains a domain name with port definition (e.g., ProxyIP = domain.com:5080), the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Note:** When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
Web/EMS: Graceful Busy Out Timeout **[sec]** CLI: graceful-bsy-out-t-out **[GracefulBusyOutTimeout]**	Defines the timeout interval (in seconds) for Out of Service (OOS) graceful shutdown mode for busy trunks (per trunk) if communication fails with a Proxy server (or Proxy Set). In such a scenario, the device rejects new calls from the PSTN (Serving Trunk Group), but maintains currently active calls for this user-defined timeout. Once this timeout elapses, the device terminates currently active calls and takes the trunk out of service (sending the PSTN busy-out signal). Trunks on which no calls are active are immediately taken out of service regardless of the timeout. The range is 0 to 3,600. The default is 0. **Note:** This parameter is applicable only to digital interfaces.
Web/EMS: Use Gateway Name for OPTIONS CLI: use-gw-name-for-opt **[UseGatewayNameForOptions]**	Determines whether the device uses its IP address or gateway name in keep-alive SIP OPTIONS messages. ▪ **[0]** No = Use the device's IP address in keep-alive OPTIONS messages (default). ▪ **[1]** Yes = Use 'Gateway Name' (SIPGatewayName) in keep-alive OPTIONS messages. The OPTIONS Request-URI host part contains either the device's IP address or a string defined by the parameter SIPGatewayName. The device uses the OPTIONS request as a keep-alive message to its primary and redundant Proxies (i.e., the parameter EnableProxyKeepAlive is set to 1).
Web/EMS: User Name CLI: user-name-4-auth **[UserName]**	Defines the user name used for registration and Basic/Digest authentication with a Proxy/Registrar server. The default value is an empty string. **Notes:** ▪ This parameter is applicable only if single device registration is used (i.e., the parameter AuthenticationMode is set to authentication per gateway). ▪ Instead of configuring this parameter, the Authentication table can be used (see Authentication on page 355).
Web/EMS: Password CLI: password-4-auth **[Password]**	Defines the password for Basic/Digest authentication with a Proxy/Registrar server. A single password is used for all device ports. The default is 'Default_Passwd'.

Parameter	Description
	Note: Instead of configuring this parameter, the Authentication table can be used (see Authentication on page 355).
Web/EMS: Cnonce CLI: cnonce-4-auth **[Cnonce]**	Defines the Cnonce string used by the SIP server and client to provide mutual authentication. The value is free format, i.e., 'Cnonce = 0a4f113b'. The default is 'Default_Cnonce'.
Web/EMS: Mutual Authentication Mode CLI: mutual-authentication **[MutualAuthenticationMode]**	Determines the device's mode of operation when Authentication and Key Agreement (AKA) Digest Authentication is used. ▪ **[0]** Optional = Incoming requests that don't include AKA authentication information are accepted (default). ▪ **[1]** Mandatory = Incoming requests that don't include AKA authentication information are rejected.
Web/EMS: Challenge Caching Mode CLI: challenge-caching **[SIPChallengeCachingMode]**	Determines the mode for Challenge Caching, which reduces the number of SIP messages transmitted through the network. The first request to the Proxy is sent without authorization. The Proxy sends a 401/407 response with a challenge. This response is saved for further uses. A new request is re-sent with the appropriate credentials. Subsequent requests to the Proxy are automatically sent with credentials (calculated from the saved challenge). If the Proxy doesn't accept the new request and sends another challenge, the old challenge is replaced with the new one. ▪ **[0]** None = Challenges are not cached. Every new request is sent without preliminary authorization. If the request is challenged, a new request with authorization data is sent. (default) ▪ **[1]** INVITE Only = Challenges issued for INVITE requests are cached. This prevents a mixture of REGISTER and INVITE authorizations. ▪ **[2]** Full = Caches all challenges from the proxies. **Note**: Challenge Caching is used with all proxies and not only with the active one.
Proxy IP Table	
Web: Proxy IP Table EMS: Proxy IP CLI: configure voip > control-network proxy-ip **[ProxyIP]**	This *para*meter table configures the Proxy Set table with Proxy Set IDs, each with up to five Proxy server IP addresses (or fully qualified domain name/FQDN). Each Proxy Set can be defined with a transport type (UDP, TCP, or TLS). The format of this parameter is as follows: [ProxyIP] FORMAT ProxyIp_Index = ProxyIp_IpAddress, ProxyIp_TransportType, ProxyIp_ProxySetId; [\ProxyIP] For example: ProxyIp 0 = 10.33.37.77, -1, 0; ProxyIp 1 = 10.8.8.10, 0, 2; ProxyIp 2 = 10.5.6.7, -1, 1; **Notes:** ▪ This parameter can include up to 32 indices (0-31). ▪ To assign various attributes (such as Proxy Load Balancing) per Proxy Set ID, use the parameter ProxySet. ▪ For configuring the Proxy Set ID table using the Web

Parameter	Description
	interface and for a detailed description of the parameters of this *ini* file table, see 'Configuring Proxy Sets Table' on page 219. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Proxy Set Table	
Web: Proxy Set Table EMS: Proxy Set CLI: configure voip > control-network proxy-set **[ProxySet]**	This *para*meter table configures the Proxy Set ID table. It is used in conjunction with the ProxyIP *ini* file table parameter, which defines the IP addresses per Proxy Set ID. The ProxySet *ini* file table parameter defines additional attributes per Proxy Set ID. This includes, for example, Proxy keep-alive and load balancing and redundancy mechanisms (if a Proxy Set contains more than one proxy address). The format of this parameter is as follows: [ProxySet] FORMAT ProxySet_Index = ProxySet_EnableProxyKeepAlive, ProxySet_ProxyKeepAliveTime, ProxySet_ProxyLoadBalancingMethod, ProxySet_IsProxyHotSwap, ProxySet_SRD, ProxySet_ClassificationInput, ProxySet_ProxyRedundancyMode; [\ProxySet] For example: ProxySet 0 = 0, 60, 0, 0, 0, , 1; ProxySet 1 = 1, 60, 1, 0, 1, , 0; **Notes:** ▪ This table parameter can include up to 32 indices (0-31). ▪ For configuring the Proxy Set IDs and their IP addresses, use the parameter ProxyIP. ▪ For configuring the Proxy Set ID table using the Web interface and for a detailed description of the parameters of this *ini* file table, see 'Configuring Proxy Sets Table' on page 219. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Registrar Parameters	
Web: Enable Registration EMS: Is Register Needed CLI: enable-registration **[IsRegisterNeeded]**	Enables the device to register to a Proxy/Registrar server. ▪ **[0]** Disable = The device doesn't register to Proxy/Registrar server (default). ▪ **[1]** Enable = The device registers to Proxy/Registrar server when the device is powered up and at every user-defined interval (configured by the parameter RegistrationTime). **Note:** The device sends a REGISTER request for each channel or for the entire device (according to the AuthenticationMode parameter).
Web/EMS: Registrar Name CLI: registrar-name **[RegistrarName]**	Defines the Registrar domain name. If specified, the name is used as the Request-URI in REGISTER messages. If it isn't specified (default), the Registrar IP address, or Proxy name or IP address is used instead. The valid range is up to 100 characters.

Parameter	Description
Web: Registrar IP Address EMS: Registrar IP CLI: ip-addrr-rgstrr **[RegistrarIP]**	Defines the IP address (or FQDN) and port number (optional) of the Registrar server. The IP address is in dotted-decimal notation, e.g., 201.10.8.1:<5080>. **Notes:** ▪ If not specified, the REGISTER request is sent to the primary Proxy server. ▪ When a port number is specified, DNS NAPTR/SRV queries aren't performed, even if the parameter DNSQueryType is set to 1 or 2. ▪ If the parameter RegistrarIP is set to an FQDN and is resolved to multiple addresses, the device also provides real-time switching (hotswap mode) between different Registrar IP addresses (the parameter IsProxyHotSwap is set to 1). If the first Registrar doesn't respond to the REGISTER message, the same REGISTER message is sent immediately to the next Proxy. To allow this mechanism, the parameter EnableProxyKeepAlive must be set to 0. ▪ When a specific transport type is defined using the parameter RegistrarTransportType, a DNS NAPTR query is not performed even if the parameter DNSQueryType is set to 2.
Web/EMS: Registrar Transport Type CLI: registrar-transport **[RegistrarTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the Registrar. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web/EMS: Registration Time CLI: registration-time **[RegistrationTime]**	Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the SIP Expires header. In addition, this parameter defines the time interval between Keep-Alive messages when the parameter EnableProxyKeepAlive is set to 2 (REGISTER). Typically, the device registers every 3,600 sec (i.e., one hour). The device resumes registration according to the parameter RegistrationTimeDivider. The valid range is 10 to 2,000,000. The default value is 180.
Web: Re-registration Timing **[%]** EMS: Time Divider CLI: re-registration-timing **[RegistrationTimeDivider]**	Defines the re-registration timing (in percentage). The timing is a percentage of the re-register timing set by the Registrar server. The valid range is 50 to 100. The default value is 50. For example: If this parameter is set to 70% and the Registration Expires time is 3600, the device re-sends its registration request after 3600 x 70% (i.e., 2520 sec). **Note**: This parameter may be overridden if the parameter RegistrationTimeThreshold is greater than 0.
Web/EMS: Registration Retry Time CLI: registration-retry-time **[RegistrationRetryTime]**	Defines the time interval (in seconds) after which a registration request is re-sent if registration fails with a 4xx response or if there is no response from the Proxy/Registrar server. The default is 30 seconds. The range is 10 to 3600.

Parameter	Description
Web: Registration Time Threshold EMS: Time Threshold CLI: registration-time-thres **[RegistrationTimeThreshold]**	Defines a threshold (in seconds) for re-registration timing. If this parameter is greater than 0, but lower than the computed re-registration timing (according to the parameter RegistrationTimeDivider), the re-registration timing is set to the following: timing set by the Registration server in the SIP Expires header minus the value of the parameter RegistrationTimeThreshold. The valid range is 0 to 2,000,000. The default value is 0.
Web: Re-register On INVITE Failure EMS: Register On Invite Failure CLI: reg-on-invite-fail **[RegisterOnInviteFailure]**	Enables immediate re-registration if no response is received for an INVITE request sent by the device. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, the device immediately expires its re-registration timer and commences re-registration to the same Proxy upon any of the following scenarios: ▪ The response to an INVITE request is 407 (Proxy Authentication Required) without an authentication header included. ▪ The remote SIP UA abandons a call before the device has received any provisional response (indicative of an outbound proxy server failure). ▪ The remote SIP UA abandons a call and the only provisional response the device has received for the call is 100 Trying (indicative of a home proxy server failure, i.e., the failure of a proxy in the route after the outbound proxy). ▪ The device terminates a call due to the expiration of RFC 3261 Timer B or due to the receipt of a 408 (Request Timeout) response and the device has not received any provisional response for the call (indicative of an outbound proxy server failure). ▪ The device terminates a call due to the receipt of a 408 (Request Timeout) response and the only provisional response the device has received for the call is the 100 Trying provisional response (indicative of a home proxy server failure).
Web: ReRegister On Connection Failure EMS: Re Register On Connection Failure CLI: reg-on-conn-failure **[ReRegisterOnConnectionFailure]**	Enables the device to perform SIP re-registration upon TCP/TLS connection failure. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Gateway Registration Name EMS: Name CLI: gw-registration-name **[GWRegistrationName]**	Defines the user name that is used in the From and To headers in SIP REGISTER messages. If no value is specified (default) for this parameter, the UserName parameter is used instead. **Note:** This parameter is applicable only for single registration per device (i.e., AuthenticationMode is set to 1). When the device registers each channel separately (i.e., AuthenticationMode is set to 0), the user name is set to the channel's phone number.
Web/EMS: Registration Mode CLI: authentication-mode	Determines the device's registration and authentication method.

Parameter	Description
[AuthenticationMode]	▪ **[0]** Per Endpoint = Registration and authentication is performed separately for each endpoint/B-channel. This is typically used for FXS interfaces, where each endpoint registers (and authenticates) separately with its user name and password. ▪ **[1]** Per Gateway = Single registration and authentication for the entire device (default). This is typically used for FXO interfaces and digital modules. ▪ **[3]** Per FXS = Registration and authentication for FXS endpoints.
Web: Set Out-Of-Service On Registration Failure EMS: Set OOS On Registration Fail CLI: set-oos-on-reg-failure **[OOSOnRegistrationFail]**	Enables setting an endpoint, trunk, or the entire device (i.e., all endpoints) to out-of-service if registration fails. ▪ **[0]** Disable (default) ▪ **[1]** Enable If the registration is per endpoint (i.e., AuthenticationMode is set to 0) or per Account (see 'Configuring Trunk Group Settings' on page 273) and a specific endpoint/Account registration fails (SIP 4xx or no response), then that endpoint is set to out-of-service until a success response is received in a subsequent registration request. When the registration is per the entire device (i.e., AuthenticationMode is set to 1) and registration fails, all endpoints are set to out-of-service. If all the Accounts of a specific Trunk Group fail registration and if the Trunk Group comprises a complete trunk, then the entire trunk is set to out-of-service. **Note:** Te out-of-service method is configured using the parameter FXSOOSBehavior.
CLI: expl-un-reg **[UnregistrationMode]**	Enables the device to perform explicit unregisters. ▪ **[0]** Disable (default) ▪ **[1]** Enable = The device sends an asterisk ("*") value in the SIP Contact header, instructing the Registrar server to remove all previous registration bindings. The device removes SIP User Agent (UA) registration bindings in a Registrar, according to RFC 3261. Registrations are soft state and expire unless refreshed, but they can also be explicitly removed. A client can attempt to influence the expiration interval selected by the Registrar. A UA requests the immediate removal of a binding by specifying an expiration interval of "0" for that contact address in a REGISTER request. UA's should support this mechanism so that bindings can be removed before their expiration interval has passed. Use of the "*" Contact header field value allows a registering UA to remove all bindings associated with an address-of-record (AOR) without knowing their precise values. **Note:** The REGISTER-specific Contact header field value of "*" applies to all registrations, but it can only be used if the Expires header field is present with a value of "0".
Web/EMS: Add Empty Authorization Header CLI: add-empty-author-hdr **[EmptyAuthorizationHeader]**	Enables the inclusion of the SIP Authorization header in initial registration (REGISTER) requests sent by the device. ▪ **[0]** Disable (default) ▪ **[1]** Enable The Authorization header carries the credentials of a user

Parameter	Description
	agent (UA) in a request to a server. The sent REGISTER message populates the Authorization header with the following parameters: ▪ username - set to the value of the private user identity ▪ realm - set to the domain name of the home network ▪ uri - set to the SIP URI of the domain name of the home network ▪ nonce - set to an empty value ▪ response - set to an empty value For example: `Authorization: Digest username=alice_private@home1.net, realm="home1.net", nonce="", response="e56131d19580cd833064787ecc"` **Note:** This registration header is according to the IMS 3GPP TS24.229 and PKT-SP-24.220 specifications.

Web: Add initial Route Header
CLI: add-init-rte-hdr
[InitialRouteHeader]
	Enables the inclusion of the SIP Route header in initial registration or re-registration (REGISTER) requests sent by the device.
	▪ **[0]** Disable (default)
	▪ **[1]** Enable
	When the device sends a REGISTER message, the Route header includes either the Proxy's FQDN, or IP address and port according to the configured Proxy Set, for example:
	`Route: <sip:10.10.10.10;lr;transport=udp>`
	or
	`Route: <sip: pcscf-gm.ims.rr.com;lr;transport=udp>`

[UsePingPongKeepAlive]	Enables the use of the carriage-return and line-feed sequences (CRLF) Keep-Alive mechanism, according to RFC 5626 "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)" for reliable, connection-orientated transport types such as TCP. ▪ **[0]** Disable (default) ▪ **[1]** Enable The SIP user agent/client (i.e., device) uses a simple periodic message as a keep-alive mechanism to keep their flow to the proxy or registrar alive (used for example, to keep NAT bindings open). For connection-oriented transports such as TCP/TLS this is based on CRLF. This mechanism uses a client-to-server "ping" keep-alive and a corresponding server-to-client "pong" message. This ping-pong sequence allows the client, and optionally the server, to tell if its flow is still active and useful for SIP traffic. If the client does not receive a pong in response to its ping, it declares the flow "dead" and opens a new flow in its place. In the CRLF Keep-Alive mechanism the client periodically (defined by the PingPongKeepAliveTime parameter) sends a double-CRLF (the "ping") then waits to receive a single CRLF (the "pong"). If the client does not receive a "pong" within an appropriate amount of time, it considers the flow failed.

Parameter	Description
	Note: The device sends a CRLF message to the Proxy Set only if the Proxy Keep-Alive feature (EnableProxyKeepAlive parameter) is enabled and its transport type is set to TCP or TLS. The device first sends a SIP OPTION message to establish the TCP/TLS connection and if it receives any SIP response, it continues sending the CRLF keep-alive sequences.
[PingPongKeepAliveTime]	Defines the periodic interval (in seconds) after which a “ping” (double-CRLF) keep-alive is sent to a proxy/registrar, using the CRLF Keep-Alive mechanism. The default range is 5 to 2,000,000. The default is 120. The device uses the range of 80-100% of this user-defined value as the actual interval. For example, if the parameter value is set to 200 sec, the interval used is any random time between 160 to 200 seconds. This prevents an “avalanche” of keep-alive by multiple SIP UAs to a specific server.

A.8.2 Network Application Parameters

The SIP network application parameters are described in the table below.

Table A-28: SIP Network Application Parameters

Parameter	Description
Signaling Routing Domain Table	
Web: SRD Settings EMS: SRD Table CLI: config voip > control-network srd **[SRD]**	This *para*meter table configures the Signaling Routing Domain (SRD) table. The format of this parameter is as follows: [SRD] FORMAT SRD_Index = SRD_Name, SRD_MediaRealm, SRD_IntraSRDMediaAnchoring, SRD_BlockUnRegUsers, SRD_MaxNumOfRegUsers, SRD_EnableUnAuthenticatedRegistrations; [\SRD] For example: SRD 1 = LAN1_SRD, Mrealm1, 0, 1, 15, 1; SRD 2 = LAN2_SRD, Mrealm2, 0, 1, 15, 1; **Notes:** ▪ This table can include up to 32 indices (where 0 is the first index). ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, see 'Configuring SRD Table' on page 207. ▪ For a description on configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
SIP Interface Table	
Web: SIP Interface Table EMS: SIP Interfaces Table CLI: configure voip > control-network sip-interface **[SIPInterface]**	This *para*meter table configures the SIP Interface table. The SIP Interface represents a SIP signaling entity, comprising ports (UDP, TCP, and TLS) and associated with a specific IP interface and an SRD ID. The format of this parameter is as follows: [SIPInterface] FORMAT SIPInterface_Index = SIPInterface_NetworkInterface, SIPInterface_ApplicationType, SIPInterface_UDPPort, SIPInterface_TCPPort, SIPInterface_TLSPort, SIPInterface_SRD; [\SIPInterface] For example: SIPInterface 0 = Voice, 2, 5060, 5060, 5061, 1; SIPInterface 1 = Voice, 2, 5070, 5070, 5071, 2; SIPInterface 2 = Voice, 0, 5090, 5000, 5081, 2; **Notes:** ▪ This table can include up to 32 indices (where 0 is the first index). ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping). ▪ You can define up to three different SIP Interfaces per SRD, where each SIP Interface pertains to a different application type (i.e., GW, SAS, and SBC). ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, see 'Configuring SIP Interface Table' on page 210. ▪ For a description on configuring *ini* file table parameters, see 'Format of ini File Table Parameters' on page 98.

Parameter	Description
NAT Translation Table	
Web: NAT Translation Table CLI: configure voip > control-network NATTranslation **[NATTranslation]**	This *para*meter table defines NAT rules for translating source IP addresses per VoIP interface (SIP control and RTP media traffic) into NAT IP addresses. This allows, for example, the separation of VoIP traffic between different ISTP's, and topology hiding (of internal IP addresses to the "public" network). Each IP interface (configured in the Multiple Interface table - InterfaceTable parameter) can be associated with a NAT rule in this table, translating the source IP address and port of the outgoing packet into the NAT address (IP address and port range). The format of this parameter is as follows: [NATTranslation] FORMAT NATTranslation_Index = NATTranslation_SourceIPInterfaceName, NATTranslation_TargetIPAddress, NATTranslation_SourceStartPort, NATTranslation_SourceEndPort, NATTranslation_TargetStartPort, NATTranslation_TargetEndPort; [\NATTranslation] Where: ▪ SourceIPInterfaceName = name of the IP interface as defined in the Multiple Interface table. ▪ TargetIPAddress = global IP address. ▪ TargetStartPort and TargetEndPort = (optional) port range (1-65536) of the global address. If no ports are required, leave this field blank. ▪ SourceStartPort and SourceEndPort = (optional) port range (1-65536) of the IP interface. If no ports are required, leave this field blank. **Notes:** ▪ This table can include up to 32 indices. ▪ The device's priority method for performing NAT is as follows (not relevant for SBC application): **a.** Uses an external STUN server (STUNServerPrimaryIP parameter) to assign a NAT address for all interfaces. **b.** Uses the StaticNATIP parameter to define one NAT IP address for all interfaces. **c.** Uses the NATTranslation parameter to define NAT per interface. ▪ If NAT is not configured (by any of the above-mentioned methods), the device sends the packet according to its IP address defined in the Multiple Interface table.

A.9 General SIP Parameters

The general SIP parameters are described in the table below.

Table A-29: General SIP Parameters

Parameter	Description
Web/EMS: Max SIP Message Length **[KB]** **[MaxSIPMessageLength]**	Defines the maximum size (in Kbytes) for each SIP message that can be sent over the network. The device rejects messages exceeding this user-defined size. The valid value range is 1 to 50. The default is 50.
[SIPForceRport]	Determines whether the device sends SIP responses to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the SIP Via header. ▪ **[0]** (default) = Disabled - the device sends the SIP response to the UDP port defined in the Via header. If the Via header contains the 'rport' parameter, the response is sent to the UDP port from where the SIP request is received. ▪ **[1]** = Enabled - SIP responses are sent to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the Via header.
Web: Max Number of Active Calls EMS: Maximum Concurrent Calls CLI: max-nb-of--act-calls **[MaxActiveCalls]**	Defines the maximum number of simultaneous active calls supported by the device. If the maximum number of calls is reached, new calls are not established. The valid range is 1 to the maximum number of supported channels. The default value is the maximum available channels (i.e., no restriction on the maximum number of calls).
Web: Number of Calls Limit **[CallLimit]**	Defines the maximum number of concurrent calls per IP Profile. If the IP Profile is set to some limit, the device maintains the number of concurrent calls (incoming and outgoing) pertaining to the specific profile. When the number of concurrent calls is equal to the limit, the device rejects any new incoming and outgoing calls belonging to that profile. This parameter can also be set to the following: ▪ **[-1]** = There is no limitation on calls for that IP Profile (default). ▪ **[0]** = Calls are rejected. **Notes:** ▪ This parameter can only be configured for an IP Profile using the IPProfile parameter (see 'Configuring IP Profiles' on page 248). ▪ For IP-to-IP calls, you can configure the device to route calls to an alternative IP Group when the maximum number of concurrent calls is reached. To do so, you need to add an alternative routing rule in the Outbound IP Routing table that reroutes the call to an alternative IP Group. You also need to add a rule to the Reason for Alternative Routing table to initiate an alternative rule for Tel-to-IP calls using cause 805.
Web: QoS statistics in SIP Release Call **[QoSStatistics]**	Enables the device to include call quality of service (QoS) statistics in SIP BYE and SIP 200 OK response to BYE, using the proprietary SIP header X-RTP-Stat. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable

Parameter	Description

	The X-RTP-Stat header provides the following statistics:
	▪ Number of received and sent voice packets
	▪ Number of received and sent voice octets
	▪ Received packet loss, jitter (in ms), and latency (in ms)
	The X-RTP-Stat header contains the following fields:
	▪ PS=<voice packets sent>
	▪ OS=<voice octets sent>
	▪ PR=<voice packets received>
	▪ OR=<voice octets received>
	▪ PL=<receive packet loss>
	▪ JI=<jitter in ms>
	▪ LA=<latency in ms>
	Below is an example of the X-RTP-Stat header in a SIP BYE message:
	`BYE sip:302@10.33.4.125 SIP/2.0`
	`Via: SIP/2.0/UDP 10.33.4.126;branch=z9hG4bKac2127550866`
	`Max-Forwards: 70`
	`From: <sip:401@10.33.4.126;user=phone>;tag=1c2113553324`
	`To: <sip:302@company.com>;tag=1c991751121`
	`Call-ID: 991750671245200001912@10.33.4.125`
	`CSeq: 1 BYE`
	`X-RTP-Stat: PS=207;OS=49680;;PR=314;OR=50240;PL=0;JI=600;LA=40;`
	`Supported: em,timer,replaces,path,resource-priority`
	`Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE`
	`User-Agent: Sip-Gateway-/v.6.2A.008.006`
	`Reason: Q.850 ;cause=16 ;text="local"`
	`Content-Length: 0`

Web/EMS: PRACK Mode CLI: prack-mode **[PrackMode]**	Determines the PRACK (Provisional Acknowledgment) mechanism mode for SIP 1xx reliable responses. ▪ **[0]** Disable ▪ **[1]** Supported (default) ▪ **[2]** Required **Notes:** ▪ The Supported and Required headers contain the '100rel' tag. ▪ The device sends PRACK messages if 180/183 responses are received with '100rel' in the Supported or Required headers.
Web/EMS: Enable Early Media CLI: early-media **[EnableEarlyMedia]**	Digital: Enables the device to send a 18x response with SDP instead of a 18x, allowing the media stream to be established prior to the answering of the call. Analog: Enables the device to send a 183 Session Progress response with SDP instead of a 180 Ringing, allowing the media stream to be established prior to the answering of the call. ▪ **[0]** Disable = Early Media is disabled (default).

Parameter	Description
	▪ **[1]** Enable = Enables Early Media. Digital: The inclusion of the SDP in the 18x response depends on the ISDN Progress Indicator (PI). The SDP is sent only if PI is set to 1 or 8 in the received Proceeding, Alerting, or Progress PRI messages. See also the ProgressIndicator2IP parameter, which if set to 1 or 8, the device behaves as if it received the ISDN messages with the PI. **Notes:** ▪ See also the IgnoreAlertAfterEarlyMedia parameter. This parameter allows, for example, to interwork Alert + PI to SIP 183 + SDP instead of 180 + SDP. ▪ You can also configure early SIP 183 response immediately upon receipt of an INVITE, using the EnableEarly183 parameter. ▪ Analog: To send a 183 response, you must also set the parameter ProgressIndicator2IP to 1. If it is equal to 0, 180 Ringing response is sent. ▪ This parameter can be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248) and per Tel profile, using the TelProfile parameter (see 'Configuring Tel Profiles' on page 246).
Web/EMS: Enable Early 183 CLI: early-183 **[EnableEarly183]**	Enables the device to send SIP 183 responses with SDP to the IP upon receipt of INVITE messages (for IP-to-Tel calls). The device sends the RTP packets only once it receives an ISDN Progress, Alerting with Progress indicator, or Connect message from the PSTN. ▪ **[0]** Disable (default) ▪ **[1]** Enable For example, if enabled and the device receives an ISDN Progress message, it starts sending RTP packets according to the initial negotiation without sending the 183 response again. Therefore, this feature reduces clipping of early media. **Notes:** ▪ To enable this feature, configure the EnableEarlyMedia parameter to 1. ▪ This feature is applicable only to ISDN interfaces.
[IgnoreAlertAfterEarlyMedia]	Determines the device's interworking of Alerting messages from PRI to SIP. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled When enabled, if the device sends a 183 response with an SDP (due to a received ISDN Progress or Proceeding with PI messages) and an Alerting message is then received from the Tel side (with or without Progress Indicator), the device does not send an additional 18x response, and the voice channel remains open. However, if the device did not send a 183 with an SDP and it receives an Alert without PI, the device sends a 180 (without SDP). If it receives an Alert with PI it sends a 183with an SDP. When disabled, the device sends additional 18x responses as a result of receiving Alerting and Progress messages, regardless of whether or not a 18x response was already sent. **Note:** This parameter is applicable only if the EnableEarlyMedia parameter is set to 1 (i.e., enabled).
Web: 183 Message Behavior	Digital interfaces: Defines the ISDN message that is sent when the

Parameter	Description
EMS: SIP 183 Behaviour CLI: 183-msg-behavior **[SIP183Behaviour]**	183 Session Progress message is received for IP-to-Tel calls. Analog interfaces: Defines the response of the device upon receipt of a SIP 183 response. ▪ **[0]** Progress = Digital interfaces: The device sends a Progress message. Analog interfaces: A 183 response (without SDP) does not cause the device to play a ringback tone (default). ▪ **[1]** Alert = Digital interfaces: The device sends an Alerting message (upon receipt of a 183 response) instead of an ISDN Progress message. Analog interfaces: 183 response is handled by the device as if a 180 Ringing response is received, and the device plays a ringback tone.
[ReleaseIP2ISDNCallOnProgressWithCause]	Typically, if an Q.931 Progress message with a Cause is received from the PSTN for an outgoing IP-to-ISDN call and the EnableEarlyMedia parameter is set to 1 (i.e., the Early Media feature is enabled), the device interworks the Progress to 183+sdp to enable the originating party to hear the PSTN announcement about the call failure. Conversely, if EnableEarlyMedia is set to 0, the device disconnects the call by sending a SIP 4xx response to the originating party. However, if the ReleaseIP2ISDNCallOnProgressWithCause parameter is set to 1, the device sends a SIP 4xx response even if the EnableEarlyMedia parameter is set to 1. ▪ **[0]** = If a Progress with Cause message is received from the PSTN for an outgoing IP-to-ISDN call, the device does not disconnect the call by sending a SIP 4xx response to the originating party (default). ▪ **[1]** = The device sends a SIP 4xx response when the EnableEarlyMedia parameter is set to 0. ▪ **[2]** = The device always sends a SIP 4xx response, even if he EnableEarlyMedia parameter is set to 1.
Web: Session-Expires Time EMS: Sip Session Expires CLI: session-expires-time **[SIPSessionExpires]**	Defines the numerical value sent in the Session-Expires header in the first INVITE request or response (if the call is answered). The valid range is 1 to 86,400 sec. The default is 0 (i.e., the Session-Expires header is disabled).
Web: Minimum Session-Expires EMS: Minimal Session Refresh Value CLI: min-session-expires **[MinSE]**	Defines the time (in seconds) that is used in the Min-SE header. This header defines the minimum time that the user agent refreshes the session. The valid range is 10 to 100,000. The default value is 90.
Web/EMS: Session Expires Method CLI: session-exp-method **[SessionExpiresMethod]**	Determines the SIP method used for session-timer updates. ▪ **[0]** Re-INVITE = Uses Re-INVITE messages for session-timer updates (default). ▪ **[1]** UPDATE = Uses UPDATE messages. **Notes:** ▪ The device can receive session-timer refreshes using both methods. ▪ The UPDATE message used for session-timer is excluded from the SDP body.
[RemoveToTagInFailureResponse]	Determines whether the device removes the 'to' header tag from final SIP failure responses to INVITE transactions. ▪ **[0]** = Do not remove tag (default).

Parameter	Description
	▪ **[1]** = Remove tag.
[EnableRTCPAttribute]	Enables the use of the 'rtcp' attribute in the outgoing SDP. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: Options User Part **[OPTIONSUserPart]**	Defines the user part value of the Request-URI for outgoing SIP OPTIONS requests. If no value is configured, the endpoint number (analog interfaces) or configuration parameter 'Username' value (digital interfaces) is used. A special value is 'empty', indicating that no user part in the Request-URI (host part only) is used. The valid range is a 30-character string. The default value is an empty string ('').
Web: TDM Over IP Minimum Calls For Trunk Activation EMS: TDM Over IP Min Calls For Trunk Activation **[TDMOverIPMinCallsForTrunkActivation]**	Defines the minimal number of SIP dialogs that must be established when using TDM Tunneling to consider the specific trunk as active. When using TDM Tunneling, if calls from this defined number of B-channels pertaining to a specific Trunk fail (i.e., SIP dialogs are not correctly set up), an AIS alarm is sent on this trunk toward the PSTN and all current calls are dropped. The originator gateway continues the INVITE attempts. When this number of calls succeed (i.e., SIP dialogs are correctly set up), the AIS alarm is cleared. The valid range is 0 to 31. The default value is 0 (i.e., don't send AIS alarms).
[TDMoIPInitiateInviteTime]	Defines the time (in msec) between the first INVITE issued within the same trunk when implementing the TDM tunneling application. The valid value range is 500 to 1000. The default is 500.
[TDMoIPInviteRetryTime]	Defines the time (in msec) between call release and a new INVITE when implementing the TDM tunneling application. The valid value range is 10,000 to 20,000. The default is 10,000.
Web: Fax Signaling Method EMS: Fax Used CLI: fax-sig-method **[IsFaxUsed]**	Determines the SIP signaling method for establishing and transmitting a fax session after a fax is detected. ▪ **[0]** No Fax = No fax negotiation using SIP signaling. Fax transport method is according to the parameter FaxTransportMode (default). ▪ **[1]** T.38 Relay = Initiates T.38 fax relay. ▪ **[2]** G.711 Transport = Initiates fax/modem using the coder G.711 A-law/Mu-law with adaptations (see Note below). ▪ **[3]** Fax Fallback = Initiates T.38 fax relay. If the T.38 negotiation fails, the device re-initiates a fax session using the coder G.711 A-law/μ-law with adaptations (see the Note below). **Notes:** ▪ Fax adaptations (for options 2 and 3): ✓ Echo Canceller = On ✓ Silence Compression = Off ✓ Echo Canceller Non-Linear Processor Mode = Off ✓ Dynamic Jitter Buffer Minimum Delay = 40 ✓ Dynamic Jitter Buffer Optimization Factor = 13 ▪ If the device initiates a fax session using G.711 (option 2 and possibly 3), a 'gpmd' attribute is added to the SDP in the following format: ✓ **For A-law:** 'a=gpmd:8 vbd=yes;ecan=on' ✓ **For μ-law:** 'a=gpmd:0 vbd=yes;ecan=on'

Parameter	Description
	▪ When this parameter is set to 1, 2, or 3, the parameter FaxTransportMode is ignored. ▪ When this parameter is set to 0, T.38 might still be used without the control protocol's involvement. To completely disable T.38, set FaxTransportMode to a value other than 1. ▪ This parameter can also be configured per IP Profile (using the IPProfile parameter). ▪ For more information on fax transport methods, see 'Fax/Modem Transport Modes' on page 156.
[HandleG711asVBD]	Enables the handling of G.711 as G.711 VBD coder. ▪ **[0]** = Disable (default). The device negotiates G.711 as a regular audio coder and sends an answer only with G.729 coder. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing only the G.729 coder. ▪ **[1]** = Enable. The device assumes that the G.711 coder received in the INVITE SDP offer is a VBD coder. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing G.729 and G.711 VBD coders, allowing a subsequent bypass (passthrough) session if fax/modem signals are detected during the call. **Note:** This parameter is applicable only if G.711 VBD coder(s) are selected for the device (using the CodersGroup parameter).
CLI: fax-vbd-behvr **[FaxVBDBehavior]**	Determines the device's fax transport behavior when G.711 VBD coder is negotiated at call start. ▪ **[0]** = If the device is configured with a VBD coder (see the CodersGroup parameter) and is negotiated OK at call start, then both fax and modem signals are sent over RTP using the bypass payload type (and no mid-call VBD or T.38 Re-INVITEs occur). (Default.) ▪ **[1]** = If the IsFaxUsed parameter is set to 1, the channel opens with the FaxTransportMode parameter set to 1 (relay). This is required to detect mid-call fax tones and to send T.38 Re-INVITE messages upon fax detection. If the remote party supports T.38, the fax is relayed over T.38. **Notes:** ▪ If VBD coder negotiation fails at call start and if the IsFaxUsed parameter is set to 1 (or 3), then the channel opens with the FaxTransportMode parameter set to 1 (relay) to allow future detection of fax tones and sending of T.38 Re-INVITES. In such a scenario, the FaxVBDBehavior parameter has no effect. ▪ This feature can be used only if the remote party supports T.38 fax relay; otherwise, the fax fails.
Web: SIP Transport Type EMS: Transport Type CLI: app-sip-transport-type **[SIPTransportType]**	Determines the default transport layer for outgoing SIP calls initiated by the device. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS (SIPS) **Notes:** ▪ It's recommended to use TLS for communication with a SIP Proxy

Parameter	Description
	and not for direct device-to-device communication. ▪ For received calls (i.e., incoming), the device accepts all these protocols. ▪ The value of this parameter is also used by the SAS application as the default transport layer for outgoing SIP calls.
Web: SIP UDP Local Port EMS: Local SIP Port CLI: sip-udp-local-port **[LocalSIPPort]**	Defines the local UDP port for SIP messages. The valid range is 1 to 65534. The default value is 5060.
Web: SIP TCP Local Port EMS: TCP Local SIP Port CLI: sip-tcp-local-port **[TCPLocalSIPPort]**	Defines the local TCP port for SIP messages. The valid range is 1 to 65535. The default value is 5060.
Web: SIP TLS Local Port EMS: TLS Local SIP Port CLI: sip-tls-local-port **[TLSLocalSIPPort]**	Defines the local TLS port for SIP messages. The valid range is 1 to 65535. The default value is 5061. **Note:** The value of this parameter must be different from the value of the parameter TCPLocalSIPPort.
Web/EMS: Enable SIPS CLI: enable-sips **[EnableSIPS]**	Enables secured SIP (SIPS URI) connections over multiple hops. ▪ **[0]** Disable (default). ▪ **[1]** Enable. When the SIPTransportType parameter is set to 2 (i.e., TLS) and the parameter EnableSIPS is disabled, TLS is used for the next network hop only. When the parameter SIPTransportType is set to 2 or 1 (i.e., TCP or TLS) and EnableSIPS is enabled, TLS is used through the entire connection (over multiple hops). **Note:** If this parameter is enabled and the parameter SIPTransportType is set to 0 (i.e., UDP), the connection fails.
Web/EMS: Enable TCP Connection Reuse CLI: tcp-conn-reuse **[EnableTCPConnectionReuse]**	Enables the reuse of the same TCP connection for all calls to the same destination. ▪ **[0]** Disable = Use a separate TCP connection for each call. ▪ **[1]** Enable = Use the same TCP connection for all calls (default).
Web/EMS: Reliable Connection Persistent Mode CLI: reliable-conn-persistent **[ReliableConnectionPersistentMode]**	Enables setting of all TCP/TLS connections as persistent and therefore, not released. ▪ **[0]** = Disable (default) - all TCP connections (except those that are set to a proxy IP) are released if not used by any SIP dialog\transaction. ▪ **[1]** = Enable - TCP connections to all destinations are persistent and not released unless the device reaches 70% of its maximum TCP resources. While trying to send a SIP message connection, reuse policy determines whether live connections to the specific destination are re-used. Persistent TCP connection ensures less network traffic due to fewer setting up and tearing down of TCP connections and reduced latency on subsequent requests due to avoidance of initial TCP handshake. For TLS, persistent connection may reduce the number of costly TLS handshakes to establish security associations, in addition to the initial TCP connection set up.

Parameter	Description
	Note: If the destination is a Proxy server, the TCP/TLS connection is persistent regardless of the settings of this parameter.
Web/EMS: TCP Timeout CLI: tcp-timeout **[SIPTCPTimeout]**	Defines the Timer B (INVITE transaction timeout timer) and Timer F (non-INVITE transaction timeout timer), as defined in RFC 3261, when the SIP Transport Type is TCP. The valid range is 0 to 40 sec. The default value is 64*SIPT1Rtx msec.
Web: SIP Destination Port EMS: Destination Port CLI: sip-dst-port **[SIPDestinationPort]**	Defines the SIP destination port for sending initial SIP requests. The valid range is 1 to 65534. The default port is 5060. **Note:** SIP responses are sent to the port specified in the Via header.
Web: Use user=phone in SIP URL EMS: Is User Phone CLI: user=phone-in-url **[IsUserPhone]**	Determines whether the 'user=phone' string is added to the SIP URI and SIP To header. ▪ **[0]** No = 'user=phone' string is not added. ▪ **[1]** Yes = 'user=phone' string is part of the SIP URI and SIP To header (default).
Web: Use user=phone in From Header EMS: Is User Phone In From CLI: phone-in-from-hdr **[IsUserPhoneInFrom]**	Determines whether the 'user=phone' string is added to the From and Contact SIP headers. ▪ **[0]** No = Doesn't add 'user=phone' string (default). ▪ **[1]** Yes = 'user=phone' string is part of the From and Contact headers.
Web: Use Tel URI for Asserted Identity CLI: uri-for-assert-id **[UseTelURIForAssertedID]**	Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers. ▪ **[0]** Disable = 'sip:' (default) ▪ **[1]** Enable = 'tel:'
Web: Tel to IP No Answer Timeout EMS: IP Alert Timeout CLI: tel2ip-no-ans-timeout **[IPAlertTimeout]**	Defines the time (in seconds) that the device waits for a 200 OK response from the called party (IP side) after sending an INVITE message. If the timer expires, the call is released. The valid range is 0 to 3600. The default value is 180.
Web: Enable Remote Party ID EMS: Enable RPI Header CLI: remote-party-id **[EnableRPIheader]**	Enables Remote-Party-Identity headers for calling and called numbers for Tel-to-IP calls. ▪ **[0]** Disable (default). ▪ **[1]** Enable = Remote-Party-Identity headers are generated in SIP INVITE messages for both called and calling numbers.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Web: Enable History-Info Header
EMS: Enable History Info
CLI: hist-info-hdr
[EnableHistoryInfo]</td><td>Enables usage of the History-Info header.
▪ [0] Disable (default)
▪ [1] Enable
User Agent Client (UAC) Behavior:
▪ Initial request: The History-Info header is equal to the Request-URI. If a PSTN Redirect number is received, it is added as an additional History-Info header with an appropriate reason.
▪ Upon receiving the final failure response, the device copies the History-Info as is, adds the reason of the failure response to the last entry, and concatenates a new destination to it (if an additional request is sent). The order of the reasons is as follows:
a. Q.850 Reason
b. SIP Reason
c. SIP Response code
▪ Upon receiving the final response (success or failure), the device searches for a Redirect reason in the History-Info (i.e., 3xx/4xx SIP reason). If found, it is passed to ISDN according to the following table:
<table><tr><th>SIP Reason Code</th><th>ISDN Redirecting Reason</th></tr><tr><td>302 - Moved Temporarily</td><td>Call Forward Universal (CFU)</td></tr><tr><td>408 - Request Timeout</td><td rowspan="3">Call Forward No Answer (CFNA)</td></tr><tr><td>480 - Temporarily Unavailable</td></tr><tr><td>487 - Request Terminated</td></tr><tr><td>486 - Busy Here</td><td rowspan="2">Call Forward Busy (CFB)</td></tr><tr><td>600 - Busy Everywhere</td></tr></table>
▪ If history reason is a Q.850 reason, it is translated to the SIP reason (according to the SIP-ISDN tables) and then to ISDN Redirect reason according to the table above.
User Agent Server (UAS) Behavior:
▪ The History-Info header is sent only in the final response.
▪ Upon receiving a request with History-Info, the UAS checks the policy in the request. If a 'session', 'header', or 'history' policy tag is found, the (final) response is sent without History-Info; otherwise, it is copied from the request.</td></tr>
<tr><td>Web: Use Tgrp Information
EMS: Use SIP Tgrp
CLI: use-tgrp-inf
[UseSIPTgrp]</td><td>Determines whether the SIP 'tgrp' parameter is used. This SIP parameter specifies the Trunk Group to which the call belongs (according to RFC 4904). For example, the SIP message below indicates that the call belongs to Trunk Group ID 1:
INVITE sip::+16305550100;tgrp=1;trunk-context=example.com@10.1.0.3;user=phone SIP/2.0
▪ [0] Disable (default) = The tgrp' parameter isn't used.
▪ [1] Send Only = The Trunk Group number is added to the 'tgrp' parameter value in the Contact header of outgoing SIP messages. If a Trunk Group number is not associated with the call, the 'tgrp' parameter isn't included. If a 'tgrp' value is specified in incoming messages, it is ignored.
▪ [2] Send and Receive = The functionality of outgoing SIP messages is identical to the functionality described in option 1. In addition, for incoming SIP INVITEs, if the Request-URI includes a</td></tr>
</table>

Parameter	Description
	'tgrp' parameter, the device routes the call according to that value (if possible). The Contact header in the outgoing SIP INVITE (Tel-to-IP call) contains "tgrp=<source trunk group ID>;trunk-context=<gateway IP address>". The <source trunk group ID> is the Trunk Group ID where incoming calls from Tel is received. For IP-Tel calls, the SIP 200 OK device's response contains "tgrp=<destination trunk group ID>;trunk-context=<gateway IP address>". The <destination trunk group ID> is the Trunk Group ID used for outgoing Tel calls. The <gateway IP address> in "trunk-context" can be configured using the parameter SIPGatewayName. ▪ **[3]** Hotline = Interworks the hotline "Off Hook Indicator" parameter between SIP and ISDN: ✓ For IP-to-ISDN calls: - The device interworks the SIP tgrp=hotline parameter (received in INVITE) to ISDN Setup with the Off Hook Indicator IE of "Voice", and "Speech" Bearer Capability IE. Note that the Off Hook Indicator IE is described in UCR 2008 specifications. - The device interworks the SIP tgrp=hotline-ccdata parameter (received in INVITE) to ISDN Setup with an Off Hook Indicator IE of "Data", and with "Unrestricted 64k" Bearer Capability IE. The following is an example of the INVITE with tgrp=hotline-ccdata: INVITE sip:1234567;tgrp=hotline-ccdata;trunk-context=dsn.mil@example.com ✓ For ISDN-to-IP calls: - The device interworks ISDN Setup with an Off Hook Indicator of "Voice" to SIP INVITE with "tgrp=hotline;trunk-context=dsn.mil" in the Contact header. - The device interworks ISDN Setup with an Off Hook indicator of "Data" to SIP INVITE with "tgrp=hotline-ccdata;trunk-context=dsn.mil" in the Contact header. - If ISDN Setup does not contain an Off Hook Indicator IE and the Bearer Capability IE contains "Unrestricted 64k", the outgoing INVITE includes "tgrp=ccdata;trunk-context=dsn.mil". If the Bearer Capability IE contains "Speech", the INVITE in this case does not contain tgrp and trunk-context parameters. ▪ **[4]** Hotline Extended = Interworks the ISDN Setup message's hotline "OffHook Indicator" Information Element (IE) to SIP INVITE's Request-URI and Contact headers. (**Note:** For IP-to-ISDN calls, the device handles the call as described in option **[3].)** ✓ The device interworks ISDN Setup with an Off Hook Indicator of "Voice" to SIP INVITE Request-URI and Contact header with "tgrp=hotline;trunk-context=dsn.mil". ✓ The device interworks ISDN Setup with an Off Hook indicator of "Data" to SIP INVITE Request-URI and Contact header with "tgrp=hotline-ccdata;trunk-context=dsn.mil". ✓ If ISDN Setup does not contain an Off Hook Indicator IE and the Bearer Capability IE contains "Unrestricted 64k", the outgoing INVITE Request-URI and Contact header includes "tgrp=ccdata;trunk-context=dsn.mil". If the Bearer Capability IE contains "Speech", the INVITE in this case does not contain tgrp and trunk-context parameters. **Note:** IP-to-Tel configuration (using the PSTNPrefix parameter) overrides the 'tgrp' parameter in incoming INVITE messages.

Parameter	Description
Web/EMS: TGRP Routing Precedence CLI: tgrp-routing-prec **[TGRProutingPrecedence]**	Determines the precedence method for routing IP-to-Tel calls - according to the Inbound IP Routing Table or according to the SIP 'tgrp' parameter. ▪ **[0]** (default) = IP-to-Tel routing is determined by the Inbound IP Routing Table (PSTNPrefix parameter). If a matching rule is not found in this table, the device uses the Trunk Group parameters for routing the call. ▪ **[1]** = The device first places precedence on the 'tgrp' parameter for IP-to-Tel routing. If the received INVITE Request-URI does not contain the 'tgrp' parameter or if the Trunk Group number is not defined, then the Inbound IP Routing Table is used for routing the call. Below is an example of an INVITE Request-URI with the 'tgrp' parameter, indicating that the IP call should be routed to Trunk Group 7: `INVITE sip:200;tgrp=7;trunk-context=example.com@10.33.2.68;user=phone SIP/2.0` **Notes:** ▪ For enabling routing based on the 'tgrp' parameter, the UseSIPTgrp parameter must be set to 2. ▪ For IP-to-Tel routing based on the 'dtg' parameter (instead of the 'tgrp' parameter), use the parameter UseBroadsoftDTG.
CLI: use-dtg **[UseBroadsoftDTG]**	Determines whether the device uses the 'dtg' parameter for routing IP-to-Tel calls to a specific Trunk Group. ▪ **[0]** Disable (default) ▪ **[1]** Enable When this parameter is enabled, if the Request-URI in the received SIP INVITE includes the 'dtg' parameter, the device routes the call to the Trunk Group according to its value. This parameter is used instead of the 'tgrp/trunk-context' parameters. The dtg' parameter appears in the INVITE Request-URI (and in the To header). For example, the received SIP message below routes the call to Trunk Group ID 56: `INVITE sip:123456@192.168.1.2;dtg=56;user=phone SIP/2.0` **Note:** If the Trunk Group is not found based on the 'dtg' parameter, the Inbound IP Routing Table is used instead for routing the call to the appropriate Trunk Group.
Web/EMS: Enable GRUU CLI: enable-gruu **[EnableGRUU]**	Determines whether the Globally Routable User Agent URIs (GRUU) mechanism is used, according to RFC 5627. This is used for obtaining a GRUU from a registrar and for communicating a GRUU to a peer within a dialog. ▪ **[0]** Disable (default) ▪ **[1]** Enable A GRUU is a SIP URI that routes to an instance-specific UA and can be reachable from anywhere. There are a number of contexts in which it is desirable to have an identifier that addresses a single UA (using GRUU) rather than the group of UA's indicated by an Address of Record (AOR). For example, in call transfer where user A is talking to user B, and user A wants to transfer the call to user C. User A sends a REFER to user C: `REFER sip:C@domain.com SIP/2.0`

Parameter	Description

	`From: sip:A@domain.com;tag=99asd`
	`To: sip:C@domain.com`
	`Refer-To: (URI that identifies B's UA)`
	The Refer-To header needs to contain a URI that user C can use to place a call to user B. This call needs to route to the specific UA instance that user B is using to talk to user A. User B should provide user A with a URI that has to be usable by anyone. It needs to be a GRUU.
	▪ Obtaining a GRUU: The mechanism for obtaining a GRUU is through registrations. A UA can obtain a GRUU by generating a REGISTER request containing a Supported header field with the value “gruu”. The UA includes a “+sip.instance” Contact header parameter of each contact for which the GRUU is desired. This Contact parameter contains a globally unique ID that identifies the UA instance. The global unique ID is created from one of the following:
	✓ If the REGISTER is per the device’s client (endpoint), it is the MAC address concatenated with the phone number of the client.
	✓ If the REGISTER is per device, it is the MAC address only.
	✓ When using TP, “User Info” can be used for registering per endpoint. Thus, each endpoint can get a unique id – its phone number. The globally unique ID in TP is the MAC address concatenated with the phone number of the endpoint.
	If the remote server doesn’t support GRUU, it ignores the parameters of the GRUU. Otherwise, if the remote side also supports GRUU, the REGISTER responses contain the “gruu” parameter in each Contact header. This parameter contains a SIP or SIPS URI that represents a GRUU corresponding to the UA instance that registered the contact. The server provides the same GRUU for the same AOR and instance-id when sending REGISTER again after registration expiration. RFC 5627 specifies that the remote target is a GRUU target if its’ Contact URL has the "gr" parameter with or without a value.
	▪ Using GRUU: The UA can place the GRUU in any header field that can contain a URI. It must use the GRUU in the following messages: INVITE request, its 2xx response, SUBSCRIBE request, its 2xx response, NOTIFY request, REFER request and its 2xx response.

EMS: Is CISCO Sce Mode **[IsCiscoSCEMode]**	Determines whether a Cisco gateway exists at the remote side. ▪ **[0]** = No Cisco gateway exists at the remote side (default). ▪ **[1]** = A Cisco gateway exists at the remote side. When a Cisco gateway exists at the remote side, the device must set the value of the 'annexb' parameter of the fmtp attribute in the SDP to 'no'. This logic is used if the parameter EnableSilenceCompression is set to 2 (enable without adaptation). In this case, Silence Suppression is used on the channel but not declared in the SDP. **Note:** The IsCiscoSCEMode parameter is applicable only when the selected coder is G.729.
Web: User-Agent Information EMS: User Agent Display Info CLI: user-agent-info **[UserAgentDisplayInfo]**	Defines the string that is used in the SIP User-Agent and Server response headers. When configured, the string <UserAgentDisplayInfo value>/software version' is used, for example: `User-Agent: myproduct/v.6.00.010.006`

Parameter	Description
	If not configured, the default string, <AudioCodes product-name>/software version' is used, for example: `User-Agent: Audiocodes-Sip-Gateway-Mediant 1000 MSBG/v.6.00.010.006` The maximum string length is 50 characters. **Note:** The software version number and preceding forward slash (/) cannot be modified. Therefore, it is recommended not to include a forward slash in the parameter's value (to avoid two forward slashes in the SIP header, which may cause problems).
Web/EMS: SDP Session Owner CLI: sdp-session-owner **[SIPSDPSessionOwner]**	Defines the value of the Owner line ('o' field) in outgoing SDP messages. The valid range is a string of up to 39 characters. The default value is 'AudiocodesGW'. For example: `o=AudiocodesGW 1145023829 1145023705 IN IP4 10.33.4.126`
CLI: sdp-ver-nego **[EnableSDPVersionNegotiation]**	Enables the device to ignore new SDP re-offers (from the media negotiation perspective) in certain scenarios (such as session expires). According to RFC 3264, once an SDP session is established, a new SDP offer is considered a new offer only when the SDP origin value is incremented. In scenarios such as session expires, SDP negotiation is irrelevant and thus, the origin field is not changed. Even though some SIP devices don't follow this behavior and don't increment the origin value even in scenarios where they want to re-negotiate, the device can assume that the remote party operates according to RFC 3264, and in cases where the origin field is not incremented, the device does not re-negotiate SDP capabilities. ▪ **[0]** Disable = The device negotiates any new SDP re-offer, regardless of the origin field (default). ▪ **[1]** Enable = The device negotiates only an SDP re-offer with an incremented origin field.
Web/EMS: Subject CLI: usr-def-subject **[SIPSubject]**	Defines the Subject header value in outgoing INVITE messages. If not specified, the Subject header isn't included (default). The maximum length is up to 50 characters.
Web: Multiple Packetization Time Format EMS: Multi Ptime Format CLI: mult-ptime-format **[MultiPtimeFormat]**	Determines whether the 'mptime' attribute is included in the outgoing SDP. ▪ **[0]** None = Disabled (default) ▪ **[1]** PacketCable = includes the 'mptime' attribute in the outgoing SDP - PacketCable-defined format The mptime' attribute enables the device to define a separate Packetization period for each negotiated coder in the SDP. The mptime' attribute is only included if this parameter is enabled, even if the remote side includes it in the SDP offer. Upon receipt, each coder receives its 'ptime' value in the following precedence: from 'mptime' attribute, from 'ptime' attribute, and then from default value.
EMS: Enable P Time **[EnablePtime]**	Determines whether the 'ptime' attribute is included in the SDP. ▪ **[0]** = Remove the 'ptime' attribute from SDP. ▪ **[1]** = Include the 'ptime' attribute in SDP (default).
Web/EMS: 3xx Behavior	Determines the device's behavior regarding call identifiers when a 3xx

Parameter	Description
CLI: 3xx-behavior **[3xxBehavior]**	response is received for an outgoing INVITE request. The device can either use the same call identifiers (Call-ID, Branch, To, and From tags) or change them in the new initiated INVITE. ▪ **[0]** Forward = Use different call identifiers for a redirected INVITE message (default). ▪ **[1]** Redirect = Use the same call identifiers.
Web/EMS: Enable P-Charging Vector CLI: p-charging-vector **[EnablePChargingVector]**	Enables the inclusion of the P-Charging-Vector header to all outgoing INVITE messages. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: Retry-After Time CLI: retry-aftr-time **[RetryAfterTime]**	Defines the time (in seconds) used in the Retry-After header when a 503 (Service Unavailable) response is generated by the device. The time range is 0 to 3,600. The default value is 0.
Web/EMS: Fake Retry After **[sec]** CLI: fake-retry-after **[FakeRetryAfter]**	Determines whether the device, upon receipt of a SIP 503 response without a Retry-After header, behaves as if the 503 response included a Retry-After header and with the period (in seconds) specified by this parameter. ▪ **[0]** Disable ▪ Any positive value (in seconds) for defining the period When enabled, this feature allows the device to operate with Proxy servers that do not include the Retry-After SIP header in SIP 503 (Service Unavailable) responses to indicate an unavailable service. The Retry-After header is used with the 503 (Service Unavailable) response to indicate how long the service is expected to be unavailable to the requesting SIP client. The device maintains a list of available proxies, by using the Keep-Alive mechanism. The device checks the availability of proxies by sending SIP OPTIONS every keep-alive timeout to all proxies. If the device receives a SIP 503 response to an INVITE, it also marks that the proxy is out of service for the defined "Retry-After" period.
Web/EMS: Enable P-Associated-URI Header CLI: p-associated-uri-hdr **[EnablePAssociatedURIHeader]**	Determines the device usage of the P-Associated-URI header. This header can be received in 200 OK responses to REGISTER requests. When enabled, the first URI in the P-Associated-URI header is used in subsequent requests as the From/P-Asserted-Identity headers value. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file).
Web/EMS: Source Number Preference CLI: src-nb-preference **[SourceNumberPreference]**	Determines from which SIP header the source (calling) number is obtained in incoming INVITE messages. ▪ If not configured (i.e., empty string) or if any string other than "From" or "Pai2" is configured, the calling number is obtained from a specific header using the following logic: **a.** P-Preferred-Identity header. **b.** If the above header is not present, then the first P-Asserted-Identity header is used. **c.** If the above header is not present, then the Remote-Party-ID header is used. **d.** If the above header is not present, then the From header is used.

Parameter	Description
	▪ "**From**" = The calling number is obtained from the From header. ▪ "**Pai2**" = The calling number is obtained using the following logic: **a.** If a P-Preferred-Identity header is present, the number is obtained from it. **b.** If no P-Preferred-Identity header is present and two P-Asserted-Identity headers are present, the number is obtained from the second P-Asserted-Identity header. **c.** If only one P-Asserted-Identity header is present, the calling number is obtained from it. **Notes:** ▪ The "From" and "Pai2" values are not case-sensitive. ▪ Once a URL is selected, all the calling party parameters are set from this header. If P-Asserted-Identity is selected and the Privacy header is set to 'id', the calling number is assumed restricted.
CLI: src-hdr-4-called-nb **[SelectSourceHeaderForCalledNumber]**	Determines the SIP header used for obtaining the called number (destination) for IP-to-Tel calls. ▪ **[0]** Request-URI header (default) = Obtains the destination number from the user part of the Request-URI. ▪ **[1]** To header = Obtains the destination number from the user part of the To header. ▪ **[2]** P-Called-Party-ID header = Obtains the destination number from the P-Called-Party-ID header.
Web/EMS: Forking Handling Mode CLI: forking-handling **[ForkingHandlingMode]**	Determines how the device handles the receipt of multiple SIP 18x forking responses for Tel-to-IP calls. The forking 18x response is the response with a different SIP to-tag than the previous 18x response. These responses are typically generated (initiated) by Proxy / Application servers that perform call forking, sending the device's originating INVITE (received from SIP clients) to several destinations, using the same CallID. ▪ **[0]** Parallel handling = If SIP 18x with SDP is received, the device opens a voice stream according to the received SDP and disregards any 18x forking responses (with or without SDP) received thereafter. If the first response is 180 without SDP, the device responds according to the PlayRBTone2TEL parameter and disregards the subsequent forking 18x responses. (default) ▪ **[1]** Sequential handling = If 18x with SDP is received, the device opens a voice stream according to the received SDP. The device re-opens the stream according to subsequently received 18x responses with SDP, or plays a ringback tone if 180 response without SDP is received. If the first received response is 180 without SDP, the device responds according to the PlayRBTone2TEL parameter and processes the subsequent 18x forking responses. **Note:** Regardless of this parameter setting, once a SIP 200 OK response is received, the device uses the RTP information and re-opens the voice stream, if necessary.
Web: Forking Timeout CLI: forking-timeout **[ForkingTimeOut]**	Defines the timeout (in seconds) that is started after the first SIP 2xx response has been received for a User Agent when a Proxy server performs call forking (Proxy server forwards the INVITE to multiple SIP User Agents). The device sends a SIP ACK and BYE in response to any additional SIP 2xx received from the Proxy within this timeout. Once this timeout elapses, the device ignores any subsequent SIP

Parameter	Description
	2xx. The number of supported forking calls per channel is 20. In other words, for an INVITE message, the device can receive up to 20 forking responses from the Proxy server. The valid range is 0 to 30. The default is 30.
Web: Tel2IP Call Forking Mode CLI: tel2ip-call-forking-mode **[Tel2IPCallForkingMode]**	Enables Tel-to-IP call forking, whereby a Tel call can be routed to multiple IP destinations. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** Once enabled, routing rules must be assigned Forking Groups in the Outbound IP Routing table.
Web/EMS: Enable Reason Header CLI: reason-header **[EnableReasonHeader]**	Enables the usage of the SIP Reason header. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Web/EMS: Gateway Name CLI: gw-name **[SIPGatewayName]**	Defines a name for the device (e.g., device123.com'). **Notes:** ▪ Ensure that the name defined is the one with which the Proxy is configured to identify the device. ▪ If specified, the device name is used as the host part of the SIP URI in the From header. If not specified, the device's IP address is used instead (default).
[ZeroSDPHandling]	Determines the device's response to an incoming SDP that includes an IP address of 0.0.0.0 in the SDP's Connection Information field (i.e., "c=IN IP4 0.0.0.0"). ▪ **[0]** = Sets the IP address of the outgoing SDP's c= field to 0.0.0.0 (default). ▪ **[1]** = Sets the IP address of the outgoing SDP c= field to the IP address of the device. If the incoming SDP doesn't contain the "a=inactive" line, the returned SDP contains the "a=recvonly" line.
Web/EMS: Enable Delayed Offer CLI: delayed-offer **[EnableDelayedOffer]**	Determines whether the device sends the initial INVITE message with or without an SDP. Sending the first INVITE without SDP is typically done by clients for obtaining the far-end's full list of capabilities before sending their own offer. (An alternative method for obtaining the list of supported capabilities is by using SIP OPTIONS, which is not supported by every SIP agent.) ▪ **[0]** Disable = The device sends the initial INVITE message with an SDP (default). ▪ **[1]** Enable = The device sends the initial INVITE message without an SDP.
[DisableCryptoLifeTimeInSDP]	Enables the device to send "a=crypto" lines without the lifetime parameter in the SDP. For example, if the SDP contains "a=crypto:12 AES_CM_128_HMAC_SHA1_80 inline:hhQe10yZRcRcpIFPkH5xYY9R1de37ogh9G1MpvNp|2^31", it removes the lifetime parameter "2^31". ▪ **[0]** Disable (default) ▪ **[1]** Enable

Parameter	Description
Web/EMS: Enable Contact Restriction CLI: contact-restriction **[EnableContactRestriction]**	Determines whether the device sets the Contact header of outgoing INVITE requests to 'anonymous' for restricted calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable
CLI: anonymous-mode **[AnonymousMode]**	Determines whether the device's IP address is used as the URI host part instead of "anonymous.invalid" in the INVITE's From header for Tel-to-IP calls. ▪ **[0]** = (default) If the device receives a call from the Tel with blocked caller ID, it sends an INVITE with From: "anonymous"<anonymous@anonymous.invalid> ▪ **[1]** = The device's IP address is used as the URI host part instead of "anonymous.invalid". This parameter may be useful, for example, for service providers who identify their SIP Trunking customers by their source phone number or IP address, reflected in the From header of the SIP INVITE. Therefore, even customers blocking their Caller ID can be identified by the service provider. Typically, if the device receives a call with blocked Caller ID from the PSTN side (e.g., Trunk connected to a PBX), it sends an INVITE to the IP with a From header as follows: From: "anonymous" <anonymous@anonymous.invalid>. This is in accordance with RFC 3325. However, when this parameter is set to 1, the device replaces the "anonymous.invalid" with its IP address.
EMS: P Asserted User Name **[PAssertedUserName]**	Defines a 'representative number' (up to 50 characters) that is used as the user part of the Request-URI in the P-Asserted-Identity header of an outgoing INVITE (for Tel-to-IP calls). The default value is null.
EMS: Use URL In Refer To Header **[UseAORInReferToHeader]**	Defines the source for the SIP URI set in the Refer-To header of outgoing REFER messages. ▪ **[0]** = Use SIP URI from Contact header of the initial call (default). ▪ **[1]** = Use SIP URI from To/From header of the initial call.
Web: Enable User-Information Usage CLI: user-inf-usage **[EnableUserInfoUsage]**	Enables the usage of the User Information, which is loaded to the device in the User Information auxiliary file. (For a description on User Information, see 'Loading Auxiliary Files' on page 681.) ▪ **[0]** Disable (default). ▪ **[1]** Enable
[HandleReasonHeader]	Determines whether the device uses the value of the incoming SIP Reason header for Release Reason mapping. ▪ **[0]** Disregard Reason header in incoming SIP messages. ▪ **[1]** Use the Reason header value for Release Reason mapping (default).
[EnableSilenceSuppInSDP]	Determines the device's behavior upon receipt of SIP Re-INVITE messages that include the SDP's 'silencesupp:off' attribute. ▪ **[0]** = Disregard the 'silecesupp' attribute (default). ▪ **[1]** = Handle incoming Re-INVITE messages that include the 'silencesupp:off' attribute in the SDP as a request to switch to the Voice-Band-Data (VBD) mode. In addition, the device includes the attribute 'a=silencesupp:off' in its SDP offer. **Note:** This parameter is applicable only if the G.711 coder is used.
[EnableRport]	Enables the usage of the 'rport' parameter in the Via header.

Parameter	Description
	▪ **[0]** = Disabled (default). ▪ **[1]** = Enabled. The device adds an 'rport' parameter to the Via header of each outgoing SIP message. The first Proxy that receives this message sets the 'rport' value of the response to the actual port from where the request was received. This method is used, for example, to enable the device to identify its port mapping outside a NAT. If the Via header doesn't include the 'rport' parameter, the destination port of the response is obtained from the host part of the Via header. If the Via header includes the 'rport' parameter without a port value, the destination port of the response is the source port of the incoming request. If the Via header includes 'rport' with a port value (e.g., rport=1001), the destination port of the response is the port indicated in the 'rport' parmeter.
Web: Enable X-Channel Header EMS: X Channel Header CLI: x-channel-header **[XChannelHeader]**	Determines whether the SIP X-Channel header is added to SIP messages for providing information on the physical Trunk/B-channel on which the call is received or placed. ▪ **[0]** Disable = X-Channel header is not used (default). ▪ **[1]** Enable = X-Channel header is generated by the device and sent in INVITE messages and 180, 183, and 200 OK SIP responses. The header includes the Trunk number, B-channel, and the device's IP address. For example, 'x-channel: DS/DS1-5/8;IP=192.168.13.1', where: ✓ 'DS/DS-1' is a constant string ✓ '5' is the Trunk number ✓ '8' is the B-channel ✓ 'IP=192.168.13.1' is the device's IP address
Web/EMS: Progress Indicator to IP CLI: prog-ind-2ip **[ProgressIndicator2IP]**	For Analog (FXS/FXO) interfaces: ▪ **[-1**] Not Configured (default) = Default values are used. The default for FXO interfaces is 1; The default for FXS interfaces is 0. ▪ **[0]** No PI = For IP-to-Tel calls, the device sends a 180 Ringing response to IP after placing a call to a phone (FXS) or PBX (FXO). ▪ **[1]** PI = 1, **[8]** PI = 8: For IP-to-Tel calls, if the parameter EnableEarlyMedia is set to 1, the device sends a 183 Session Progress message with SDP immediately after a call is placed to a phone/PBX. This is used to cut-through the voice path before the remote party answers the call. This allows the originating party to listen to network Call Progress Tones (such as ringback tone or other network announcements). For Digital interfaces: ▪ **[-1**] Not Configured = for ISDN spans, the progress indicator (PI) that is received in ISDN Proceeding, Progress, and Alerting messages is used as described in the options below. (default) ▪ **[0]** No PI = For IP-to-Tel calls, the device sends 180 Ringing SIP response to IP after receiving ISDN Alerting or (for CAS) after placing a call to PBX/PSTN. ▪ **[1]** PI =1, **[8]** PI =8: For IP-to-Tel calls, if the parameter EnableEarlyMedia is set to 1, the device sends 180 Ringing with SDP in response to an ISDN Alerting or it sends a 183 Session Progress message with SDP in response to only the first received ISDN Proceeding or Progress message after a call is placed to PBX/PSTN over the trunk.

Parameter	Description
	Note: This parameter can also be configured per IP Profile (using the IPProfile parameter) and Tel Profile (using the TelProfile parameter).
[EnableRekeyAfter181]	Enables the device to send a Re-INVITE with a new (different) SRTP key (in the SDP) upon receipt of a SIP 181 response ("call is being forwarded"). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only if SRTP is used.
[NumberOfActiveDialogs]	Defines the maximum number of active SIP dialogs that are not call related (i.e., REGISTER and SUBSCRIBE). This parameter is used to control the Registration/Subscription rate. The valid range is 1 to 20. The default value is 20.
[TransparentCoderOnDataCall]	▪ **[0]** = Only use coders from the coder list (default). ▪ **[1]** = Use Transparent coder for data calls (according to RFC 4040). The Transparent' coder can be used on data calls. When the device receives a Setup message from the ISDN with 'TransferCapabilities = data', it can initiate a call using the coder 'Transparent' (even if the coder is not included in the coder list). The initiated INVITE includes the following SDP attribute: `a=rtpmap:97 CLEARMODE/8000` The default payload type is set according to the CodersGroup parameter. If the Transparent coder is not defined, the default value is set to 56. The payload type is negotiated with the remote side, i.e., the selected payload type is according to the remote side selection. The receiving device must include the 'Transparent' coder in its coder list.
Web: Enable IP2IP Application CLI: enable-ip2ip **[EnableIP2IPApplication]**	Enables the IP-to-IP Call Routing application. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
[IP2IPTranscodingMode]	Defines the voice transcoding mode (media negotiation) between two user agents for the IP-to-IP application. ▪ **[0]** Only if Required = Do not force transcoding. Many of the media settings (such as gain control) are not implemented on the voice stream. The device passes packets RTP to RTP packets without any processing. ▪ **[1]** Force = Force transcoding on the outgoing IP leg. The device interworks the media by implementing DSP transcoding. (default)
Web: Enable RFC 4117 Transcoding CLI: rfc4117-trnsc-enbl **[EnableRFC4117Transcoding]**	Enables transcoding of calls according to RFC 4117. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For more information on transcoding, see Transcoding using Third-Party Call Control on page 503.
Web/EMS: Default Release Cause	Defines the default Release Cause (sent to IP) for IP-to-Tel calls when the device initiates a call release and an explicit matching

Parameter	Description
CLI: dflt-release-cse **[DefaultReleaseCause]**	cause for this release is not found. The default release cause is NO_ROUTE_TO_DESTINATION (3). Other common values include NO_CIRCUIT_AVAILABLE (34), DESTINATION_OUT_OF_ORDER (27), etc. **Notes:** ▪ The default release cause is described in the Q.931 notation and is translated to corresponding SIP 40x or 50x values (e.g., 3 to SIP 404, and 34 to SIP 503). ▪ For analog interfaces: For information on mapping PSTN release causes to SIP responses, see Mapping PSTN Release Cause to SIP Response on page 302. ▪ When the Trunk is disconnected or is not synchronized, the internal cause is 27. This cause is mapped, by default, to SIP 502. ▪ For mapping SIP-to-Q.931 and Q.931-to-SIP release causes, see Configuring Release Cause Mapping on page 288. ▪ For a list of SIP responses-Q.931 release cause mapping, see 'Release Reason Mapping' on page 262.
Web: Enable Microsoft Extension CLI: microsoft-ext **[EnableMicrosoftExt]**	Enables the modification of the called and calling number for numbers received with Microsoft's proprietary "ext=xxx" parameter in the SIP INVITE URI user part. Microsoft Office Communications Server sometimes uses this proprietary parameter to indicate the extension number of the called or calling party. ▪ **[0]** Disable (default) ▪ **[1]** Enable For example, if a calling party makes a call to telephone number 622125519100 Ext. 104, the device receives the SIP INVITE (from Microsoft's application) with the URI user part as INVITE sip:622125519100;ext=104@10.1.1.10 (or INVITE tel:622125519100;ext=104). If the parameter EnableMicrosofExt is enabled, the device modifies the called number by adding an "e" as the prefix, removing the "ext=" parameter, and adding the extension number as the suffix (e.g., **e**622125519100**104**). Once modified, the device can then manipulate the number further, using the Number Manipulation tables to leave only the last 3 digits (for example) for sending to a PBX.
EMS: Use SIP URI For Diversion Header **[UseSIPURIForDiversionHeader]**	Defines the URI format in the SIP Diversion header. ▪ **[0]** = 'tel:' (default) ▪ **[1]** = 'sip:'
[TimeoutBetween100And18x]	Defines the timeout (in msec) between receiving a 100 Trying response and a subsequent 18x response. If a 18x response is not received within this timeout period, the call is disconnected. The valid range is 0 to 180,000 (i.e., 3 minutes). The default value is 32000 (i.e., 32 sec).
[EnableImmediateTrying]	Determines if and when the device sends a 100 Trying in response to an incoming INVITE request. ▪ **[0]** = 100 Trying response is sent upon receipt of a Proceeding message from the PSTN. ▪ **[1]** = 100 Trying response is sent immediately upon receipt of INVITE request (default).
[TransparentCoderPresentation]	Determines the format of the Transparent coder representation in the SDP.

Parameter	Description
	▪ **[0]** = clearmode (default) ▪ **[1]** = X-CCD
[IgnoreRemoteSDPMKI]	Determines whether the device ignores the Master Key Identifier (MKI) if present in the SDP received from the remote side. ▪ **[0]** Disable (default) ▪ **[1]** Enable
[TrunkStatusReportingMode]	Determines whether the device responds to SIP OPTIONS if all the trunks pertaining to Trunk Group #1 are down or busy. ▪ **[0]** Disable (default) ▪ **[1]** Enable = If all the trunks pertaining to Trunk Group #1 are down or busy, the device does not respond to received SIP OPTIONS.
Web: Comfort Noise Generation Negotiation EMS: Comfort Noise Generation CLI: com-noise-gen-nego **[ComfortNoiseNegotiation]**	Enables negotiation and usage of Comfort Noise (CN). ▪ **[0]** Disable ▪ **[1]** Enable (default) The use of CN is indicated by including a payload type for CN on the media description line of the SDP. The device can use CN with a codec whose RTP time stamp clock rate is 8,000 Hz (G.711/G.726). The static payload type 13 is used. The use of CN is negotiated between sides. Therefore, if the remote side doesn't support CN, it is not used. Regardless of the device's settings, it always attempts to adapt to the remote SIP UA's request for CNG, as described below. To determine CNG support, the device uses the ComfortNoiseNegotiation parameter and the codec's SCE (silence suppression setting) using the CodersGroup parameter. If the ComfortNoiseNegotiation parameter is enabled, then the following occurs: ▪ If the device is the initiator, it sends a "CN" in the SDP only if the SCE of the codec is enabled. If the remote UA responds with a "CN" in the SDP, then CNG occurs; otherwise, CNG does not occur. ▪ If the device is the receiver and the remote SIP UA does not send a "CN" in the SDP, then no CNG occurs. If the remote side sends a "CN", the device attempts to be compatible with the remote side and even if the codec's SCE is disabled, CNG occurs. If the ComfortNoiseNegotiation parameter is disabled, then the device does not send "CN" in the SDP. However, if the codec's SCE is enabled, then CNG occurs.
Web/EMS: First Call Ringback Tone ID CLI: 1st-call-rbt-id **[FirstCallRBTId]**	Defines the index of the first Ringback Tone in the CPT file. This option enables an Application server to request the device to play a distinctive Ringback tone to the calling party according to the destination of the call. The tone is played according to the Alert-Info header received in the 180 Ringing SIP response (the value of the Alert-Info header is added to the value of this parameter). The valid range is -1 to 1,000. The default value is -1 (i.e., play standard Ringback tone). **Notes:** ▪ It is assumed that all Ringback tones are defined in sequence in the CPT file. ▪ In case of an MLPP call, the device uses the value of this parameter plus 1 as the index of the Ringback tone in the CPT file

Parameter	Description
	(e.g., if this value is set to 1, then the index is 2, i.e., 1 + 1).
Web: Reanswer Time EMS: Regret Time CLI: reanswer-time **[RegretTime]**	For Analog interfaces: Defines the time interval from when the user hangs up the phone until the call is disconnected (FXS). This allows the user to hang up and then pick up the phone (before this timeout) to continue the call conversation. Thus, it's also referred to as regret time. For Digital interfaces: Defines the time period the device waits for an MFC R2 Resume (Reanswer) signal once a Suspend (Clear back) signal is received from the PBX. If this timer expires, the call is released. Note that this is applicable only to the MFC-R2 CAS Brazil variant. The valid range is 0 to 255 (in seconds). The default value is 0.
Web: Enable Reanswering Info CLI: reans-info-enbl **[EnableReansweringINFO]**	Enables the device to send a SIP INFO message with the On-Hook/Off-Hook parameter when the FXS phone goes on-hook during an ongoing call and then off-hook again, within the user-defined regret timeout (configured by the parameter RegretTime). Therefore, the device notifies the far-end that the call has been re-answered. ▪ **[0]** Disable (default) ▪ **[1]** Enable This parameter is typically implemented for incoming IP-to-Tel collect calls to the FXS port. If the FXS user does not wish to accept the collect call, the user disconnects the call by on-hooking the phone. The device notifies the softswitch (or Application server) of the unanswered collect call (on-hook) by sending a SIP INFO message. As a result, the softswitch disconnects the call (sends a BYE message to the device). If the call is a regular incoming call and the FXS user on-hooks the phone without intending to disconnect the call, the softswitch does not disconnect the call (during the regret time). The INFO message format is as follows: INFO sip:12345@10.50.228.164:5082 SIP/2.0 Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK_05_905924040-90579 From: <sip:+551137077803@ims.acme.com.br:5080;user=phone>;tag=008277765 To: <sip:notavailable@unknown.invalid>;tag=svw-0-1229428367 Call-ID: ConorCCR-0-LU-1229417827103300@dtas-stdn.fs5000group0-000.l CSeq: 1 INFO Contact: sip:10.20.7.70:5060 Content-Type: application/On-Hook (application/Off-Hook) Content-Length: 0 **Notes:** ▪ This parameter is applicable only if the parameter RegretTime is configured. ▪ This parameter is applicable only to FXS interfaces.
Web: PSTN Alert Timeout EMS: Trunk PSTN Alert Timeout CLI: pstn-alert-timeout **[PSTNAlertTimeout]**	For digital interfaces: Defines the Alert Timeout (in seconds) for calls sent to the PSTN. This timer is used between the time a Setup message is sent to the Tel side (IP-to-Tel call establishment) and a Connect message is received. If an Alerting message is received, the timer is restarted. If the timer expires before the call is answered, the device disconnects the call and sends a SIP 408 request timeout response to the SIP party that initiated the call. For analog interfaces: Defines the Alert Timeout (in seconds) for calls

Parameter	Description
	to the Tel side. This timer is used between the time a ring is generated (FXS) or a line is seized (FXO), until the call is connected. For example: If the FXS device receives an INVITE, it generates a ring to the phone and sends a SIP 180 Ringing response to the IP. If the phone is not answered within the time interval set by this parameter, the device cancels the call by sending a SIP 408 response. The valid value range is 1 to 600 (in seconds). The default is 180. **Note:** If per trunk configuration (using TrunkPSTNAlertTimeout) is set to other than default, the PSTNAlertTimeout parameter value is overridden.
Web: RTP Only Mode CLI: rtp-only-mode **[RTPOnlyMode]**	Enables the device to send and receive RTP packets to and from remote endpoints without the need to establish a SIP session. The remote IP address is determined according to the Outbound IP Routing table (Prefix parameter). The port is the same port as the local RTP port (configured by the BaseUDPPort parameter and the channel on which the call is received). ▪ **[0]** Disable (default) ▪ **[1]** Transmit & Receive = Send and receive RTP packets ▪ **[2]** Transmit Only= Send RTP packets only ▪ **[3]** Receive Only= Receive RTP packets only **Notes:** ▪ To activate the RTP Only feature without using ISDN / CAS signaling, you must do the following: ✓ Configure E1/T1 Transparent protocol type (set the ProtocoType parameter to 5 or 6). ✓ Enable the TDM-over-IP feature (set the EnableTDMoverIP parameter to 1). ▪ To configure the RTP Only mode per trunk, use the RTPOnlyModeForTrunk_ID parameter. ▪ If per trunk configuration (using the RTPOnlyModeForTrunk_ID parameter) is set to a value other than the default, the RTPOnlyMode parameter value is ignored.
[RTPOnlyModeForTrunk_ID]	Enables the RTP Only feature per trunk, where ID denotes the trunk number (0 is the first trunk). For more information, see the RTPOnlyMode parameter. **Note:** For using the global parameter (i.e., setting the RTP Only feature for all trunks), set this parameter to -1 (default).
Web/EMS: SIT Q850 Cause **[SITQ850Cause]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when a Special Information Tone (SIT) is detected on an IP-to-Tel call. The valid range is 0 to 127. The default value is 34. **Note:** For mapping specific SIT tones, you can use the SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO parameters.
Web/EMS: SIT Q850 Cause For NC **[SITQ850CauseForNC]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-NC (No Circuit Found Special Information Tone) is detected from the PSTN for IP-to-Tel calls. The valid range is 0 to 127. The default value is 34. **Note:** When not configured (i.e., default), the SITQ850Cause

Parameter	Description
	parameter is used.
Web/EMS: SIT Q850 Cause For IC **[SITQ850CauseForIC]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-IC (Operator Intercept Special Information Tone) is detected from the PSTN for IP-to-Tel calls. The valid range is 0 to 127. The default value is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
Web/EMS: SIT Q850 Cause For VC **[SITQ850CauseForVC]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-VC (Vacant Circuit - non-registered number Special Information Tone) is detected from the PSTN for IP-to-Tel calls. The valid range is 0 to 127. The default value is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
Web/EMS: SIT Q850 Cause For RO **[SITQ850CauseForRO]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-RO (Reorder - System Busy Special Information Tone) is detected from the PSTN for IP-to-Tel calls. The valid range is 0 to 127. The default value is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
[GWInboundManipulationSet]	Selects the Manipulation Set ID for manipulating all inbound INVITE messages. The Manipulation Set is defined using the MessageManipulations parameter. By default, no manipulation is done (i.e. Manipulation Set ID is set to -1). **Note:** This parameter is applicable only for the Gateway/IP2IP application.
[GWOutboundManipulationSet]	Selects the Manipulation Set ID for manipulating all outbound INVITE messages. The Manipulation Set is defined using the MessageManipulations parameter. By default, no manipulation is done (i.e. Manipulation Set ID is set to -1). **Notes:** ▪ This parameter is used only if the Outbound Message Manipulation Set parameter of the destination IP Group is not set. ▪ This parameter is applicable only for the Gateway/IP2IP application.
Out-of-Service (Busy Out) Parameters	
Web/EMS: Enable Busy Out CLI: busy-out **[EnableBusyOut]**	Enables the Busy Out feature. ▪ **[0]** Disable = 'Busy out' feature is not used (default). ▪ **[1]** Enable = 'Busy out' feature is enabled. When Busy Out is enabled and certain scenarios exist, the device performs the following: For analog interfaces: A reorder tone (configured by the parameter FXSOOSBehavior) is played when the phone is off-hooked. For digital interface: All E1/T1 trunks are automatically taken out of service by taking down the D-Channel or by sending a Service Out message for T1 PRI trunks supporting these messages (NI-2, 4/5-ESS, DMS-100, and Meridian). These behaviors are performed upon one of the following scenarios:

Parameter	Description
	▪ Physically disconnected from the network (i.e., Ethernet cable is disconnected). ▪ The Ethernet cable is connected, but the device can't communicate with any host. Note that LAN Watch-Dog must be activated (the parameter EnableLANWatchDog set to 1). ▪ The device can't communicate with the proxy (according to the Proxy Keep-Alive mechanism) and no other alternative route exists to send the call. ▪ The IP Connectivity mechanism is enabled (using the parameter AltRoutingTel2IPEnable) and there is no connectivity to any destination IP address. **Notes:** ▪ For Analog interfaces: The FXSOOSBehavior parameter determines the behavior of the FXS endpoints when a Busy Out or Graceful Lock occurs. ▪ For Analog interfaces: FXO endpoints during Busy Out and Lock are inactive. ▪ For Analog interfaces: See the LifeLineType parameter for complementary optional behavior. ▪ For Digital interfaces: The Busy Out behavior varies between different protocol types. ▪ For Digital interfaces: The Busy-Out condition can also be applied to a specific Trunk Group. If there is no connectivity to the Serving IP Group of a specific Trunk Group (defined in the Trunk Group Settings table), all the physical trunks pertaining to that Trunk Group are set to the Busy-Out condition. Each trunk uses the proper Out-Of-Service method according to the selected ISDN/CAS variant. ▪ For Digital interfaces: You can use the parameter DigitalOOSBehavior to select the method for setting digital trunks to Out-Of-Service.
Web: Out-Of-Service Behavior EMS:FXS OOS Behavior CLI: oos-behavior **[FXSOOSBehavior]**	Determines the behavior of undefined FXS endpoints and all FXS endpoints when a Busy Out condition exists. ▪ **[0]** None = Normal operation. No response is provided to undefined endpoints. A dial tone is played to FXS endpoints when a Busy Out condition exists. ▪ **[1]** Reorder Tone = The device plays a reorder tone to the connected phone/PBX (default). ▪ **[2]** Polarity Reversal = The device reverses the polarity of the endpoint marking it unusable (relevant, for example, for PBX DID lines). This option can't be configured on-the-fly. ▪ **[3]** Reorder Tone + Polarity Reversal = Same as 2 and 3 combined. This option can't be configured on-the-fly. ▪ **[4]** Current Disconnect = The device disconnects the current of the FXS endpoint. This option can't be configured on-the-fly. **Note:** This parameter is applicable only to FXS interfaces.
Retransmission Parameters	
Web: SIP T1 Retransmission Timer **[msec]** EMS: T1 RTX CLI: t1-re-tx-time	Defines the time interval (in msec) between the first transmission of a SIP message and the first retransmission of the same message. The default is 500. **Note:** The time interval between subsequent retransmissions of the

Parameter	Description
[SipT1Rtx]	same SIP message starts with SipT1Rtx. For INVITE requests, it is multiplied by two for each new retransmitted message. For all other SIP messages, it is multiplied by two until SipT2Rtx. For example, assuming SipT1Rtx = 500 and SipT2Rtx = 4000: ▪ The first retransmission is sent after 500 msec. ▪ The second retransmission is sent after 1000 (2*500) msec. ▪ The third retransmission is sent after 2000 (2*1000) msec. ▪ The fourth retransmission and subsequent retransmissions until SIPMaxRtx are sent after 4000 (2*2000) msec.
Web: SIP T2 Retransmission Timer **[msec]** EMS: T2 RTX CLI: t2-re-tx-time **[SipT2Rtx]**	Defines the maximum interval (in msec) between retransmissions of SIP messages (except for INVITE requests). The default is 4000. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx.
Web: SIP Maximum RTX EMS: Max RTX CLI: sip-max-rtx **[SIPMaxRtx]**	Defines the maximum number of UDP transmissions (first transmission plus retransmissions) of SIP messages. The range is 1 to 30. The default value is 7.
Web: Number of RTX Before Hot-Swap EMS: Proxy Hot Swap Rtx CLI: nb-of-rtx-b4-hot-swap **[HotSwapRtx]**	Defines the number of retransmitted INVITE/REGISTER messages before the call is routed (hot swap) to another Proxy/Registrar. The valid range is 1 to 30. The default value is 3. **Note:** This parameter is also used for alternative routing using the Outbound IP Routing Table'. If a domain name in the table is resolved into two IP addresses, and if there is no response for HotSwapRtx retransmissions to the INVITE message that is sent to the first IP address, the device immediately initiates a call to the second IP address.
Message Manipulations Table	
Web: Message Manipulations EMS: Message Manipulations CLI: configure voip > sbc manipulations message-manipulations **[MessageManipulations]**	This parameter table defines manipulation rules for SIP header messages. The format of this parameter is as follows: [MessageManipulations] FORMAT MessageManipulations_Index = MessageManipulations_ManSetID, MessageManipulations_MessageType, MessageManipulations_Condition, MessageManipulations_ActionSubject, MessageManipulations_ActionType, MessageManipulations_ActionValue, MessageManipulations_RowRole; [\MessageManipulations] For example, the below configuration changes the user part of the SIP From header to 200: MessageManipulations 1 = 0, Invite.Request, , Header.From.Url.User, 2, 200, 0; **Notes:** ▪ This table can include up to 200 indices (where 1 is the first index). ▪ You must enclose a string in a single apostrophe. If you are using multiple strings, then the entire string must also be enclosed in double apostrophe, for example, "<sip:' + header.from.url.user + '@domain.com>"'.

Parameter	Description
	▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, see 'Configuring Message Manipulations' on page 464. ▪ For a description on configuring ini file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.10 Coders and Profile Parameters

The profile parameters are described in the table below.

Table A-30: Profile Parameters

Parameter	Description
Coders Table / Coder Groups Table	
Web: Coders Table/Coder Group Settings EMS: Coders Group CLI: config voip > coders-and-profiles coders-group **[CodersGroup0]** **[CodersGroup1]** **[CodersGroup2]** **[CodersGroup3]** **[CodersGroup4]**	This *para*meter table defines the device's coders. Up to five groups of coders can be defined, where each group can consist of up to 10 coders. The first Coder Group is the default coder list and the default Coder Group. These Coder Groups can later be assigned to IP or Tel Profiles. The format of this parameter is as follows: **[CodersGroup0**] FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime, CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce; **[** \CodersGroup0] Where, ▪ Index = Coder entry 0-9, i.e., up to 10 coders per group. ▪ Name = Coder name. ▪ Ptime = Packetization time (ptime) - how many coder payloads are combined into a single RTP packet. ▪ Rate = Packetization rate. ▪ PayloadType = Identifies the format of the RTP payload. ▪ Sce = Enables silence suppression: ✓ **[0]** Disabled (default) ✓ **[1]** Enabled For example, below are defined two Coder Groups (0 and 1): **[CodersGroup0**] FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime, CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce; CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0; CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0; CodersGroup0 2 = eg711Ulaw, 10, 0, 71, 0; **[** \CodersGroup0] **[CodersGroup1**] FORMAT CodersGroup1_Index = CodersGroup1_Name, CodersGroup1_pTime, CodersGroup1_rate, CodersGroup1_PayloadType, CodersGroup1_Sce; CodersGroup1 0 = Transparent, 20, 0, 56, 0; CodersGroup1 1 = g726, 20, 0, 23, 0; **[** \CodersGroup1]

Parameter	Description

The table below lists the supported coders:

Coder Name	Packetization Time (msec)	Rate (kbps)	Payload Type	Silence Suppression
G.711 A-law **[g711Alaw64k]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	Always 64	Always 8	Disable **[0]** Enable **[1]**
G.711 U-law **[g711Ulaw64k]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	Always 64	Always 0	Disable **[0]** Enable **[1]**
G.711A-law_VBD **[g711AlawVbd]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	Always 64	Dynamic (0-127)	N/A
G.711U-law_VBD **[g711UlawVbd]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	Always 64	Dynamic (0-127)	N/A
G.722 **[g722]**	20 (default), 40, 60, 80, 100, 120	64 (default)	Always 9	N/A
G.723.1 **[g7231]**	30 (default), 60, 90, 120	5.3 **[0]** (default), 6.3 **[1]**	Always 4	Disable **[0]** Enable **[1]**
G.726 **[g726]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	16 **[0]**, 24 **[1]**, 32 **[2]** (default), 40 **[3]**	Dynamic (0-127) Default is 23	Disable **[0]** Enable **[1]**
G.727 ADPCM	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	16, 24, 32, 40	Dynamic (0-127)	Disable **[0]** Enable **[1]**
G.729 **[g729]**	10, 20 (default), 30, 40, 50, 60, 80, 100	Always 8	Always 18	Disable **[0]** Enable **[1]** Enable w/o Adaptations **[2]**
GSM-FR **[gsmFullRate]**	20 (default), 40, 60, 80	Always 13	Always 3	Disable **[0]** Enable **[1]**
GSM-EFR **[gsmEnhancedFullRate]**	0, 20 (default), 30, 40, 50, 60, 80, 100	12.2	Dynamic (0-127)	Disable **[0]** Enable **[1]**
MS-GSM **[gsmMS]**	40 (default)	Always 13	Always 3	Disable **[0]** Enable **[1]**
AMR **[Amr]**	20 (default)	4.75 **[0]**, 5.15 **[1]**, 5.90 **[2]**, 6.70 **[3]**, 7.40 **[4]**, 7.95 **[5]**, 10.2 **[6]**, 12.2 **[7]** (default)	Dynamic (0-127)	Disable **[0]** Enable **[1]**
QCELP **[QCELP]**	20 (default), 40, 60, 80, 100, 120	Always 13	Always 12	Disable **[0]** Enable **[1]**
EVRC **[Evrc]**	20 (default), 40,60, 80, 100	Variable **[0]** (default), 1/8 **[1]**, 1/2 **[3]**, Full **[4]**	Dynamic (0-127)	Disable **[0]** Enable **[1]**
iLBC **[iLBC]**	20 (default), 40, 60, 80, 100, 120	15 (default)	Dynamic (0-127)	Disable **[0]** Enable **[1]**

Parameter	Description

	30 (default), 60, 90, 120	13		
Transparent **[Transparent]**	10, 20 (default), 40, 60, 80, 100, 120	Always 64	Dynamic (0-127)	Disable **[0]** Enable **[1]**
T.38 **[t38fax]**	N/A	N/A	N/A	N/A

Notes:

- The coder name is case-sensitive.
- Each coder type can appear only once per Coder Group.
- Only the packetization time of the first coder in the defined coder list is declared in INVITE/200 OK SDP, even if multiple coders are defined.
- The device always uses the packetization time requested by the remote side for sending RTP packets. If not specified, the packetization time is assigned the default value.
- The value of several fields is hard-coded according to common standards (e.g., payload type of G.711 U-law is always 0). Other values can be set dynamically. If no value is specified for a dynamic field, a default value is assigned. If a value is specified for a hard-coded field, the value is ignored.
- If silence suppression is not defined for a specific coder, the value defined by the parameter EnableSilenceCompression is used.
- If G.729 is selected and silence suppression is enabled (for this coder), the device includes the string 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception is when the remote device is a Cisco gateway (IsCiscoSCEMode).
- The coder G.722 provides Packet Loss Concealment (PLC) capabilities, ensuring higher voice quality.
- Both GSM-FR and MS-GSM coders use Payload Type 3. When using SDP, it isn't possible to differentiate between the two. Therefore, it is recommended not to select both coders simultaneously.
- A Coder Group can be assigned to an IP Profile (using the IPProfile parameter) and/or to a Tel Profile (using the TelProfile parameter).
- For information on V.152 (and implementation of T.38 and VBD coders), see 'V.152 Support' on page 162.
- For a description of using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

IP Profile Table

Web: IP Profile Settings
EMS: Protocol Definition > IP Profile
CLI: config voip > coders-and-profiles ip-profile
[IPProfile]

This *para*meter table configures the IP Profile table. Each IP Profile ID includes a set of parameters (which are typically configured separately using their individual "global" parameters). You can later assign these IP Profiles to outbound IP routing rules (Prefix parameter), inbound IP routing rules (PSTNPrefix parameter), and IP Groups (IPGroup parameter).

The format of this parameter is as follows:
[IPProfile]
FORMAT IpProfile_Index = IpProfile_ProfileName, IpProfile_IpPreference, IpProfile_CodersGroupID, IpProfile_IsFaxUsed, IpProfile_JitterBufMinDelay, IpProfile_JitterBufOptFactor, IpProfile_IPDiffServ, IpProfile_SigIPDiffServ, IpProfile_SCE, IpProfile_RTPRedundancyDepth, IpProfile_RemoteBaseUDPPort, IpProfile_CNGmode, IpProfile_VxxTransportType, IpProfile_NSEMode, IpProfile_IsDTMFUsed, IpProfile_PlayRBTone2IP, IpProfile_EnableEarlyMedia,

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>IpProfile_ProgressIndicator2IP, IpProfile_EnableEchoCanceller, IpProfile_CopyDest2RedirectNumber, IpProfile_MediaSecurityBehaviour, IpProfile_CallLimit, IpProfile_DisconnectOnBrokenConnection, IpProfile_FirstTxDtmfOption, IpProfile_SecondTxDtmfOption, IpProfile_RxDTMFOption, IpProfile_EnableHold, IpProfile_InputGain, IpProfile_VoiceVolume, IpProfile_AddIEInSetup, IpProfile_SBCExtensionCodersGroupID, IpProfile_MediaIPVersionPreference, IpProfile_TranscodingMode, IpProfile_SBCAllowedCodersGroupID, IpProfile_SBCAllowedCodersMode, IpProfile_SBCMediaSecurityBehaviour, IpProfile_SBCRFC2833Behavior, IpProfile_SBCAlternativeDTMFMethod, IpProfile_SBCAssertIdentity, IpProfile_AMDSensitivityParameterSuit, IpProfile_AMDSensitivityLevel, IpProfile_AMDMaxGreetingTime, IpProfile_AMDMaxPostSilenceGreetingTime, IpProfile_SBCDiversionMode, IpProfile_SBCHistoryInfoMode, IpProfile_EnableQSIGTunneling, IpProfile_SBCFaxCodersGroupID, IpProfile_SBCFaxBehavior, IpProfile_SBCFaxOfferMode, IpProfile_SBCFaxAnswerMode;
[\IPProfile]
For example:
IPProfile 1 = ITSP, 1, 0, 0, 10, 10, 46, 40, 0, 0, 0, 0, 2, 0, 0, 0, 0, -1, 1, 0, 0, -1, 1, 4, -1, 1, 1, 0, 0, , -1, 0, 0, -1, 0, 0, 0, 0, -1, 0, 8, 300, 400, -1, -1;
Notes:
▪ You can configure up to nine IP Profiles (i.e., indices 1 through 9).
▪ To use the settings of the corresponding "global" parameter, enter the value -1 (or in the Web interface, the option 'Not Configured').
▪ For a detailed description of each parameter, see its corresponding global parameter:
<table><tr><th>IPProfile Field</th><th>Web Name</th><th>Global Parameter</th></tr><tr><td>IpProfile_Index</td><td>Profile ID</td><td>-</td></tr><tr><td>IpProfile_ProfileName</td><td>Profile Name</td><td>-</td></tr><tr><td>IpProfile_IpPreference</td><td>Profile Preference</td><td>-</td></tr><tr><td>IpProfile_CodersGroupID</td><td>Coder Group</td><td>CodersGroup</td></tr><tr><td>IpProfile_IsFaxUsed</td><td>Fax Signaling Method</td><td>IsFaxUsed</td></tr><tr><td>IpProfile_JitterBufMinDelay</td><td>Dynamic Jitter Buffer Minimum Delay</td><td>DJBufMinDelay</td></tr><tr><td>IpProfile_JitterBufOptFactor</td><td>Dynamic Jitter Buffer Optimization Factor</td><td>DJBufOptFactor</td></tr><tr><td>IpProfile_IPDiffServ</td><td>RTP IP DiffServ</td><td>PremiumServiceClassMediaDiffServ</td></tr><tr><td>IpProfile_SigIPDiffServ</td><td>Signaling DiffServ</td><td>PremiumServiceClassControlDiffServ</td></tr><tr><td>IpProfile_SCE</td><td>-</td><td>EnableSilenceCompression</td></tr><tr><td>IpProfile_RTPRedundancyDepth</td><td>RTP Redundancy Depth</td><td>RTPRedundancyDepth</td></tr></table></td></tr>
</table>

Parameter	Description		
	IpProfile_RemoteBaseUDPPort	Remote RTP Base UDP Port	RemoteBaseUDPPort
	IpProfile_CNGmode	CNG Detector Mode	CNGDetectorMode
	IpProfile_VxxTransportType	Modems Transport Type	V21ModemTransportType; V22ModemTransportType; V23ModemTransportType; V32ModemTransportType; V34ModemTransportType
	IpProfile_NSEMode	NSE Mode	NSEMode
	IpProfile_PlayRBTone2IP	Play Ringback Tone to IP	PlayRBTone2IP
	IpProfile_EnableEarlyMedia	Enable Early Media	EnableEarlyMedia
	IpProfile_ProgressIndicator2IP	Progress Indicator to IP	ProgressIndicator2IP
	IpProfile_EnableEchoCanceller	Echo Canceler	EnableEchoCanceller
	IpProfile_CopyDest2RedirectNumber	Copy Destination Number to Redirect Number	CopyDest2RedirectNumber
	IpProfile_MediaSecurityBehaviour	Media Security Behavior	MediaSecurityBehaviour
	IpProfile_CallLimit	Number of Calls Limit	-
	IpProfile_DisconnectOnBrokenConnection	Disconnect on Broken Connection	DisconnectOnBrokenConnection
	IpProfile_FirstTxDtmfOption	First Tx DTMF Option	TxDTMFOption
	IpProfile_SecondTxDtmfOption	Second Tx DTMF Option	TxDTMFOption
	IpProfile_RxDTMFOption	Declare RFC 2833 in SDP	RxDTMFOption
	IpProfile_EnableHold	Enable Hold	EnableHold
	IpProfile_InputGain	Input Gain	InputGain
	IpProfile_VoiceVolume	Voice Volume	VoiceVolume
	IpProfile_AddIEInSetup	Add IE In SETUP	AddIEinSetup
	IpProfile_SBCExtensionCodersGroupID	Extension Coders Group ID	SBCExtensionCodersGroupID
	IpProfile_MediaIPVersionPreference	Media IP Version Preference	MediaIPVersionPreference

Parameter	Description
	IpProfile_TranscodingMode — Transcoding Mode — TranscodingMode
	IpProfile_SBCAllowedCodersGroupID — Allowed Coders Group ID — -
	IpProfile_SBCAllowedCodersMode — Allowed Coders Mode — AllowedCodersGroup0
	IpProfile_SBCMediaSecurityBehaviour — - — SBCMediaSecurityBehaviour
	IpProfile_SBCRFC2833Behavior — - — -
	IpProfile_SBCAlternativeDTMFMethod — - — -
	IpProfile_SBCAssertIdentity — - — SBCAssertIdentity
	IpProfile_EnableQSIGTunneling — - — EnableQSIGTunneling
	IpProfile_AMDSensitivityParameterSuit — AMD Sensitivity Level — AMDSensitivityLevel
	IpProfile_AMDSensitivityLevel — AMD Sensitivity Level — AMDSensitivityLevel
	IpProfile_AMDMaxGreetingTime — AMD Max Greeting Time — AMDMaxGreetingTime
	IpProfile_AMDMaxPostSilenceGreetingTime — AMD Max Post Silence Greeting Time — AMDMaxPostGreetingSilenceTime
	IpProfile_SBCDiversionMode — Diversion Mode — -
	IpProfile_SBCHistoryInfoMode — History Info Mode — -

- The parameter IpPreference determines the priority of the IP Profile (1 to 20, where 20 is the highest preference). If both IP and Tel Profiles apply to the same call, the coders and common parameters (i.e., parameters configurable in both IP and Tel Profiles) of the preferred profile are applied to that call. If the Tel and IP Profiles are identical, the Tel Profile parameters take precedence.
- The parameter CallLimit defines the maximum number of concurrent calls allowed for that Profile. If the Profile is set to some limit, the device maintains the number of concurrent calls (incoming and outgoing) pertaining to the specific Profile. A limit value of **[-1]** indicates that there is no limitation on calls (default). A limit value of **[0]** indicates that all calls are rejected. When the number of concurrent calls is equal to the limit, the device rejects any new incoming and outgoing calls pertaining to that profile.
- RxDTMFOption configures the received DTMF negotiation method: **[-1]** not configured, use the global parameter; **[0]** don't declare RFC 2833; **[1]** declare RFC 2833 payload type is SDP.
- FirstTxDtmfOption and SecondTxDtmfOption configures the transmit DTMF negotiation method: **[-1]** not configured, use the global parameter; for the

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>remaining options, see the global parameter.
▪ The VxxTransportType parameter configures the modem transport type per IP Profile for the following parameters: V21ModemTransportType, V22ModemTransportType, V23ModemTransportType, V32ModemTransportType, and V34ModemTransportType.
▪ IP Profiles can also be used when operating with a Proxy server (set the parameter AlwaysUseRouteTable to 1).
▪ The following parameters are not applicable: IsDTMFUsed (deprecated), MediaIPVersionPreference
▪ For a description of using ini file table parameters, see 'Configuring ini File Table Parameters' on page 98.</td></tr>
<tr><td colspan="2">Tel Profile Table</td></tr>
<tr><td>Web: Tel Profile Settings
EMS: Protocol Definition > Telephony Profile
CLI: config voip > coders-and-profiles tel-profile
[TelProfile]</td><td>This parameter table configures the Tel Profile table. Each Tel Profile ID includes a set of parameters (which are typically configured separately using their individual, "global" parameters). You can later assign these Tel Profile IDs to other elements such as in the Trunk Group Table (TrunkGroup parameter). Therefore, Tel Profiles allow you to apply the same settings of a group of parameters to multiple channels, or apply specific settings to different channels. The format of this parameter is as follows:
[TelProfile]
FORMAT TelProfile_Index = TelProfile_ProfileName, TelProfile_TelPreference, TelProfile_CodersGroupID, TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay, TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ, TelProfile_SigIPDiffServ, TelProfile_DtmfVolume, TelProfile_InputGain, TelProfile_VoiceVolume, TelProfile_EnableReversePolarity, TelProfile_EnableCurrentDisconnect, TelProfile_EnableDigitDelivery, TelProfile_EnableEC, TelProfile_MWIAnalog, TelProfile_MWIDisplay, TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia, TelProfile_ProgressIndicator2IP, TelProfile_TimeForReorderTone, TelProfile_EnableDIDWink, TelProfile_IsTwoStageDial, TelProfile_DisconnectOnBusyTone, TelProfile_EnableVoiceMailDelay, TelProfile_DialPlanIndex, TelProfile_Enable911PSAP, TelProfile_SwapTelToIpPhoneNumbers, TelProfile_EnableAGC, TelProfile_ECNlpMode; TelProfile_DigitalCutThrough; [\TelProfile]
For example:
TelProfile 1 = ITSP_audio, 1, 0, 0, 10, 10, 46, 40, -11, 0, 0, 0, 0, 0, 1, 0, 0, 700, 0, -1, 255, 0, 1, 1, 1, -1, 1, 0, 0, 0, 0;
Notes:
▪ You can configure up to nine Tel Profiles (i.e., indices 1 through 9).
▪ To use the settings of the corresponding global parameter, enter the value -1 (or in the Web interface, the option 'Not Configured').
▪ For a detailed description of each parameter, see its corresponding "global" parameter:
<table><tr><th>TelProfile Field</th><th>Web Name</th><th>Global Parameter</th></tr><tr><td>TelProfile_ProfileName</td><td>Profile Name</td><td>-</td></tr><tr><td>TelProfile_TelPreference</td><td>Profile Preference</td><td>-</td></tr></table></td></tr>
</table>

Parameter	Description		
	TelProfile_CodersGroupID	Coder Group	CodersGroup0
	TelProfile_IsFaxUsed	Fax Signaling Method	IsFaxUsed
	TelProfile_JitterBufMinDelay	Dynamic Jitter Buffer Minimum Delay	DJBufMinDelay
	TelProfile_JitterBufOptFactor	Dynamic Jitter Buffer Optimization Factor	DJBufOptFactor
	TelProfile_IPDiffServ	RTP IP DiffServ	PremiumServiceClassMediaDiffServ
	TelProfile_SigIPDiffServ	Signaling DiffServ	PremiumServiceClassControlDiffServ
	TelProfile_DtmfVolume	DTMF Volume	DTMFVolume
	TelProfile_InputGain	Input Gain	InputGain
	TelProfile_VoiceVolume	Voice Volume	VoiceVolume
	TelProfile_EnableReversePolarity	Enable Polarity Reversal	EnableReversalPolarity
	TelProfile_EnableCurrentDisconnect	Enable Current Disconnect	EnableCurrentDisconnect
	TelProfile_EnableDigitDelivery	Enable Digit Delivery	EnableDigitDelivery
	TelProfile_EnableEC	Echo Canceler	EnableEchoCanceller
	TelProfile_MWIAnalog	MWI Analog Lamp	MWIAnalogLamp
	TelProfile_MWIDisplay	MWI Display	MWIDisplay
	TelProfile_FlashHookPeriod	Flash Hook Period	FlashHookPeriod
	TelProfile_EnableEarlyMedia	Enable Early Media	EnableEarlyMedia
	TelProfile_ProgressIndicator2IP	Progress Indicator to IP	ProgressIndicator2IP
	TelProfile_TimeForReorderTone	Time For Reorder Tone	TimeForReorderTone
	TelProfile_EnableDIDWink	Enable DID Wink	EnableDIDWink
	TelProfile_IsTwoStageDial	Dialing Mode	IsTwoStageDial
	TelProfile_DisconnectOnBusyTone	Disconnect Call on Detection of Busy Tone	DisconnectOnBusyTone

Parameter	Description
	TelProfile_EnableVoiceMailDelay — Enable Voice Mail Delay — -
	TelProfile_DialPlanIndex — Dial Plan Index — DialPlanIndex
	TelProfile_Enable911PSAP — Enable 911 PSAP — Enable911PSAP
	TelProfile_SwapTelToIpPhoneNumbers — Swap Tel To IP Phone Numbers — SwapTEl2IPCalled&Calling Numbers
	TelProfile_EnableAGC — Enable AGC — EnableAGC
	TelProfile_ECNlpMode — EC NLP Mode — ECNLPMode
	TelProfile_DigitalCutThrough — - — DigitalCutThrough
	▪ The following parameters are applicable only to analog interfaces: EnableReversePolarity, EnableCurrentDisconnect, MWIAnalog, MWIDisplay, EnableDIDWink, IsTwoStageDial, DisconnectOnBusyTone, and Enable911PSAP. ▪ The parameter IpPreference determines the priority of the Tel Profile (1 to 20, where 20 is the highest preference). If both IP and Tel Profiles apply to the same call, the coders and common parameters (i.e., parameters configurable in both IP and Tel Profiles) of the preferred profile are applied to that call. If the Tel and IP Profiles are identical, the Tel Profile parameters take precedence. ▪ The parameter EnableVoiceMailDelay is applicable only if voice mail is enabled globally (using the VoiceMailInterface parameter). ▪ For a description of using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.11 Channel Parameters

This subsection describes the device's channel parameters.

A.11.1 Voice Parameters

The voice parameters are described in the table below.

Table A-31: Voice Parameters

Parameter	Description
Web/EMS: Input Gain CLI: input-gain **[InputGain]**	Defines the pulse-code modulation (PCM) input gain control (in decibels). This parameter sets the level for the received (Tel/PSTN-to-IP) signal. The valid range is -32 to 31 dB. The default value is 0 dB. **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248) and per Tel Profile, using the TelProfile parameter (see 'Configuring Tel Profiles' on page 246).
Web: Voice Volume EMS: Volume (dB) CLI: voice-volume **[VoiceVolume]**	Defines the voice gain control (in decibels). This parameter sets the level for the transmitted (IP-to-Tel/PSTN) signal. The valid range is -32 to 31 dB. The default value is 0 dB. **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248) and per Tel Profile, using the TelProfile parameter (see 'Configuring Tel Profiles' on page 246).
EMS: Payload Format CLI: G726-voice-payload-format **[VoicePayloadFormat]**	Determines the bit ordering of the G.726/G.727 voice payload format. ▪ **[0]** = Little Endian (default) ▪ **[1]** = Big Endian **Note:** To ensure high voice quality when using G.726/G.727, both communicating ends should use the same endianness format. Therefore, when the device communicates with a third-party entity that uses the G.726/G.727 voice coder and voice quality is poor, change the settings of this parameter (between Big Endian and Little Endian).
Web: MF Transport Type CLI: MF-transport-type **[MFTransportType]**	Currently, not supported.
Web: Enable Answer Detector **[EnableAnswerDetector]**	Currently, not supported.
Web: Answer Detector Activity Delay CLI: answer-detector-activativity-delay **[AnswerDetectorActivityDelay]**	Defines the time (in 100-msec resolution) between activating the Answer Detector and the time that the detector actually starts to operate. The valid range is 0 to 1023. The default is 0.
Web: Answer Detector Silence Time **[AnswerDetectorSilenceTime]**	Currently, not supported.
Web: Answer Detector Redirection	Currently, not supported.

Parameter	Description
[AnswerDetectorRedirection]	
Web: Answer Detector Sensitivity EMS: Sensitivity CLI: answer-detector-sensitivity **[AnswerDetectorSensitivity]**	Defines the Answer Detector sensitivity. The range is 0 (most sensitive) to 2 (least sensitive). The default is 0.
Web: Silence Suppression EMS: Silence Compression Mode CLI: silence-compression-mode **[EnableSilenceCompression]**	Determines the Silence Suppression support. Silence Suppression is a method for conserving bandwidth on VoIP calls by not sending packets when silence is detected. ▪ **[0]** Disable = Silence Suppression is disabled (default). ▪ **[1]** Enable = Silence Suppression is enabled. ▪ **[2]** Enable without Adaptation = A single silence packet is sent during a silence period (applicable only to G.729). **Note:** If the selected coder is G.729, the value of the 'annexb' parameter of the fmtp attribute in the SDP is determined by the following rules: ▪ If EnableSilenceCompression is 0: 'annexb=no'. ▪ If EnableSilenceCompression is 1: 'annexb=yes'. ▪ If EnableSilenceCompression is 2 and IsCiscoSCEMode is 0: 'annexb=yes'. ▪ If EnableSilenceCompression is 2 and IsCiscoSCEMode is 1: 'annexb=no'. **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: Echo Canceler EMS: Echo Canceller Enable CLI: echo-canceller-enable **[EnableEchoCanceller]**	Enables echo cancellation (i.e., echo from voice calls is removed). ▪ **[0]** Disable ▪ **[1]** Enable (default) **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248) and per Tel Profile, using the TelProfile parameter (see 'Configuring Tel Profiles' on page 246).

Parameter	Description
Web: Max Echo Canceller Length CLI: max-echo-canceller-length **[MaxEchoCancellerLength]**	Defines the maximum Echo Canceler Length (in msec), which is the maximum echo path delay (tail length) for which the echo canceller is designed to operate: ▪ **[0]** Default = based on various internal device settings to attain maximum channel capacity (default) ▪ **[11]** 64 msec ▪ **[22]** 128 msec **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Using 128 msec may reduce channel capacity. For example: with DSP Template 0 and number of spans 4, the capacity is reduced from 120 to 100. The reduction depends on the combination of "DSP Template" and "Number of Spans". For accurate figures, see DSP Templates on page 1061. ▪ When housed with an analog/BRI module, the device can use a max. echo canceller length of 64 msec. ▪ When housed with PRI TRUNKS module, the device can use a max. echo canceller length of 128 msec. ▪ When housed with an MPM module (in Slot #6), no channel reduction occurs. ▪ It is unnecessary to configure the parameter EchoCancellerLength, as it automatically acquires its value from this parameter.
EMS: Echo Canceller Hybrid Loss CLI: echo-canceller-hybrid-loss **[ECHybridLoss]**	Defines the four-wire to two-wire worst-case Hybrid loss, the ratio between the signal level sent to the hybrid and the echo level returning from the hybrid. ▪ **[0]** = 6 dB (default) ▪ **[1]** = N/A ▪ **[2]** = 0 dB ▪ **[3]** = 3 dB
CLI: echo-canceller-NLP-mode **[ECNLPMode]**	Defines the echo cancellation Non-Linear Processing (NLP) mode. ▪ **[0]** = NLP adapts according to echo changes (default). ▪ **[1]** = Disables NLP. ▪ **[2]** = Silence output NLP. **Note:** This parameter can also be configured per Tel Profile, using the TelProfile parameter (see 'Configuring Tel Profiles' on page 246).
CLI: echo-canceller-aggressive-NLP **[EchoCancellerAggressiveNLP]**	Enables the Aggressive NLP at the first 0.5 second of the call. ▪ **[0]** = Disable ▪ **[1]** = Enable (default). The echo is removed only in the first half of a second of the incoming IP signal. **Note:** For this parameter to take effect, a device reset is required.
CLI: number-of-SID-coefficients **[RTPSIDCoeffNum]**	Defines the number of spectral coefficients added to an SID packet being sent according to RFC 3389. The valid values are **[0]** (default), **[4]**, **[6]**, **[8]** and **[10]**.

A.11.2 Coder Parameters

The coder parameters are described in the table below.

Table A-32: Coder Parameters

Parameter	Description
CLI: EVRC-VAD-enable **[EnableEVRCVAD]**	Enables the EVRC voice activity detector. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** Supported for EVRC and EVRC-B coders.
EMS: VBR Coder DTX Min CLI: EVRC-dtx-min **[EVRCDTXMin]**	Defines the minimum gap between two SID frames when using the EVRC voice activity detector. Units are in EVRC frame size (20 msec). The range is 0 to 20000. The default value is 12. **Note:** Supported for EVRC and EVRC-B coders.
EMS: VBR Coder DTX Max CLI: EVRC-dtx-max **[EVRCDTXMax]**	Defines the maximum gap between two SID frames when using the EVRC voice activity detector. Units are in EVRC frame size (20 msec). The range is 0 to 20000. The default value is 32. **Note:** This parameter is applicable only to EVRC and EVRC-B coders.
EMS: VBR Coder Header Format CLI: VBR-coder-header-format **[VBRCoderHeaderFormat]**	Determines the format of the RTP header for VBR coders. ▪ **[0]** = Payload only (no header, TOC, or m-factor) - similar to RFC 3558 Header Free format (default). ▪ **[1]** = Supports RFC 2658 - 1 byte for interleaving header (always 0), TOC, no m-factor. ▪ **[2]** = Payload including TOC only, allow m-factor. ▪ **[3]** = RFC 3558 Interleave/Bundled format.
EMS: VBR Coder Hangover CLI: VBR-coder-hangover **[VBRCoderHangover]**	Defines the required number of silence frames at the beginning of each silence period when using the VBR coder silence suppression. The range is 0 to 255. The default value is 1.
Web: DSP Template Mix Table EMS: VoP Media Provisioning > General Settings	
[DSPTemplates]	This parameter table allows the device to use a combination of two DSP templates and determines the percentage of DSP resources allocated per DSP template. The format of this parameter is as follows: **[DspTemplates]** FORMAT DspTemplates_Index = DspTemplates_DspTemplateNumber, DspTemplates_DspResourcesPercentage; **[\DspTemplates]** For example, to load DSP Template 1 to 50% of the DSPs, and DSP Template 2 to the remaining 50%, the table is configured as follows: DspTemplates 0 = 1, 50; DspTemplates 1 = 2, 50; **Notes:**

Parameter	Description
	▪ The DSPVersionTemplateNumber parameter is ignored when the DSPTemplates parameter is configured. ▪ For a list of supported DSP templates, see DSP Templates on page 1061.
Web: DSP Version Template Number EMS: Version Template Number CLI: DSP-version-template-number **[DSPVersionTemplateNumber]**	Determines the DSP template used by the device. Each DSP template supports specific coders, channel capacity, and features. The default is DSP template 0. You can load different DSP templates to analog and digital modules using the syntax DSPVersionTemplateNumber=xy where: ▪ x = 0 or 1 for DSP templates of analog modules ▪ y = 0 to 5 for DSP templates of digital and MPM modules **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a list of supported DSP templates, see DSP Templates on page 1061.

A.11.3 DTMF Parameters

The dual-tone multi-frequency (DTMF) parameters are described in the table below.

Table A-33: DTMF Parameters

Parameter	Description
Web/EMS: DTMF Transport Type CLI: DTMF-transport-type **[DTMFTransportType]**	Determines the DTMF transport type. ▪ **[0]** DTMF Mute = Erases digits from voice stream and doesn't relay to remote. ▪ **[2]** Transparent DTMF = Digits remain in voice stream. ▪ **[3]** RFC 2833 Relay DTMF = Erases digits from voice stream and relays to remote according to RFC 2833 (default). ▪ **[7]** RFC 2833 Relay Rcv Mute = DTMFs are sent according to RFC 2833 and muted when received. **Note:** This parameter is automatically updated if the parameters TxDTMFOption or RxDTMFOption are configured.
Web: DTMF Volume (-31 to 0 dB) EMS: DTMF Volume (dBm) CLI: DTMF-volume **[DTMFVolume]**	Defines the DTMF gain control value (in decibels) to the PSTN or analog side. The valid range is -31 to 0 dB. The default value is -11 dB. **Note:** This parameter can also be configured per Tel Profile, using the TelProfile parameter.
Web: DTMF Generation Twist EMS: DTMF Twist Control CLI: DTMF-generation-twist **[DTMFGenerationTwist]**	Defines the range (in decibels) between the high and low frequency components in the DTMF signal. Positive decibel values cause the higher frequency component to be stronger than the lower one. Negative values cause the opposite effect. For any parameter value, both components change so that their average is constant. The valid range is -10 to 10 dB. The default value is 0 dB. **Note:** For this parameter to take effect, a device reset is required.
EMS: DTMF Inter Interval (msec) CLI: inter-digit-interval	Defines the time (in msec) between generated DTMF digits to PSTN side (if TxDTMFOption = 1, 2 or 3). The default value is 100 msec. The valid range is 0 to 32767.

Parameter	Description
[DTMFInterDigitInterval]	
EMS: DTMF Length (msec) **[DTMFDigitLength]**	Defines the time (in msec) for generating DTMF tones to the PSTN side (if TxDTMFOption = 1, 2 or 3). It also configures the duration that is sent in INFO (Cisco) messages. The valid range is 0 to 32767. The default value is 100.
EMS: Rx DTMF Relay Hang Over Time (msec) CLI: default-dtmf-signal-duration **[RxDTMFHangOverTime]**	Defines the Voice Silence time (in msec) after playing DTMF or MF digits to the Tel/PSTN side that arrive as Relay from the IP side. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
EMS: Tx DTMF Relay Hang Over Time (msec)| CLI: digit-hangover-time-tx **[TxDTMFHangOverTime]**	Defines the Voice Silence time (in msec) after detecting the end of DTMF or MF digits at the Tel/PSTN side when the DTMF Transport Type is either Relay or Mute. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
Web/EMS: NTE Max Duration CLI: telephony-events-max-duration **[NTEMaxDuration]**	Defines the maximum time for sending Named Telephony Events / NTEs (RFC 4733/2833 DTMF relay) to the IP side, regardless of the DTMF signal duration on the TDM side. The range is -1 to 200,000,000 msec. The default is -1 (i.e., NTE stops only upon detection of an End event).

A.11.4 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

Table A-34: RTP/RTCP and T.38 Parameters

Parameter	Description
Web: Dynamic Jitter Buffer Minimum Delay EMS: Minimal Delay (dB) CLI: jitter-buffer-minimum-delay **[DJBufMinDelay]**	Defines the minimum delay (in msec) for the Dynamic Jitter Buffer. The valid range is 0 to 150. The default delay is 10. **Notes:** ▪ This parameter can also be configured per IP Profile or Tel Profile, using the IPProfile and TelProfile parameters respectively. ▪ For more information on Jitter Buffer, see 'Dynamic Jitter Buffer Operation' on page 165.
Web: Dynamic Jitter Buffer Optimization Factor EMS: Opt Factor CLI: jitter-buffer-optimization-factor **[DJBufOptFactor]**	Defines the Dynamic Jitter Buffer frame error/delay optimization factor. The valid range is 0 to 12. The default factor is 10. **Notes:** ▪ For data (fax and modem) calls, set this parameter to 12. ▪ This parameter can also be configured per IP Profile or Tel Profile, using the IPProfile and TelProfile parameters respectively. ▪ For more information on Jitter Buffer, see 'Dynamic Jitter Buffer Operation' on page 165.
Web/EMS: Analog Signal Transport Type	Determines the analog signal transport type. ▪ **[0]** Ignore Analog Signals = Ignore (default).

Parameter	Description
[AnalogSignalTransportType]	▪ **[1]** RFC 2833 Analog Signal Relay = Transfer hookflash using RFC 2833.
Web: RTP Redundancy Depth EMS: Redundancy Depth CLI: RTP-redundancy-depth **[RTPRedundancyDepth]**	Enables the device to generate RFC 2198 redundant packets. This can be used for packet loss where the missing information (audio) can be reconstructed at the receiver's end from the redundant data that arrives in subsequent packets. This is required, for example, in wireless networks where a high percentage (up to 50%) of packet loss can be experienced. ▪ **[0]** 0 = Disable (default) ▪ **[1]** 1 = Enable - previous voice payload packet is added to current packet. **Notes:** ▪ When enabled, you can configure the payload type, using the RFC2198PayloadType parameter. ▪ The RTP redundancy dynamic payload type can be included in the SDP, by using the EnableRTPRedundancyNegotiation parameter. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter.
Web: Enable RTP Redundancy Negotiation CLI: rtp-rdcy-nego-enbl **[EnableRTPRedundancyNegotiation]**	Enables the device to included the RTP redundancy dynamic payload type in the SDP, according to RFC 2198. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, the device includes in the SDP message the RTP payload type "RED" and the payload type configured by the parameter RFC2198PayloadType. `a=rtpmap:<PT> RED/8000` Where *<PT>* is the payload type as defined by RFC2198PayloadType. The device sends the INVITE message with "a=rtpmap:<PT> RED/8000" and responds with a 18x/200 OK and "a=rtpmap:<PT> RED/8000" in the SDP. **Notes:** ▪ For this feature to be functional, you must also set the parameter RTPRedundancyDepth to 1 (i.e., enabled). ▪ Currently, the negotiation of “RED” payload type is not supported and therefore, it should be configured to the same PT value for both parties.
Web: RFC 2198 Payload Type EMS: Redundancy Payload Type CLI: RTP-redundancy-payload-type **[RFC2198PayloadType]**	Defines the RTP redundancy packet payload type according to RFC 2198. The range is 96 to 127. The default is 104. **Note:** This parameter is applicable only if the parameter RTPRedundancyDepth is set to 1.
Web: Packing Factor EMS: Packetization Factor **[RTPPackingFactor]**	N/A. Controlled internally by the device according to the selected coder.
Web/EMS: Basic RTP Packet Interval **[BasicRTPPacketInterval]**	N/A. Controlled internally by the device according to the selected coder.

Parameter	Description
Web: RTP Directional Control **[RTPDirectionControl]**	N/A. Controlled internally by the device according to the selected coder.
Web/EMS: RFC 2833 TX Payload Type CLI: telephony-events-payload-type-tx **[RFC2833TxPayloadType]**	Defines the Tx RFC 2833 DTMF relay dynamic payload type. The valid range is 96 to 99, and 106 to 127. The default is 96. The 100, 102 to 105 range is allocated for proprietary usage. **Notes:** ▪ Certain vendors (e.g., Cisco) use payload type 101 for RFC 2833. ▪ When RFC 2833 payload type negotiation is used (i.e., the parameter TxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.
Web/EMS: RFC 2833 RX Payload Type CLI: telephony-events-payload-type-rx **[RFC2833RxPayloadType]**	Defines the Rx RFC 2833 DTMF relay dynamic payload type. The valid range is 96 to 99, and 106 to 127. The default is 96. The 100, 102 to 105 range is allocated for proprietary usage. **Notes:** ▪ Certain vendors (e.g., Cisco) use payload type 101 for RFC 2833. ▪ When RFC 2833 payload type negotiation is used (i.e., the parameter TxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.
[EnableDetectRemoteMACChange]	Determines whether the device changes the RTP packets according to the MAC address of received RTP packets and according to Gratuitous Address Resolution Protocol (GARP) messages. ▪ **[0]** = Nothing is changed. ▪ **[1]** = If the device receives RTP packets with a different source MAC address (than the MAC address of the transmitted RTP packets), then it sends RTP packets to this MAC address and removes this IP entry from the device's ARP cache table. ▪ **[2]** = The device uses the received GARP packets to change the MAC address of the transmitted RTP packets (default). ▪ **[3]** = Options 1 and 2 are used. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If the device is located in a network subnet which is connected to other gateways using a router that uses Virtual Router Redundancy Protocol (VRRP) for redundancy, then set this parameter to 0 or 2.

Parameter	Description
Web: RTP Base UDP Port EMS: Base UDP Port **[BaseUDPport]**	Defines the lower boundary of the UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2). For example, if the Base UDP Port is set to 6000, then one channel may use the ports RTP 6000, RTCP 6001, and T.38 6002, while another channel may use RTP 6010, RTCP 6011, and T.38 6012, and so on. The range of possible UDP ports is 6,000 to 64,000. The default base UDP port is 6000. Once this parameter is configured, the UDP port range (lower to upper boundary) is calculated as follows: ▪ BaseUDPport to (BaseUDPport + 329*10) **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The UDP ports are allocated randomly to channels. ▪ You can define a UDP port range per Media Realm (see Configuring Media Realms on page 181). ▪ If RTP Base UDP Port is not a factor of 10, the following message is generated: 'invalid local RTP port'. ▪ For more information on the default RTP/RTCP/T.38 port allocation, refer to the *Product Reference Manual*.
EMS: No Op Enable CLI: no-operation-enable **[NoOpEnable]**	Enables the transmission of RTP or T.38 No-Op packets. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods.
EMS: No Op Interval **[NoOpInterval]**	Defines the time interval in which RTP or T.38 No-Op packets are sent in the case of silence (no RTP/T.38 traffic) when No-Op packet transmission is enabled. The valid range is 20 to 65,000 msec. The default is 10,000. **Note:** To enable No-Op packet transmission, use the NoOpEnable parameter.
EMS: No Op Payload Type CLI: no-operation-interval **[RTPNoOpPayloadType]**	Defines the payload type of No-Op packets. The valid range is 96 to 127 (for the range of Dynamic RTP Payload Type for all types of non hard-coded RTP Payload types, refer to RFC 3551). The default value is 120. **Note:** When defining this parameter, ensure that it doesn't cause collision with other payload types.
CLI: rtcp-act-mode **[RTCPActivationMode]**	Disables RTCP traffic when there is no RTP traffic. This feature is useful, for example, to stop RTCP traffic that is typically sent when calls are put on hold (by an INVITE with 'a=inactive' in the SDP). ▪ **[0]** Active Always = RTCP is active even during inactive RTP periods, i.e., when the media is in 'recvonly' or 'inactive' mode. (default) ▪ **[1]** Inactive Only If RTP Inactive = No RTCP is sent when RTP is inactive.

Parameter	Description
RTP Control Protocol Extended Reports (RTCP XR) Parameters (**Note:** For a detailed description of RTCP XR reports, refer to the Product Reference Manual.)	
Web: Enable RTCP XR EMS: RTCP XR Enable CLI: voice-quality-monitoring-enable **[VQMonEnable]**	Enables voice quality monitoring and RTCP Extended Reports (RTCP XR), according to Internet-Draft draft-ietf-sipping-rtcp-summary-13. ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enables **Note:** For this parameter to take effect, a device reset is required.
Web: Minimum Gap Size EMS: GMin **[VQMonGMin]**	Defines the voice quality monitoring - minimum gap size (number of frames). The default is 16.
Web/EMS: Burst Threshold **[VQMonBurstHR]**	Defines the voice quality monitoring - excessive burst alert threshold. if set to -1 (default), no alerts are issued.
Web/EMS: Delay Threshold **[VQMonDelayTHR]**	Defines the voice quality monitoring - excessive delay alert threshold. if set to -1 (default), no alerts are issued.
Web: R-Value Delay Threshold EMS: End of Call Rval Delay Threshold **[VQMonEOCRValTHR]**	Defines the voice quality monitoring - end of call low quality alert threshold. if set to -1 (default), no alerts are issued.
Web: RTCP XR Packet Interval EMS: Packet Interval CLI: rtcp-interval **[RTCPInterval]**	Defines the time interval (in msec) between adjacent RTCP reports. The interval range is 0 to 65,535. The default interval is 5,000.
Web: Disable RTCP XR Interval Randomization EMS: Disable Interval Randomization CLI: disable-RTCP-randomization **[DisableRTCPRandomize]**	Determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval. ▪ **[0]** Disable = Randomize (default) ▪ **[1]** Enable = No Randomize
EMS: RTCP XR Collection Server Transport Type **[RTCPXRESCTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the RTCP-XR Collection Server. ▪ **[-1**] Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web: RTCP XR Collection Server EMS: Esc IP CLI: rtcp-xr-coll-srvr **[RTCPXREscIP]**	Defines the IP address of the Event State Compositor (ESC). The device sends RTCP XR reports to this server, using SIP PUBLISH messages. The address can be configured as a numerical IP address or as a domain name.

Parameter	Description
Web: RTCP XR Report Mode EMS: Report Mode CLI: rtcp-xr-rep-mode **[RTCPXRReportMode]**	Determines whether RTCP XR reports are sent to the Event State Compositor (ESC) and defines the interval in which they are sent. ▪ **[0]** Disable = RTCP XR reports are not sent to the ESC (default). ▪ **[1]** End Call = RTCP XR reports are sent to the ESC at the end of each call. ▪ **[2]** End Call & Periodic = RTCP XR reports are sent to the ESC at the end of each call and periodically according to the parameter RTCPInterval.

A.12 Gateway and IP-to-IP Parameters

A.12.1 Fax and Modem Parameters

The fax and modem parameters are described in the table below.

Table A-35: Fax and Modem Parameters

Parameter	Description
Web: Fax Transport Mode EMS: Transport Mode CLI: fax-transport-mode **[FaxTransportMode]**	Determines the fax transport mode used by the device. ▪ **[0]** Disable = transparent mode ▪ **[1]** T.38 Relay (default) ▪ **[2]** Bypass ▪ **[3]** Events Only **Note:** This parameter is overridden by the parameter IsFaxUsed. If the parameter IsFaxUsed is set to 1 (T.38 Relay) or 3 (Fax Fallback), then FaxTransportMode is always set to 1 (T.38 relay).
Web: V.21 Modem Transport Type EMS: V21 Transport CLI: V21-modem-transport-type **[V21ModemTransportType]**	Determines the V.21 modem transport type. ▪ **[0]** Disable = Disable (Transparent) - default ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass. ▪ **[3]** Events Only = Transparent with Events **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: V.22 Modem Transport Type EMS: V22 Transport CLI: V22-modem-transport-type **[V22ModemTransportType]**	Determines the V.22 modem transport type. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: V.23 Modem Transport Type EMS: V23 Transport CLI: V23-modem-transport-type	Determines the V.23 modem transport type. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default)

Parameter	Description
[V23ModemTransportType]	▪ **[3]** Events Only = Transparent with Events **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: V.32 Modem Transport Type EMS: V32 Transport CLI: V32-modem-transport-type **[V32ModemTransportType]**	Determines the V.32 modem transport type. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events **Notes:** ▪ This parameter applies only to V.32 and V.32bis modems. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: V.34 Modem Transport Type EMS: V34 Transport CLI: V34-modem-transport-type **[V34ModemTransportType]**	Determines the V.90/V.34 modem transport type. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
EMS: Bell Transport Type CLI: bell-modem-transport-type **[BellModemTransportType]**	Determines the Bell modem transport method. ▪ **[0]** = Transparent (default) ▪ **[2]** = Bypass ▪ **[3]** = Transparent with events
Web/EMS: Fax CNG Mode **[FaxCNGMode]**	Determines the device's behavior upon detection of a CNG tone. ▪ **[0]** = Does not send a SIP Re-INVITE upon detection of a fax CNG tone when the parameter CNGDetectorMode is set to 1 (default). ▪ **[1]** = Sends a SIP Re-INVITE upon detection of a fax CNG tone when the parameter CNGDetectorMode is set to 1.
Web/EMS: CNG Detector Mode CLI: coder **[CNGDetectorMode]**	Determines whether the device detects the fax Calling tone (CNG). ▪ **[0]** Disable = The originating device doesn't detect CNG; the CNG signal passes transparently to the remote side (default). ▪ **[1]** Relay = CNG is detected on the originating side. CNG packets are sent to the remote side according to T.38 (if IsFaxUsed = 1) and the fax session is started. A SIP Re-INVITE message isn't sent and the fax session starts by the terminating device. This option is useful, for example, when the originating device is located behind a firewall that blocks incoming T.38 packets on ports that have not yet received T.38 packets from the internal network (i.e., originating device). To also send a Re-INVITE message upon detection of a fax CNG tone in this mode, set the parameter FaxCNGMode to 1. ▪ **[2]** Events Only = CNG is detected on the originating side and a fax session is started by the originating side using the Re-

Parameter	Description
	INVITE message. Usually, T.38 fax session starts when the 'preamble' signal is detected by the answering side. Some SIP devices don't support the detection of this fax signal on the answering side and thus, in these cases it is possible to configure the device to start the T.38 fax session when the CNG tone is detected by the originating side. However, this mode is not recommended. **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: Fax Relay Enhanced Redundancy Depth EMS: Enhanced Relay Redundancy Depth CLI: enhanced-redundancy-depth **[FaxRelayEnhancedRedundancyDepth]**	Defines the number of times that control packets are retransmitted when using the T.38 standard. The valid range is 0 to 4. The default value is 2.
Web: Fax Relay Redundancy Depth EMS: Relay Redundancy Depth CLI: redundancy-depth **[FaxRelayRedundancyDepth]**	Defines the number of times that each fax relay payload is retransmitted to the network. ▪ **[0]** = No redundancy (default). ▪ **[1]** = One packet redundancy. ▪ **[2]** = Two packet redundancy. **Note:** This parameter is applicable only to non-V.21 packets.
Web: Fax Relay Max Rate (bps) EMS: Relay Max Rate CLI: max-rate **[FaxRelayMaxRate]**	Defines the maximum rate (in bps) at which fax relay messages are transmitted (outgoing calls). ▪ **[0]** 2400 = 2.4 kbps ▪ **[1]** 4800 = 4.8 kbps ▪ **[2]** 7200 = 7.2 kbps ▪ **[3]** 9600 = 9.6 kbps ▪ **[4]** 12000 = 12.0 kbps ▪ **[5]** 14400 = 14.4 kbps (default) **Note:** The rate is negotiated between both sides (i.e., the device adapts to the capabilities of the remote side). Negotiation of the T.38 maximum supported fax data rate is provided in SIP's SDP T38MaxBitRate parameter. The negotiated T38MaxBitRate is the minimum rate supported between the local and remote endpoints.
Web: Fax Relay ECM Enable EMS: Relay ECM Enable CLI: ecm-mode **[FaxRelayECMEnable]**	Enables Error Correction Mode (ECM) mode during fax relay. ▪ **[0]** Disable. ▪ **[1]** Enable (default).
Web: Fax/Modem Bypass Coder Type EMS: Coder Type **[FaxModemBypassCoderType]**	Determines the coder used by the device when performing fax/modem bypass. Typically, high-bit-rate coders such as G.711 should be used. ▪ **[0]** G.711Alaw= G.711 A-law 64 (default). ▪ **[1]** G.711Mulaw = G.711 μ-law.
Web: Fax/Modem Bypass Packing Factor EMS: Packetization Period CLI: packing-factor **[FaxModemBypassM]**	Defines the number (20 msec) of coder payloads used to generate a fax/modem bypass packet. The valid range is 1, 2, or 3 coder payloads. The default value is 1 coder payload.

Parameter	Description
CLI: fax-modem-telephony-events-mode **[FaxModemNTEMode]**	Determines whether the device sends RFC 2833 ANS/ANSam events upon detection of fax and/or modem Answer tones (i.e., CED tone). ▪ **[0]** = Disabled (default). ▪ **[1]** = Enabled. **Note:** This parameter is applicable only when the fax or modem transport type is set to bypass or Transparent-with-Events.
Web/EMS: Fax Bypass Payload Type CLI: fax-bypass-payload-type **[FaxBypassPayloadType]**	Defines the fax bypass RTP dynamic payload type. The valid range is 96 to 120. The default value is 102.
EMS: Modem Bypass Payload Type CLI: modem-bypass-payload-type **[ModemBypassPayloadType]**	Defines the modem bypass dynamic payload type. The range is 0-127. The default value is 103.
EMS: Relay Volume (dBm) CLI: volume **[FaxModemRelayVolume]**	Defines the fax gain control. The range is -18 to -3, corresponding to -18 dBm to -3 dBm in 1-dB steps. The default is -6 dBm fax gain control.
Web/EMS: Fax Bypass Output Gain CLI: fax-bypass-output-gain **[FaxBypassOutputGain]**	Defines the fax bypass output gain control. The range is -31 to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain).
Web/EMS: Modem Bypass Output Gain **[ModemBypassOutputGain]**	Defines the modem bypass output gain control. The range is -31 dB to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain).
EMS: Basic Packet Interval CLI: modem-bypass-output-gain **[FaxModemBypassBasicRTPPacketInterval]**	Defines the basic frame size used during fax/modem bypass sessions. ▪ **[0]** = Determined internally (default) ▪ **[1]** = 5 msec (not recommended) ▪ **[2]** = 10 msec ▪ **[3]** = 20 msec **Note:** When set to 5 msec (1), the maximum number of simultaneous channels supported is 120.
EMS: Dynamic Jitter Buffer Minimal Delay (dB) CLI: jitter-buffer-minimum-delay **[FaxModemBypasDJBufMinDelay]**	Defines the Jitter Buffer delay (in milliseconds) during fax and modem bypass session. The range is 0 to 150 msec. The default is 40.
EMS: Enable Inband Network Detection CLI: enable-fax-modem-inband-network-detection **[EnableFaxModemInbandNetworkDetection]**	Enables in-band network detection related to fax/modem. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable. When this parameter is enabled on Bypass and transparent with events mode (VxxTransportType is set to 2 or 3), a detection of an Answer Tone from the network triggers a switch to bypass mode in addition to the local Fax/Modem tone detections. However, only a high bit-rate coder voice session effectively detects the Answer Tone sent by a remote endpoint. This can be useful when, for example, the payload of voice and bypass is the same, allowing the originator to

Parameter	Description
	switch to bypass mode as well.
EMS: NSE Mode CLI: NSE-mode **[NSEMode]**	Enables Cisco compatible fax and modem bypass mode. ▪ **[0]** = NSE disabled (default) ▪ **[1]** = NSE enabled In NSE bypass mode, the device starts using G.711 A-Law (default) or G.711μ-Law according to the FaxModemBypassCoderType parameter. The payload type used with these G.711 coders is a standard one (8 for G.711 A-Law and 0 for G.711 μ-Law). The parameters defining payload type for the 'old' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass. The bypass packet interval is selected according to the FaxModemBypassBasicRtpPacketInterval parameter. **Notes:** ▪ This feature can be used only if the VxxModemTransportType parameter is set to 2 (Bypass). ▪ If NSE mode is enabled, the SDP contains the following line: 'a=rtpmap:100 X-NSE/8000'. ▪ To use this feature: ✓ The Cisco gateway must include the following definition: 'modem passthrough nse payload-type 100 codec g711alaw'. ✓ Set the Modem transport type to Bypass mode (VxxModemTransportType is set to 2) for all modems. ✓ Configure the gateway parameter NSEPayloadType = 100. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
EMS: NSE Payload Type CLI: NSE-payload-type **[NSEPayloadType]**	Defines the NSE payload type for Cisco Bypass compatible mode. The valid range is 96-127. The default value is 105. **Note:** Cisco gateways usually use NSE payload type of 100.
EMS: T38 Use RTP Port **[T38UseRTPPort]**	Defines the port (with relation to RTP port) for sending and receiving T.38 packets. ▪ **[0]** = Use the RTP port +2 to send/receive T.38 packets (default). ▪ **[1]** = Use the same port as the RTP port to send/receive T.38 packets. **Notes:** ▪ For this parameter to take effect, you must reset the device. ▪ When the device is configured to use V.152 to negotiate audio and T.38 coders, the UDP port published in SDP for RTP and for T38 must be different. Therefore, set the T38UseRTPPort parameter to 0.
Web/EMS: T.38 Max Datagram Size CLI: t38-mx-datagram-sz **[T38MaxDatagramSize]**	Defines the maximum size of a T.38 datagram that the device can receive. This value is included in the outgoing SDP when T.38 is used. The valid range is 120 to 600. The default value is 238.

Parameter	Description
Web/EMS: T38 Fax Max Buffer CLI: t38-fax-mx-buff **[T38FaxMaxBufferSize]**	Defines the maximum size (in bytes) of the device's T.38 buffer. This value is included in the outgoing SDP when T.38 is used for fax relay over IP. The valid range is 500 to 3000. The default value is 1024.
Web/EMS: Enable Fax Re-Routing CLI: fax-re-routing **[EnableFaxReRouting]**	Enables re-routing of Tel-to-IP calls that are identified as fax calls. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enabled. If a CNG tone is detected on the Tel side of a Tel-to-IP call, the prefix "FAX" is appended to the destination number before routing and manipulations. A value of "FAX" entered as the destination number in the Outbound IP Routing Table is then used to route the call and the destination number manipulation mechanism is used to remove the "FAX" prefix, if required. If the initial INVITE used to establish the voice call (not fax) was already sent, a CANCEL (if not connected yet) or a BYE (if already connected) is sent to tear down the voice call. **Notes:** ▪ To enable this feature, set the parameter CNGDetectorMode to 2 and the parameter IsFaxUsed to 1, 2, or 3. ▪ The "FAX" prefix in routing and manipulation tables is case-sensitive.
Web: Detect Fax on Answer Tone EMS: Enables Detection of FAX on Answer Tone CLI: det-fax-on-ans-tone **[DetFaxOnAnswerTone]**	Determines when the device initiates a T.38 session for fax transmission. ▪ **[0]** Initiate T.38 on Preamble = The device to which the called fax is connected initiates a T.38 session on receiving HDLC Preamble signal from the fax (default). ▪ **[1]** Initiate T.38 on CED = The device to which the called fax is connected initiates a T.38 session on receiving a CED answer tone from the fax. This option can only be used to relay fax signals, as the device sends T.38 Re-INVITE on detection of any fax/modem Answer tone (2100 Hz, amplitude modulated 2100 Hz, or 2100 Hz with phase reversals). The modem signal fails when using T.38 for fax relay. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameters is applicable only if the parameter IsFaxUsed is set to 1 (T.38 Relay) or 3 (Fax Fallback).
CLI: t38-sess-imm-strt **[T38FaxSessionImmediateStart]**	Enables fax transmission of T.38 “no-signal” packets to the terminating fax machine. ▪ **[0]** Disable (default) ▪ **[1]** Enable This is used for transmission from fax machines (connected to the device) located inside a Network Address Translation (NAT). Generally, the firewall blocks T.38 (and other) packets received from the WAN, unless the device behind NAT sends at least one IP packet from the LAN to the WAN through the firewall. If the firewall blocks T.38 packets sent from the termination IP fax, the fax fails. To overcome this, the device sends No-Op (“no-signal”) packets to open a pinhole in the NAT for the answering fax machine. The originating fax does not wait for an answer, but immediately starts

Parameter	Description
	sending T.38 packets to the terminating fax machine. **Note:** To enable No-Op packet transmission, use the NoOpEnable and NoOpInterval parameters.

A.12.2 DTMF and Hook-Flash Parameters

The DTMF and hook-flash parameters are described in the table below.

Table A-36: DTMF and Hook-Flash Parameters

Parameter	Description
Hook-Flash Parameters	
Web/EMS: Hook-Flash Code CLI: hook-flash-code **[HookFlashCode]**	For analog interfaces: Defines the digit pattern that when received from the Tel side, indicates a Hook Flash event. For digital interfaces: Defines the digit pattern used by the PBX to indicate a Hook Flash event. When this pattern is detected from the Tel side, the device responds as if a Hook Flash event has occurred and sends a SIP INFO message if the HookFlashOption parameter is set to 1, 5, 6, or 7 (indicating a Hook Flash). If configured and a Hook Flash indication is received from the IP side, the device generates this pattern to the Tel side. The valid range is a 25-character string. The default is a null string. **Note:** This parameter can also be configured per Tel Profile, using the TelProfile parameter.
Web/EMS: Hook-Flash Option CLI: hook-flash-option **[HookFlashOption]**	Determines the hook-flash transport type (i.e., method by which hook-flash is sent and received). For digital interfaces (E1/T1): This feature is applicable only if the HookFlashCode parameter is configured. ▪ **[0]** Not Supported = Hook-Flash indication is not sent (default). ▪ **[1]** INFO = Sends proprietary INFO message (Broadsoft) with Hook-Flash indication. The device sends the INFO message as follows: Content-Type: application/broadsoft; version=1.0 Content-Length: 17 event flashhook ▪ **[4]** RFC 2833 = This option is currently not supported. ▪ **[5]** INFO (Lucent) = Sends proprietary SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows: Content-Type: application/hook-flash Content-Length: 11 signal=hf ▪ **[6]** INFO (NetCentrex) = Sends proprietary SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows: Content-Type: application/dtmf-relay Signal=16

Parameter	Description
	Where 16 is the DTMF code for hook flash. ▪ **[7]** INFO (HUAWAEI) = Sends a SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows: Content-Length: 17 Content-Type: application/sscc event=flashhook **Notes:** ▪ The device can interwork DTMF HookFlashCode to SIP INFO messages with Hook Flash indication (for digital interfaces) ▪ FXO interfaces support only the receipt of RFC 2833 Hook-Flash signals and INFO **[1]** type. ▪ FXS interfaces send Hook-Flash signals only if the EnableHold parameter is set to 0.
Web: Min. Flash-Hook Detection Period **[msec]** EMS: Min Flash Hook Time CLI: min-flash-hook-time **[MinFlashHookTime]**	Defines the minimum time (in msec) for detection of a hook-flash event. Detection is guaranteed for hook-flash periods of at least 60 msec (when setting the minimum time to 25). Hook-flash signals that last a shorter period of time are ignored. The valid range is 25 to 300. The default value is 300. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces. ▪ It's recommended to reduce the detection time by 50 msec from the desired value. For example, if you want to set the value to 200 msec, then enter 150 msec (i.e., 200 minus 50).
Web: Max. Flash-Hook Detection Period **[msec]** EMS: Flash Hook Period CLI: flash-hook-period **[FlashHookPeriod]**	Defines the hook-flash period (in msec) for both Tel and IP sides (per device). For the IP side, it defines the hook-flash period that is reported to the IP. For the analog side, it defines the following: ▪ FXS interfaces: ✓ Maximum hook-flash detection period. A longer signal is considered an off-hook or on-hook event. ✓ Hook-flash generation period upon detection of a SIP INFO message containing a hook-flash signal. ▪ FXO interfaces: Hook-flash generation period. The valid range is 25 to 3,000. The default value is 700. **Notes:** ▪ For this parameter to take effect, you need to reset the device. ▪ For FXO interfaces, a constant of 100 msec must be added to the required hook-flash period. For example, to generate a 450 msec hook-flash, set this parameter to 550. ▪ This parameter can also be configured per Tel Profile, using the TelProfile parameter.

Parameter	Description
DTMF Parameters	
EMS: Use End of DTMF CLI: notify-on-sig-end **[MGCPDTMFDetectionPoint]**	Determines when the detection of DTMF events is notified. ▪ **[0]** = DTMF event is reported at the end of a detected DTMF digit. ▪ **[1]** = DTMF event is reported at the start of a detected DTMF digit (default).
Web: Declare RFC 2833 in SDP EMS: Rx DTMF Option CLI: rfc-2833-in-sdp **[RxDTMFOption]**	Defines the supported receive DTMF negotiation method. ▪ **[0]** No = Don't declare RFC 2833 telephony-event parameter in SDP. ▪ **[3]** Yes = Declare RFC 2833 telephony-event parameter in SDP (default). The device is always receptive to RFC 2833 DTMF relay packets. Therefore, it is always correct to include the 'telephony-event' parameter as default in the SDP. However, some devices use the absence of the 'telephony-event' in the SDP to decide to send DTMF digits in-band using G.711 coder. If this is the case, you can set this parameter to 0. **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Tx DTMF Option Table	
Web/EMS: Tx DTMF Option CLI: config voip > gw dtmf-and-suppl dtmf-and-dialing > dtmf-options **[TxDTMFOption]**	This parameter table configures up to two preferred transmit DTMF negotiation methods. The format of this parameter is as follows: **[TxDTMFOption]** FORMAT TxDTMFOption_Index = TxDTMFOption_Type; **[\TxDTMFOption]** Where Type is: ▪ **[0]** Not Supported = No negotiation - DTMF digits are sent according to the parameters DTMFTransportType and RFC2833PayloadType (default). ▪ **[1]** INFO (Nortel) = Sends DTMF digits according to IETF Internet-Draft draft-choudhuri-sip-info-digit-00. ▪ **[2]** NOTIFY = Sends DTMF digits according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01. ▪ **[3]** INFO (Cisco) = Sends DTMF digits according to Cisco format. ▪ **[4]** RFC 2833. ▪ **[5]** INFO (Korea) = Sends DTMF digits according to Korea Telecom format. For example: TxDTMFOption 0 = 1; TxDTMFOption 1 = 3; **Notes:** ▪ DTMF negotiation methods are prioritized according to the order of their appearance. ▪ When out-of-band DTMF transfer is used (**[1], [2], [3],** or **[5]),** the parameter DTMFTransportType is automatically set to 0 (DTMF digits are erased from

Parameter	Description
	the RTP stream). ▪ When RFC 2833 (4) is selected, the device: **a.** Negotiates RFC 2833 payload type using local and remote SDPs. **b.** Sends DTMF packets using RFC 2833 payload type according to the payload type in the received SDP. **c.** Expects to receive RFC 2833 packets with the same payload type as configured by the parameter RFC2833PayloadType. **d.** Removes DTMF digits in transparent mode (as part of the voice stream). ▪ When TxDTMFOption is set to 0, the RFC 2833 payload type is set according to the parameter RFC2833PayloadType for both transmit and receive. ▪ If an ISDN phone user presses digits (e.g., for interactive voice response / IVR applications such as retrieving voice mail messages), ISDN Information messages received by the device for each digit are sent in the voice channel to the IP network as DTMF signals, according to the settings of the TxDTMFOption parameter. ▪ The *ini* file table parameter TxDTMFOption can be repeated twice for configuring the DTMF transmit methods. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
[DisableAutoDTMFMute]	Enables the automatic muting of DTMF digits when out-of-band DTMF transmission is used. ▪ **[0]** = Automatic mute is used (default). ▪ **[1]** = No automatic mute of in-band DTMF. When this parameter is set to 1, the DTMF transport type is set according to the parameter DTMFTransportType and the DTMF digits aren't muted if out-of-band DTMF mode is selected (TxDTMFOption set to 1, 2 or 3). This enables the sending of DTMF digits in-band (transparent of RFC 2833) in addition to out-of-band DTMF messages. **Note:** Usually this mode is not recommended.
Web/EMS: Enable Digit Delivery to IP CLI: digit-delivery-2ip **[EnableDigitDelivery2IP]**	Enables the Digit Delivery feature whereby DTMF digits are sent to the destination IP address after the Tel-to-IP call is answered. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable digit delivery to IP. To enable this feature, modify the called number to include at least one 'p' character. The device uses the digits before the 'p' character in the initial INVITE message. After the call is answered, the device waits for the required time (number of 'p' multiplied by 1.5 seconds), and then sends the rest of the DTMF digits using the method chosen (in-band or out-of-band). **Notes:**

Parameter	Description
	▪ For this parameter to take effect, a device reset is required. ▪ The called number can include several 'p' characters (1.5 seconds pause), for example, 1001pp699, 8888p9p300.
Web: Enable Digit Delivery to Tel EMS: Enable Digit Delivery CLI: digit-delivery-2tel **[EnableDigitDelivery]**	Enables the Digit Delivery feature, which sends DTMF digits of the called number to the device's port (analog)/B-channel (digital) (phone line) after the call is answered (i.e., line is off-hooked for FXS, or seized for FXO) for IP-to-Tel calls. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable Digit Delivery feature for the FXO/FXS device. For digital interfaces: If the called number in IP-to-Tel call includes the characters 'w' or 'p', the device places a call with the first part of the called number (before 'w' or 'p') and plays DTMF digits after the call is answered. If the character 'w' is used, the device waits for detection of a dial tone before it starts playing DTMF digits. For example, if the called number is '1007766p100', the device places a call with 1007766 as the destination number, then after the call is answered it waits 1.5 seconds ('p') and plays the rest of the number (100) as DTMF digits. Additional examples: 1664wpp102, 66644ppp503, and 7774w100pp200. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For analog interfaces: The called number can include characters 'p' (1.5 seconds pause) and 'd' (detection of dial tone). If character 'd' is used, it must be the first 'digit' in the called number. The character 'p' can be used several times. For example (for FXS/FXO interfaces), the called number can be as follows: d1005, dpp699, p9p300. To add the 'd' and 'p' digits, use the usual number manipulation rules. ▪ For analog interfaces: To use this feature with FXO interfaces, configure the device to operate in one-stage dialing mode. ▪ If this parameter is enabled, it is possible to configure the FXS/FXO interface to wait for dial tone per destination phone number (before or during dialing of destination phone number). Therefore, the parameter IsWaitForDialTone (configurable for the entire device) is ignored. ▪ For analog interfaces: The FXS interface send SIP 200 OK responses only after the DTMF dialing is complete. ▪ This parameter can also be configured per Tel Profile, using the TelProfile parameter.
CLI: replace-nb-sign-w-esc **[ReplaceNumberSignWithEscapeChar]**	Determines whether to replace the number sign (#) with the escape character (%23) in outgoing SIP messages for

Parameter	Description
	Tel-to-IP calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable = All number signs #, received in the dialed DTMF digits are replaced in the outgoing SIP Request-URI and To headers with the escape sign %23. **Notes:** ▪ This parameter is applicable only if the parameter IsSpecialDigits is set 1. ▪ This parameter is applicable only to analog interfaces.
Web: Special Digit Representation EMS: Use Digit For Special DTMF CLI: special-digit-rep **[UseDigitForSpecialDTMF]**	Defines the representation for 'special' digits ('*' and '#') that are used for out-of-band DTMF signaling (using SIP INFO/NOTIFY). ▪ **[0]** Special = Uses the strings '*' and '#' (default). ▪ **[1]** Numeric = Uses the numerical values 10 and 11.

A.12.3 Digit Collection and Dial Plan Parameters

The digit collection and dial plan parameters are described in the table below.

Table A-37: Digit Collection and Dial Plan Parameters

Parameter	Description
Web/EMS: Dial Plan Index CLI: dial-plan-index **[DialPlanIndex]**	Defines the Dial Plan index to use in the external Dial Plan file. The Dial Plan file is loaded to the device as a .dat file (converted using the DConvert utility). The Dial Plan index can be defined globally or per Tel Profile. The valid value range is 0 to 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The default is -1, indicating that no Dial Plan file is used. **Notes:** ▪ If this parameter is configured to select a Dial Plan index, the settings of the parameter DigitMapping are ignored. ▪ If this parameter is configured to select a Dial Plan index from an external Dial Plan file, the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map rules configured by the DigitMapping parameter. ▪ This parameter is applicable also to ISDN with overlap dialing. ▪ For E1 CAS MFC-R2 variants (which don't support terminating digit for the called party number, usually I-15), this parameter and the DigitMapping parameter are ignored. Instead, you can define a Dial Plan template per trunk using the parameter CasTrunkDialPlanName_x (or in the Trunk Settings page). ▪ This parameter can also be configured per Tel Profile, using the TelProfile parameter. ▪ For more information on the Dial Plan file, see 'External Dial Plan File' on page 377.
CLI: tel2ip-src-nb-map-dial-	Defines the Dial Plan index in the external Dial Plan file for the Tel-

Parameter	Description
index **[Tel2IPSourceNumberMappingDialPlanIndex]**	to-IP Source Number Mapping feature. The valid value range is 0 to 7, defining the Dial Plan index **[Plan x]** in the Dial Plan file. The default is -1 (disabled). For more information on this feature, see 'Modifying ISDN-to-IP Calling Party Number' on page 378.
Web: Digit Mapping Rules EMS: Digit Map Pat terns CLI: default-dm **[DigitMapping]**	Defines the digit map pattern (used to reduce the dialing period when ISDN overlap dialing for digital interfaces). If the digit string (i.e., dialed number) matches one of the patterns in the digit map, the device stops collecting digits and establishes a call with the collected number. The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar (|). The maximum length of the entire digit pattern is 152 characters. The available notations include the following: ▪ **[n-m]:** Range of numbers (not letters). ▪ **.** (single dot): Repeat digits until next notation (e.g., T). ▪ **x:** Any single digit. ▪ **T:** Dial timeout (configured by the TimeBetweenDigits parameter). ▪ **S:** Short timer (configured by the TimeBetweenDigits parameter; default is two seconds) that can be used when a specific rule is defined after a more general rule. For example, if the digit map is 99|998, then the digit collection is terminated after the first two 9 digits are received. Therefore, the second rule of 998 can never be matched. But when the digit map is 99s|998, then after dialing the first two 9 digits, the device waits another two seconds within which the caller can enter the digit 8. An example of a digit map is shown below: 11xS|00T|**[1-**7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T In the example above, the last rule can apply to International numbers: 9 for dialing tone, 011 Country Code, and then any number of digits for the local number ('x.'). **Notes:** ▪ For ISDN interfaces, the digit map mechanism is applicable only when ISDN overlap dialing is used (ISDNRxOverlap is set to 1). ▪ If the DialPlanIndex parameter is configured (to select a Dial Plan index), then the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map rules configured by the DigitMapping parameter. ▪ For more information on digit mapping, see 'Digit Mapping' on page 375.
Web: Max Digits in Phone Num EMS: Max Digits in Phone Number CLI: mxdig-b4-dialing **[MaxDigits]**	Defines the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side (analog) when Tel-to-IP ISDN overlap dialing is performed (digital). When the number of collected digits reaches this maximum, the device uses these digits for the called destination number. The valid range is 1 to 49. The default value is 5 for analog and 30for digital. **Notes:** ▪ Instead of using this parameter, Digit Mapping rules can be configured. ▪ Dialing ends when any of the following scenarios occur: ✓ Maximum number of digits is dialed

Parameter	Description
	✓ Interdigit Timeout (TimeBetweenDigits) expires ✓ Pound (#) key is pressed ✓ Digit map pattern is matched
Web: Inter Digit Timeout for Overlap Dialing **[sec]** EMS: Interdigit Timeout (Sec) CLI: time-btwn-dial-digs **[TimeBetweenDigits]**	For analog interfaces: Defines the time (in seconds) that the device waits between digits that are dialed by the user. For ISDN overlap dialing: Defines the time (in seconds) that the device waits between digits that are received from the PSTN or IP during overlap dialing. When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number. The valid range is 1 to 10. The default value is 4.
Web: Enable Special Digits EMS: Use '#' For Dial Termination CLI: special-digits **[IsSpecialDigits]**	Determines whether the asterisk (*) and pound (#) digits can be used in DTMF. ▪ **[0]** Disable = Use '*' or '#' to terminate number collection (refer to the parameter UseDigitForSpecialDTMF). (Default.) ▪ **[1]** Enable = Allows '*' and '#' for telephone numbers dialed by a user or for the endpoint telephone number. **Note:** These symbols can always be used as the first digit of a dialed number even if you disable this parameter.

A.12.4 Voice Mail Parameters

The voice mail parameters are described in the table below. For more information on the Voice Mail application, refer to the *CPE Configuration Guide for Voice Mail*.

Table A-38: Voice Mail Parameters

Parameter	Description
Web/EMS: Voice Mail Interface CLI: vm-interface **[VoiceMailInterface]**	Enables the device's Voice Mail application and determines the communication method used between the PBX and the device. ▪ **[0]** None (default) ▪ **[1]** DTMF ▪ **[2]** SMDI ▪ **[3]** QSIG ▪ **[4]** SETUP Only = For ISDN ▪ **[5]** MATRA/AASTRA QSIG ▪ **[6]** QSIG SIEMENS = QSIG MWI activate and deactivate messages include Siemens Manufacturer Specific Information (MSI) ▪ **[7]** IP2IP = The device's IP2IP application is used for interworking between an IP Voice Mail server and the device. This is implemented for sending unsolicited SIP NOTIFY messages received from the Voice Mail server to an IP Group (configured using the parameter NotificationIPGroupID). ▪ **[8]** ETSI = Euro ISDN, according to ETS 300 745-1 V1.2.4, section 9.5.1.1. Enables MWI interworking from IP to Tel, typically used for BRI phones. **Note:** To disable voice mail per Trunk Group, you can use a Tel Profile ID (using the TelProfile parameter) that is configured with

Parameter	Description
	the EnableVoiceMailDelay parameter to disabled (0). This eliminates the phenomenon of call delay on Trunks not implementing voice mail when voice mail is enabled using this global parameter.
Web: Enable VoiceMail URI EMS: Enable VMURI CLI: voicemail-uri **[EnableVMURI]**	Enables the interworking of target and cause for redirection from Tel to IP and vice versa, according to RFC 4468. ▪ **[0]** Disable (default). ▪ **[1]** Enable. Upon receipt of an ISDN Setup message with Redirect values, the device maps the Redirect phone number to the SIP 'target' parameter and the Redirect number reason to the SIP 'cause' parameter in the Request-URI. **Redirecting Reason >> SIP Response Code** Unknown >> 404 User busy >> 486 No reply >> 408 Deflection >> 487/480 Unconditional >> 302 Others >> 302 If the device receives a Request-URI that includes a 'target' and 'cause' parameter, the 'target' is mapped to the Redirect phone number and the 'cause' is mapped to the Redirect number reason.
[WaitForBusyTime]	Defines the time (in msec) that the device waits to detect busy and/or reorder tones. This feature is used for semi-supervised PBX call transfers (i.e., the LineTransferMode parameter is set to 2). The valid value range is 0 to 20000 (i.e., 20 sec). The default is 2000 (i.e., 2 sec).
Web/EMS: Line Transfer Mode CLI: line-transfer-mode **[LineTransferMode]**	Defines the call transfer method used by the device. This parameter is applicable to FXO call transfer as well as E1/T1 CAS call transfer if the TrunkTransferMode_x parameter is set to 3 (CAS Normal) or 1 (CAS NFA). ▪ **[0]** None = IP (default). ▪ **[1]** Blind = PBX blind transfer: ✓ Analog (FXO): After receiving a SIP REFER message from the IP side, the device (FXO) sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then immediately releases the line (i.e., on-hook). The PBX performs the transfer internally. ✓ E1/T1 CAS: When a SIP REFER message is received, the device performs a blind transfer, by performing a CAS wink, waiting a user-defined time (configured by the WaitForDialTime parameter), dialing the Refer-To number, and then releasing the call. The PBX performs the transfer internally. ▪ **[2]** Semi Supervised = PBX semi-supervised transfer: ✓ Analog (FXO): After receiving a SIP REFER message from the IP side, the device sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). If no busy or reorder tones are detected (within the user-defined interval set by the

Parameter	Description
	WaitForBusyTime parameter), the device completes the call transfer by releasing the line. If these tones are detected, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected), and generates an additional hook-flash toward the FXO line to restore connection to the original call. ✓ E1/T1 CAS: The device performs a CAS wink, waits a user-defined time (configured by the WaitForDialTime parameter), and then dials the Refer-To number. If during the user-defined interval set by the WaitForBusyTime parameter, no busy or reorder tones are detected, the device completes the call transfer by releasing the line. If during this interval, the device detects these tones, the transfer operation is cancelled, the device sends a SIP NOTIFY message with a failure reason (e.g., 486 if a busy tone is detected), and then generates an additional wink toward the CAS line to restore connection with the original call. ▪ **[3]** Supervised = PBX Supervised transfer: ✓ Analog (FXO): After receiving a SIP REFER message from the IP side, the device sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). The device waits for connection of the transferred call and then completes the call transfer by releasing the line. If speech is not detected, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected) and generates an additional hook-flash toward the FXO line to restore connection to the original call. ✓ E1/T1 CAS: The device performs a supervised transfer to the PBX. The device performs a CAS wink, waits a user-defined time (configured by the WaitForDialTime parameter), and then dials the Refer-To number. The device completes the call transfer by releasing the line only after detection of the transferred party answer. To enable answer supervision, you also need to do the following: 1) Enable voice detection (i.e., set the EnableVoiceDetection parameter to 1). 2) Set the EnableDSPIPMDetectors parameter to 1. 3) Install the IPMDetector DSP option Feature Key.
Message Waiting Indication (MWI) Parameters	
Web: MWI Off Digit Pattern EMS: MWI Off Code CLI: mwi-off-dig-ptrn **[MWIOffCode]**	Defines the digit code used by the device to notify the PBX that there are no messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.
Web: MWI On Digit Pattern EMS: MWI On Code CLI: mwi-on-dig-ptrn **[MWIOnCode]**	Defines the digit code used by the device to notify the PBX of messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.

Parameter	Description
Web: MWI Suffix Pattern EMS: MWI Suffix Code CLI: mwi-suffix-pattern **[MWISuffixCode]**	Defines the digit code used by the device as a suffix for 'MWI On Digit Pattern' and 'MWI Off Digit Pattern'. This suffix is added to the generated DTMF string after the extension number. The valid range is a 25-character string.
Web: MWI Source Number EMS: MWI Source Name CLI: mwi-source-number **[MWISourceNumber]**	Defines the calling party's phone number used in the Q.931 MWI Setup message to PSTN. If not configured, the channel's phone number is used as the calling number.

CLI: mwi-subs-ipgrpid
[MWISubscribeIPGroupID]
	Defines the IP Group ID used when subscribing to an MWI server. The 'The SIP Group Name' field value of the IP Group table is used as the Request-URI host name in the outgoing MWI SIP SUBSCRIBE message. The request is sent to the IP address defined for the Proxy Set that is associated with the IP Group. The Proxy Set's capabilities such as proxy redundancy and load balancing are also applied to the message.
	For example, if the 'SIP Group Name' field of the IP Group is set to "company.com", the device sends the following SUBSCRIBE message:
	`SUBSCRIBE sip:company.com...`
	Instead of:
	`SUBSCRIBE sip:10.33.10.10...`
	Note: If this parameter is not configured, the MWI SUBSCRIBE message is sent to the MWI server as defined by the MWIServerIP parameter.

[NotificationIPGroupID]	Defines the IP Group ID to which the device sends SIP NOTIFY MWI messages. **Notes:** ▪ This is used for MWI Interrogation. For more information on the interworking of QSIG MWI to IP, see Message Waiting Indication on page 321. ▪ To determine the handling method of MWI Interrogation messages, use the MWIInterrogationType parameter.
[MWIQsigMsgCentreIdIDPartyNumber]	Defines the Message Centred ID party number used for QSIG MWI messages. If not configured (default), the parameter is not included in MWI (activate and deactivate) QSIG messages. The value is a string.
Digit Patterns The following digit pattern parameters apply only to voice mail applications that use the DTMF communication method. For the available pattern syntaxes, refer to the *CPE Configuration Guide for Voice Mail*.	
Web: Forward on Busy Digit Pattern (Internal) EMS: Digit Pattern Forward On Busy CLI: fwd-bsy-dig-ptrn-int **[DigitPatternForwardOnBusy]**	Defines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an internal extension. The valid range is a 120-character string.

Parameter	Description
Web: Forward on No Answer Digit Pattern (Internal) EMS: Digit Pattern Forward On No Answer CLI: fwd-no-ans-dig-pat-int **[DigitPatternForwardOnNoAnswer]**	Defines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on Do Not Disturb Digit Pattern (Internal) EMS: Digit Pattern Forward On DND CLI: fwd-dnd-dig-ptrn-int **[DigitPatternForwardOnDND]**	Defines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on No Reason Digit Pattern (Internal) EMS: Digit Pattern Forward No Reason CLI: fwd-no-rsn-dig-ptrn-int **[DigitPatternForwardNoReason]**	Defines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on Busy Digit Pattern (External) EMS: VM Digit Pattern On Busy External CLI: fwd-bsy-dig-ptrn-ext **[DigitPatternForwardOnBusyExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on No Answer Digit Pattern (External) EMS: VM Digit Pattern On No Answer Ext CLI: fwd-no-ans-dig-pat-ext **[DigitPatternForwardOnNoAnswerExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on Do Not Disturb Digit Pattern (External) EMS: VM Digit Pattern On DND External CLI: fwd-dnd-dig-ptrn-ext **[DigitPatternForwardOnDNDExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on No Reason Digit Pattern (External) EMS: VM Digit Pattern No Reason External CLI: fwd-no-rsn-dig-ptrn-ext **[DigitPatternForwardNoReasonExt]**	Defines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Internal Call Digit Pattern EMS: Digit Pattern Internal Call CLI: int-call-dig-ptrn **[DigitPatternInternalCall]**	Defines the digit pattern used by the PBX to indicate an internal call. The valid range is a 120-character string.

Parameter	Description
Web: External Call Digit Pattern EMS: Digit Pattern External Call CLI: ext-call-dig-ptrn **[DigitPatternExternalCall]**	Defines the digit pattern used by the PBX to indicate an external call. The valid range is a 120-character string.
Web: Disconnect Call Digit Pattern EMS: Tel Disconnect Code CLI: disc-call-dig-ptrn **[TelDisconnectCode]**	Defines a digit pattern that when received from the Tel side, indicates the device to disconnect the call. The valid range is a 25-character string.
Web: Digit To Ignore Digit Pattern EMS: Digit To Ignore CLI: dig-to-ignore-dig-pattern **[DigitPatternDigitToIgnore]**	Defines a digit pattern that if received as Src (S) or Redirect (R) numbers is ignored and not added to that number. The valid range is a 25-character string.

A.12.5 Supplementary Services Parameters

This subsection describes the device's supplementary telephony services parameters.

A.12.5.1 Caller ID Parameters

The caller ID parameters are described in the table below.

Table A-39: Caller ID Parameters

Parameter	Description
Web: Caller ID Permissions Table EMS: SIP Endpoints > Caller ID	
[EnableCallerID]	This parameter table configures Caller ID permissions. It allows you to enable or disable (per port) Caller ID generation (for FXS interfaces) and detection (for FXO interfaces). The format of this parameter is as follows: [EnableCallerID] FORMAT EnableCallerID_Index = EnableCallerID_IsEnabled, EnableCallerID_Module, EnableCallerID_Port; [\EnableCallerID] Where, ▪ IsEnabled: ✓ **[0]** Disable = disables Caller ID (default). ✓ **[1]** Enable = enables Caller ID generation (FXS) or detection (FXO). ▪ Module = Module number (where 1 denotes the module in Slot 1). ▪ Port = Port number (where 1 denotes Port 1 of a module). For example: EnableCallerID 0 = 1,3,1; (caller ID enabled on Port 1 of Module 3) EnableCallerID 1 = 0,3,2; (caller ID disabled on Port 2 of Module 3) **Notes:** ▪ The indexing of this parameter starts at 0. ▪ If a port is not configured, its Caller ID generation/detection is determined according to the global parameter EnableCallerID. ▪ For configuring this table using the Web interface, see Configuring Caller ID Permissions on page 360. ▪ For configuring ini file table parameters, see Configuring ini File Table Parameters on page 98.
Web: Caller Display Information Table EMS: SIP Endpoints > Caller ID	
[CallerDisplayInfo]	This parameter table enables the device to send Caller ID information to IP when a call is made. The called party can use this information for caller identification. The information configured in this table is sent in the SIP INVITE message's From header. The format of this parameter is as follows: [CallerDisplayInfo] FORMAT CallerDisplayInfo_Index = CallerDisplayInfo_DisplayString, CallerDisplayInfo_IsCidRestricted, CallerDisplayInfo_Module, CallerDisplayInfo_Port;

Parameter	Description
	[\CallerDisplayInfo] Where, ▪ DisplayString = Caller ID string (up to 18 characters). ▪ IsCidRestricted = ✓ **[0]** Allowed = sends the defined caller ID string when a Tel-to-IP call is made using the corresponding device port (default). ✓ **[1]** Restricted = does not send the defined caller ID string. ▪ Module = Module number (where 1 denotes the module in Slot 1). ▪ Port = Port number (where 1 denotes Port 1 of a module). For example: CallerDisplayInfo 0 = Susan C.,0,1,1; ("Susan C." is sent as the Caller ID for Port 1 of Module 1) CallerDisplayInfo 1 = Mark M.,0,1,2; ("Mark M." is sent as Caller ID for Port 2 of Module 1) **Notes:** ▪ The indexing of this ini file table parameter starts at 0. ▪ When FXS ports receive 'Private' or 'Anonymous' strings in the SIP From header, the calling name or number is not sent to the Caller ID display. ▪ If the Caller ID name is detected on an FXO line (the parameter EnableCallerID is set to 1), it is used instead of the Caller ID name defined in this table parameter. ▪ When the parameter CallerDisplayInfo_IsCidRestricted is set to 1 (Restricted), the Caller ID is sent to the remote side using only the SIP headers P-Asserted-Identity and P-Preferred-Identity (AssertedIdMode). ▪ The value of the parameter CallerDisplayInfo_IsCidRestricted is overridden by the parameter SourceNumberMapIp2Tel_IsPresentationRestricted in the Source Number Manipulation table (table parameter SourceNumberMapIP2Tel). ▪ For configuring this table using the Web interface, see Configuring Caller Display Information on page 357. ▪ For configuring ini file table parameters, see Configuring ini File Table Parameters on page 98.
Web/EMS: Enable Caller ID CLI: enable-caller-id **[EnableCallerID]**	Enables Caller ID. ▪ **[0]** Disable (default). ▪ **[1]** Enable. If the Caller ID service is enabled, then for FXS interfaces, calling number and Display text (from IP) are sent to the device's port. For FXO interfaces, the Caller ID signal is detected and sent to IP in the SIP INVITE message (as 'Display' element). For information on the Caller ID table, see Configuring Caller Display Information on page 357. To disable/enable caller ID generation per port, see Configuring Call Forward on page 358.
Web: Caller ID Type EMS: Caller id Types CLI: caller-ID-type **[CallerIDType]**	Determines the standard used for detection (FXO) and generation (FXS) of Caller ID, and detection (FXO) / generation (FXS) of MWI (when specified) signals: ▪ **[0]** Standard Bellcore = Caller ID and MWI (default)

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>

[1] Standard ETSI = Caller ID and MWI
[2] Standard NTT
[4] Standard BT = Britain
[16] Standard DTMF Based ETSI
[17] Standard Denmark = Caller ID and MWI
[18] Standard India
[19] Standard Brazil

Notes:

Typically, the Caller ID signals are generated/detected between the first and second rings. However, sometimes the Caller ID is detected before the first ring signal (in such a scenario, configure the parameter RingsBeforeCallerID to 0).
Caller ID detection for Britain [4] is not supported on the device’s FXO ports. Only FXS ports can generate the Britain [4] Caller ID.
To select the Bellcore Caller ID sub standard, use the parameter BellcoreCallerIDTypeOneSubStandard. To select the ETSI Caller ID substandard, use the parameter ETSICallerIDTypeOneSubStandard.
To select the Bellcore MWI sub standard, use the parameter BellcoreVMWITypeOneStandard. To select the ETSI MWI sub standard, use the parameter ETSIVMWITypeOneStandard.
If you define Caller ID Type as NTT [2], you need to define the NTT DID signaling form (FSK or DTMF) using the parameter NTTDIDSignallingForm.

</td></tr>
<tr><td>Web: Enable FXS Caller ID Category Digit For Brazil Telecom
CLI: fxs-callid-cat-brazil
[AddCPCPrefix2BrazilCallerID]</td><td>
Enables the interworking of Calling Party Category (cpc) code from SIP INVITE messages to FXS Caller ID first digit.

[0] Disable (default)
[1] Enable

When this parameter is enabled, the device sends the Caller ID number (calling number) with the cpc code (received in the SIP INVITE message) to the device's FXS port. The cpc code is added as a prefix to the caller ID (after IP-to-Tel calling number manipulation). For example, assuming that the incoming INVITE contains the following From (or P-Asserted-Id) header:
<pre>From:<sip:+551137077801;cpc=payphone@10.20.7.35>;tag=53700</pre>
The calling number manipulation removes "+55" (leaving 10 digits), and then adds the prefix 7, the cpc code for payphone user. Therefore, the Caller ID number that is sent to the FXS port, in this example is 71137077801.
If the incoming INVITE message doesn't contain the 'cpc' parameter, nothing is added to the Caller ID number.
<table>
<tr><th>CPC Value in Received INVITE</th><th>CPC Code Prefixed to Caller ID (Sent to FXS Endpoint)</th><th>Description</th></tr>
<tr><td>cpc=unknown</td><td>1</td><td>Unknown user</td></tr>
<tr><td>cpc=subscribe</td><td>1</td><td>-</td></tr>
<tr><td>cpc=ordinary</td><td>1</td><td>Ordinary user</td></tr>
</table>
</td></tr>
</table>

Parameter	Description
	cpc=priority 2 Pre-paid user cpc=test 3 Test user cpc=operator 5 Operator cpc=data 6 Data call cpc=payphone 7 Payphone user **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ For this parameter to be enabled, you must also set the parameter EnableCallingPartyCategory to 1.
[EnableCallerIDTypeTwo]	Disables the generation of Caller ID type 2 when the phone is off-hooked. Caller ID type 2 (also known as off-hook Caller ID) is sent to a currently busy telephone to display the caller ID of the waiting call. ▪ **[0]** = Caller ID type 2 isn't played. ▪ **[1]** = Caller ID type 2 is played (default).
EMS: Caller ID Timing Mode CLI: caller-id-timing-mode **[AnalogCallerIDTimingMode]**	Determines when Caller ID is generated. ▪ **[0]** = Caller ID is generated between the first two rings (default). ▪ **[1]** = The device attempts to find an optimized timing to generate the Caller ID according to the selected Caller ID type. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If this parameter is set to 1 and used with distinctive ringing, the Caller ID signal doesn't change the distinctive ringing timing. ▪ For this parameter to take effect, a device reset is required.
EMS: Bellcore Caller ID Type One Sub Standard CLI: bellcore-callerid-type-one-sub-standard **[BellcoreCallerIDTypeOneSubStandard]**	Determines the Bellcore Caller ID sub-standard. ▪ **[0]** = Between rings (default). ▪ **[1]** = Not ring related. **Note:** For this parameter to take effect, a device reset is required.
EMS: ETSI Caller ID Type One Sub Standard CLI: etsi-callerid-type-one-sub-standard **[ETSICallerIDTypeOneSubStandard]**	Determines the ETSI FSK Caller ID Type 1 sub-standard (FXS only). ▪ **[0]** = ETSI between rings (default). ▪ **[1]** = ETSI before ring DT_AS. ▪ **[2]** = ETSI before ring RP_AS. ▪ **[3]** = ETSI before ring LR_DT_AS. ▪ **[4]** = ETSI not ring related DT_AS. ▪ **[5]** = ETSI not ring related RP_AS. ▪ **[6]** = ETSI not ring related LR_DT_AS. **Note:** For this parameter to take effect, a device reset is required.
Web: Asserted Identity Mode EMS: Asserted ID Mode CLI: asserted-identity-m **[AssertedIdMode]**	Determines whether the SIP header P-Asserted-Identity or P-Preferred-Identity is used in the generated INVITE request for Caller ID (or privacy). ▪ **[0]** Disabled = None (default) ▪ **[1]** Adding PAsserted Identity ▪ **[2]** Adding PPreferred Identity

Parameter	Description
	This parameter determines the header (P-Asserted-Identity or P-Preferred-Identity) used in the generated INVITE request. The header also depends on the calling Privacy (allowed or restricted). These headers are used to present the originating party's Caller ID. The Caller ID is composed of a Calling Number and (optionally), a Calling Name. These headers are used together with the Privacy header. If Caller ID is restricted (i.e., P-Asserted-Identity is not sent), the Privacy header includes the value 'id' ('Privacy: id'). Otherwise, for allowed Caller ID, 'Privacy: none' is used. If Caller ID is restricted (received from Tel or configured in the device), the From header is set to <anonymous@anonymous.invalid>. The 200 OK response can contain the connected party CallerID - Connected Number and Connected Name. For example, if the call is answered by the device, the 200 OK response includes the P-Asserted-Identity with Caller ID. The device interworks (in some ISDN variants), the Connected Party number and name from Q.931 Connect message to SIP 200 OK with the P-Asserted-Identity header. In the opposite direction, if the ISDN device receives a 200 OK with P-Asserted-Identity header, it interworks it to the Connected party number and name in the Q.931 Connect message, including its privacy.
Web: Use Destination As Connected Number **[UseDestinationAsConnectedNumber]**	Determines whether the device includes the Called Party Number from outgoing Tel calls (after number manipulation) in the SIP P-Asserted-Identity header. The device includes the SIP P-Asserted-Identity header in 180 Ringing and 200 OK responses for IP-to-Tel calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ If the received Q.931 Connect message contains a Connected Party Number, this number is used in the P-Asserted-Identity header in 200 OK response. ▪ For this feature, you must also enable the device to include the P-Asserted-Identity header in 180/200 OK responses, by setting the parameter AssertedIDMode to 1. ▪ This parameter is applicable to ISDN, CAS, and/or FXO interfaces.
Web: Caller ID Transport Type EMS: Transport Type CLI: caller-ID-transport-type **[CallerIDTransportType]**	Determines the device's behavior for Caller ID detection. ▪ **[0]** Disable = The caller ID signal is not detected - DTMF digits remain in the voice stream. ▪ **[1]** Relay = (Currently not applicable.) ▪ **[3]** Mute = The caller ID signal is detected from the Tel/PSTN side and then erased from the voice stream (default). **Note:** Caller ID detection is applicable only to FXO interfaces.

A.12.5.2 Call Waiting Parameters

The call waiting parameters are described in the table below.

Table A-40: Call Waiting Parameters

Parameter	Description
Web/EMS: Enable Call Waiting CLI: call-waiting **[EnableCallWaiting]**	Determines whether Call Waiting is enabled. ▪ **[0]** Disable = Disable the Call Waiting service. ▪ **[1]** Enable = Enable the Call Waiting service (default). **Digital interfaces:** If enabled and the device initiates a Tel-to-IP call to a destination that is busy, it plays a Call Waiting Ringback tone to the caller. The tone is played only if the destination returns a 182 "Queued" SIP response. **FXS interfaces:** If enabled, when an FXS interface receives a call on a busy endpoint, it responds with a 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that initiated the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received. **Notes:** ▪ The device's Call Progress Tones (CPT) file must include a Call Waiting Ringback tone (caller side) and a Call Waiting tone (called side, FXS only). ▪ **FXS interfaces:** The EnableHold parameter must be enabled on both the calling and the called side. ▪ **Analog interfaces:** You can use the parameter table CallWaitingPerPort to enable Call Waiting per port. ▪ **Analog interfaces:** For information on the Call Waiting feature, see 'Call Waiting' on page 320.
EMS: Send 180 For Call Waiting **[Send180ForCallWaiting]**	Determines the SIP response code for indicating Call Waiting. ▪ **[0]** = Use 182 Queued response to indicate call waiting (default). ▪ **[1]** = Use 180 Ringing response to indicate call waiting.
Web: Call Waiting Table **EMS: SIP Endpoints > Call Waiting**	
[CallWaitingPerPort]	This parameter table configures call waiting per FXS port. The format of this parameter is as follows: [CallWaitingPerPort] FORMAT CallWaitingPerPort_Index = CallWaitingPerPort_IsEnabled, CallWaitingPerPort_Module, CallWaitingPerPort_Port; [\CallWaitingPerPort] Where, ▪ IsEnabled: ✓ **[0]** Disable = no call waiting for the specific port. ✓ **[1]** Enable = enables call waiting for the specific port. When the FXS device receives a call on a busy endpoint (port), it responds with a SIP 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that

Parameter	Description
	initiates the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received. ▪ Port = Port number. ▪ Module = Module number. For example: CallWaitingPerPort 0 = 0,1,1; (call waiting disabled for Port 1 of Module 1) CallWaitingPerPort 1 = 1,1,2; (call waiting enabled for Port 2 of Module 1) **Notes:** ▪ This parameter is applicable only to FXS ports. ▪ If this parameter is not configured (default), call waiting is determined according to the global parameter EnableCallWaiting. ▪ The device's CPT file must include a 'call waiting Ringback' tone (caller side) and a 'call waiting' tone (called side, FXS interfaces only). ▪ The EnableHold parameter must be enabled on both the calling and the called sides. ▪ For configuring this table using the Web interface, see Configuring Call Waiting on page 361. ▪ For a description on using ini file table parameters, see Configuring ini File Table Parameters on page 98.
Web: Number of Call Waiting Indications EMS: Call Waiting Number of Indications CLI: nb-of-cw-ind **[NumberOfWaitingIndications]**	Defines the number of call waiting indications that are played to the called telephone that is connected to the device for Call Waiting. The valid range is 1 to 100 indications. The default value is 2. **Note:** This parameter is applicable only to FXS ports.
Web: Time Between Call Waiting Indications EMS: Call Waiting Time Between Indications CLI: time-between-cw **[TimeBetweenWaitingIndications]**	Defines the time (in seconds) between consecutive call waiting indications for call waiting. The valid range is 1 to 100. The default value is 10. **Note:** This parameter is applicable only to FXS ports.
Web/EMS: Time Before Waiting Indications CLI: time-b4-cw-ind **[TimeBeforeWaitingIndications]**	Defines the interval (in seconds) before a call waiting indication is played to the port that is currently in a call. The valid range is 0 to 100. The default time is 0 seconds. **Note:** This parameter is applicable only to FXS ports.
Web/EMS: Waiting Beep Duration CLI: waiting-beep-dur **[WaitingBeepDuration]**	Defines the duration (in msec) of call waiting indications that are played to the port that is receiving the call. The valid range is 100 to 65535. The default value is 300. **Note:** This parameter is applicable only to FXS ports.
EMS: First Call Waiting Tone ID **[FirstCallWaitingToneID]**	Defines the index of the first Call Waiting Tone in the CPT file. This feature enables the called party to distinguish between different call origins (e.g., external versus internal calls). There are three ways to use the distinctive call waiting tones: ▪ Playing the call waiting tone according to the SIP Alert-Info header in the received 180 Ringing SIP response. The value

Parameter	Description
	of the Alert-Info header is added to the value of the FirstCallWaitingToneID parameter. ▪ Playing the call waiting tone according to PriorityIndex in the ToneIndex parameter table. ▪ Playing the call waiting tone according to the parameter “CallWaitingTone#' of a SIP INFO message. The device plays the tone received in the 'play tone CallWaitingTone#' parameter of an INFO message plus the value of this parameter minus 1. The valid range is -1 to 1,000. The default value is -1 (i.e., not used). **Notes:** ▪ This parameter is applicable only to analog interfaces. ▪ It is assumed that all Call Waiting Tones are defined in sequence in the CPT file. ▪ SIP Alert-Info header examples: ✓ Alert-Info:<Bellcore-dr2> ✓ Alert-Info:<http://…/Bellcore-dr2> (where "dr2" defines call waiting tone #2) ▪ The SIP INFO message is according to Broadsoft's application server definition. Below is an example of such an INFO message: INFO sip:06@192.168.13.2:5060 SIP/2.0 Via:SIP/2.0/UDP 192.168.13.40:5060;branch=z9hG4bK040066422630 From: <sip:4505656002@192.168.13.40:5060>;tag=1455352915 To: <sip:06@192.168.13.2:5060> Call-ID:0010-0008@192.168.13.2 CSeq:342168303 INFO Content-Length:28 Content-Type:application/broadsoft play tone CallWaitingTone1

A.12.5.3 Call Forwarding Parameters

The call forwarding parameters are described in the table below.

Table A-41: Call Forwarding Parameters

Parameter	Description
Web: Enable Call Forward CLI: call-forward **[EnableForward]**	Enables the Call Forwarding feature. ▪ **[0]** Disable = Disable the Call Forward service. ▪ **[1]** Enable = Enable Call Forward service (using REFER) (default). For FXS interfaces, the Call Forward table (FwdInfo parameter) must be defined to use the Call Forward service. The device uses REFER messages for call forwarding. **Note:** To use this service, the devices at both ends must support this option.
Web: Call Forwarding Table EMS: SIP Endpoints > Call Forward	
[FwdInfo]	This parameter table forwards (redirects) IP-to-Tel calls (using SIP 302 response) to other device ports or an IP destination, based on the device's port to which the call was originally routed. The format of this parameter is as follows: [FwdInfo] FORMAT FwdInfo_Index = FwdInfo_Type, FwdInfo_Destination, FwdInfo_NoReplyTime, FwdInfo_Module, FwdInfo_Port; [\FwdInfo] Where, ▪ Type = the scenario for forwarding the call: ✓ **[0]** Deactivate = Don't forward incoming calls (default). ✓ **[1]** On Busy = Forward incoming calls when the port is busy. ✓ **[2]** Unconditional = Always forward incoming calls. ✓ **[3]** No Answer = Forward incoming calls that are not answered within the time specified in the 'Time for No Reply Forward' field. ✓ **[4]** On Busy or No Answer = Forward incoming calls when the port is busy or when calls are not answered within the time specified in the 'Time for No Reply Forward' field. ✓ **[5]** Do Not Disturb = Immediately reject incoming calls. ▪ Destination = Telephone number or URI (<number>@<IP address>) to where the call is forwarded. ▪ NoReplyTime = Timeout (in seconds) for No Reply. If you have set the Forward Type for this port to No Answer **[3],** enter the number of seconds the device waits before forwarding the call to the specified phone number. ▪ Module = Module number (where 1 denotes the module in Slot 1). ▪ Port = Port number (where 1 denotes Port 1 of a module). For example: ▪ Below configuration forwards calls originally destined to Port 1 of Module 1 to "1001" upon On Busy: FwdInfo 0 = 1,1001,30,1,1; ▪ Below configuration forwards calls originally destined to Port 2 of Module 1 to an IP address upon On Busy: FwdInfo 1 = 1,2003@10.5.1.1,30,1,2; **Notes:**

Parameter	Description
	▪ The indexing of this parameter starts at 0. ▪ Ensure that the Call Forward feature is enabled (default) for the settings of this table parameter to take effect. To enable Call Forwarding, use the parameter EnableForward. ▪ If the parameter FwdInfo_Destination only contains a telephone number and a Proxy isn't used, the 'forward to' phone number must be specified in the Outbound IP Routing Table (Prefix ini file parameter). ▪ For configuring this table using the Web interface, see Configuring Call Forward on page 358. ▪ For configuring ini file table parameters, see Configuring ini File Table Parameters on page 98.
Call Forward Reminder Ring Parameters **Notes:** ▪ These parameters are applicable only to FXS interfaces. ▪ For a description of this feature, see Call Forward Reminder Ring on page 318.	
Web: Enable NRT Subscription CLI: nrt-subscription **[EnableNRTSubscription]**	Enables endpoint subscription for Ring reminder event notification feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: AS Subscribe IPGroupID CLI: as-subs-ipgroupid **[ASSubscribeIPGroupID]**	Defines the IP Group ID that contains the Application server for Subscription. The valid value range is 1 to 8. The default is -1 (i.e., not configured).
Web: NRT Retry Subscription Time **[NRTSubscribeRetryTime]**	Defines the Retry period (in seconds) for Dialog subscription if a previous request failed. The valid value range is 10 to 7200. The default is 120.
Web: Call Forward Ring Tone ID CLI: cfe-ring-tone-id **[CallForwardRingToneID]**	Defines the ringing tone type played when call forward notification is accepted. The valid value range is 1 to 5. The default is 1.

A.12.5.4 Message Waiting Indication Parameters

The message waiting indication (MWI) parameters are described in the table below.

Table A-42: MWI Parameters

Parameter	Description
Web: Enable MWI EMS: MWI Enable CLI: enable-mwi **[EnableMWI]**	Enables Message Waiting Indication (MWI). ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = MWI service is enabled. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ The device supports only the receipt of SIP MWI NOTIFY messages (the device doesn't generate these messages). ▪ For more information on MWI, see 'Message Waiting Indication' on page 321.
Web/EMS: MWI Analog Lamp CLI: mwi-analog-lamp **[MWIAnalogLamp]**	Enables the visual display of MWI. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enables visual MWI by supplying line voltage of approximately 100 VDC to activate the phone's lamp. **Notes:** ▪ This parameter is applicable only for FXS interfaces. ▪ This parameter can also be configured per Tel Profile (using the TelProfile parameter).
Web/EMS: MWI Display CLI: enable-mwi **[MWIDisplay]**	Enables sending MWI information to the phone display. ▪ **[0]** Disable = MWI information isn't sent to display (default). ▪ **[1]** Enable = The device generates an MWI message (determined by the parameter CallerIDType), which is displayed on the MWI display. **Note:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter can also be configured per Tel Profile (using the TelProfile parameter).
Web: Subscribe to MWI EMS: Enable MWI Subscription CLI: subscribe-to-mwi **[EnableMWISubscription]**	Enables subscription to an MWI server. ▪ **[0]** No = Disables MWI subscription (default). ▪ **[1]** Yes = Enables subscription to an MWI server (defined by the parameter MWIServerIP address). **Note:** To configure whether the device subscribes per endpoint or per the entire device, use the parameter SubscriptionMode.
Web: MWI Server IP Address EMS: MWI Server IP CLI: mwi-srvr-ip-addr **[MWIServerIP]**	Defines the MWI server's IP address. If provided, the device subscribes to this IP address. The MWI server address can be configured as a numerical IP address or as a domain name. If not configured, the Proxy IP address is used instead.
Web/EMS: MWI Server Transport Type CLI: mwi-srvr-transp-type **[MWIServerTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the MWI server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS

Parameter	Description
	Note: When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web: MWI Subscribe Expiration Time EMS: MWI Expiration Time CLI: mwi-subs-expr-time **[MWIExpirationTime]**	Defines the MWI subscription expiration time in seconds. The default is 7200 seconds. The range is 10 to 2,000,000.
Web: MWI Subscribe Retry Time EMS: Subscribe Retry Time CLI: mwi-subs-rtry-time **[SubscribeRetryTime]**	Defines the subscription retry time (in seconds) after last subscription failure. The default is 120 seconds. The range is 10 to 2,000,000.
Web: Subscription Mode CLI: subscription-mode **[SubscriptionMode]**	Determines the method the device uses to subscribe to an MWI server. ▪ **[0]** Per Endpoint = Each endpoint subscribes separately - typically used for FXS interfaces (default). ▪ **[1]** Per Gateway = Single subscription for the entire device - typically used for FXO interfaces.
EMS: ETSI VMWI Type One Standard CLI: etsi-vmwi-type-one-standard **[ETSIVMWITypeOneStandard]**	Determines the ETSI Visual Message Waiting Indication (VMWI) Type 1 sub-standard. ▪ **[0]** = ETSI VMWI between rings (default) ▪ **[1]** = ETSI VMWI before ring DT_AS ▪ **[2]** = ETSI VMWI before ring RP_AS ▪ **[3]** = ETSI VMWI before ring LR_DT_AS ▪ **[4]** = ETSI VMWI not ring related DT_AS ▪ **[5]** = ETSI VMWI not ring related RP_AS ▪ **[6]** = ETSI VMWI not ring related LR_DT_AS **Note:** For this parameter to take effect, a device reset is required.
EMS: Bellcore VMWI Type One Standard CLI: bellcore-vmwi-type-one-standard **[BellcoreVMWITypeOneStandard]**	Determines the Bellcore VMWI sub-standard. ▪ **[0]** = Between rings (default). ▪ **[1]** = Not ring related. **Note:** For this parameter to take effect, a device reset is required.

A.12.5.5 Call Hold Parameters

The call hold parameters are described in the table below.

Table A-43: Call Hold Parameters

Parameter	Description
Web/EMS: Enable Hold CLI: hold **[EnableHold]**	For digital interfaces: Enables interworking of the Hold/Retrieve supplementary service from PRI to SIP. ▪ **[0]** Disable ▪ **[1]** Enable (default) For analog interfaces: If the Hold service is enabled, a user can place the call on hold (or remove from hold) using the Hook Flash button. On receiving a Hold request, the remote party is placed on hold and hears the hold tone. **Notes:** ▪ For digital interfaces: To support interworking of the Hold/Retrieve supplementary service from SIP to ISDN (for QSIG and Euro ISDN), set the parameter EnableHold2ISDN to 1. ▪ For analog interfaces: To use this service, the devices at both ends must support this option. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web/EMS: Hold Format CLI: hold-format **[HoldFormat]**	Determines the format of the SDP in the Re-INVITE hold request. ▪ **[0]** 0.0.0.0 = The SDP "c=" field contains the IP address "0.0.0.0" and the "a=inactive" attribute (default). ▪ **[1]** Send Only = The SDP "c=" field contains the device's IP address and the "a=sendonly" attribute. **Notes:** ▪ The device does not send any RTP packets when it is in hold state (for both hold formats). ▪ For digital interfaces: This parameter is applicable only to QSIG and Euro ISDN protocols.
Web/EMS:Held Timeout CLI: held-timeout **[HeldTimeout]**	Defines the time interval that the device allows for a call to remain on hold. If a Resume (un-hold Re-INVITE) message is received before the timer expires, the call is renewed. If this timer expires, the call is released (terminated). ▪ **[-1]** = The call is placed on hold indefinitely until the initiator of the on hold retrieves the call again (default). ▪ **[0 - 2400]** = Time to wait (in seconds) after which the call is released.
Web: Call Hold Reminder Ring Timeout EMS: CHRR Timeout CLI: call-hold-remnd-rng **[CHRRTimeout]**	Defines the duration (in seconds) that the Call Hold Reminder Ring is played. If a user hangs up while a call is still on hold or there is a call waiting, then the FXS interface immediately rings the extension for the duration specified by this parameter. If the user off-hooks the phone, the call becomes active. The valid range is 0 to 600. The default value is 30. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This Reminder Ring feature can be disabled using the DisableReminderRing parameter.
CLI: dis-reminder-ring	Disables the reminder ring, which notifies the FXS user of a call on hold

Parameter	Description
[DisableReminderRing]	or a waiting call when the phone is returned to on-hook position. ▪ **[0]** = (default) The reminder ring feature is active. In other words, if a call is on hold or there is a call waiting, and the phone is changed from offhook to onhook, the phone rings (for a duration defined by the CHRRTimeout parameter) to "remind" you of the call hold or call waiting. ▪ **[1]** = Disables the reminder ring. If a call is on hold or there is a call waiting and the phone is changed from offhook to onhook, the call is released (and the device sends a SIP BYE to the IP). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter is typically used for MLPP, allowing preemption to clear held calls.
CLI: dtmf-during-hold **[PlayDTMFduringHold]**	Determines whether the device sends DTMF signals (or DTMF SIP INFO message) when a call is on hold. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable - If the call is on hold, the device stops playing the Held tone (if it is played) and sends DTMF: ✓ To Tel side: plays DTMF digits according to the received SIP INFO message(s). (The stopped Held tone is not played again.) ✓ To IP side: sends DTMF SIP INFO messages to an IP destination if it detects DTMF digits from the Tel side.

A.12.5.6 Call Transfer Parameters

The call transfer parameters are described in the table below.

Table A-44: Call Transfer Parameters

Parameter	Description
Web/EMS: Enable Transfer CLI: enable-transfer **[EnableTransfer]**	Enables the Call Transfer feature. ▪ **[0]** Disable = Disable the call transfer service. ▪ **[1]** Enable = The device responds to a REFER message with the Referred-To header to initiate a call transfer (default). For analog interfaces: If the transfer service is enabled, the user can activate Transfer using hook-flash signaling. If this service is enabled, the remote party performs the call transfer. **Notes:** ▪ To use call transfer, the devices at both ends must support this option. ▪ To use call transfer, set the parameter EnableHold to 1.
Web: Transfer Prefix EMS: Logical Prefix For Transferred Call CLI: transfer-prefix **[xferPrefix]**	Defines the string that is added as a prefix to the transferred/forwarded called number when the REFER/3xx message is received. **Notes:** ▪ The number manipulation rules apply to the user part of the Refer-To and/or Contact URI before it is sent in the INVITE message. ▪ This parameter can be used to apply different

Parameter	Description
	manipulation rules to differentiate transferred/forwarded (only for analog interfaces) number from the originally dialed number.
Web: Transfer Prefix IP 2 Tel **[XferPrefixIP2Tel]**	Defines the prefix that is added to the destination number received in the SIP Refer-To header (for IP-to-Tel calls). This parameter is applicable to FXO/CAS blind transfer modes, i.e., LineTransferMode = 1, 2 or 3, and TrunkTransferMode = 1 or 3 (for CAS). The valid range is a string of up to 9 characters. The default is an empty string. **Note:** This parameter is also applicable to ISDN Blind Transfer, according to AT&T Toll Free Transfer Connect Service (TR 50075) “Courtesy Transfer-Human-No Data”. To support this transfer mode, you need to configure the parameter XferPrefixIP2Tel to "*8" and the parameter TrunkTransferMode to 5.
Web/EMS: Enable Semi-Attended Transfer CLI: semi-att-transfer **[EnableSemiAttendedTransfer]**	Determines the device behavior when Transfer is initiated while in Alerting state. ▪ **[0]** Disable = Send REFER with the Replaces header (default). ▪ **[1]** Enable = Send CANCEL, and after a 487 response is received, send REFER without the Replaces header.
Web: Blind EMS: Blind Transfer CLI: blind-transfer **[KeyBlindTransfer]**	Defines the keypad sequence to activate blind transfer for established Tel-to-IP calls. The Tel user can perform blind transfer by dialing the KeyBlindTransfer digits, followed by a transferee destination number. After the KeyBlindTransfer DTMF digits sequence is dialed, the current call is put on hold (using a Re-INVITE message), a dial tone is played to the channel, and then the phone number collection starts. After the destination phone number is collected, it is sent to the transferee in a SIP REFER request in a Refer-To header. The call is then terminated and a confirmation tone is played to the channel. If the phone number collection fails due to a mismatch, a reorder tone is played to the channel. **Note:** For FXS/FXO interfaces, it is possible to configure whether the KeyBlindTransfer code is added as a prefix to the dialed destination number, by using the parameter KeyBlindTransferAddPrefix.
EMS: Blind Transfer Add Prefix **[KeyBlindTransferAddPrefix]**	Determines whether the device adds the Blind Transfer code (defined by the KeyBlindTransfer parameter) to the dialed destination number. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This parameter is applicable only to FXO and FXS interfaces.

Parameter	Description
EMS: Blind Transfer Disconnect Timeout **[BlindTransferDisconnectTimeout]**	Defines the duration (in milliseconds) for which the device waits for a disconnection from the Tel side after the Blind Transfer Code (KeyBlindTransfer) has been identified. When this timer expires, a SIP REFER message is sent toward the IP side. If this parameter is set to 0, the REFER message is immediately sent. The valid value range is 0 to 1,000,000. The default is 0.
Web: QSIG Path Replacement Mode CLI: qsig-path-replacement-md **[QSIGPathReplacementMode]**	Enables QSIG transfer for IP-to-Tel and Tel-to-IP calls. ▪ **[0]** IP2QSIGTransfer = Enables IP-to-QSIG transfer. (default) ▪ **[1]** QSIG2IPTransfer = Enables QSIG-to-IP transfer.
CLI: replace-tel2ip-calnum-to **[ReplaceTel2IPCallingNumTimeout]**	Defines the maximum duration (timeout) to wait between call Setup and Facility with Redirecting Number for replacing the calling number (for Tel-to-IP calls). The valid value range is 0 to 10,000 msec. The default is 0. The interworking of the received Setup message to a SIP INVITE is suspended when this parameter is set to any value greater than 0. This means that the redirecting number in the Setup message is not checked. When a subsequent Facility with Call Transfer Complete/Update is received with a non-empty Redirection Number, the Calling Number is replaced with the received redirect number in the sent INVITE message. If the timeout expires, the device sends the INVITE without changing the calling number. **Notes:** ▪ The suspension of the INVITE message occurs for all calls. ▪ This parameter is applicable to QSIG.

A.12.5.7 Three-Way Conferencing Parameters

The three-way conferencing parameters are described in the table below.

Table A-45: Three-Way Conferencing Parameters

Parameter	Description
Web: Enable 3-Way Conference EMS: Enable 3 Way CLI: enable-3w-conf **[Enable3WayConference]**	Enables the 3-Way Conference feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: 3-Way Conference Mode EMS: 3 Way Mode CLI: 3w-conf-mode **[3WayConferenceMode]**	Determines the mode of operation when the 3-Way Conference feature is used. ▪ **[0]** AudioCodes Media Server = The Conference-initiating INVITE (sent by the device) uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. This conference mode is used when operating with AudioCodes IPMedia conferencing server. (Default) ▪ **[1]** Non-AudioCodes Media Server = The Conference-initiating

Parameter	Description
	INVITE (sent by the device) uses only the ConferenceID as the Request-URI. The conference server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is then included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the conference using this conference URI. ▪ **[2]** On Board = On-board three-way conference. The conference is established on the device without the need of an external Conference server. The device sets up the conference call using its IP media channels. These channels are obtained from the IP media module (i.e., MPM module). Note that the device must be housed with MPM module(s) to support three-way conferencing. The device supports up to five simultaneous, on-board three-way conference calls. **Notes:** ▪ This parameter is applicable only to FXS and BRI interfaces. ▪ Three-way conferencing using an external conference server is supported only by FXS interfaces. ▪ When using an external conference server (options **[0]** or **[1]),** a conference call with up to six participants can be established.
Web: Establish Conference Code EMS: Establish Code CLI: estb-conf-code **[ConferenceCode]**	Defines the DTMF digit pattern, which upon detection generates the conference call when three-way conferencing is enabled (Enable3WayConference is set to 1). The valid range is a 25-character string. The default is “!” (Hook-Flash). **Note:** If the FlashKeysSequenceStyle parameter is set to 1 or 2, the setting of the ConferenceCode parameter is overridden.
Web/EMS: Conference ID CLI: conf-id **[ConferenceID]**	Defines the Conference Identification string (up to 16 characters). The default value is 'conf'. For 3-way conferencing using an external media server: The device uses this identifier in the Conference-initiating INVITE that is sent to the media server when Enable3WayConference is set to 1. When using the Media Processing Module (MPM): To join a conference, the INVITE URI must include the Conference ID string, preceded by the number of the participants in the conference, and terminated by a unique number. For example: INVITE sip:4MyConference1234@10.1.10.10. INVITE messages with the same URI join the same conference. For example: ConferenceID = MyConference.

A.12.5.8 Emergency Call Parameters

The emergency call parameters are described in the table below.

Table A-46: Emergency Call Parameters

Parameter	Description
[E911Gateway]	Enables Enhanced 9-1-1 support for ELIN handling in Microsoft Lync Server 2010 environment. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
[E911CallbackTimeout]	Defines the maximum interval within which the PSAP can use the ELIN to call back the E9-1-1 caller. This interval starts from when the initial call established with the PSAP is terminated. The valid range is 1 to 60 (minutes). The default value is 30.
[EmergencySpecialReleaseCause]	Enables the device to send a SIP 503 "Service Unavailable" response if an emergency call cannot be established (i.e., rejected). This can occur due to the PSTN (for example, the destination is busy or not found) or ELIN Gateway (for example, lack of resources or an internal error). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: Enable 911 PSAP **[Enable911PSAP]**	Enables the support for the E911 DID protocol, according to the Bellcore GR-350-CORE standard. This protocol defines signaling between E911 Tandem Switches and the PSAP, using analog loop-start lines. The FXO device can be installed instead of an E911 switch, connected directly to PSAP DID loop-start lines. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ This parameter can also be configured per Tel Profile, using the TelProfile parameter.
Web/EMS: Emergency Numbers CLI: emerg-nbs **[EmergencyNumbers]**	Defines a list of “emergency” numbers. When one of these numbers is dialed, the outgoing INVITE message includes the SIP Priority and Resource-Priority headers. If the user places the phone on-hook, the call is not disconnected. Instead, a Hold Re-INVITE request is sent to the remote party. Only if the remote party disconnects the call (i.e., a BYE is received) or a timer expires (set by the EmergencyRegretTimeout parameter) is the call terminated. This scenario is applicable only to FXS interfaces. These emergency numbers can also be used for pre-emption of E911 IP-to-Tel calls when there are unavailable (busy) channels. In this scenario, the device terminates one of the busy channels and sends the E911 call to this channel. This feature is enabled by setting the CallPriorityMode parameter to 2 (“Emergency”) and by defining an emergency number value of “911” for the EmergencyNumbers parameter. For a description of this feature, see 'Pre-empting Existing Call for E911 IP-to-Tel Call' on page 332. This scenario is applicable to FXS/FXO, CAS, and ISDN interfaces.

Parameter	Description
	The list can include up to four different numbers, where each number can be up to four digits long. Example: EmergencyNumbers = '100','911','112'
Web: Emergency Calls Regret Timeout EMS: Emergency Regret Timeout CLI: emerg-calls-regrt-t-out **[EmergencyRegretTimeout]**	Defines the time (in minutes) that the device waits before tearing-down an emergency call (defined by the parameter EmergencyNumbers). Until this time expires, an emergency call can only be disconnected by the remote party, typically, by a Public Safety Answering Point (PSAP). The valid range is 1 to 30. The default value is 10. **Note:** This parameter is applicable only to FXS interfaces.

A.12.5.9 Call Cut-Through Parameters

The call cut-through parameters are described in the table below.

Table A-47: Call Cut-Through Parameters

Parameter	Description
Web: Enable Calls Cut Through EMS: Cut Through CLI: calls-cut-through **[CutThrough]**	Enables FXS endpoints to receive incoming IP calls while the port is in off-hook state. ▪ **[0]** Disable (default) ▪ **[1]** Enable If enabled, the FXS interface answers the call and 'cuts through' the voice channel if there is no other active call on the port, even if the port is in off-hook state. When the call is terminated (by the remote IP party), the device plays a reorder tone for a user-defined time (configured by the CutThroughTimeForReorderTone parameter) and is then ready to answer the next incoming call without on-hooking the phone. The waiting call is automatically answered by the device when the current call is terminated (configured by setting the parameter EnableCallWaiting to 1). **Note:** This feature is applicable only to FXS interfaces.
CLI: cut-through-anable **[DigitalCutThrough]**	Enables PSTN CAS channels/endpoints to receive incoming IP calls even if the B-channels are in off-hook state. ▪ **[0]** Disabled (default) ▪ **[1]** Enabled When enabled, this feature operates as follows: 1 A Tel-to-IP call is established (connected) by the device for a B-channel. 2 The device receives a SIP BYE (i.e., IP side ends the call) and plays a reorder tone to the PSTN side for the duration set by the CutThroughTimeForReOrderTone parameter. The device releases the call towards the IP side (sends a SIP 200 OK). 3 The PSTN side, for whatever reason, remains off-hook. 4 If a new IP call is received for this B-channel after the reorder tone has ended, the device "cuts through" the channel and connects the call immediately (despite the B-channel being in physical off-hook state) without playing a ring tone. If an IP call is received while the reorder

Parameter	Description
	tone is played, the device rejects the call. **Notes:** ▪ If this parameter is disabled and the PSTN side remains in off-hook state after the IP call ends the call, the device releases the call after 60 seconds. ▪ A special CAS table can be used to report call status events (Active/Idle) to the PSTN side during Cut Through mode. ▪ The Digital Cut-Through feature can also be configured as a Tel Profile (using the TelProfile parameter) and therefore, assigned to specific B-channels that use specific CAS tables.

A.12.5.10 Automatic Dialing Parameters

The automatic dialing upon off-hook parameters are described in the table below.

Table A-48: Automatic Dialing Parameters

Parameter	Description
Web: Automatic Dialing Table **EMS: SIP Endpoints > Auto Dial**	
[TargetOfChannel]	This *para*meter table defines telephone numbers that are automatically dialed when a specific FXS or FXO port is used (i.e., telephone is off-hooked). The format of this parameter is as follows: **[TargetOfChannel]** FORMAT TargetOfChannel_Index = TargetOfChannel_Destination, TargetOfChannel_Type, TargetOfChannel_Module, TargetOfChannel_Port, TargetOfChannel_HotLineToneDuration; **[\TargetOfChannel]** Where, ▪ Destination = Destination phone number that you want dialed. ▪ Type: ✓ **[0]** Disable = automatic dialing is disabled. ✓ **[1]** Enable = Destination phone number is automatically dialed if phone is off-hooked (for FXS interface) or ring signal is applied to port (FXO interface). ✓ **[2]** Hotline = enables the Hotline feature where if the phone is off-hooked and no digit is pressed for a user-defined duration (configured by the parameter HotLineToneDuration), the destination phone number is automatically dialed. ▪ Module = Module number (where 1 denotes the module in Slot 1). ▪ Port = Port number (where 1 denotes the Port 1 of the module). ▪ HotLineToneDuration = if Hotline is enabled and the phone (connected to the specific port) is off-hooked and no digit is pressed for this user-defined duration (timeout), the device automatically initiates a call to the user-defined destination phone number. The value range is 0 to 60 seconds, with default as 16. Note that you can use the "global" HotLineToneDuration parameter to define this interval for all ports. For example, the below configuration defines automatic dialing of phone number 911 when the phone that is connected to Port 1 of Module 1 is off-hooked for over 10 seconds: TargetOfChannel 0 = 911, 1, 1, 1 ,10; (phone number "911" is automatically dialed for Port 1 of Module 1 after being off-hooked for 10

Parameter	Description
	seconds) **Notes:** ▪ This is parameter is applicable to FXS and FXO interfaces. ▪ The indexing of this *ini* file table parameter starts at 0. ▪ Define this parameter for each device port that implements Automatic Dialing. ▪ After a ring signal is detected on an 'Enabled' FXO port, the device initiates a call to the destination number without seizing the line. The line is seized only after the call is answered. After a ring signal is detected on a 'Disabled' or 'Hotline' FXO port, the device seizes the line. ▪ For configuring this table using the Web interface, see 'Configuring Automatic Dialing' on page 356. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.12.5.11 Direct Inward Dialing Parameters

The Direct Inward Dialing (DID) parameters are described in the table below.

Table A-49: DID Parameters

Parameter	Description
Web/EMS: Enable DID Wink CLI: did-wink-enbl **[EnableDIDWink]**	Enables Direct Inward Dialing (DID) using Wink-Start signaling. ▪ **[0]** Disable = Disables DID Wink(default). ▪ **[1]** Enable = Enables DID Wink. If enabled, the device can be used for connection to EIA/TIA-464B DID Loop Start lines. Both FXO (detection) and FXS (generation) are supported. An FXO interface dials DTMF digits after a Wink signal is detected (instead of a Dial tone). An FXS interface generates the Wink signal after the detection of off-hook (instead of playing a Dial tone). **Note:** This parameter can also be configured per Tel Profile, using the TelProfile parameter.
Web/EMS: Delay Before DID Wink CLI: delay-b4-did-wink **[DelayBeforeDIDWink]**	Defines the time interval (in msec) between detection of off-hook and generation of a DID Wink. The valid range is 0 to 1,000. The default value is 0. **Note:** This parameter is applicable only to FXS interfaces.
EMS: NTT DID Signalling Form CLI: NTT-DID-signaling-form **[NTTDIDSignallingForm]**	Determines the type of DID signaling support for NTT (Japan) modem: DTMF- or Frequency Shift Keying (FSK)-based signaling. The devices can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. ▪ **[0]** = FSK-based signaling (default) ▪ **[1]** = DTMF-based signaling **Note:** This parameter is applicable only to FXS interfaces.
EMS: Enable DID **[EnableDID]**	This *para*meter table enables support for Japan NTT 'Modem' DID. FXS interfaces can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. The DID signal can be sent alone or combined with an NTT Caller ID signal.

Parameter	Description
	The format of this parameter is as follows: **[EnableDID]** FORMAT EnableDID_**Index** = EnableDID_**IsEnable**, EnableDID_Port, EnableDID_Module; **[\EnableDID]** Where, ▪ IsEnable = Enables **[1]** or disables **[0]** (default) Japan NTT Modem DID support. ▪ Port = Port number. ▪ Module = Module number. For example: EnableDID 0 = 1,1,2; (DID is enabled on Port 1 of Module 2) **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
CLI: wink-time **[WinkTime]**	Defines the time (in msec) elapsed between two consecutive polarity reversals. This parameter can be used for DID signaling, for example, E911 lines to the Public Safety Answering Point (PSAP), according to the Bellcore GR-350-CORE standard (refer to the ini file parameter Enable911PSAP). The valid range is 0 to 4,294,967,295. The default is 200. **Notes:** ▪ This parameter is applicable to FXS and FXO interfaces. ▪ For this parameter to take effect, a device reset is required.

A.12.5.12 MLPP Parameters

The Multilevel Precedence and Preemption (MLPP) parameters are described in the table below.

Table A-50: MLPP Parameters

Parameter	Description
Web/EMS: Call Priority Mode CLI: call-prio-mode **[CallPriorityMode]**	Enables priority calls handling. ▪ **[0]** Disable = Disable (default). ▪ **[1]** MLPP = MLPP Priority Call handling is enabled. MLPP prioritizes call handling whereby the relative importance of various kinds of communications is strictly defined, allowing higher precedence communication at the expense of lower precedence communications. Higher priority calls override less priority calls when, for example, congestion occurs in a network. ▪ **[2]** Emergency = Preemption of IP-to-Tel E911 emergency calls. If the device receives an E911 call and there are unavailable channels to receive the call, the device terminates one of the channel calls and sends the E911 call to that channel. The preemption is done only on a channel pertaining to the same Trunk Group for which the E911 call was initially destined and if the channel select mode (configured by the ChannelSelectMode parameter) is set to other than “By Dest Number” (0). The preemption is done only if the incoming IP-to-Tel call is identified as an emergency call. The device identifies

Parameter	Description
	emergency calls by one of the following: ✓ The destination number of the IP call matches one of the numbers defined by the EmergencyNumbers parameter. (For E911, you must define this parameter with the value "911".) ✓ The incoming SIP INVITE message contains the "emergency" value in the Priority header. **Notes:** ✓ Applicable to FXS/FXO, CAS, and ISDN interfaces. ✓ For FXO interfaces, the preemption is done only on existing IP-to-Tel calls. In other words, if all the current FXO channels are busy with calls that were initiated by the FXO (i.e., Tel-to-IP calls), new incoming emergency IP-to-Tel calls are dropped. ✓ For more information, see 'Pre-empting Existing Call for E911 IP-to-Tel Call' on page 332.
Web: MLPP Default Namespace EMS: Default Name Space CLI: mlpp-dflt-namespace **[MLPPDefaultNamespace]**	Determines the namespace used for MLPP calls received from the ISDN side and destined for the Application server. The namespace value is not present in the Precedence IE of the PRI Setup message. Therefore, the value is used in the Resource-Priority header of the outgoing SIP INVITE request. ▪ **[1]** DSN = DSN (default) ▪ **[2]** DOD = DOD ▪ **[3]** DRSN = DRSN ▪ **[5]** UC = UC
Web/EMS: Default Call Priority CLI: dflt-call-prio **[SIPDefaultCallPriority]**	Determines the default call priority for MLPP calls. ▪ **[0]** 0 = ROUTINE (default) ▪ **[2]** 2 = PRIORITY ▪ **[4]** 4 = IMMEDIATE ▪ **[6]** 6 = FLASH ▪ **[8]** 8 = FLASH-OVERRIDE ▪ **[9]** 9 = FLASH-OVERRIDE-OVERRIDE If the incoming SIP INVITE request doesn't contain a valid priority value in the SIP Resource-Priority header, the default value is used in the Precedence IE (after translation to the relevant ISDN Precedence value) of the outgoing PRI Setup message. If the incoming PRI Setup message doesn't contain a valid Precedence Level value, the default value is used in the Resource-Priority header of the outgoing SIP INVITE request. In this scenario, the character string is sent without translation to a numerical value.
Web: MLPP DiffServ EMS: Diff Serv CLI: mlpp-diffserv **[MLPPDiffserv]**	Defines the DiffServ value (differentiated services code point/DSCP) used in IP packets containing SIP messages that are related to MLPP calls. This parameter defines DiffServ for incoming and outgoing MLPP calls with the Resource-Priority header. The valid range is 0 to 63. The default value is 50.
Web/EMS: Preemption Tone Duration CLI: preemp-tone-dur **[PreemptionToneDuration]**	Defines the duration (in seconds) in which the device plays a preemption tone to the Tel and IP sides if a call is preempted. The valid range is 0 to 60. The default is 3. **Note:** If set to 0, no preemption tone is played.

Parameter	Description
Web: MLPP Normalized Service Domain EMS: Normalized Service Domain CLI: mlpp-norm-ser-dmn **[MLPPNormalizedServiceDomain]**	Defines the MLPP normalized service domain string. If the device receives an MLPP ISDN incoming call, it uses the parameter (if different from 'FFFFFF') as a Service domain in the SIP Resource-Priority header in outgoing INVITE messages. If the parameter is configured to 'FFFFFF', the Resource-Priority header is set to the MLPP Service Domain obtained from the Precedence IE. The valid value is 6 hexadecimal digits. The default is '000000'. **Note:** This parameter is applicable only to the MLPP NI-2 ISDN variant with CallPriorityMode set to 1.
CLI: mlpp-nwrk-id **[MLPPNetworkIdentifier]**	Defines the MLPP network identifier (i.e., International prefix or Telephone Country Code/TCC) for IP-to-ISDN calls, according to the UCR 2008 and ITU Q.955 specifications. The valid range is 1 to 999. The default is 1 (i.e., USA). The MLPP network identifier is sent in the Facility IE of the ISDN Setup message. For example: ▪ MLPPNetworkIdentifier set to default (i.e., USA, 1): PlaceCall- MLPPNetworkID:0100 MlppServiceDomain:123abc, MlppPrecLevel:5 Fac(1c): 91 a1 15 02 01 05 02 01 19 30 0d 0a 01 05 0a 01 01 04 05 01 00 12 3a bc ▪ MLPPNetworkIdentifier set to 490: PlaceCall- MLPPNetworkID:9004 MlppServiceDomain:123abc, MlppPrecLevel:5 Fac(1c): 91 a1 15 02 01 0a 02 01 19 30 0d 0a 01 05 0a 01 01 04 05 90 04 12 3a bc
Web: MLPP Default Service Domain EMS: Default Service Domain CLI: mlpp-dflt-srv-domain **[MLPPDefaultServiceDomain]**	Defines the MLPP default service domain string. If the device receives a non-MLPP ISDN incoming call (without a Precedence IE), it uses the parameter (if different than "FFFFFF") as a Service domain in the SIP Resource-Priority header in outgoing (Tel-to-IP calls) INVITE messages. This parameter is used in conjunction with the parameter SIPDefaultCallPriority. If MLPPDefaultServiceDomain is set to 'FFFFFF', the device interworks the non-MLPP ISDN call to non-MLPP SIP call, and the outgoing INVITE does not contain the Resource-Priority header. The valid value is a 6 hexadecimal digits. The default is "000000". **Note:** This parameter is applicable only to the MLPP NI-2 ISDN variant with CallPriorityMode set to 1.
Web/EMS: Precedence Ringing Type CLI: precedence-ringing **[PrecedenceRingingType]**	Defines the index of the Precedence Ringing tone in the Call Progress Tones (CPT) file. This tone is used when the parameter CallPriorityMode is set to 1 and a Precedence call is received from the IP side. The valid range is -1 to 16. The default value is -1 (i.e., plays standard Ringing tone). **Note:** This parameter is applicable only to analog interfaces.
EMS: E911 MLPP Behavior CLI: e911-mlpp-bhvr **[E911MLPPBehavior]**	Defines the E911 (or Emergency Telecommunication Services/ETS) MLPP Preemption mode: ▪ **[0]** Standard Mode - ETS calls have the highest priority and preempt any MLPP call (default). ▪ **[1]** Treat as routine mode - ETS calls are handled as routine calls. **Note:** This parameter is applicable only to analog interfaces.

Parameter	Description
CLI: resource-prio-req **[RPRequired]**	Determines whether the SIP resource-priority tag is added in the SIP Require header of the INVITE message for Tel-to-IP calls. ▪ **[0]** Disable = Excludes the SIP resource-priority tag from the SIP Require header. ▪ **[1]** Enable (default) = Adds the SIP resource-priority tag in the SIP Require header. **Note:** This parameter is applicable only to MLPP priority call handling (i.e., only when the CallPriorityMode parameter is set to 1).
Multiple Differentiated Services Code Points (DSCP) per MLPP Call Priority Level (Precedence) Parameters The MLPP service allows placement of priority calls, where properly validated users can preempt (terminate) lower-priority phone calls with higher-priority calls. For each MLPP call priority level, the DSCP can be set to a value from 0 to 63. The Resource Priority value in the Resource-Priority SIP header can be one of the following: **MLPP Precedence Level** — **Precedence Level in Resource-Priority SIP Header** 0 (lowest) — routine 2 — priority 4 — immediate 6 — flash 8 — flash-override 9 (highest) — flash-override-override	
Web/EMS: RTP DSCP for MLPP Routine CLI: dscp-4-mlpp-rtn **[MLPPRoutineRTPDSCP]**	Defines the RTP DSCP for MLPP Routine precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Priority CLI: dscp-4-mlpp-prio **[MLPPPriorityRTPDSCP]**	Defines the RTP DSCP for MLPP Priority precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Immediate CLI: dscp-4-mlpp-immed **[MLPPImmediateRTPDSCP]**	Defines the RTP DSCP for MLPP Immediate precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Flash CLI: dscp-4-mlpp-flsh **[MLPPFlashRTPDSCP]**	Defines the RTP DSCP for MLPP Flash precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Flash Override CLI: dscp-4-mlpp-flsh-ov **[MLPPFlashOverRTPDSCP]**	Defines the RTP DSCP for MLPP Flash-Override precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP

Parameter	Description
	Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Flash-Override-Override CLI: dscp-4-mlpp-flsh-ov-ov **[MLPPFlashOverOverRTPDSCP]**	Defines the RTP DSCP for MLPP Flash-Override-Override precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).

A.12.5.13 ISDN BRI Parameters

The automatic dialing upon off-hook parameters are described in the table below.

Table A-51: Automatic Dialing Parameters

Parameter	Description
Web: ISDN Supp Services Table	
[ISDNSuppServ]	This *para*meter table defines BRI phone extension numbers per BRI port and configures various ISDN supplementary services per BRI endpoint. The format of this parameter is as follows: [ISDNSuppServ] FORMAT ISDNSuppServ_Index = ISDNSuppServ_PhoneNumber, ISDNSuppServ_Module, ISDNSuppServ_Port, ISDNSuppServ_UserId, ISDNSuppServ_UserPassword, ISDNSuppServ_CallerID, ISDNSuppServ_IsPresentationRestricted, ISDNSuppServ_IsCallerIDEnabled; [\ISDNSuppServ] For example: ISDNSuppServ 0 = 400, 1, 1, user, pass, callerid, 0, 1; ISDNSuppServ 1 = 401, 1, 1, user, pass, callerid, 0, 1; **Notes:** ▪ For a description of each table parameter and for configuring the table using the Web interface, see 'Configuring ISDN Supplementary Services' on page 346. ▪ For configuring ini file table parameters, see 'Configuring ini File Table Parameters' on page 98.
BRI-to-SIP Supplementary Services Codes for Call Forward **Note:** Upon receipt of an ISDN Facility message for call forward from the BRI phone, the device sends a SIP INVITE to the softswitch with a user-defined code in the SIP To header, representing the reason for the call forward. For more information on BRI call forwarding, see 'BRI Call Forwarding' on page 319.	
Call Forward Unconditional **[SuppServCodeCFU]**	Defines the prefix code for activating Call Forward Unconditional sent to the softswitch. The valid value is a string. The default is an empty string. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').
Call Forward Unconditional Deactivation **[SuppServCodeCFUDeact]**	Defines the prefix code for deactivating Call Forward Unconditional Deactivation sent to the softswitch. The valid value is a string. The default is an empty string. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').

Parameter	Description
Call Forward on Busy **[SuppServCodeCFB]**	Defines the prefix code for activating Call Forward on Busy sent to the softswitch. The valid value is a string. The default is an empty string. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').
Call Forward on Busy Deactivation **[SuppServCodeCFBDeact]**	Defines the prefix code for deactivating Call Forward on Busy Deactivation sent to the softswitch. The valid value is a string. The default is an empty string. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').
Call Forward on No Reply **[SuppServCodeCFNR]**	Defines the prefix code for activating Call Forward on No Reply sent to the softswitch. The valid value is a string. The default is an empty string. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').
Call Forward on No Reply Deactivation **[SuppServCodeCFNRDeact]**	Defines the prefix code for deactivating Call Forward on No Reply Deactivation sent to the softswitch. The valid value is a string. The default is an empty string. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').

A.12.6 PSTN Parameters

This subsection describes the device's PSTN parameters.

A.12.6.1 General Parameters

The general PSTN parameters are described in the table below.

Table A-52: General PSTN Parameters

Parameter	Description
Web/EMS: Protocol Type CLI: protocol **[ProtocolType]**	Defines the PSTN protocol for all the Trunks. To configure the protocol type for a specific Trunk, use the *ini* file parameter ProtocolType_x: ▪ **[0]** NONE ▪ **[1]** E1 EURO ISDN = ISDN PRI Pan-European (CTR4) protocol ▪ **[2]** T1 CAS = Common T1 robbed bits protocols including E&M wink start, E&M immediate start, E&M delay dial/start and loop-start and ground start. ▪ **[3]** T1 RAW CAS ▪ **[4]** T1 TRANSPARENT = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 24 of all trunks are mapped to DSP channels. ▪ **[5]** E1 TRANSPARENT 31 = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 31 of each trunk are mapped to DSP channels. ▪ **[6]** E1 TRANSPARENT 30 = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 31, excluding time slot 16 of all trunks are mapped to DSP channels. ▪ **[7]** E1 MFCR2 = Common E1 MFC/R2 CAS protocols (including line signaling and compelled register signaling). ▪ **[8]** E1 CAS = Common E1 CAS protocols (including line signaling and MF/DTMF address transfer). ▪ **[9]** E1 RAW CAS ▪ **[10]** T1 NI2 ISDN = National ISDN 2 PRI protocol ▪ **[11]** T1 4ESS ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 4ESS switch. ▪ **[12]** T1 5ESS 9 ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 5ESS-9 switch. ▪ **[13]** T1 5ESS 10 ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 5ESS-10 switch. ▪ **[14]** T1 DMS100 ISDN = ISDN PRI protocol for the Nortel™ DMS switch. ▪ **[15]** J1 TRANSPARENT ▪ **[16]** T1 NTT ISDN = ISDN PRI protocol for the Japan - Nippon Telegraph Telephone (known also as INS 1500). ▪ **[17]** E1 AUSTEL ISDN = ISDN PRI protocol for the Australian Telecom. ▪ **[18]** E1 HKT ISDN = ISDN PRI (E1) protocol for the Hong Kong - HKT. ▪ **[19]** E1 KOR ISDN = ISDN PRI protocol for Korean Operator (similar to ETSI). ▪ **[20]** T1 HKT ISDN = ISDN PRI (T1) protocol for the Hong Kong

Parameter	Description
	- HKT. ▪ **[21]** E1 QSIG = ECMA 143 QSIG over E1 ▪ **[22]** E1 TNZ = ISDN PRI protocol for Telecom New Zealand (similar to ETSI) ▪ **[23]** T1 QSIG = ECMA 143 QSIG over T1 ▪ **[30]** E1 FRENCH VN6 ISDN = France Telecom VN6 ▪ **[31]** E1 FRENCH VN3 ISDN = France Telecom VN3 ▪ **[34]** T1 EURO ISDN =ISDN PRI protocol for Euro over T1 ▪ **[35]** T1 DMS100 Meridian ISDN = ISDN PRI protocol for the Nortel™ DMS Meridian switch ▪ **[36]** T1 NI1 ISDN = National ISDN 1 PRI protocol ▪ **[40]** E1 NI2 ISDN = National ISDN 2 PRI protocol over E1 ▪ **[50]** BRI EURO ISDN = Euro ISDN over BRI ▪ **[54]** BRI QSIG = QSIG over BRI ▪ **[55]** BRI FRENCH VN6 ISDN = VN6 over BRI ▪ **[56]** BRI NTT = BRI ISDN Japan (Nippon Telegraph) **Notes:** ▪ All PRI trunks must be configured as the same line type (either E1 or T1). The device can support different variants of CAS and PRI protocols on different E1/T1 spans (no more than four simultaneous PRI variants). ▪ BRI trunks can operate with E1 or T1 trunks.
[ProtocolType_x]	Defines the protocol type for a specific trunk ID (where x denotes the Trunk ID and 0 is the first trunk). For more information, see the ProtocolType parameter.
[ISDNTimerT310]	Defines the T310 override timer for DMS, Euro ISDN, and ISDN NI2 variants. An ISDN timer is started when a Q.931 Call Proceeding message is received. The timer is stopped when a Q.931 Alerting, Connect, or Disconnect message is received from the other end. If no ISDN Alerting, Progress, or Connect message is received within the duration of T310 timer, the call clears. The valid value range is 0 to 600 seconds. The default is 0 (i.e., use the default timer value according to the protocol's specifications). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When both the parameters ISDNDmsTimerT310 and ISDNTimerT310 are configured, the value of the parameter ISDNTimerT310 prevails.
[ISDNDMSTimerT310]	Overrides the T310 timer for the DMS-100 ISDN variant. T310 defines the timeout between the receipt of a Proceeding message and the receipt of an Alerting/Connect message. The valid range is 10 to 30. The default value is 10 (seconds). **Notes:** ▪ Instead of configuring this parameter, it is recommended to use the parameter ISDNTimerT310. ▪ This parameter is applicable only to Nortel DMS and Nortel MERIDIAN PRI variants (ProtocolType = 14 and 35).
[ISDNJapanNTTTimerT3JA]	Defines the T3_JA timer (in seconds). This parameter overrides the internal PSTN T301 timeout on the Users Side (TE side).

Parameter	Description
	If an outgoing call from the device to ISDN is not answered during this timeout, the call is released. The valid range is 10 to 240. The default value is 50. **Notes:** ▪ This timer is also affected by the parameter PSTNAlertTimeout. ▪ This parameter is applicable only to the Japan NTT PRI variant (ProtocolType = 16).
Web/EMS: Trace Level **[TraceLevel]**	Defines the trace level: ▪ **[0]** No Trace (default) ▪ **[1]** Full ISDN Trace ▪ **[2]** Layer 3 ISDN Trace ▪ **[3]** Only ISDN Q.931 Messages Trace ▪ **[4]** Layer 3 ISDN No Duplication Trace
Web/EMS: Framing Method CLI: framing **[FramingMethod]**	Determines the physical framing method for the trunk. ▪ **[0]** Extended Super Frame = (Default) Depends on protocol type: ✓ E1: E1 CRC4 MultiFrame Format extended G.706B (same as c) ✓ T1: T1 Extended Super Frame with CRC6 (same as D) ▪ **[1]** Super Frame = T1 SuperFrame Format (as B). ▪ **[a]** E1 FRAMING DDF = E1 DoubleFrame Format - CRC4 is forced to off ▪ **[b]** E1 FRAMING MFF CRC4 = E1 CRC4 MultiFrame Format - CRC4 is always on ▪ **[c]** E1 FRAMING MFF CRC4 EXT = E1 CRC4 MultiFrame Format extended G.706B - auto negotiation is on. If the negotiation fails, it changes automatically to CRC4 off (ddf) ▪ **[A]** T1 FRAMING F4 = T1 4-Frame multiframe. ▪ **[B]** T1 FRAMING F12 = T1 12-Frame multiframe (D4). ▪ **[C]** T1 FRAMING ESF = T1 Extended SuperFrame without CRC6 ▪ **[D]** T1 FRAMING ESF CRC6 = T1 Extended SuperFrame with CRC6 ▪ **[E]** T1 FRAMING F72 = T1 72-Frame multiframe (SLC96) ▪ **[F]** T1 FRAMING ESF CRC6 J2 = J1 Extended SuperFrame with CRC6 (Japan) **Note:** This parameter is not configurable for BRI interfaces; the device automatically uses the BRI framing method.
[FramingMethod_x]	Same as the description for parameter FramingMethod, but for a specific trunk ID (where *x* denotes the Trunk ID and 0 is the first Trunk).
Web/EMS: Clock Master CLI: clock-master **[ClockMaster]**	Determines the Tx clock source of the E1/T1 line. ▪ **[0]** Recovered = Generate the clock according to the Rx of the E1/T1 line (default). ▪ **[1]** Generated = Generate the clock according to the internal TDM bus. **Notes:** ▪ The source of the internal TDM bus clock is determined by the parameter TDMBusClockSource.

Parameter	Description
[ClockMaster_x]	Same as the description for parameter ClockMaster, but for a specific Trunk ID (where *x* denotes the Trunk ID and 0 is the first Trunk).
Web/EMS: Line Code CLI: line-code **[LineCode]**	Selects B8ZS or AMI for T1 spans, and HDB3 or AMI for E1 spans. ▪ **[0]** B8ZS = use B8ZS line code (for T1 trunks only) default. ▪ **[1]** AMI = use AMI line code. ▪ **[2]** HDB3 = use HDB3 line code (for E1 trunks only). **Note:** This parameter is not configurable for BRI interfaces; the device automatically uses the Modified Alternate Mark Invert (MAMI) line code.
[LineCode_x]	Same as the description for parameter LineCode, but for a specific trunk ID (where 0 denotes the first trunk).
[TrunkLifeLineType]	Determines the scenarios upon which the PSTN Fallback (lifeline) feature is activated. This feature redirects IP calls to the PSTN upon a power outage or lack of IP connectivity (i.e., no ping), thereby guaranteeing call continuity. PSTN Fallback is supported if the device houses one or two E1/T1 ("TRUNKS") modules, where each module provides two or four spans. In the event of a PSTN fallback, the module's metalic relay switch automatically connects trunk Port 1 (I) to Port 2 (II), and / or trunk Port 3 (III) to Port 4 (IIII), of the same module. Therefore, if for example, a PBX trunk is connected to Port 1 and the PSTN network is connected to Port 2, when PSTN fallback is activated, calls from the PBX are routed directly to the PSTN through Port 2. ▪ **[0]** = Activate PSTN Fallback upon power outage (default). ▪ **[2]** = Activate PSTN Fallaback on power outage or loss of ping (i.e., no IP connectivity). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ PSTN Fallback is supported only between ports on the same module. ▪ PSTN Fallback is supported only for ISDN when the number of supported channels (e.g., 30) is less than the maximum number of possible channels provided by the physical ports (e.g., two E1 trunks). When the number of supported channels (e.g., 60) equals the maximum number of channels provided by the physical ports (e.g., two E1 trunks), then other protocols such as CAS are also supported. ▪ The PSTN Fallback feature has no relation to the PSTN Fallback Software Upgrade Key.
[AdminState]	Defines the administrative state for all trunks. ▪ **[0]** = Lock the trunk; stops trunk traffic to configure the trunk protocol type. ▪ **[1]** = Shutting down (read only). ▪ **[2]** = Unlock the trunk (default); enables trunk traffic. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When the device is locked from the Web interface, this

Parameter	Description
	parameter changes to 0. ▪ To define the administrative state per trunk, use the TrunkAdministrativeState parameter.
[TrunkAdministrativeState_x]	Defines the administrative state per trunk, where *x* denotes the trunk number. ▪ **[0]** = Lock the trunk; stops trunk traffic to configure the trunk protocol type. ▪ **[1]** = shutting down (read only). ▪ **[2]** = Unlock the trunk (default); enables trunk traffic.
Web/EMS: Line Build Out Loss CLI: line-build-out-loss **[LineBuildOut.**Loss]	Defines the line build out loss for the selected T1 trunk. ▪ **[0]** 0 dB (default) ▪ **[1]** -7.5 dB ▪ **[2]** -15 dB ▪ **[3]** -22.5 dB **Note:** This parameter is applicable only to T1 trunks.
[TDMHairPinning]	Defines static TDM hair-pinning (cross-connection) performed at initialization. The connection is between trunks with an option to exclude a single B-Channel in each trunk. Format example: T0-T1/B3,T2-T3,T4-T5/B2. **Note:** For this parameter to take effect, a device reset is required.
Web: Enable TDM Tunneling EMS: TDM Over IP CLI: tdm-tunneling **[EnableTDMoverIP]**	Enables TDM tunneling. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = TDM Tunneling is enabled. When TDM Tunneling is enabled, the originating device automatically initiates SIP calls from all enabled B-channels pertaining to E1/T1/J1 spans that are configured with the 'Transparent' protocol. The called number of each call is the internal phone number of the B-channel from where the call originates. The 'The Inbound IP Routing Table is used to define the destination IP address of the terminating device. The terminating device automatically answers these calls if its E1/T1 protocol is set to 'Transparent' (ProtocolType = 5). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For an overview on TDM tunneling, see 'TDM Tunneling' on page 258.

A.12.6.2 TDM Bus and Clock Timing Parameters

The TDM Bus parameters are described in the table below.

Table A-53: TDM Bus and Clock Timing Parameters

Parameter	Description
TDM Bus Parameters	
Web/EMS: PCM Law Select **[PCMLawSelect]**	Determines the type of pulse-code modulation (PCM) companding algorithm law in input and output TDM bus. ▪ **[1]** Alaw = A-law

Parameter	Description
	▪ **[3]** MuLaw = Mu-Law The default value is automatically selected according to the Protocol Type of the selected trunk: E1 defaults to ALaw, T1 defaults to MuLaw. If the Protocol Type is set to NONE, the default is MuLaw. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Typically, A-Law is used for E1 spans and Mu-Law for T1/J1 spans.
Web/EMS: Idle PCM Pattern CLI: idle-pcm-pattern **[IdlePCMPattern]**	Defines the PCM Pattern that is applied to the E1/T1 timeslot (B-channel) when the channel is idle. The range is 0 to 255. The default is set internally according to the Law select 1 (0xFF for Mu-Law; 0x55 for A-law). **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Idle ABCD Pattern **[IdleABCDPattern]**	Defines the ABCD (CAS) Pattern that is applied to the CAS signaling bus when the channel is idle. The valid range is 0x0 to 0xF. The default is -1 (i.e., default pattern is 0000). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only when using PSTN interface with CAS protocols.
Web/EMS: TDM Bus Clock Source **[TDMBusClockSource]**	Determines the clock source to which the device synchronizes. ▪ **[1]** Internal = Generate clock from local source (default). ▪ **[4]** Network = Recover clock from PSTN line.
EMS/Web: TDM Bus Local Reference **[TDMBusLocalReference]**	Defines the physical Trunk ID from which the device recovers (receives) its clock synchronization. The range is 0 to the maximum number of Trunks. The default is 0. **Note:** This parameter is applicable only if the parameter TDMBusClockSource is set to 4 and the parameter TDMBusPSTNAutoClockEnable is set to 0.
Web/EMS: TDM Bus Enable Fallback **[TDMBusEnableFallback]**	Defines the automatic fallback of the clock. ▪ **[0]** Manual (default) ▪ **[1]** Auto Non-Revertive ▪ **[2]** Auto Revertive
Web: TDM Bus Fallback Clock Source EMS: TDM Bus Fallback Clock **[TDMBusFallbackClock]**	Determines the fallback clock source on which the device synchronizes in the event of a clock failure. ▪ **[4]** Network (default) ▪ **[8]** H.110_A ▪ **[9]** H.110_B ▪ **[10]** NetReference1 ▪ **[11]** NetReference2
Web/EMS: TDM Bus Net Reference Speed **[TDMBusNetrefSpeed]**	Defines the NetRef frequency (for both generation and synchronization). ▪ **[0]** 8 kHz (default) ▪ **[1]** 1.544 MHz ▪ **[2]** 2.048 MHz

Parameter	Description
Web: TDM Bus PSTN Auto FallBack Clock EMS: TDM Bus Auto Fall Back Enable **[TDMBusPSTNAutoClockEnable]**	Enables the PSTN trunk Auto-Fallback Clock feature. ▪ **[0]** Disable (default) = Recovers the clock from the E1/T1 line defined by the parameter TDMBusLocalReference. ▪ **[1]** Enable = Recovers the clock from any connected synchronized slave E1/T1 line. If this trunk loses its synchronization, the device attempts to recover the clock from the next trunk. Note that initially, the device attempts to recover the clock from the trunk defined by the parameter TDMBusLocalReference. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is relevant only if the parameter TDMBusClockSource is set to 4.
Web: TDM Bus PSTN Auto Clock Reverting EMS: TDM Bus Auto Fall Back Reverting Enable **[TDMBusPSTNAutoClockRevertingEnable]**	Enables the PSTN trunk Auto-Fallback Reverting feature. If enabled and a trunk returning to service has an AutoClockTrunkPriority parameter value that is higher than the priority of the local reference trunk (set in the TDMBusLocalReference parameter), the local reference reverts to the trunk with the higher priority that has returned to service for the device's clock source. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only when the TDMBusPSTNAutoClockEnable parameter is set to 1.
Web: Auto Clock Trunk Priority EMS: Auto Trunk Priority CLI: clock-priority **[AutoClockTrunkPriority]**	Defines the trunk priority for auto-clock fallback (per trunk parameter). ▪ 0 to 99 = priority, where 0 (default) is the highest. ▪ 100 = the SW never performs a fallback to that trunk (usually used to mark untrusted source of clock). **Note:** Fallback is enabled when the TDMBusPSTNAutoClockEnable parameter is set to 1.

A.12.6.3 CAS Parameters

The Common Channel Associated (CAS) parameters are described in the table below. Note that CAS is not applicable to BRI interfaces.

Table A-54: CAS Parameters

Parameter	Description
Web: CAS Transport Type EMS: CAS Relay Transport Mode CLI: CAS-transport-type **[CASTransportType]**	Determines the ABCD signaling transport type over IP. ▪ **[0]** CAS Events Only = Disable CAS relay (default). ▪ **[1]** CAS RFC2833 Relay = Enable CAS relay mode using RFC 2833. The CAS relay mode can be used with the TDM tunneling feature to enable tunneling over IP for both voice and CAS signaling bearers.

Parameter	Description
[CASAddressingDelimiters]	Enables the addition of delimiters to the received address or received ANI digits string. ▪ **[0]** = Disable (default). The address and ANI strings remain without delimiters. ▪ **[1]** = Enable. Delimiters such as '*', '#', and 'ST' are added to the received address or received ANI digits string.
CLI: cas-delimiters-types **[CASDelimitersPaddingUsage]**	Defines the digits string delimiter padding usage per trunk. ▪ **[0]** (default) = default address string padding: '*XXX#' (where XXX is the digit string that begins with '*' and ends with '#', when using padding). ▪ **[1]** = special use of asterisks delimiters: '*XXX*YYY*' (where XXX is the address, YYY is the source phone number, and '*' is the only delimiter padding). **Note:** For this parameter to take effect, a device reset is required.

Web: CAS Table per Trunk
EMS: Trunk CAS Table Index
CLI: cas-table-index
[CASTableIndex_x]
	Defines the CAS protocol per trunk (where *x* denotes the trunk ID) from a list of CAS protocols defined by the parameter CASFileName_x.
	For example, the below configuration specifies Trunks 0 and 1 to use the E&M Winkstart CAS (E_M_WinkTable.dat) protocol, and Trunks 2 and 3 to use the E&M Immediate Start CAS (E_M_ImmediateTable.dat) protocol:
	`CASFileName_0 = 'E_M_WinkTable.dat'`
	`CASFileName_1 = 'E_M_ImmediateTable.dat'`
	`CASTableIndex_0 = 0`
	`CASTableIndex_1 = 0`
	`CASTableIndex_2 = 1`
	`CASTableIndex_3 = 1`
	Note: You can define CAS tables per B-channel using the parameter CASChannelIndex.

Web: Dial Plan
EMS: Dial Plan Name
CLI: cas-dial-plan-name
[CASTrunkDialPlanName_x]
	Defines the CAS Dial Plan name that is used on a specific trunk (where *x* denotes the trunk ID).
	The range is up to 11 characters.
	For example, the below configures E1_MFCR2 trunk with a single protocol (Trunk 5):
	`ProtocolType_5 = 7`
	`CASFileName_0='R2_Korea_CP_ANI.dat'`
	`CASTableIndex_5 = 0`
	`DialPlanFileName = 'DialPlan_USA.dat'`
	`CASTrunkDialPlanName_5 = 'AT_T'`

[CASFileName_x]	Defines the CAS file name (e.g., 'E_M_WinkTable.dat') that defines the CAS protocol, where x denotes the CAS file ID (0-7). It is possible to define up to eight different CAS files by repeating this parameter. Each CAS file can be associated with one or more of the device's trunks, using the parameter CASTableIndex_x. **Note:** For this parameter to take effect, a device reset is required.
Web: CAS Table per Channel CLI: cas-channel-index **[CASChannelIndex]**	Defines the loaded CAS protocol table index per B-channel pertaining to a CAS trunk. This parameter is assigned a string value and can be set in one of the following two formats:

Parameter	Description

	▪ **CAS table per channel:** Each channel is separated by a comma and the value entered denotes the CAS table index used for that channel. The syntax is <CAS index>,<CAS index> (e.g., "1,2,1,2…"). For this format, 31 indices must be defined for E1 trunks (including dummy for B-channel 16), or 24 indices for T1 trunks. Below is an example for configuring a T1 CAS trunk (Trunk 5) with several CAS variants
	`ProtocolType_5 = 7`
	`CASFILENAME_0='E_M_FGBWinkTable.dat'`
	`CASFILENAME_1='E_M_FGDWinkTable.dat'`
	`CASFILENAME_2='E_M_WinkTable.txt'`
	`CasChannelIndex_5 =`
	`'0,0,0,1,1,1,2,2,2,0,0,0,1,1,1,0,1,2,0,2,1,2,2,2'`
	`CASDelimitersPaddingUsage_5 = 1`
	▪ **CAS table per channel group:** Each channel group is separated by a colon and each channel is separated by a comma. The syntax is <x-y channel range>:<CAS table index>, (e.g., "1-10:1,11-31:3"). Every B-channel (including 16 for E1) must belong to a channel group. Below is an example for configuring an E1 CAS trunk (Trunk 5) with several CAS variants:
	`ProtocolType_5 = 8`
	`CASFILENAME_2='E1_R2D'`
	`CASFILENAME_7= E_M_ImmediateTable_A-Bit.txt'`
	`CasChannelIndex_5 = '1-10:2,11-20:7,21-31:2'`
	Notes:
	▪ To configure this parameter, the trunk must first be stopped.
	▪ Only one of these formats can be implemented; not both.
	▪ When this parameter is not configured, a single CAS table for the entire trunk is used, configured by the parameter CASTableIndex.

[CASTablesNum]	Defines how many CAS protocol configurations files are loaded. The valid range is 1 to 8. **Note:** For this parameter to take effect, a device reset is required.
CAS State Machines Parameters **Note:** For configuring the CAS State Machine table using the Web interface, see 'Configuring CAS State Machines' on page 251. The CAS state machine can be configured only through the Web-based management tool.	
Web: Generate Digit On Time **[CASStateMachineGenerateDigitOnTime]**	Generates digit on-time (in msec). The value must be a positive value. The default value is -1.
Web: Generate Inter Digit Time **[CASStateMachineGenerateInterDigitTime]**	Generates digit off-time (in msec). The value must be a positive value. The default value is -1.
Web: DTMF Max Detection Time **[CASStateMachineDTMFMaxOnDetectionTime]**	Detects digit maximum on time (according to DSP detection information event) in msec units. The value must be a positive value. The default value is -1.
Web: DTMF Min Detection Time **[CASStateMachineDTMFMinOnDetectionTime]**	Detects digit minimum on time (according to DSP detection information event) in msec units. The digit time length must be longer than this value to receive a detection. Any number may

Parameter	Description
	be used, but the value must be less than CasStateMachineDTMFMaxOnDetectionTime. The value must be a positive value. The default value is -1.
Web: MAX Incoming Address Digits **[CASStateMachineMaxNumOfIncomingAddressDigits]**	Defines the limitation for the maximum address digits that need to be collected. After reaching this number of digits, the collection of address digits is stopped. The value must be an integer. The default value is -1.
Web: MAX Incoming ANI Digits **[CASStateMachineMaxNumOfIncomingANIDigits]**	Defines the limitation for the maximum ANI digits that need to be collected. After reaching this number of digits, the collection of ANI digits is stopped. The value must be an integer. The default value is -1.
Web: Collect ANI **[CASStateMachineCollectANI]**	In some cases, when the state machine handles the ANI collection (not related to MFCR2), you can enable the state machine to collect ANI or discard ANI. ▪ **[0]** No = Don't collect ANI. ▪ **[1]** Yes = Collect ANI. ▪ **[-1]** Default = Default value.
Web: Digit Signaling System **[CASStateMachineDigitSignalingSystem]**	Defines which Signaling System to use in both directions (detection\generation). ▪ **[0]** DTMF = Uses DTMF signaling. ▪ **[1]** MF = Uses MF signaling (default). ▪ **[-1]** Default = Default value.

A.12.6.4 ISDN Parameters

The ISDN parameters are described in the table below.

Table A-55: ISDN Parameters

Parameter	Description
Web: ISDN Termination Side EMS: Termination Side CLI: isdn-termination-side **[TerminationSide]**	Determines the ISDN termination side. ▪ **[0]** User side = ISDN User Termination Equipment (TE) side (default) ▪ **[1]** Network side = ISDN Network Termination (NT) side **Note:** Select 'User side' when the PSTN or PBX side is configured as 'Network side' and vice versa. If you don't know the device's ISDN termination side, choose 'User side'. If the D-channel alarm is indicated, choose 'Network Side'. The BRI module supports the ITU-T I.430 standard, which defines the ISDN-BRI layer 1 specification. The BRI and PRI ports are configured similarly, using this parameter. When an NT port is active, it drives a 38-V line and sends an INFO1 signal (as defined in ITU-T I.430 Table 4) on the data line to synchronize to a TE port that might be connected to it. To stop the voltage and the INFO1 signal on the line, stop the trunk using the Stop Trunk button.
[TerminationSide_x]	Same as the description for parameter TerminationSide, but for a specific trunk ID (where *x* denotes the Trunk ID and 0 is the first Trunk).

Parameter	Description
BRI Layer 2 Mode CLI: isdn-layer2-mode **[BriLayer2Mode]**	Determines whether Point-to-Point or Point-to-Multipoint mode for BRI ports. ▪ **[0]** Point to Point (default) ▪ **[1]** Point to Multipoint = Must be configured for Network side.
Web/EMS: B-channel Negotiation CLI: b-ch-negotiation **[BchannelNegotiation]**	Determines the ISDN B-Channel negotiation mode. ▪ **[0]** Preferred. ▪ **[1]** Exclusive (default). ▪ **[2]** Any. **Notes:** ▪ This parameter is applicable only to ISDN protocols. ▪ For some ISDN variants, when 'Any' (2) is selected, the Setup message excludes the Channel Identification IE. ▪ The Any' (2) option is applicable only if the following conditions are met: ✓ The parameter TerminationSide is set to 0 ('User side'). ✓ The PSTN protocol type (ProtocolType) is configured as Euro ISDN.
NFAS Parameters **Note:** These parameters are applicable to PRI interfaces.	
Web: NFAS Group Number EMS: Group Number CLI: isdn-nfas-group-number **[NFASGroupNumber_x]**	Defines the NFAS group number (NFAS member) for the selected trunk, where *x* denotes the Trunk ID. ▪ 0 = Non-NFAS trunk (default) ▪ 1 to 12 = NFAS group number Trunks that belong to the same NFAS group have the same number. With ISDN Non-Facility Associated Signaling you can use single D-channel to control multiple PRI interfaces. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to T1 ISDN protocols. ▪ For more information on NFAS, see 'ISDN Non-Facility Associated Signaling (NFAS)' on page 268.
Web/EMS: D-channel Configuration CLI: isdn-nfas-dchannel-type **[DChConfig_x]**	Defines primary, backup (optional), and B-channels only, per trunk (where x denotes the Trunk ID). ▪ **[0]** PRIMARY= Primary Trunk (default) - contains a D-channel that is used for signaling. ▪ **[1]** BACKUP = Backup Trunk - contains a backup D-channel that is used if the primary D-channel fails. ▪ **[2]** NFAS = NFAS Trunk - contains only 24 B-channels, without a signaling D-channel. **Note:** This parameter is applicable only to T1 ISDN protocols.
Web: NFAS Interface ID EMS: ISDN NFAS Interface ID CLI: isdn-nfas-interface-id **[ISDNNFASInterfaceID_x]**	Defines a different Interface ID for each T1 trunk (where x denotes the trunk ID). The valid range is 0 to 100. The default interface ID equals the trunk's ID. **Notes:** ▪ To set the NFAS interface ID, configure ISDNIBehavior_x to include '512' feature per T1 trunk. ▪ For more information on NFAS, see 'ISDN Non-Facility

Parameter	Description
	Associated Signaling (NFAS)' on page 268.
Web: Enable ignoring ISDN Disconnect with PI CLI: ign-isdn-disc-w-pi **[KeepISDNCallOnDisconnectWithPI]**	Allows the device to ignore ISDN Disconnect messages with PI 1 or 8. ▪ **[1]** = The call (in connected state) is not released if a Q.931 Disconnect with PI (PI = 1 or 8) message is received during the call. ▪ **[0]** = The call is disconnected (default).
Web: PI For Setup Message CLI: pi-4-setup-msg **[PIForSetupMsg]**	Determines whether and which Progress Indicator (PI) information element (IE) is added to the sent ISDN Setup message. Some ISDN protocols such as NI-2 or Euro ISDN can optionally contain PI = 1 or PI = 3 in the Setup message. ▪ **[0]** = PI is not added (default). ▪ **[1]** = PI 1 is added to a sent ISDN Setup message - call is not end-to-end ISDN. ▪ **[3]** = PI 3 is added to a sent ISDN Setup message - calling equipment is not ISDN.
ISDN Flexible Behavior Parameters ISDN protocol is implemented in different switches/PBXs by different vendors. Several implementations may vary slightly from the specification. Therefore, to provide a flexible interface that supports these ISDN variants, the ISDN behavior parameters can be used.	
Web/EMS: Incoming Calls Behavior CLI: isdn-bits-incoming-calls-behavior **[ISDNInCallsBehavior]**	Determines the bit-field used to determine several behavior options that influence how the ISDN Stack INCOMING calls behave. ▪ **[32]** DATA CONN RS = The device sends a Connect (answer) message on not incoming Tel calls. ▪ **[64]** VOICE CONN RS = The device sends a Connect (answer) message on incoming Tel calls. ▪ **[2048]** CHAN ID IN FIRST RS = The device sends Channel ID in the first response to an incoming Q.931 Call Setup message. Otherwise, the Channel ID is sent only if the device requires changing the proposed Channel ID (default). ▪ **[8192]** CHAN ID IN CALL PROC = The device sends Channel ID in a Q.931 Call Proceeding message. ▪ **[65536]** PROGR IND IN SETUP ACK = The device includes Progress Indicator (PI=8) in Setup ACK message if an empty called number is received in an incoming Setup message. This option is applicable to the overlap dialing mode. The device also plays a dial tone (for TimeForDialTone) until the next called number digits are received. ▪ **[262144]** = NI-2 second redirect number. You can select and use (in INVITE messages) the NI-2 second redirect number if two redirect numbers are received in Q.931 Setup for incoming Tel-to-IP calls. ▪ **[2147483648]** CC_USER_SCREEN_INDICATOR = When the device receives two Calling Number IE's in the Setup message, the device by default, uses only one of the numbers according to the following: ✓ Network provided, Network provided - the first calling number is used ✓ Network provided, User provided: the first one is used ✓ User provided, Network provided: the second one is used

Parameter	Description
	✓ User provided, user provided: the first one is used When this bit is configured, the device behaves as follows: ✓ Network provided, Network provided: the first calling number is used ✓ Network provided, User provided: the second one is used ✓ User provided, Network provided: the first one is used ✓ User provided, user provided: the first one is used **Note:** When using the *ini* file to configure the device to support several ISDNInCallsBehavior features, enter a summation of the individual feature values. For example, to support both **[2048]** and **[65536]** features, set ISDNInCallsBehavior = 67584 (i.e., 2048 + 65536).
[ISDNInCallsBehavior_x]	Same as the description for the parameter ISDNInCallsBehavior, but per trunk (i.e., where *x* denotes the Trunk ID).
Web/EMS: Q.931 Layer Response Behavior CLI: isdn-bits-ns-behavior **[ISDNIBehavior]**	Bit-field used to determine several behavior options that influence the behaviour of the Q.931 protocol. ▪ **[0]** = Disable (default) ▪ **[1]** NO STATUS ON UNKNOWN IE = Q.931 Status message isn't sent if Q.931 received message contains an unknown/unrecognized IE. By default, the Status message is sent. **Note:** This value is applicable only to ISDN variants in which sending of Status message is optional. ▪ **[2]** NO STATUS ON INV OP IE = Q.931 Status message isn't sent if an optional IE with invalid content is received. By default, the Status message is sent. **Note:** This option is applicable only to ISDN variants in which sending of Status message is optional. ▪ **[4]** ACCEPT UNKNOWN FAC IE = Accepts unknown/unrecognized Facility IE. Otherwise, the Q.931 message that contains the unknown Facility IE is rejected. **Note:** This option is applicable only to ISDN variants where a complete ASN1 decoding is performed on Facility IE. ▪ **[128]** SEND USER CONNECT ACK = The Connect ACK message is sent in response to received Q.931 Connect; otherwise, the Connect ACK is not sent (default). **Note:** This option is applicable only to Euro ISDN User side outgoing calls. ▪ **[512]** EXPLICIT INTERFACE ID = Enables to configure T1 NFAS Interface ID (refer to the parameter ISDNNFASInterfaceID_x). **Note:** This value is applicable only to 4/5ESS, DMS, NI-2 and HKT variants. ▪ **[2048]** ALWAYS EXPLICIT = Always set the Channel Identification IE to explicit Interface ID, even if the B-channel is on the same trunk as the D-channel. **Note:** This value is applicable only to 4/5ESS, DMS and NI-2 variants. ▪ **[32768]** ACCEPT MU LAW =Mu-Law is also accepted in ETSI. ▪ **[65536]** EXPLICIT PRES SCREENING = The calling party number (octet 3a) is always present even when presentation and screening are at their default. **Note:** This option is applicable only to ETSI, NI-2, and 5ESS.

Parameter	Description
	▪ **[131072]** STATUS INCOMPATIBLE STATE = Clears the call on receipt of Q.931 Status with incompatible state. Otherwise, no action is taken (default). ▪ **[262144]** STATUS ERROR CAUSE = Clear call on receipt of Status according to cause value. ▪ **[524288]** ACCEPT A LAW =A-Law is also accepted in 5ESS. ▪ **[2097152]** RESTART INDICATION = Upon receipt of a Restart message, acEV_PSTN_RESTART_CONFIRM is generated. ▪ **[4194304]** FORCED RESTART = On data link (re)initialization, send RESTART if there is no call. ▪ **[67108864]** NS ACCEPT ANY CAUSE = Accept any Q.850 Cause IE from ISDN. **Note:** This option is applicable only to Euro ISDN. ▪ **[134217728]** NS_BRI_DL_ALWAYS_UP (0x08000000) = By default, the BRI D-channel goes down if there are no active calls. If this option is configured, the BRI D-channel is always up and synchronized. ▪ **[536870912]** Alcatel coding for redirect number and display name is accepted by the device. **Note:** This option is applicable only to QSIG (and relevant for specific Alcatel PBXs such as OXE). ▪ **[1073741824]** QSI ENCODE INTEGER = If this bit is set, INTEGER ASN.1 type is used in operator coding (compliant to new ECMA standards); otherwise, OBJECT IDENTIFIER ASN.1 type is used. **Note:** This option is applicable only to QSIG. ▪ **[2147483648]** 5ESS National Mode For Bch Maintenance = Use the National mode of AT&T 5ESS for B-channel maintenance. **Notes:** ▪ To configure the device to support several ISDNIBehavior features, enter a summation of the individual feature values. For example, to support both **[512]** and **[2048]** features, set the parameter ISDNIBehavior is set to 2560 (i.e., 512 + 2048). ▪ When configuring in the Web interface, to select the options click the arrow button and then for each required option select 1 to enable.
[ISDNIBehavior_x]	Same as the description for parameter ISDNIBehavior, but for a specific trunk ID.
Web: General Call Control Behavior EMS: General CC Behavior CLI: isdn-bits-cc-behavior **[ISDNGeneralCCBehavior]**	Bit-field for determining several general CC behavior options. To select the options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[2]** = Data calls with interworking indication use 64 kbps B-channels (physical only). ▪ **[8]** REVERSE CHAN ALLOC ALGO = Channel ID allocation algorithm. ▪ **[16]** = The device clears down the call if it receives a NOTIFY message specifying 'User-Suspended'. A NOTIFY (User-Suspended) message is used by some networks (e.g., in Italy or Denmark) to indicate that the remote user has cleared the call, especially in the case of a long distance voice call.

Parameter	Description
	▪ **[32]** CHAN ID 16 ALLOWED = Applies only to ETSI E1 lines (30B+D). Enables handling the differences between the newer QSIG standard (ETS 300-172) and other ETSI-based standards (ETS 300-102 and ETS 300-403) in the conversion of B-channel ID values into timeslot values: ✓ In 'regular ETSI' standards, the timeslot is identical to the B-channel ID value, and the range for both is 1 to 15 and 17 to 31. The D-channel is identified as channel-id #16 and carried into the timeslot #16. ✓ In newer QSIG standards, the channel-id range is 1 to 30, but the timeslot range is still 1 to 15 and 17 to 31. The D-channel is not identified as channel-id #16, but is still carried into the timeslot #16. When this bit is set, the channel ID #16 is considered as a valid B-channel ID, but timeslot values are converted to reflect the range 1 to 15 and 17 to 31. This is the new QSIG mode of operation. When this bit is not set (default), the channel_id #16 is not allowed, as for all ETSI-like standards. ▪ **[64]** USE T1 PRI = PRI interface type is forced to T1. ▪ **[128]** USE E1 PRI = PRI interface type is forced to E1. ▪ **[256]** START WITH B CHAN OOS = B-channels start in the Out-Of-Service state (OOS). ▪ **[512]** CHAN ALLOC LOWEST = CC allocates B-channels starting from the lowest available B-channel id. ▪ **[1024]** CHAN ALLOC HIGHEST = CC allocates B-channels starting from the highest available B-channel id. ▪ **[16384]** CC_TRANSPARENT_UUI bit: The UUI-protocol implementation of CC is disabled allowing the application to freely send UUI elements in any primitive, regardless of the UUI-protocol requirements (UUI Implicit Service 1). This allows more flexible application control on the UUI. When this bit is not set (default behavior), CC implements the UUI-protocol as specified in the ETS 300-403 standards for Implicit Service 1. ▪ **[65536]** GTD5 TBCT = CC implements the VERIZON-GTD-5 Switch variant of the TBCT Supplementary Service, as specified in FSD 01-02-40AG Feature Specification Document from Verizon. Otherwise, TBCT is implemented as specified in GR-2865-CORE specification (default behavior). **Note:** When using the *ini* file to configure the device to support several ISDNGeneralCCBehavior features, add the individual feature values. For example, to support both **[16]** and **[32]** features, set ISDNGeneralCCBehavior = 48 (i.e., 16 + 32).
Web/EMS: Outgoing Calls Behavior CLI: isdn-bits-outgoing-calls-behavior **[ISDNOutCallsBehavior]**	Determines several behaviour options (bit fields) that influence the behaviour of the ISDN Stack outgoing calls. To select options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[2]** USER SENDING COMPLETE =The device doesn't automatically generate the Sending-Complete IE in the Setup message. If this bit is not set, the device generates it automatically in the Setup message only. ▪ **[16]** USE MU LAW = The device sends G.711-m-Law in outgoing voice calls. When disabled, the device sends G.711-A-Law in outgoing voice calls.

Parameter	Description
	Note: This option is applicable only to the Korean variant. ▪ **[128]** DIAL WITH KEYPAD = The device uses the Keypad IE to store the called number digits instead of the CALLED_NB IE. **Note:** This option is applicable only to the Korean variant (Korean network). This is useful for Korean switches that don't accept the CALLED_NB IE. ▪ **[256]** STORE CHAN ID IN SETUP = The device forces the sending of a Channel-Id IE in an outgoing Setup message even if it's not required by the standard (i.e., optional) and no Channel-Id has been specified in the establishment request. This is useful for improving required compatibility with switches. On BRI lines, the Channel-Id IE indicates 'any channel'. On PRI lines, it indicates an unused channel ID, preferred only. ▪ **[572]** USE A LAW = The device sends G.711 A-Law in outgoing voice calls. When disabled, the device sends the default G.711-Law in outgoing voice calls. **Note:** This option is applicable only to the E10 variant. ▪ **[1024]** = Numbering plan/type for T1 IP-to-Tel calling numbers are defined according to the manipulation tables or according to the RPID header (default). Otherwise, the plan/type for T1 calls are set according to the length of the calling number. ▪ **[2048]** = The device accepts any IA5 character in the called_nb and calling_nb strings and sends any IA5 character in the called_nb, and is not restricted to extended digits only (i.e., 0-9,*,#). ▪ **[16384]** DLCI REVERSED OPTION = Behavior bit used in the IUA interface groups to indicate that the reversed format of the DLCI field must be used. **Note:** When using the *ini* file to configure the device to support several ISDNOutCallsBehavior features, add the individual feature values. For example, to support both **[2]** and **[16]** features, set ISDNOutCallsBehavior = 18 (i.e., 2 + 16).
[ISDNOutCallsBehavior_x]	Same as the description for parameter ISDNOutCallsBehavior, but for a specific trunk ID.
Web: ISDN NS Behaviour 2 CLI: isdn-bits-ns-extension-behavior **[ISDNNSBehaviour2]**	Bit-field to determine several behavior options that influence the behavior of the Q.931 protocol. ▪ **[8]** NS_BEHAVIOUR2_ANY_UUI: any User to User Information Element (UUIE) is accepted for any protocol discriminator. This is useful for interoperability with non-standard switches.

Parameter	Description
[PSTNExtendedParams]	Determines the bit map for special PSTN behavior parameters: ▪ **[0]** (default) = Applicable to NI-2 ISDN and QSIG "Networking Extensions". This bit (i.e., bit #0) is responsible for the Invoke ID size: ✓ If this bit is not set (default), then the Invoke ID size is always one byte, with a value of 01 to 7f. ✓ If this bit is set, then the Invoke ID size is one or two bytes according to the Invoke ID value. ▪ **[2]** = Applicable to ROSE format (according to the old QSIG specifications). This bit (bit #1) is responsible for the QSIG octet 3. According to the ECMA-165 new version, octet 3 in all QSIG supplementary services Facility messages should be 0x9F = Networking Extensions. However, according to the old version, the value should be 0x91 = ROSE: ✓ If this bit is not set (default): 0x9F = Networking Extensions. ✓ If this bit is set: 0x91 = ROSE. ▪ **[3]** = Use options [0] and [2]. **Note:** For this parameter to take effect, a device reset is required.

A.12.7 ISDN and CAS Interworking Parameters

The ISDN and CAS interworking parameters are described in the table below.

Table A-56: ISDN and CAS Interworking Parameters

Parameter	Description
ISDN Parameters	
Web: Send Local Time To ISDN Connect **[SendLocalTimeToISDNConnect]**	Enables the device to send the date and time in the ISDN Connect message (Date / Time Information Element) if the received SIP 200 OK message is received without the SIP Date header. The device obtains the date and time from its internal clock. This feature is applicable only to Tel-to-IP calls. ▪ **[0]** Disable (default) = If the SIP 200 OK contains the Date header, the device sends its value in the ISDN Connect Date / Time IE. If the 200 OK does not include this header, it does not add the Date / Time IE to the sent ISDN Connect message. ▪ **[1]** Enable = If the SIP 200 OK contains the Date header, the device sends its value (i.e. date and time) in the ISDN Connect Date / Time IE. If the 200 OK does not include this header, the device uses its internal, local date and time for the Date / Time IE, which it adds to the sent ISDN Connect message. **Note:** For IP-to-Tel calls, this parameter is not applicable. Only if the incoming ISDN Connect message contains the Date / Time IE does the device add the Date header to the sent SIP 200 OK message.
Web/EMS: Min Routing Overlap Digits CLI: min-dg-b4-routing	Defines the minimum number of overlap digits to collect (for ISDN overlap dialing) before sending the first SIP message for routing Tel-to-IP calls.

Parameter	Description
[MinOverlapDigitsForRouting]	The valid value range is 0 to 49. The default is 1. **Note:** This parameter is applicable when the ISDNRxOverlap parameter is set to **[2].**
Web/EMS: ISDN Overlap IP to Tel Dialing CLI: isdn-tx-overlap **[ISDNTxOverlap]**	Enables ISDN overlap dialing for IP-to-Tel calls. This feature is part of ISDN-to-SIP overlap dialing according to RFC 3578. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, for each received INVITE of the same dialog session, the device sends an ISDN Setup (and subsequent ISDN Info Q.931 messages) with the collected digits to the Tel side. For all subsequent INVITEs received, the device sends a SIP 484 Address Incomplete response in order to maintain the current dialog session and receive additional digits from subsequent INVITEs. **Note:** When IP-to-Tel overlap dialing is enabled, to send ISDN Setup messages without the Sending Complete IE, the ISDNOutCallsBehavior parameter must be set to USER SENDING COMPLETE (2).
Web: Enable Receiving of Overlap Dialing CLI: ovrlp-rcving-type **[ISDNRxOverlap_x]**	Determines the receiving (Rx) type of ISDN overlap dialing for Tel-to-IP calls. ▪ **[0]** None (default) = Disabled. ▪ **[1]** Local receiving = ISDN Overlap Dialing - the complete number is sent in the INVITE Request-URI user part. The device receives ISDN called number that is sent in the 'Overlap' mode. The ISDN Setup message is sent to IP only after the number (including the Sending Complete IE) is fully received (via Setup and/or subsequent Info Q.931 messages). In other words, the device waits until it has received all the ISDN signaling messages containing parts of the called number, and only then it sends a SIP INVITE with the entire called number in the Request-URI. ▪ **[2]** Through SIP = Interworking of ISDN Overlap Dialing to SIP, based on RFC 3578. The device interworks ISDN to SIP by sending digits each time they are received (from Setup and subsequent Info Q.931 messages) to the IP, using subsequent SIP INVITE messages. **Notes:** ▪ When option **[2]** is configured, you can define the minimum number of overlap digits to collect before sending the first SIP message for routing the call, using the MinOverlapDigitsForRouting parameter. ▪ When option **[2]** is configured, even if SIP 4xx responses are received during this ISDN overlap receiving, the device does not release the call. ▪ The MaxDigits parameter can be used to limit the length of the collected number for ISDN overlap dialing (if Sending Complete is not received). ▪ If a digit map pattern is defined (using the DigitMapping or DialPlanIndex parameters), the device collects digits until a match is found (e.g., for closed numbering schemes) or until a timer expires (e.g., for open numbering schemes). If a match is found (or the timer expires), the digit collection

Parameter	Description
	process is terminated even if Sending Complete is not received. ▪ For enabling ISDN overlap dialing for IP-to-Tel calls, use the ISDNTxOverlap parameter. ▪ For more information on ISDN overlap dialing, see 'ISDN Overlap Dialing' on page 266.
CLI: ovrlp-rcving-type **[ISDNRxOverlap]**	Same as the description for parameter ISDNRxOverlap_x, but for all trunks.
Web/EMS: Mute DTMF In Overlap **[MuteDTMFInOverlap]**	Enables the muting of in-band DTMF detection until the device receives the complete destination number from the ISDN (for Tel-to-IP calls). In other words, the device does not accept DTMF digits received in the voice stream from the PSTN, but only accepts digits from ISDN Info messages. ▪ **[0]** Don't Mute (default) ▪ **[1]** Mute DTMF in Overlap Dialing = The device ignores in-band DTMF digits received during ISDN overlap dialing (disables the DTMF in-band detector). **Note:** This parameter is applicable to ISDN Overlap mode only when dialed numbers are sent using Q.931 Information messages.
[ConnectedNumberType]	Defines the Numbering Type of the ISDN Q.931 Connected Number IE that the device sends in the Connect message to the ISDN (for Tel-to-IP calls). This is interworked from the P-Asserted-Identity header in SIP 200 OK. The default is **[0]** (i.e., unknown).
[ConnectedNumberPlan]	Defines the Numbering Plan of the ISDN Q.931 Connected Number IE that the device sends in the Connect message to the ISDN (for Tel-to-IP calls). This is interworked from the P-Asserted-Identity header in SIP 200 OK. The default is **[0]** (i.e., unknown).
Web/EMS: Enable ISDN Tunneling Tel to IP CLI: isdn-tnl-tel2ip **[EnableISDNTunnelingTel2IP]**	Enables ISDN Tunneling. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Using Header = Enable ISDN Tunneling from ISDN PRI to SIP using a proprietary SIP header. ▪ **[2]** Using Body = Enable ISDN Tunneling from ISDN PRI to SIP using a dedicated message body. When ISDN Tunneling is enabled, the device sends all ISDN PRI messages using the correlated SIP messages. The ISDN Setup message is tunneled using SIP INVITE, all mid-call messages are tunneled using SIP INFO, and ISDN Disconnect/Release message is tunneled using SIP BYE messages. The raw data from the ISDN is inserted into a proprietary SIP header (X-ISDNTunnelingInfo) or a dedicated message body (application/isdn) in the SIP messages. **Notes:** ▪ For this feature to function, you must set the parameter ISDNDuplicateQ931BuffMode to 128 (i.e., duplicate all messages). ▪ ISDN tunneling is applicable for all ISDN variants as well as QSIG.
Web/EMS: Enable ISDN Tunneling	Enables ISDN Tunneling to the Tel side.

Parameter	Description
IP to Tel CLI: isdn-tnl-ip2tel **[EnableISDNTunnelingIP2Tel]**	▪ **[0]** Disable (default) ▪ **[1]** Enable ISDN Tunneling from IP to ISDN When ISDN Tunneling is enabled, the device extracts raw data received in the proprietary SIP header, x-isdntunnelinginfo, or a dedicated message body (application/isdn) in the SIP message and then sends the data in an ISDN message to the PSTN. If the raw data in this SIP header is suffixed with the string "ADDE", then the raw data is extracted and added as Informational Elements (IE) in the outgoing Q.931 message. The tunneling of the x-isdntunnelinginfo SIP header with IEs is converted from INVITE, 180, and 200 OK SIP messages to Q.931 SETUP, ALERT, and CONNECT respectively. For example, if the following SIP header is received, `x-isdntunnelinginfo: ADDE1C269FAA 06 800100820100A10F020136 0201F0A00702010102021F69` then it is added as an IE to the outgoing Q.931 message as 1C269FAA 06 800100820100A10F020136 0201F0A00702010102021F69, where, for example, "1C269F" is a 26 byte length Facility IE. **Note:** This feature is similar to that of the AddIEinSetup parameter. If both parameters are configured, the x-isdntunneling parameter takes precedence.
Web/EMS: Enable QSIG Tunneling CLI: qsig-tunneling **[EnableQSIGTunneling]**	Enables QSIG tunneling-over-SIP for all calls. This is according to IETF Internet-Draft draft-elwell-sipping-qsig-tunnel-03 and ECMA-355 and ETSI TS 102 345. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable QSIG tunneling from QSIG to SIP and vice versa. All QSIG messages are sent as raw data in corresponding SIP messages using a dedicated message body. **Notes:** ▪ You can enable QSIG tunneling per specific calls by enabling QSIG tunneling for an IP Profile. ▪ QSIG tunneling must be enabled on originating and terminating devices. ▪ To enable this function, set the ISDNDuplicateQ931BuffMode parameter to 128 (i.e., duplicate all messages). ▪ To define the format of encapsulated QSIG messages, use the QSIGTunnelingMode parameter. ▪ Tunneling according to ECMA-355 is applicable to all ISDN variants (in addition to the QSIG protocol). ▪ For more information on QSIG tunneling, see 'QSIG Tunneling' on page 261.
[QSIGTunnelingMode]	Defines the format of encapsulated QSIG message data in the SIP message MIME body. ▪ **[0]** = ASCII presentation of Q.931 QSIG message (default). ▪ **[1]** = Binary encoding of Q.931 QSIG message (according to ECMA-355, RFC 3204, and RFC 2025). **Note:** This parameter is applicable only if the QSIG Tunneling

Parameter	Description
	feature is enabled (using the EnableQSIGTunneling parameter).
Web: Enable Hold to ISDN EMS: Enable Hold 2 ISDN CLI: hold-to-isdn **[EnableHold2ISDN]**	Enables SIP-to-ISDN interworking of the Hold/Retrieve supplementary service. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ This parameter is applicable to Euro ISDN variants - from TE (user) to NT (network). ▪ This parameter is applicable also to QSIG BRI. ▪ If the parameter is disabled, the device plays a Held tone to the Tel side when a SIP request with 0.0.0.0 or "inactive" in SDP is received. An appropriate CPT file with the Held tone should be used.
EMS: Duplicate Q931 Buff Mode **[ISDNDuplicateQ931BuffMode]**	Determines the activation/deactivation of delivering raw Q.931 messages. ▪ **[0]** = ISDN messages aren't duplicated (default). ▪ **[128]** = All ISDN messages are duplicated. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: ISDN SubAddress Format CLI: isdn-subaddr-frmt **[ISDNSubAddressFormat]**	Determines the encoding format of the SIP Tel URI parameter 'isub', which carries the encoding type of ISDN subaddresses. This is used to identify different remote ISDN entities under the same phone number (ISDN Calling and Called numbers) for interworking between ISDN and SIP networks. ▪ **[0]** = ASCII - IA5 format that allows up to 20 digits. Indicates that the 'isub' parameter value needs to be encoded using ASCII characters (default) ▪ **[1]** = BCD (Binary Coded Decimal) - allows up to 40 characters (digits and letters). Indicates that the 'isub' parameter value needs to be encoded using BCD when translated to an ISDN message. ▪ **[2]** = User Specified For IP-to-Tel calls, if the incoming SIP INVITE message includes subaddress values in the 'isub' parameter for the Called Number (in the Request-URI) and/or the Calling Number (in the From header), these values are mapped to the outgoing ISDN Setup message. If the incoming ISDN Setup message includes 'subaddress' values for the Called Number and/or the Calling Number, these values are mapped to the outgoing SIP INVITE message's 'isub' parameter in accordance with RFC 4715.
[IgnoreISDNSubaddress]	Determines whether the device ignores the Subaddress from the incoming ISDN Called and Calling numbers when sending to IP. ▪ **[0]** = If an incoming ISDN Q.931 Setup message contains a Called/Calling Number Subaddress, the Subaddress is interworked to the SIP 'isub' parameter according to RFC (default). ▪ **[1]** = The device removes the ISDN Subaddress and does not include the 'isub' parameter in the Request-URI and

Parameter	Description
	does not process INVITEs with this parameter.
[ISUBNumberOfDigits]	Defines the number of digits (from the end) that the device takes from the called number (received from the IP) for the isub number (in the sent ISDN Setup message). This feature is only applicable for IP-to-ISDN calls. The valid value range is 0 to 36. The default value is 0. This feature operates as follows: 1 If an isub parameter is received in the Request-URI, for example, INVITE sip:9565645;**isub**=1234@host.domain:user=phone SIP/2.0 then the isub value is sent in the ISDN Setup message as the destination subaddress. 2 If the isub parameter is not received in the user part of the Request-URI, the device searches for it in the URI parameters of the To header, for example, To: "Alex" <sip: 9565645@host.domain;**isub**=1234> If present, the isub value is sent in the ISDN Setup message as the destination subaddress. 3 If the isub parameter is not present in the Request-URI header nor To header, the device does the following: ✓ If the called number (that appears in the user part of the Request-URI) starts with zero (0), for example, INVITE sip:**0**5694564@host.domain:user=phone SIP/2.0 then the device maps this called number to the destination number of the ISDN Setup message, and the destination subaddress in this ISDN Setup message remains empty. ✓ If the called number (that appears in the user part of the Request-URI) does not start with zero, for example, INVITE sip:5694564@host.domain:user=phone SIP/2.0 then the device maps this called number to the destination number of the ISDN Setup message, and the destination subaddress in this ISDN Setup message then contains *y* digits from the end of the called number. The *y* number of digits can be configured using the ISUBNumberOfDigits parameter. The default value of ISUBNumberOfDigits is 0, thus, if this parameter is not configured, and 1) and 2) scenarios (described above) have not provided an isub value, the subaddress remains empty.
Web: Digital Out-Of-Service Behavior EMS: Digital OOS Behavior For Trunk Value CLI: dig-oos-behavior **[DigitalOOSBehaviorFor Trunk_ID]**	Determines the method for setting digital trunks to Out-Of-Service state per trunk. ▪ **[-1]** Not Configured = Use the settings of the DigitalOOSBehavior parameter for per device (default). ▪ **[0]** Default = Uses default behavior for each trunk (see note below). ▪ **[1]** Service = Sends ISDN In or Out of Service (only for ISDN protocols that support Service message). ▪ **[2]** D-Channel = Takes D-Channel down or up (ISDN only). ▪ **[3]** Alarm = Sends or clears PSTN AIS Alarm (ISDN and CAS).

Parameter	Description
	▪ **[4]** Block = Blocks trunk (CAS only). **Notes:** ▪ This parameter is applicable only if the parameter EnableBusyOut is set to 1. ▪ The default behavior (value 0) is as follows: ✓ **ISDN:** Use Service messages on supporting variants and use Alarm on non-supporting variants. ✓ **CAS:** Use Alarm. ▪ When updating this parameter value at run-time, you must stop the trunk and then restart it for the update to take effect. ▪ To determine the method for setting Out-Of-Service state for all trunks (i.e., per device), use the DigitalOOSBehavior parameter. ▪ The *ID* in the *ini* file parameter name represents the trunk number, where 0 is the first trunk.
Web: Digital Out-Of-Service Behavior CLI: dig-oos-behavior **[DigitalOOSBehavior]**	Determines the method for setting digital trunks to Out-Of-Service state per device. For a description, see the DigitalOOSBehaviorFor Trunk_ID parameter. **Note:** To configure the method for setting Out-Of-Service state per trunk, use the DigitalOOSBehaviorForTrunk_ID parameter.
Web: Default Cause Mapping From ISDN to SIP CLI: dflt-cse-map-isdn2sip **[DefaultCauseMapISDN2IP]**	Defines a single default ISDN release cause that is used (in ISDN-to-IP calls) instead of all received release causes, except when the following Q.931 cause values are received: Normal Call Clearing (16), User Busy (17), No User Responding (18), or No Answer from User (19). The range is any valid Q.931 release cause (0 to 127). The default value is 0 (i.e., not configured - static mapping is used).
Release Cause Mapping from ISDN to SIP Table	
Web: Release Cause Mapping Table EMS: ISDN to SIP Cause Mapping CLI: configure voip > gw manipulations CauseMapIsdn2Sip **[CauseMapISDN2SIP]**	This *para*meter table maps ISDN Q.850 Release Causes to SIP responses. The format of this parameter is as follows: [CauseMapISDN2SIP] FORMAT CauseMapISDN2SIP_Index = CauseMapISDN2SIP_IsdnReleaseCause, CauseMapISDN2SIP_SipResponse; [\CauseMapISDN2SIP] Where, ▪ IsdnReleaseCause = Q.850 Release Cause ▪ SipResponse = SIP Response For example: CauseMapISDN2SIP 0 = 50,480; CauseMapISDN2SIP 0 = 6,406; When a Release Cause is received (from the PSTN side), the device searches this mapping table for a match. If the Q.850 Release Cause is found, the SIP response assigned to it is sent to the IP side. If no match is found, the default static mapping is used. **Notes:** ▪ This parameter can appear up to 12 times. ▪ For configuring *ini* file table parameters, see 'Configuring ini

Parameter	Description
	File Table Parameters' on page 98.
Release Cause Mapping from SIP to ISDN Table	
Web: Release Cause Mapping Table EMS: SIP to ISDN Cause Mapping CLI: configure voip > gw manipulations CauseMapSip2Isdn **[CauseMapSIP2ISDN]**	This *para*meter table maps SIP responses to Q.850 Release Causes. The format of this parameter is as follows: [CauseMapSIP2ISDN] FORMAT CauseMapSIP2ISDN_Index = CauseMapSIP2ISDN_SipResponse, CauseMapSIP2ISDN_IsdnReleaseCause; [\CauseMapSIP2ISDN] Where, ▪ SipResponse = SIP Response ▪ IsdnReleaseCause = Q.850 Release Cause For example: CauseMapSIP2ISDN 0 = 480,50; CauseMapSIP2ISDN 0 = 404,3; When a SIP response is received (from the IP side), the device searches this mapping table for a match. If the SIP response is found, the Q.850 Release Cause assigned to it is sent to the PSTN. If no match is found, the default static mapping is used. **Notes:** ▪ This parameter can appear up to 12 times. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Web/EMS: Enable Calling Party Category CLI: ni2-cpc **[EnableCallingPartyCategory]**	Determines whether Calling Party Category (CPC) is mapped between SIP and PRI. ▪ **[0]** Disable = Don't relay the CPC between SIP and PRI (default). ▪ **[1]** Enable = The CPC is relayed between SIP and PRI. If enabled, the CPC received in the Originating Line Information (OLI) IE of an incoming ISDN Setup message is relayed to the From/P-Asserted-Identity headers using the 'cpc' parameter in the outgoing INVITE message, and vice versa. For example (calling party is a payphone): From:<sip:2000;cpc=payphone@10.8.23.70>;tag=1c1806157451 **Note:** This feature is applicable only to the NI-2 PRI variant.
CLI: usr2usr-hdr-frmt **[UserToUserHeaderFormat]**	Determines the format of the User-to-User SIP header in the INVITE message for interworking the ISDN User to User (UU) IE data to SIP. ▪ **[0]** = Format: X-UserToUser (default). ▪ **[1]** = Format: User-to-User with Protocol Discriminator (pd) attribute. User-to-User=30303734353137343136353533b313233343b3834;pd=4. (This format is according to IETF Internet-Draft draft-johnston-sipping-cc-uui-04.) ▪ **[2]** = Format: User-to-User with encoding=hex at the end and pd embedded as the first byte. User-to-User=04303037343531373431363535 3b313233343b3834;encoding=hex. Where "04" at the beginning of this message

Parameter	Description
	is the pd. (This format is according to IETF Internet-Draft draft-johnston-sipping-cc-uui-03.)
Web/EMS: Remove CLI when Restricted CLI: rmv-cli-when-restr **[RemoveCLIWhenRestricted]**	Determines (for IP-to-Tel calls) whether the Calling Number and Calling Name IEs are removed from the ISDN Setup message if the presentation is set to Restricted. ▪ **[0]** No = IE's are not removed (default). ▪ **[1]** Yes = IE's are removed.
Web/EMS: Remove Calling Name CLI: rmv-calling-name **[RemoveCallingName]**	Enables the device to remove the Calling Name from SIP-to-ISDN calls for all trunks. ▪ **[0]** Disable = Does not remove Calling Name (default). ▪ **[1]** Enable = Removes Calling Name.
Web: Remove Calling Name EMS: Remove Calling Name For Trunk Mode **[RemoveCallingNameForTrunk_x]**	Enables the device to remove the Calling Name per trunk (where *x* denotes the trunk number) for SIP-to-ISDN calls. ▪ **[-1]** Use Global Parameter = Settings of the global parameter RemoveCallingName are used (default). ▪ **[0]** Disable = Does not remove Calling Name. ▪ **[1]** Enable = Remove Calling Name.
Web/EMS: Progress Indicator to ISDN CLI: pi-to-isdn **[ProgressIndicator2ISDN_ID]**	Determines the Progress Indicator (PI) to ISDN. The *ID* in the *ini* file parameter denotes the trunk number, where 0 is the first trunk. ▪ **[-1]** Not Configured = The PI in ISDN messages is set according to the parameter PlayRBTone2Tel (default). ▪ **[0]** No PI = PI is not sent to ISDN. ▪ **[1]** PI = 1; **[8]** PI = 8: The PI value is sent to PSTN in Q.931/Proceeding and Alerting messages. Typically, the PSTN/PBX cuts through the audio channel without playing local Ringback tone, enabling the originating party to hear remote Call Progress Tones or network announcements.
Web: Set PI in Rx Disconnect Message EMS: Set PI For Disconnect Msg CLI: pi-in-rx-disc-msg **[PIForDisconnectMsg_ID]**	Defines the device's behavior when a Disconnect message is received from the ISDN before a Connect message is received. The *ID* in the *ini* file parameter denotes the trunk number, where 0 is the first trunk. ▪ **[-1]** Not Configured = Sends a 183 SIP response according to the received progress indicator (PI) in the ISDN Disconnect message. If PI = 1 or 8, the device sends a 183 response, enabling the PSTN to play a voice announcement to the IP side. If there isn't a PI in the Disconnect message, the call is released (default). ▪ **[0]** No PI = Doesn't send a 183 response to IP. The call is released. ▪ **[1]** PI = 1; **[8]** PI = 8: Sends a 183 response to IP.
EMS: Connect On Progress Ind **[ConnectOnProgressInd]**	Enables the play of announcements from IP to PSTN without the need to answer the Tel-to-IP call. It can be used with PSTN networks that don't support the opening of a TDM channel before an ISDN Connect message is received. ▪ **[0]** = Connect message isn't sent after SIP 183 Session Progress message is received (default). ▪ **[1]** = Connect message is sent after SIP 183 Session Progress message is received.
Web: Local ISDN Ringback Tone Source	Determines whether the Ringback tone is played to the ISDN

Parameter	Description
EMS: Local ISDN RB Source CLI: local-isdn-rbt-src **[LocalISDNRBSource_ID]**	by the PBX/PSTN or by the device. ▪ **[0]** PBX = PBX/PSTN (default). ▪ **[1]** Gateway = device plays the Ringback tone. This parameter is applicable to ISDN protocols. It is used simultaneously with the parameter PlayRBTone2Trunk. The *ID* in the *ini* file parameter denotes the trunk number, where 0 is the first trunk.
Web/EMS: PSTN Alert Timeout CLI: pstn-alrt-timeout **[TrunkPSTNAlertTimeout_ID]**	Defines the Alert Timeout (ISDN T301 timer) in seconds for outgoing calls to PSTN. This timer is used between the time that an ISDN Setup message is sent to the Tel side (IP-to-Tel call establishment) and a Connect message is received. If Alerting is received, the timer is restarted. In the *ini* file parameter, *ID* denotes the trunk number, where 0 is the first trunk. The range is 1 to 600. The default is 180.
Web: B-Channel Negotiation EMS: B-Channel Negotiation For Trunk Mode **[BChannelNegotiationForTrunk_x]**	Determines the ISDN B-channel negotiation mode. ▪ **[-1]** Not Configured = use per device configuration of the BChannelNegotiation parameter (default). ▪ **[0]** Preferred = Preferred. ▪ **[1]** Exclusive = Exclusive. ▪ **[2]** Any = Any. **Notes:** ▪ This parameter is applicable to ISDN protocols. ▪ The option 'Any' is only applicable if TerminationSide is set to 0 (i.e., User side). ▪ The *x* represents the trunk number, where 0 is the first trunk.
CLI: snd-isdn-ser-aftr-restart **[SendISDNServiceAfterRestart]**	Enables the device to send an ISDN SERVice message per trunk upon device reset. The messsage (transmitted on the trunk's D-channel) indicates the availability of the trunk's B-channels (i.e., trunk in service). ▪ **[0]** = Disable (default). ▪ **[0]** = Enable.
EMS: Support Redirect InFacility **[SupportRedirectInFacility]**	Determines whether the Redirect Number is retrieved from the Facility IE. ▪ **[0]** = Not supported (default). ▪ **[1]** = Supports partial retrieval of Redirect Number (number only) from the Facility IE in ISDN Setup messages. This is applicable to Redirect Number according to ECMA-173 Call Diversion Supplementary Services. **Note:** To enable this feature, the parameter ISDNDuplicateQ931BuffMode must be set to 1.
CLI: call-re-rte-mode **[CallReroutingMode]**	Determines whether ISDN call rerouting (call forward) is performed by the PSTN instead of by the SIP side. This call forwarding is based on Call Deflection for Euro ISDN (ETS-300-207-1) and QSIG (ETSI TS 102 393). ▪ **[0]** Disable (default) ▪ **[1]** Enable = Enables ISDN call rerouting. When the device sends the INVITE message to the remote SIP entity and receives a SIP 302 response with a Contact header containing a URI host name that is the same as the device's

Parameter	Description
	IP address, the device sends a Facility message with a Call Rerouting invoke method to the ISDN and waits for the PSTN side to disconnect the call. **Note:** When this parameter is enabled, ensure that you configure in the Inbound IP Routing Table (PSTNPrefix *ini* file parameter) a rule to route the redirected call (using the user part from the 302 Contact header) to the same Trunk Group from where the incoming Tel-to-IP call was received.
EMS: Enable CIC **[EnableCIC]**	Determines whether the Carrier Identification Code (CIC) is relayed to ISDN. ▪ **[0]** = Do not relay the Carrier Identification Code (CIC) to ISDN (default). ▪ **[1]** = CIC is relayed to the ISDN in Transit Network Selection (TNS) IE. If enabled, the CIC code (received in an INVITE Request-URI) is included in a TNS IE in the ISDN Setup message. For example: INVITE sip:555666;cic=2345@100.2.3.4 sip/2.0. **Notes:** ▪ This feature is supported only for SIP-to-ISDN calls. ▪ The parameter AddCicAsPrefix can be used to add the CIC as a prefix to the destination phone number for routing IP-to-Tel calls.
EMS: Enable AOC **[EnableAOC]**	Determines whether ISDN Advice of Charge (AOC) messages are interworked with SIP. ▪ **[0]** = Not used (default). ▪ **[1]** = AOC messages are interworked to SIP (in receive direction) and sent to the PSTN in the transmit direction. The device supports both the receipt and sending of ISDN (Euro ISDN) AOC messages: ▪ AOC messages can be received during a call (Facility messages) or at the end of a call (Disconnect or Release messages). The device converts the AOC messages into SIP INFO (during a call) and BYE (end of a call) messages, using a proprietary AOC SIP header. The device supports both Currency and Pulse AOC messages. ▪ AOC messages can be sent during a call (Facility messages) or at the end of a call (Disconnect or Release messages). This is done by assigning the Charge Code index to the desired routing rule in the Outbound IP Routing table. For more information, see 'Advice of Charge Services for Euro ISDN' on page 349.
Web: IPMedia Detectors EMS: DSP Detectors Enable CLI: IPM-detectors-enable **[EnableDSPIPMDetectors]**	Enables the device's DSP detectors. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The device's Software Upgrade Key must contain the 'IPMDetector' DSP option. ▪ When enabled (1), the number of available channels is reduced.
Web: Add IE in SETUP	Adds an optional Information Element (IE) data (in hex format)

Parameter	Description
EMS: IE To Be Added In Q.931 Setup CLI: add-ie-in-setup **[AddIEinSetup]**	to ISDN Setup messages. For example, to add IE '0x20,0x02,0x00,0xe1', enter the value "200200e1". **Notes:** ▪ This IE is sent from the Trunk Group IDs that are defined by the parameter SendIEonTG. ▪ You can configure different IE data for Trunk Groups by defining this parameter for different IP Profile IDs (using the IPProfile parameter) and then assigning the required IP Profile ID in the Inbound IP Routing Table (PSTNPrefix). ▪ This feature is similar to that of the EnableISDNTunnelingIP2Tel parameter. If both parameters are configured, the EnableISDNTunnelingIP2Tel parameter takes precedence.
Web: Trunk Groups to Send IE EMS: List Of Trunk Groups To Send IE CLI: trkgrps-to-snd-ie **[SendIEonTG]**	Defines Trunk Group IDs (up to 50 characters) from where the optional ISDN IE (defined by the parameter AddIEinSetup) is sent. For example: '1,2,4,10,12,6'. **Notes:** ▪ You can configure different IE data for Trunk Groups by defining this parameter for different IP Profile IDs (using the parameter IPProfile), and then assigning the required IP Profile ID in the Inbound IP Routing Table (PSTNPrefix). ▪ When IP Profiles are used for configuring different IE data for Trunk Groups, this parameter is ignored.
Web: Enable User-to-User IE for Tel to IP EMS: Enable UUI Tel 2 Ip CLI: uui-ie-for-tel2ip **[EnableUUITel2IP]**	Enables ISDN PRI-to-SIP interworking. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable transfer of User-to-User (UU) IE from PRI to SIP. The device supports the following ISDN PRI-to-SIP interworking: Setup to SIP INVITE, Connect to SIP 200 OK, User Information to SIP INFO, Alerting to SIP 18x response, and Disconnect to SIP BYE response messages. **Note:** The interworking of ISDN User-to-User IE to SIP INFO is applicable only to the Euro ISDN, QSIG, and 4ESS PRI variants.
Web: Enable User-to-User IE for IP to Tel EMS: Enable UUI Ip 2 Tel CLI: uui-ie-for-ip2tel **[EnableUUIIP2Tel]**	Enables SIP-to-PRI ISDN interworking. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable transfer of User-to-User (UU) IE from SIP INVITE message to PRI Setup message. The device supports the following SIP-to-PRI ISDN interworking: SIP INVITE to Setup, SIP 200 OK to Connect, SIP INFO to User Information, SIP 18x to Alerting, and SIP BYE to Disconnect. **Notes:** ▪ The interworking of ISDN User-to-User IE to SIP INFO is applicable only to the Euro ISDN, QSIG, and 4ESS PRI variants. ▪ To interwork the UUIE header from SIP-to-ISDN messages with the 4ESS ISDN variant, the parameter ISDNGeneralCCBehavior must be set to 16384.
[Enable911LocationIdIP2Tel]	Enables interworking of Emergency Location Identification from

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>SIP to PRI.
▪ [0] = Disabled (default)
▪ [1] = Enabled
When enabled, the From header received in the SIP INVITE is translated into the following ISDN IE's:
▪ Emergency Call Control.
▪ Generic Information - to carry the Location Identification Number information.
▪ Generic Information - to carry the Calling Geodetic Location information.
Note: This capability is applicable only to the NI-2 ISDN variant.</td></tr>
<tr><td>[EarlyAnswerTimeout]</td><td>Defines the time (in seconds) that the device waits for an ISDN Connect message from the called party (Tel side) after sending a Setup message. If the timer expires, the call is answered by sending a SIP 200 OK message (IP side).
The valid range is 0 to 600. The default value is 0 (i.e., disabled).</td></tr>
<tr><td>Web/EMS: Trunk Transfer Mode
CLI: trk-xfer-mode-type
[TrunkTransferMode]</td><td>Determines the trunk transfer method (for all trunks) when a SIP REFER message is received. The transfer method depends on the Trunk's PSTN protocol (configured by the parameter ProtocolType) and is applicable only when one of these protocols are used:
<table>
<tr><th>PSTN Protocol</th><th>Transfer Method (Described Below)</th></tr>
<tr><td>E1 Euro ISDN [1]</td><td>ECT [2] or InBand [5]</td></tr>
<tr><td>E1 QSIG [21],
T1 QSIG [23]</td><td>Single Step Transfer [4], Path Replacement Transfer [2], or InBand [5]</td></tr>
<tr><td>T1 NI2 ISDN [10],
T1 4ESS ISDN [11],
T1 5ESS 9 ISDN [12]</td><td>TBCT [2] or InBand [5]</td></tr>
<tr><td>T1 DMS-100 ISDN [14]</td><td>RTL [2] or InBand [5]</td></tr>
<tr><td>T1 RAW CAS [3], T1 CAS [2], E1 CAS [8], E1 RAW CAS [9]</td><td>[1] CAS NFA DMS-100 or [3] CAS Normal transfer</td></tr>
<tr><td>T1 DMS-100 Meridian ISDN [35]</td><td>RTL [2] or InBand [5]</td></tr>
</table>
The valid values of this parameter are described below:
▪ [0] = Not supported (default).
▪ [1] = Supports CAS NFA DMS-100 transfer. When a SIP REFER message is received, the device performs a Blind Transfer by executing a CAS Wink, waits for an acknowledged Wink from the remote side, dials the Refer-to number to the switch, and then releases the call.
Note: A specific NFA CAS table is required.
▪ [2] = Supports ISDN (PRI/BRI) transfer - Release Link Trunk (RLT) (DMS-100), Two B Channel Transfer (TBCT) (NI2),</td></tr>
</table>

Parameter	Description
	Explicit Call Transfer (ECT) (EURO ISDN), and Path Replacement (QSIG). When a SIP REFER message is received, the device performs a transfer by sending Facility messages to the PBX with the necessary information on the call's legs to be connected. The different ISDN variants use slightly different methods (using Facility messages) to perform the transfer. **Notes:** ✓ For RLT ISDN transfer, the parameter SendISDNTransferOnConnect must be set to 1. ✓ The parameter SendISDNTransferOnConnect can be used to define if the TBCT/ECT transfer is performed after receipt of Alerting or Connect messages. For RLT, the transfer is always done after receipt of Connect (SendISDNTransferOnConnect is set to 1). ✓ This transfer can be performed between B-channels from different trunks or Trunk Groups, by using the parameter EnableTransferAcrossTrunkGroups. ✓ The device initiates the ECT process after receiving a SIP REFER message only for trunks that are configured to User side. ▪ **[3]** = Supports CAS Normal transfer. When a SIP REFER message is received, the device performs a Blind Transfer by executing a CAS Wink, dialing the Refer-to number to the switch, and then releasing the call. ▪ **[4]** = Supports QSIG Single Step transfer (PRI/BRI): IP-to-Tel: When a SIP REFER message is received, the device performs a transfer by sending a Facility message to the PBX, initiating Single Step transfer. Once a success return result is received, the transfer is completed. Tel-to-IP: When a Facility message initiating Single Step transfer is received from the PBX, a SIP REFER message is sent to the IP side. ▪ **[5]** = IP-to-Tel Blind Transfer mode supported for ISDN (PRI/BRI) protocols and implemented according to AT&T Toll Free Transfer Connect Service (TR 50075) “Courtesy Transfer-Human-No Data”. When the device receives a SIP REFER message, it performs a blind transfer by first dialing the DTMF digits (transfer prefix) defined by the parameter XferPrefixIP2Tel (configured to "*8" for AT&T service), and then (after 500 msec) the device dials the DTMF of the number (referred) from the Refer-To header sip:URI userpart. If the hostpart of the Refer-To sip:URI contains the device's IP address, and if the Trunk Group selected according to the IP to Tel Routing table is the same Trunk Group as the original call, then the device performs the in-band DTMF transfer; otherwise, the device sends the INVITE according to regular transfer rules. After completing the in-band transfer, the device waits for the ISDN Disconnect message. If the Disconnect message is received during the first 5 seconds, the device sends a SIP NOTIFY with 200 OK message; otherwise, the device sends a NOTIFY with 4xx message. ▪ **[6]** = Supports AT&T toll free out-of-band blind transfer for

Parameter	Description
	trunks configured with the 4ESS ISDN protocol. AT&T courtesy transfer is a supplementary service which enables a user (e.g., user "A") to transform an established call between it and user "B" into a new call between users "B" and "C", whereby user "A" does not have a call established with user "C" prior to call transfer. The device handles this feature as follows: ✓ IP-to-Tel (user side): When a SIP REFER message is received, the device initiates a transfer by sending a Facility message to the PBX. ✓ Tel-to-IP (network side): When a Facility message initiating an out-of-band blind transfer is received from the PBX, the device sends a SIP REFER message to the IP side (if the EnableNetworkISDNTransfer parameter is set to 1). **Note:** For configuring trunk transfer mode per trunk, use the parameter TrunkTransferMode_X.
[TrunkTransferMode_X]	Determines the trunk transfer mode per trunk (where *x* is the Trunk ID). For configuring trunk transfer mode for all trunks and for a description of the parameter options, refer to the parameter TrunkTransferMode.
[EnableTransferAcrossTrunkGroups]	Determines whether the device allows ISDN ECT, RLT or TBCT IP-to-Tel call transfers between B-channels of different Trunk Groups. ▪ **[0]** = Disable - ISDN call transfer is only between B-channels of the same Trunk Group (default). ▪ **[1]** = Enable - the device performs ISDN transfer between any two PSTN calls (between any Trunk Group) handled by the device. **Note:** The ISDN transfer also requires that you configure the parameter TrunkTransferMode_x to 2.
Web: ISDN Transfer Capabilities EMS: Transfer Capability To ISDN CLI: isdn-xfer-cab **[ISDNTransferCapability_ID]**	Defines the IP-to-ISDN Transfer Capability of the Bearer Capability IE in ISDN Setup messages. The *ID* in the ini file parameter denotes the trunk number, where 0 is the first trunk. ▪ **[-1]** Not Configured ▪ **[0]** Audio 3.1 = Audio (default). ▪ **[1]** Speech = Speech. ▪ **[2]** Data = Data. ▪ Audio 7 = Currently not supported. **Note:** If this parameter isn't configured or equals to '-1', Audio 3.1 capability is used.
[TransferCapabilityForDataCalls]	Defines the ISDN Transfer Capability for data calls. ▪ **[0]** = (Default) ISDN Transfer Capability for data calls is 64k unrestricted (data). ▪ **[1]** = ISDN Transfer Capability for data calls is determined according to the ISDNTransferCapability parameter.
Web: ISDN Transfer On Connect EMS: Send ISDN Transfer On Connect CLI: isdn-trsfr-on-conn **[SendISDNTransferOnConnect]**	This parameter is used for the ECT/TBCT/RLT/Path Replacement ISDN transfer methods. Usually, the device requests the PBX to connect an incoming and outgoing call. This parameter determines if the outgoing call (from the device to the PBX) must be connected before the transfer is initiated.

Parameter	Description
	▪ **[0]** Alert = Enables ISDN Transfer if the outgoing call is in Alerting or Connect state (default). ▪ **[1]** Connect = Enables ISDN Transfer only if the outgoing call is in Connect state. **Note:** For RLT ISDN transfer (TrunkTransferMode = 2 and ProtocolType = 14 DMS-100), this parameter must be set to 1.
[ISDNTransferCompleteTimeout]	Defines the timeout (in seconds) for determining ISDN call transfer (ECT, RLT, or TBCT) failure. If the device does not receive any response to an ISDN transfer attempt within this user-defined time, the device identifies this as an ISDN transfer failure and subsequently performs a hairpin TDM connection or sends a SIP NOTIFY message with a SIP 603 response (depending whether hairpin is enabled or disabled, using the parameter DisableFallbackTransferToTDM). The valid range is 1 to 10. The default is 4.
Web/EMS: Enable Network ISDN Transfer CLI: network-isdn-xfer **[EnableNetworkISDNTransfer]**	Determines whether the device allows interworking of network-side received ECT/TBCT Facility messages (NI2 TBCT - Two B-channel Transfer and ETSI ECT - Explicit Call Transfer) to SIP REFER. ▪ **[0]** Disable = Rejects ISDN transfer requests. ▪ **[1]** Enable (default) = The device sends a SIP REFER message to the remote call party if ECT/TBCT Facility messages are received from the ISDN side (e.g., from a PBX).
[DisableFallbackTransferToTDM]	Enables "hairpin" TDM transfer upon ISDN (ECT, RLT, or TBCT) call transfer failure. When this feature is enabled and an ISDN call transfer failure occurs, the device sends a SIP NOTIFY message with a SIP 603 Decline response. ▪ **[0]** = device performs a hairpin TDM transfer upon ISDN call transfer (default). ▪ **[1]** = Hairpin TDM transfer is disabled.
Web: Enable QSIG Transfer Update CLI: qsig-xfer-update **[EnableQSIGTransferUpdate]**	Determines whether the device interworks QSIG Facility messages with callTranferComplete invoke application protocol data unit (APDU) to SIP UPDATE messages with P-Asserted-Identity and optional Privacy headers. This feature is supported for IP-to-Tel and Tel-to-IP calls. ▪ **[0]** Disable (default) = Ignores QSIG Facility message with callTranferComplete invoke ▪ **[1]** Enable For example, assume A and C are PBX call parties, and B is the SIP IP phone: 1 A calls B; B answers the call. 2 A places B on hold, and calls C; C answers the call. 3 A performs a call transfer (the transfer is done internally by the PBX); B and C are connected to one another. In the above example, the PBX updates B that it is now talking with C. The PBX updates this by sending a QSIG Facility message with callTranferComplete invoke APDU. The device interworks this message to a SIP UPDATE message containing a P-Asserted-Identity header with the number and name derived from QSIG callTranferComplete redirectionNumber and

Parameter	Description
	redirectionName. **Note:** For IP-to-Tel calls, the redirectionNumber and redirectionName in the callTRansferComplete invoke is derived from the P-Asserted-Identity and Privacy headers.

CLI: is-cas-sndhook-flsh
[CASSendHookFlash]
	Enables sending Wink signal toward CAS trunks.
	▪ **[0]** = Disable (default).
	▪ **[1]** = Enable.
	If the device receives a mid-call SIP INFO message with flashhook event body (as shown below) and this parameter is set to 1, the device generates a wink signal toward the CAS trunk. The CAS wink signal is done by changing the A bit from 1 to 0, and then back to 1 for 450 msec.
	`INFO sip:4505656002@192.168.13.40:5060 SIP/2.0`
	`Via: SIP/2.0/UDP 192.168.13.2:5060`
	`From: <sip:06@192.168.13.2:5060>`
	`To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-1040067870294`
	`Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2`
	`CSeq:2 INFO`
	`Content-Type: application/broadsoft`
	`Content-Length: 17`
	`event flashhook`
	Note: This parameter is applicable only to T1 CAS protocols.

A.12.8 Answer and Disconnect Supervision Parameters

The answer and disconnect supervision parameters are described in the table below.

Table A-57: Answer and Disconnect Parameters

Parameter	Description
Web: Answer Supervision EMS: Enable Voice Detection CLI: answer-supervision **[EnableVoiceDetection]**	Enables the sending of SIP 200 OK upon detection of speech, fax, or modem. ▪ **[1]** Yes = The device sends a SIP 200 OK (in response to an INVITE message) when speech, fax, or modem is detected from the Tel side. ▪ **[0]** No = The device sends a SIP 200 OK only after it completes dialing to the Tel side (default). Typically, this feature is used only when early media (enabled using the EnableEarlyMedia parameter) is used to establish the voice path before the call is answered. **Notes:** ▪ FXO interfaces: This feature is applicable only to one-stage dialing (FXO). ▪ Digital interfaces: To activate this feature, set the EnableDSPIPMDetectors parameter to 1. ▪ Digital interfaces: This feature is applicable only when the protocol type is CAS.
Web/EMS: Max Call Duration (min) CLI: mx-call-duration **[MaxCallDuration]**	Defines the maximum duration (in minutes) of a call. If this duration is reached, the device terminates the call. This feature is useful for ensuring available resources for new calls, by ensuring calls are properly terminated. The valid range is 0 to 35,791. The default is 0 (i.e., no limitation).
Web/EMS: Disconnect on Dial Tone CLI: disc-on-dial-tone **[DisconnectOnDialTone]**	Determines whether the device disconnects a call when a dial tone is detected from the PBX. ▪ **[0]** Disable = Call is not released (default). ▪ **[1]** Enable = Call is released if dial tone is detected on the device's FXO port. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ This option is in addition to the mechanism that disconnects a call when either busy or reorder tones are detected.
Web: Send Digit Pattern on Connect EMS: Connect Code CLI: digit-pttrn-on-conn **[TelConnectCode]**	Defines a digit pattern to send to the Tel side after a SIP 200 OK is received from the IP side. The digit pattern is a user-defined DTMF sequence that is used to indicate an answer signal (e.g., for billing). The valid range is 1 to 8 characters. **Note:** This parameter is applicable to FXO/CAS.
Web: Disconnect on Broken Connection EMS: Disconnect Calls on Broken Connection CLI: disc-broken-conn **[DisconnectOnBrokenConnectio**	Determines whether the device releases the call if RTP packets are not received within a user-defined timeout. ▪ **[0]** No ▪ **[1]** Yes (default) **Notes:**

Parameter	Description
n]	▪ The timeout is configured by the BrokenConnectionEventTimeout parameter. ▪ This feature is applicable only if the RTP session is used without Silence Compression. If Silence Compression is enabled, the device doesn't detect a broken RTP connection. ▪ During a call, if the source IP address (from where the RTP packets are received) is changed without notifying the device, the device filters these RTP packets. To overcome this, set the DisconnectOnBrokenConnection parameter to 0; the device doesn't detect RTP packets arriving from the original source IP address and switches (after 300 msec) to the RTP packets arriving from the new source IP address. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: Broken Connection Timeout EMS: Broken Connection Event Timeout CLI: broken-connection-event-timeout **[BrokenConnectionEventTimeout]**	Defines the time period (in 100-msec units) after which a call is disconnected if an RTP packet is not received. The valid range is from 3 (i.e., 300 msec) to an unlimited value (e.g., 20 hours). The default value is 100 (i.e., 10000 msec or 10 seconds). **Notes:** ▪ This parameter is applicable only if the parameter DisconnectOnBrokenConnection is set to 1. ▪ Currently, this feature functions only if Silence Suppression is disabled.
Web: Disconnect Call on Silence Detection EMS: Disconnect On Detection Of Silence CLI: disc-on-silence-det **[EnableSilenceDisconnect]**	Determines whether calls are disconnected after detection of silence. ▪ **[1]** Yes = The device disconnects calls in which silence occurs (in both call directions) for more than a user-defined time. ▪ **[0]** No = Call is not disconnected when silence is detected (default). The silence duration can be configured by the FarEndDisconnectSilencePeriod parameter (default 120). **Note:** To activate this feature, set the parameters EnableSilenceCompression and FarEndDisconnectSilenceMethod to 1.
Web: Silence Detection Period **[sec]** EMS: Silence Detection Time Out **[FarEndDisconnectSilencePeriod]**	Defines the duration of the silence period (in seconds) after which the call is disconnected. The range is 10 to 28,800 (i.e., 8 hours). The default is 120 seconds. **Note:** For this parameter to take effect, a device reset is required.
Web: Silence Detection Method **[FarEndDisconnectSilenceMethod]**	Determines the silence detection method. ▪ **[0]** None = Silence detection option is disabled. ▪ **[1]** Packets Count = According to packet count. ▪ **[2]** Voice/Energy Detectors = N/A. ▪ **[3]** All = N/A. **Note:** For this parameter to take effect, a device reset is required.
[FarEndDisconnectSilenceThres	Defines the threshold of the packet count (in percentages)

Parameter	Description
hold]	below which is considered silence by the device. The valid range is 1 to 100%. The default is 8%. **Notes:** ▪ This parameter is applicable only if silence is detected according to packet count (FarEndDisconnectSilenceMethod is set to 1). ▪ For this parameter to take effect, a device reset is required.
[BrokenConnectionDuringSilence]	Enables the generation of the BrokenConnection event during a silence period if the channel's NoOp feature is enabled (using the parameter NoOpEnable) and if the channel stops receiving NoOp RTP packets. ▪ **[0]** Disable (default). ▪ **[1]** Enable.
Web: Trunk Alarm Call Disconnect Timeout CLI: trk-alrm-call-disc-to **[TrunkAlarmCallDisconnectTimeout]**	Defines the time (in seconds) to wait (in seconds) after an E1/T1 trunk "red" alarm (LOS/LOF) is raised before the device disconnects the SIP call. Once this user-defined time elapses, the device sends a SIP BYE message to terminate the call. If the alarm is cleared before this timeout elapses, the call is not terminated and continues as normal. The range is 1 to 80. The default is 0 (20 for E1 and 40 for T1).
Web: Disconnect Call on Busy Tone Detection (ISDN) EMS: Isdn Disconnect On Busy Tone CLI: disc-on-bsy-tone-i **[ISDNDisconnectOnBusyTone]**	Determines whether a call is disconnected upon detection of a busy tone (for ISDN). ▪ **[0]** Disable = Do not disconnect call upon detection of busy tone (default). ▪ **[1]** Enable = Disconnect call upon detection of busy tone. **Notes:** ▪ This parameter is applicable only to ISDN protocols. ▪ IP-to-ISDN calls are disconnected on detection of SIT tones only in call alert state. If the call is in connected state, the SIT does not disconnect the calls. Detection of Busy or Reorder tones disconnects the IP-to-ISDN calls also in call connected state. ▪ For IP-to-CAS calls, detection of Busy, Reorder or SIT tones disconnect the calls in any call state.
Web: Disconnect Call on Busy Tone Detection (CAS) EMS: Disconnect On Detection End Tones CLI: disc-on-bsy-tone-c **[DisconnectOnBusyTone]**	Determines whether a call is disconnected upon detection of a busy tone (for CAS). ▪ **[0]** Disable = Do not disconnect call on detection of busy tone. ▪ **[1]** Enable = Call is released if busy or reorder (fast busy) tone is detected on the device's FXO port (default). **Notes:** ▪ Digital interfaces: This parameter is applicable only to CAS protocols. ▪ Analog interfaces: This parameter is applicable only to FXO interfaces. ▪ This parameter can also be configured per Tel Profile, using the TelProfile parameter.
Polarity (Current) Reversal for Call Release (Analog Interfaces) Parameters	
CLI: fxs-fxo > default-linepolarity-	Defines the FXO line polarity, required for DID signaling.

Parameter	Description
state **[SetDefaultLinePolarityState]**	▪ **[0]** = Positive line polarity ▪ **[1]** = Negative line polarity ▪ **[2]** = Auto (default). The device detects the polarity upon power-up or upon insertion of the RJ-11 cable, and uses it as a reference polarity. Typically, if the RJ-11 cabling is connected correctly (without crossing, Tip to Tip, Ring to Ring), the Tip line is positive compared to the Ring line. In this case, set this parameter to 0. With this configuration, the device assumes that the idle line polarity is Tip line positive. When the device receives a SIP INVITE, it detects the FXO line polarity. If the polarity is "Reversed", it skips this FXO line and goes to the next line. **Notes:** ▪ To take advantage of this new feature, configure all FXO lines as a single Trunk Group with ascending or descending channel select mode, and configure routing rules to route incoming INVITE messages to this Trunk Group. ▪ This parameter is applicable only to FXO interfaces.
Web: Enable Polarity Reversal EMS: Enable Reversal Polarity CLI: polarity-rvrsl **[EnableReversalPolarity]**	Enables the polarity reversal feature for call release. ▪ **[0]** Disable = Disable the polarity reversal service (default). ▪ **[1]** Enable = Enable the polarity reversal service. If the polarity reversal service is enabled, the FXS interface changes the line polarity on call answer and then changes it back on call release. The FXO interface sends a 200 OK response when polarity reversal signal is detected (applicable only to one-stage dialing) and releases a call when a second polarity reversal signal is detected. **Note:** This parameter can also be configured per Tel Profile, using the TelProfile parameter.
Web/EMS: Enable Current Disconnect CLI: current-disc **[EnableCurrentDisconnect]**	Enables call release upon detection of a Current Disconnect signal. ▪ **[0]** Disable = Disable the current disconnect service (default). ▪ **[1]** Enable = Enable the current disconnect service. If the current disconnect service is enabled: ▪ The FXO releases a call when a current disconnect signal is detected on its port. ▪ The FXS interface generates a 'Current Disconnect Pulse' after a call is released from IP. The current disconnect duration is configured by the CurrentDisconnectDuration parameter. The current disconnect threshold (FXO only) is configured by the CurrentDisconnectDefaultThreshold parameter. The frequency at which the analog line voltage is sampled is configured by the TimeToSampleAnalogLineVoltage parameter. **Note:** This parameter can also be configured per Tel Profile, using the TelProfile parameter.
EMS: Polarity Reversal Type CLI: polarity-reversal-type	Defines the voltage change slope during polarity reversal or wink.

Parameter	Description
[PolarityReversalType]	▪ **[0]** = Soft reverse polarity (default). ▪ **[1]** = Hard reverse polarity. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ Some Caller ID signals use reversal polarity and/or Wink signals. In these cases, it is recommended to set the parameter PolarityReversalType to 1 (Hard). ▪ For this parameter to take effect, a device reset is required.
EMS: Current Disconnect Duration CLI: current-disconnect-duration **[CurrentDisconnectDuration]**	Defines the duration (in msec) of the current disconnect pulse. The range is 200 to 1500. The default is 900. **Notes:** ▪ This parameter is applicable for FXS and FXO interfaces. ▪ The FXO interface detection window is 100 msec below the parameter's value and 350 msec above the parameter's value. For example, if this parameter is set to 400 msec, then the detection window is 300 to 750 msec. ▪ For this parameter to take effect, a device reset is required.
[CurrentDisconnectDefaultThreshold]	Defines the line voltage threshold at which a current disconnect detection is considered. The valid range is 0 to 20 Volts. The default value is 4 Volts. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ For this parameter to take effect, a device reset is required.
CLI: time-to-sample-analog-line-voltage **[TimeToSampleAnalogLineVoltage]**	Defines the frequency at which the analog line voltage is sampled (after offhook), for detection of the current disconnect threshold. The valid range is 100 to 2500 msec. The default value is 1000 msec. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ For this parameter to take effect, a device reset is required.

A.12.9 Tone Parameters

This subsection describes the device's tone parameters.

A.12.9.1 Telephony Tone Parameters

The telephony tone parameters are described in the table below.

Table A-58: Tone Parameters

Parameter	Description
CLI: help-tone-4-ip2ip **[PlayHeldToneForIP2IP]**	Enables playing a Held tone to an IP-to-IP leg instead of putting it on hold. ▪ **[0]** = Disabled. The device interworks the re-INVITE with a=inactive from one SIP leg to another SIP leg. (default) ▪ **[1]** = Enabled. The device plays a Held tone to the IP if it receives a re-INVITE with a=inactive in the SDP from the party initiating the call hold. The Held tone must be configured in the CPT or PRT file. **Note:** This parameter is applicable only to the IP-to-IP application (enabled using the parameter EnableIP2IPApplication).
Web/EMS: Dial Tone Duration **[sec]** CLI: dt-duration **[TimeForDialTone]**	Defines the duration (in seconds) that the dial tone is played (for digital interfaces, to an ISDN terminal). For digital interfaces: This parameter is applicable for overlap dialing when ISDNInCallsBehavior is set to 65536. The dial tone is played if the ISDN Setup message doesn't include the called number. The valid range is 0 to 60. The default is 5. For analog interfaces: FXS interfaces play the dial tone after the phone is picked up (off-hook). FXO interfaces play the dial tone after the port is seized in response to ringing (from PBX/PSTN). The valid range is 0 to 60. The default time is 16. Notes for analog interfaces: ▪ During play of dial tone, the device waits for DTMF digits. ▪ This parameter is not applicable when Automatic Dialing is enabled.
Web/EMS: Stutter Tone Duration CLI: sttr-tone-duration **[StutterToneDuration]**	Defines the duration (in msec) of the Confirmation tone. A Stutter tone is played (instead of a regular dial tone) when a Message Waiting Indication (MWI) is received. The Stutter tone is composed of a Confirmation tone (Tone Type #8), which is played for the defined duration (StutterToneDuration) followed by a Stutter Dial tone (Tone Type #15). Both these tones are defined in the CPT file. The range is 1,000 to 60,000. The default is 2,000 (i.e., 2 seconds). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If you want to configure the duration of the Confirmation tone to longer than 16 seconds, you must increase the value of the parameter TimeForDialTone accordingly. ▪ The MWI tone takes precedence over the Call Forwarding Reminder tone. For more information on MWI, see

Parameter	Description
	Message Waiting Indication on page 321.
Web: FXO AutoDial Play BusyTone EMS: Auto Dial Play Busy Tone CLI: fxo-autodial-play-bsytn **[FXOAutoDialPlayBusyTone]**	Determines whether the device plays a Busy/Reorder tone to the PSTN side if a Tel-to-IP call is rejected by a SIP error response (4xx, 5xx or 6xx). If a SIP error response is received, the device seizes the line (off-hook), and then plays a Busy/Reorder tone to the PSTN side (for the duration defined by the parameter TimeForReorderTone). After playing the tone, the line is released (on-hook). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only to FXO interfaces.
Web: Hotline Dial Tone Duration EMS: Hot Line Tone Duration CLI: hotline-dt-dur **[HotLineToneDuration]**	Defines the duration (in seconds) of the Hotline dial tone. If no digits are received during this duration, the device initiates a call to a user-defined number (configured in the Automatic Dialing table - TargetOfChannel - see Configuring Automatic Dialing on page 356). The valid range is 0 to 60. The default is 16. **Notes:** ▪ This parameter is applicable to FXS and FXO interfaces. ▪ You can define the Hotline duration per FXS/FXO port using the Automatic Dialing table.
Web/EMS: Reorder Tone Duration **[sec]** CLI: reorder-tone-duration **[TimeForReorderTone]**	For Analog: Defines the duration (in seconds) that the device plays a Busy or Reorder tone duration before releasing the line. The valid range is 0 to 254. The default is 0 seconds. Typically, after playing a Reorder/Busy tone for the specified duration, the device starts playing an Offhook Warning tone. For Digital: Defines the duration (in seconds) that the CAS device plays a Busy or Reorder Tone before releasing the line. The valid range is 0 to 254. The default value is 10. **Notes:** ▪ The selection of Busy or Reorder tone is performed according to the release cause received from IP. ▪ This parameter is also applicable for ISDN when PlayBusyTone2ISDN is set to 2. ▪ This parameter can also be configured per Tel Profile, using the TelProfile parameter).
Web: Time Before Reorder Tone **[sec]** EMS: Time For Reorder Tone CLI: time-b4-reordr-tn **[TimeBeforeReorderTone]**	Defines the delay interval (in seconds) from when the device receives a SIP BYE message (i.e., remote party terminates call) until the device starts playing a Reorder tone to the FXS phone. The valid range is 0 to 60. The default is 0. **Note:** This parameter is applicable only to FXS interfaces.
Web: Cut Through Reorder Tone Duration **[sec]** CLI: cut-thru-reord-dur **[CutThroughTimeForReOrderTone]**	Defines the duration (in seconds) of the Reorder tone played to the PSTN side after the IP call party releases the call, for the Cut-Through feature. After the tone stops playing, an incoming call is immediately answered if the FXS is off-hooked (for analog interfaces) or the PSTN is connected (for digital interfaces). The valid values are 0 to 30. The default is 0 (i.e., no Reorder

Parameter	Description
	tone is played). **Note:** To enable the Cut-Through feature, use the DigitalCutThrough (for CAS channels) or CutThrough (for FXS channels) parameters.
Web/EMS: Enable Comfort Tone CLI: comfort-tone **[EnableComfortTone]**	Determines whether the device plays a Comfort Tone (Tone Type #18) to the FXS/FXO endpoint after a SIP INVITE is sent and before a SIP 18x response is received. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This parameter is applicable to FXS and FXO interfaces.
[WarningToneDuration]	Defines the duration (in seconds) for which the Off-Hook Warning Tone is played to the user. The valid range is -1 to 2,147,483,647. The default is 600. **Notes:** ▪ A negative value indicates that the tone is played infinitely. ▪ This parameter is applicable only to analog interfaces.
Web: Play Busy Tone to Tel CLI: play-bsy-tone-2tel **[PlayBusyTone2ISDN]**	Enables the device to play a busy or reorder tone to the PSTN after a Tel-to-IP call is released. ▪ **[0]** Don't Play = Immediately sends an ISDN Disconnect message (default). ▪ **[1]** Play when Disconnecting = Sends an ISDN Disconnect message with PI = 8 and plays a busy or reorder tone to the PSTN (depending on the release cause). ▪ **[2]** Play before Disconnect = Delays the sending of an ISDN Disconnect message for a user-defined time (configured by the TimeForReorderTone parameter) and plays a busy or reorder tone to the PSTN. This is applicable only if the call is released from the IP **[Busy Here** (486) or Not Found (404)] before it reaches the Connect state; otherwise, the Disconnect message is sent immediately and no tones are played. **Note**: This parameter is applicable only to PSTN digital interfces.
Web: Play Ringback Tone to Trunk CLI: play-rbt-to-trk **[PlayRBTone2Trunk_ID]**	Determines the playing method of the ringback tone to the trunk side per trunk, where ID denotes the trunk number and where 0 is the first trunk. ▪ **[-1]** Not configured = The settings of the PlayRBTone2Tel parameter is used (default). ▪ **[0]** Don't Play = When the device is configured for ISDN / CAS, it doesn't play a ringback tone. No Progress Indicator (PI) is sent to the ISDN unless the ProgressIndicator2ISDN_ID parameter is configured differently. ▪ **[1]** Play on Local = When the device is configured for CAS, it plays a local ringback tone to the PSTN upon receipt of a SIP 180 Ringing response (with or without SDP). Note that the receipt of a SIP 183 response does not cause the device configured for CAS to play a ringback tone (unless the SIP183Behaviour parameter is set to 1). When the device is configured for ISDN, it operates according to the LocalISDNRBSource parameter, as

Parameter	Description
	follows: ✓ If the device receives a SIP 180 Ringing response (with or without SDP) and the LocalISDNRBSource parameter is set to 1, it plays a ringback tone and sends an ISDN Alert with PI = 8 (unless the ProgressIndicator2ISDN_ID parameter is configured differently). ✓ If the LocalISDNRBSource parameter is set to 0, the device doesn't play a ringback tone and an Alert message without PI is sent to the ISDN. In this case, the PBX / PSTN plays the ringback tone to the originating terminal. Note that the receipt of a 183 response does not cause the device to play a ringback tone; the device sends a Progress message (unless SIP183Behaviour is set to 1). If the SIP183Behaviour parameter is set to 1, the 183 response is handled the same way as a 180 Ringing response. ▪ **[2]** Prefer IP = Plays according to 'Early Media'. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device configured for ISDN / CAS doesn't play the ringback tone; PI = 8 is sent in an ISDN Alert message (unless the ProgressIndicator2ISDN_ID parameter is configured differently). If a 180 response is received, but the 'early media' voice channel is not opened, the device configured for CAS plays a ringback tone to the PSTN. The device configured for ISDN operates according to the LocalISDNRBSource parameter: ✓ If LocalISDNRBSource is set to 1, the device plays a ringback tone and sends an ISDN Alert with PI = 8 to the ISDN (unless the ProgressIndicator2ISDN_ID parameter is configured differently). ✓ If LocalISDNRBSource is set to 0, the device doesn't play a ringback tone. No PI is sent in the ISDN Alert message (unless the ProgressIndicator2ISDN_ID parameter is configured differently). In this case, the PBX / PSTN plays a ringback tone to the originating terminal. Note that the receipt of a 183 response results in an ISDN Progress message (unless SIP183Behaviour is set to 1). If SIP183Behaviour is set to 1 (183 is handled the same way as a 180 with SDP), the device sends an Alert message with PI = 8 without playing a ringback tone. ▪ **[3]** Play Local Until Remote Media Arrive = Plays tone according to received media. The behaviour is similar to option [2]. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device plays a local ringback tone if there are no prior received RTP packets. The device stops playing the local ringback tone as soon as it starts receiving RTP packets. At this stage, if the device receives additional 18x responses, it does not resume playing the

Parameter	Description
	local ringback tone. Note that for ISDN trunks, this option is applicable only if LocalISDNRBSource is set to 1. **Notes:** ▪ This parameter is applicable only to the Gateway (GW) application (i.e., not the IP-to-IP application). ▪ This parameter is applicable only to PSTN digital interfces.
Web: Play Ringback Tone to Tel EMS: Play Ring Back Tone To Tel CLI: play-rbt2tel **[PlayRBTone2Tel]**	Determines the playing method of the ringback tone to the Tel (for analog interfaces) or Trunk (for digital interfaces) side. For digital interfaces: This parameter applies to all trunks that are not configured by the PlayRBTone2Trunk parameter (which defines ringback tone per Trunk). ▪ **[0]** Don't Play = ✓ **Analog Interfaces:** Ringback tone is not played. ✓ **Digital Interfaces:** The device configured for ISDN / CAS doesn't play a ringback tone. No PI is sent to the ISDN unless the ProgressIndicator2ISDN_ID parameter is configured differently. ▪ **[1]** Play on Local = ✓ **Analog Interfaces:** Plays a ringback tone to the Tel side of the call when a SIP 180/183 response is received. ✓ **Digital Interfaces:** The device configured for CAS plays a local ringback tone to the PSTN upon receipt of a SIP 180 Ringing response (with or without SDP). Note that the receipt of a 183 response does not cause the device configured for CAS to play a ringback tone (unless the SIP183Behaviour parameter is set to 1). The device configured for ISDN operates according to the LocalISDNRBSource parameter: 1) If the device receives a 180 Ringing response (with or without SDP) and the LocalISDNRBSource parameter is set to 1, it plays a ringback tone and sends an ISDN Alert with PI = 8 (unless the ProgressIndicator2ISDN_ID parameter is configured differently). 2) If the LocalISDNRBSource parameter is set to 0, the device doesn't play a ringback tone and an Alert message without PI is sent to the ISDN. In this case, the PBX / PSTN plays the ringback tone to the originating terminal. Note that the receipt of a 183 response does not cause the device configured for ISDN to play a ringback tone; the device issues a Progress message (unless SIP183Behaviour is set to 1). If the SIP183Behaviour parameter is set to 1, the 183 response is handled the same way as a 180 Ringing response. ▪ **[2]** Prefer IP = (default) ✓ **Analog Interfaces:** Plays a ringback tone to the Tel side only if a 180/183 response without SDP is received. If 180/183 with SDP message is received, the device cuts through the voice channel and doesn't play the ringback tone. ✓ **Digital Interfaces:** Plays according to 'Early Media'. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media

Parameter	Description
	response or due to an SDP in the current 180 response), the device configured for ISDN / CAS doesn't play the ringback tone; PI = 8 is sent in an ISDN Alert message (unless the ProgressIndicator2ISDN_ID parameter is configured differently). If a 180 response is received, but the 'early media' voice channel is not opened, the device configured for CAS plays a ringback tone to the PSTN. The device configured for ISDN operates according to the LocalISDNRBSource parameter: 1) If LocalISDNRBSource is set to 1, the device plays a ringback tone and sends an ISDN Alert with PI = 8 to the ISDN (unless the ProgressIndicator2ISDN_ID parameter is configured differently). 2) If LocalISDNRBSource is set to 0, the device doesn't play a ringback tone. No PI is sent in the ISDN Alert message (unless the ProgressIndicator2ISDN_ID parameter is configured differently). In this case, the PBX / PSTN plays a ringback tone to the originating terminal. Note that the receipt of a 183 response results in an ISDN Progress message (unless SIP183Behaviour is set to 1). If SIP183Behaviour is set to 1 (183 is handled the same way as a 180 + SDP), the device sends an Alert message with PI = 8, without playing a ringback tone. ▪ **[3]** Play Local Until Remote Media Arrive = Plays a ringback tone according to received media. The behaviour is similar to [2]. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device plays a local ringback tone if there are no prior received RTP packets. The device stops playing the local ringback tone as soon as it starts receiving RTP packets. At this stage, if the device receives additional 18x responses, it does not resume playing the local ringback tone. Note that for ISDN trunks, this option is applicable only if the LocalISDNRBSource parameter is set to 1. **Note:** This parameter is applicable to the Gateway and IP-to-IP applications.
Web: Play Ringback Tone to IP EMS: Play Ring Back Tone To IP CLI: play-rbt-2ip **[PlayRBTone2IP]**	Determines whether the device plays a ringback tone to the IP side for IP-to-Tel calls. ▪ **[0]** Don't Play = Ringback tone isn't played (default). ▪ **[1]** Play = Ringback tone is played after SIP 183 session progress response is sent. For digital modules: If configured to 1 ('Play') and EnableEarlyMedia is set to 1, the device plays a ringback tone according to the following: ▪ For CAS interfaces: the device opens a voice channel, sends a 183+SDP response, and then plays a ringback tone to IP. ▪ For ISDN interfaces: if a Progress or an Alerting message with PI (1 or 8) is received from the ISDN, the device opens a voice channel, sends a 183+SDP or 180+SDP

Parameter	Description
	response, but doesn't play a ringback tone to IP. If PI (1 or 8) is received from the ISDN, the device assumes that ringback tone is played by the ISDN switch. Otherwise, the device plays a ringback tone to IP after receiving an Alerting message from the ISDN. It sends a 180+SDP response, signaling to the calling party to open a voice channel to hear the played ringback tone. **Notes:** ▪ To enable the device to send a 183/180+SDP responses, set the EnableEarlyMedia parameter to 1. ▪ If the EnableDigitDelivery parameter is set to 1, the device doesn't play a ringback tone to IP and doesn't send 183 or 180+SDP responses. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: Play Local RBT on ISDN Transfer EMS: Play RBT On ISDN Transfer CLI: play-l-rbt-isdn-trsfr **[PlayRBTOnISDNTransfer]**	Determines whether the device plays a local ringback tone for ISDN's Two B Channel Transfer (TBCT), Release Line Trunk (RLT), or Explicit Call Transfer (ECT) call transfers to the originator when the second leg receives an ISDN Alerting or Progress message. ▪ **[0]** Don't Play (default). ▪ **[1]** Play. **Notes:** ▪ For Blind transfer, the local ringback tone is played to first call PSTN party when the second leg receives the ISDN Alerting or Progress message. ▪ For Consulted transfer, the local ringback tone is played when the second leg receives ISDN Alerting or Progress message if the Progress message is received after a SIP REFER. ▪ This parameter is applicable only if the parameter SendISDNTransferOnConnect is set to 1.
Web: MFC R2 Category EMS: R2 Category CLI: mfcr2-category **[R2Category]**	Defines the tone for MFC R2 calling party category (CPC). The parameter provides information on the calling party such as National or International call, Operator or Subscriber and Subscriber priority. The value range is 1 to 15 (defining one of the MFC R2 tones). The default value is 1.
Tone Index Table	
[ToneIndex]	This parameter table configures the Tone Index table, which allows you to define Distinctive Ringing and Call Waiting tones per FXS endpoint (or for a range of FXS endpoints). This is based on calling number (source number prefix) and/or called (destination number/prefix) for IP-to-Tel calls. This allows different tones to be played for an FXS endpoint depending on the source or destination number of the IP-to-Tel call. The format of this parameter is as follows: [ToneIndex] FORMAT ToneIndex_Index = ToneIndex_FXSPort_First, ToneIndex_FXSPort_Last, ToneIndex_SourcePrefix,

Parameter	Description
	ToneIndex_DestinationPrefix, ToneIndex_PriorityIndex; [\ToneIndex] Where, ▪ FXSPort_First = starting range of FXS ports (where 1 is the first port). ▪ FXSPort_Last = end range of FXS ports. ▪ SourcePrefix = prefix of the calling number. ▪ DestinationPrefix = prefix of the called number. ▪ PriorityIndex = index for Distinctive Ringing and Call Waiting tones (default is 0): ✓ Ringing tone index = index in the CPT file for playing the ring tone. ✓ Call Waiting tone index = priority index + FirstCallWaitingToneID(*). For example, if you want to select the Call Waiting tone defined in the CPT file at Index #9, then you can enter 1 as the priority index and the value 8 for FirstCallWaitingToneID. The summation of these values equals 9, i.e., index #9. For example, the configuration below plays the tone Index #3 to FXS ports 1 and 2 if the source number prefix of the received call is 20. ToneIndex 1 = 1, 2, 20*, , 3; **Notes:** ▪ You can define up to 50 indices. ▪ This parameter is applicable only to FXS interfaces. ▪ Typically, the Ringing and/or Call Waiting tone played is indicated in the SIP Alert-Info header field of the received INVITE message. If this header is not present, then the tone played is according to the settings of this table. ▪ For denoting a range of FXS ports, use the syntax x-y (e.g., "1-4" for ports 1 through 4). ▪ You can configure multiple entries with different source and/or destination prefixes and tones for the same FXS port.

A.12.9.2 Tone Detection Parameters

The signal tone detection parameters are described in the table below.

Table A-59: Tone Detection Parameters

Parameter	Description
EMS: DTMF Enable CLI: DTMF-detector-enable **[DTMFDetectorEnable]**	Enables the detection of DTMF signaling. ▪ **[0]** = Disable ▪ **[1]** = Enable (default)
EMS: MF R1 Enable CLI: MFR1-detector-enable **[MFR1DetectorEnable]**	Enables the detection of MF-R1 signaling. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: R1.5 Detection Standard	Determines the MF-R1 protocol used for detection.

Parameter	Description
[R1DetectionStandard]	▪ **[0]** = ITU (default) ▪ **[1]** = R1.5 **Note:** For this parameter to take effect, a device reset is required.
EMS: User Defined Tone Enable CLI: user-defined-tones-detector-enable **[UserDefinedToneDetectorEnable]**	Enables the detection of User Defined Tones signaling, applicable for Special Information Tone (SIT) detection. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: SIT Enable CLI: SIT-detector-enable **[SITDetectorEnable]**	Enables SIT detection according to the ITU-T recommendation E.180/Q.35. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. To disconnect IP-to-ISDN calls when a SIT tone is detected, the following parameters must be configured: ▪ SITDetectorEnable = 1 ▪ UserDefinedToneDetectorEnable = 1 ▪ ISDNDisconnectOnBusyTone = 1 (applicable for Busy, Reorder and SIT tones) Another parameter for handling the SIT tone is SITQ850Cause, which determines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when a SIT tone is detected on an IP-to-Tel call. To disconnect IP-to-CAS calls when a SIT tone is detected, the following parameters must be configured (applicable to FXO interfaces): ▪ SITDetectorEnable = 1 ▪ UserDefinedToneDetectorEnable = 1 ▪ DisconnectOnBusyTone = 1 (applicable for Busy, Reorder and SIT tones) **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The IP-to-ISDN call is disconnected on detection of a SIT tone only in call alert state. If the call is in connected state, the SIT does not disconnect the call. Detection of Busy or Reorder tones disconnect these calls also in call connected state. ▪ For IP-to-CAS calls, detection of Busy, Reorder, or SIT tones disconnect the call in any call state.
EMS: UDT Detector Frequency Deviation CLI: UDT-detector-frequency-deviation **[UDTDetectorFrequencyDeviation]**	Defines the deviation (in Hz) allowed for the detection of each signal frequency. The valid range is 1 to 50. The default value is 50. **Note:** For this parameter to take effect, a device reset is required.
EMS: CPT Detector Frequency Deviation CLI: CPT-detector-frequency-deviation **[CPTDetectorFrequencyDeviation]**	Defines the deviation (in Hz) allowed for the detection of each CPT signal frequency. The valid range is 1 to 30. The default value is 10. **Note:** For this parameter to take effect, a device reset is required.

A.12.9.3 Metering Tone Parameters

The metering tone parameters are described in the table below.

Table A-60: Metering Tone Parameters

Parameter	Description
Web: Generate Metering Tones EMS: Metering Mode CLI: gen-mtr-tones **[PayPhoneMeteringMode]**	Determines the method used to configure the metering tones that are generated to the Tel side. ▪ **[0]** Disable = Metering tones aren't generated (default). ▪ **[1]** Internal Table = Metering tones are generated according to the internal table configured by the parameter ChargeCode. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If you select 'Internal Table', you must configure the 'Charge Codes Table' (see 'Configuring Charge Codes Table' on page 353).
Web: Analog Metering Type EMS: Metering Type CLI: metering-type **[MeteringType]**	Determines the metering method for generating pulses (sinusoidal metering burst frequency) by the FXS port. ▪ **[0]** 12 KHz (default) = 12 kHz sinusoidal bursts ▪ **[1]** 16 KHz = 16 kHz sinusoidal bursts ▪ **[2]** = Polarity Reversal pulses **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces.
Web: Analog TTX Voltage Level EMS: TTX Voltage Level **[AnalogTTXVoltageLevel]**	Determines the metering signal/pulse voltage level (TTX). ▪ **[0]** 0V = 0 Vrms sinusoidal bursts ▪ **[1]** 0.5V = 0.5 Vrms sinusoidal bursts (default) ▪ **[2]** 1V = 1 Vrms sinusoidal bursts **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces.
Charge Codes Table	
Web: Charge Codes Table EMS: Charge Codes CLI: configure voip > gw analoggw ChargeCode **[ChargeCode]**	This *para*meter table configures metering tones (and their time intervals) that the device's FXS interface generates to the Tel side or the E1 trunk (EuroISDN) sends in AOC Facility messages to the PSTN (i.e., PBX). The format of this parameter is as follows: [ChargeCode] FORMAT ChargeCode_Index = ChargeCode_EndTime1, ChargeCode_PulseInterval1, ChargeCode_PulsesOnAnswer1, ChargeCode_EndTime2, ChargeCode_PulseInterval2, ChargeCode_PulsesOnAnswer2, ChargeCode_EndTime3, ChargeCode_PulseInterval3, ChargeCode_PulsesOnAnswer3, ChargeCode_EndTime4, ChargeCode_PulseInterval4, ChargeCode_PulsesOnAnswer4; [\ChargeCode] Where, ▪ EndTime = Period (1 - 4) end time. ▪ PulseInterval = Period (1 - 4) pulse interval. ▪ PulsesOnAnswer = Period (1 - 4) pulses on answer.

Parameter	Description
	For example: ChargeCode 1 = 7,30,1,14,20,2,20,15,1,0,60,1; ChargeCode 2 = 5,60,1,14,20,1,0,60,1; ChargeCode 3 = 0,60,1; ChargeCode 0 = 6, 3, 1, 12, 2, 1, 18, 5, 2, 0, 2, 1; **Notes:** ▪ The parameter can include up to 25 indices (i.e., up to 25 different metering rules can be defined). ▪ To associate a charge code to an outgoing Tel-to-IP call, use the Outbound IP Routing Table'. ▪ To configure the Charge Codes table using the Web interface, see 'Configuring Charge Codes Table' on page 353. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.12.10 Telephone Keypad Sequence Parameters

The telephony keypad sequence parameters are described in the table below.

Table A-61: Keypad Sequence Parameters

Parameter	Description
Prefix for External Line	
[Prefix2ExtLine]	Defines a string prefix (e.g., '9' dialed for an external line) that when dialed, the device plays a secondary dial tone (i.e., stutter tone) to the FXS line and then starts collecting the subsequently dialed digits from the FXS line. The valid range is a one-character string. The default is an empty string. **Notes:** ▪ You can enable the device to add this string as the prefix to the collected (and sent) digits, using the parameter AddPrefix2ExtLine. ▪ This parameter is applicable only to FXS interfaces.
CLI: prefix-2-ext-line **[AddPrefix2ExtLine]**	Determines whether the prefix string for accessing an external line (defined by the parameter Prefix2ExtLine) is added to the dialed number as the prefix and together sent to the IP destination (Tel-to-IP calls). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable For example, if this parameter is enabled and the prefix string for the external line is defined as "9" (using the parameter Prefix2ExtLine) and the FXS user wants to make a call to destination "123", the device collects and sends all the dialed digits, including the prefix string, as "9123" to the IP destination number. **Note:** This parameter is applicable only to FXS interfaces.
Hook Flash Parameters	
Web: Flash Keys Sequence Style CLI: flash-key-seq-style	Determines the hook-flash key sequence for FXS interfaces. ▪ **[0]** 0 = Flash hook (default) - only the phone's Flash button

Parameter	Description
[FlashKeysSequenceStyle]	is used, according to the following scenarios: ✓ During an existing call, if the user presses the Flash button, the call is put on hold; a dial tone is heard and the user is able to initiate a second call. Once the second call is established, on-hooking transfers the first (held) call to the second call. ✓ During an existing call, if a call comes in (call waiting), pressing the Flash button places the active call on hold and answers the waiting call; pressing Flash again toggles between these two calls. ▪ **[1]** 1 = Sequence of Flash hook and digit: ✓ Flash + 1: holds a call or toggles between two existing calls ✓ Flash + 2: makes a call transfer. ✓ Flash + 3: makes a three-way conference call (if the Three-Way Conference feature is enabled, i.e., the parameter Enable3WayConference is set to 1 and the parameter 3WayConferenceMode is set to 2). ▪ **[2]** 2 = Sequence of Flash Hook and digit: ✓ Flash Hook only: places a call on hold. ✓ Flash + 2: places a call on hold and answers a call-waiting call, or toggles between active and on-hold calls. ✓ Flash + 3: makes a three-way conference call (if the Enable3WayConference parameter is set to 1 and the 3WayConferenceMode parameter is set to 2, and the device houses the MPM modules). Note that the settings of the ConferenceCode parameter are ignored. ✓ Flash + 4: makes a call transfer.
Web: Flash Keys Sequence Timeout CLI: flash-key-seq-tmout **[FlashKeysSequenceTimeout]**	Defines the Flash keys sequence timeout - the time (in msec) that the device waits for digits after the user presses the Flash button (Flash Hook + Digit mode - when the parameter FlashKeysSequenceStyle is set to 1 or 2). The valid range is 100 to 5,000. The default is 2,000.
Keypad Feature - Call Forward Parameters	
Web: Unconditional EMS: Call Forward Unconditional CLI: fwd-unconditional **[KeyCFUnCond]**	Defines the keypad sequence to activate the immediate call forward option.
Web: No Answer EMS: Call Forward No Answer CLI: fwd-no-answer **[KeyCFNoAnswer]**	Defines the keypad sequence to activate the forward on no answer option.
Web: On Busy EMS: Call Forward Busy CLI: fwd-on-busy **[KeyCFBusy]**	Defines the keypad sequence to activate the forward on busy option.
Web: On Busy or No Answer EMS: CF Busy Or No Answer CLI: fwd-busy-or-no-ans **[KeyCFBusyOrNoAnswer]**	Defines the keypad sequence to activate the forward on 'busy or no answer' option.

Parameter	Description
Web: Do Not Disturb EMS: CF Do Not Disturb CLI: fwd-dnd **[KeyCFDoNotDisturb]**	Defines the keypad sequence to activate the Do Not Disturb option (immediately reject incoming calls).
To activate the required forward method from the telephone: 1 Dial the user-defined sequence number on the keypad; a dial tone is heard. 2 Dial the telephone number to which the call is forwarded (terminate the number with #); a confirmation tone is heard.	
Web: Deactivate EMS: Call Forward Deactivation CLI: fwd-deactivate **[KeyCFDeact]**	Defines the keypad sequence to deactivate any of the call forward options. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Caller ID Restriction Parameters	
Web: Activate EMS: CLIR CLI: id-restriction-act **[KeyCLIR]**	Defines the keypad sequence to activate the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Web: Deactivate EMS: CLIR Deactivation CLI: id-restriction-deact **[KeyCLIRDeact]**	Defines the keypad sequence to deactivate the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Hotline Parameters	
Web: Activate EMS: Hot Line CLI: hotline-act **[KeyHotLine]**	Defines the keypad sequence to activate the delayed hotline option. To activate the delayed hotline option from the telephone, perform the following: 1 Dial the user-defined sequence number on the keypad; a dial tone is heard. 2 Dial the telephone number to which the phone automatically dials after a configurable delay (terminate the number with #); a confirmation tone is heard.
Web: Deactivate EMS: Hot Line Deactivation CLI: hotline-deact **[KeyHotLineDeact]**	Defines the keypad sequence to deactivate the delayed hotline option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Transfer Parameters **Note:** See the description of the KeyBlindTransfer parameter for this feature.	
Keypad Feature - Call Waiting Parameters	
Web: Activate EMS: Keypad Features CW CLI: cw-act **[KeyCallWaiting]**	Defines the keypad sequence to activate the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Web: Deactivate EMS: Keypad Features CW Deact CLI: cw-deact **[KeyCallWaitingDeact]**	Defines the keypad sequence to deactivate the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Reject Anonymous Call Parameters	

Parameter	Description
Web: Activate EMS: Reject Anonymous Call **[KeyRejectAnonymousCall]**	Defines the keypad sequence to activate the reject anonymous call option, whereby the device rejects incoming anonymous calls. After the sequence is pressed, a confirmation tone is heard.
Web: Deactivate EMS: Reject Anonymous Call Deact **[KeyRejectAnonymousCallDeact]**	Defines the keypad sequence that de-activate the reject anonymous call option. After the sequence is pressed, a confirmation tone is heard.
[RejectAnonymousCallPerPort]	This *para*meter table determines whether the device rejects incoming anonymous calls on FXS interfaces. The format of this parameter is as follows: [RejectAnonymousCallPerPort] FORMAT RejectAnonymousCallPerPort_Index = RejectAnonymousCallPerPort_Enable, RejectAnonymousCallPerPort_Port, RejectAnonymousCallPerPort_Module; [\RejectAnonymousCallPerPort] Where, ▪ Enable = accept **[0]** (default) or reject **[1]** incoming anonymous calls. ▪ Port = Port number. ▪ Module = Module number. For example: RejectAnonymousCallPerPort 0 = 0,1,1; RejectAnonymousCallPerPort 1 = 1,2,1; If enabled, when a device's FXS interface receives an anonymous call, it responds with a 433 (Anonymity Disallowed) SIP response. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter is per FXS port. ▪ For configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.12.11 General FXO Parameters

The general FXO parameters are described in the table below.

Table A-62: General FXO Parameters

Parameter	Description
Web: FXO Coefficient Type EMS: Country Coefficients CLI: fxo-country-coefficients **[CountryCoefficients]**	Determines the FXO line characteristics (AC and DC) according to USA or TBR21 standard. ▪ **[66]** Europe = TBR21 ▪ **[70]** USA = United States (default) **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
CLI: fxo-dc-termination **[FXODCTermination]**	Defines the FXO line DC termination (i.e., resistance). ▪ **[0]** = DC termination is set to 50 Ohms (default). ▪ **[1]** = DC termination set to 800 Ohms. The termination changes from 50 to 800 Ohms only when moving from onhook to offhook. **Note:** For this parameter to take effect, a device reset is required.
CLI: enable-fxo-current-limit **[EnableFXOCurrentLimit]**	Enables limiting the FXO loop current to a maximum of 60 mA (according to the TBR21 standard). ▪ **[0]** = FXO line current limit is disabled (default). ▪ **[1]** = FXO loop current is limited to a maximum of 60 mA. **Note:** For this parameter to take effect, a device reset is required.
[FXONumberOfRings]	Defines the number of rings before the device's FXO interface answers a call by seizing the line. The valid range is 0 to 10. The default is 0. When set to 0, the FXO seizes the line after one ring. When set to 1, the FXO seizes the line after two rings. **Notes:** ▪ This parameter is applicable only if automatic dialing is not used. ▪ If caller ID is enabled and if the number of rings defined by the parameter RingsBeforeCallerID is greater than the number of rings defined by this parameter, the greater value is used.
Web/EMS: Dialing Mode CLI: dialing-mode **[IsTwoStageDial]**	Determines the dialing mode for IP-to-Tel (FXO) calls. ▪ **[0]** One Stage = One-stage dialing. In this mode, the device seizes one of the available lines (according to the ChannelSelectMode parameter), and then dials the destination phone number received in the INVITE message. To specify whether the dialing must start after detection of the dial tone or immediately after seizing the line, use the IsWaitForDialTone parameter. ▪ **[1]** Two Stages = Two-stage dialing (default). In this mode, the device seizes one of the PSTN/PBX lines without performing any dialing, connects the remote IP user to the PSTN/PBX, and all further signaling (dialing and Call Progress Tones) is performed directly with the PBX without the device's intervention. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ This parameter can also be configured per Tel Profile, using the TelProfile parameter.
Web/EMS: Waiting For Dial Tone CLI: waiting-4-dial-tone **[IsWaitForDialTone]**	Determines whether the device waits for a dial tone before dialing the phone number for IP-to-Tel (FXO) calls. ▪ **[0]** No = Don't wait for dial tone. ▪ **[1]** Yes = Wait for dial tone (default). When one-stage dialing and this parameter are enabled, the device dials the phone number (to the PSTN/PBX line) only after it detects a dial tone. If this parameter is disabled, the device immediately dials the

Parameter	Description
	phone number after seizing the PSTN/PBX line without 'listening' for a dial tone. **Notes:** ▪ The correct dial tone parameters must be configured in the CPT file. ▪ The device may take 1 to 3 seconds to detect a dial tone (according to the dial tone configuration in the CPT file). If the dial tone is not detected within 6 seconds, the device releases the call and sends a SIP 500 "Server Internal Error" response. ▪ This parameter is applicable only to FXO interfaces.
Web: Time to Wait before Dialing **[msec]** EMS: Time Before Dial CLI: time-wait-b4-dialing **[WaitForDialTime]**	For digital interfaces: Defines the delay after hook-flash is generated and until dialing begins. Applies to call transfer (i.e., the parameter TrunkTransferMode is set to 3) on CAS protocols. For Analog interfaces: Defines the delay before the device starts dialing on the FXO line in the following scenarios: ▪ The delay between the time the line is seized and dialing begins during the establishment of an IP-to-Tel call. **Note:** Applicable only for one-stage dialing when the parameter IsWaitForDialTone is disabled. ▪ The delay between detection of a Wink and the start of dialing during the establishment of an IP-to-Tel call (for DID lines, EnableDIDWink is set to 1). ▪ For call transfer - the delay after hook-flash is generated and dialing begins. The valid range (in milliseconds) is 0 to 20,000 (i.e., 20 seconds). The default value is 1,000 (i.e., 1 second).
Web: Ring Detection Timeout **[sec]** EMS: Timeout Between Rings CLI: ring-detection-tout **[FXOBetweenRingTime]**	Defines the timeout (in seconds) for detecting the second ring after the first detected ring. If automatic dialing is not used and Caller ID is enabled, the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP. If automatic dialing is used, the deviceinitiates a call to IP when the ringing signal is detected. The FXO line is seized only if the remote IP party answers the call. If the remote party doesn't answer the call and the second ring signal is not received within this timeout, the device releases the IP call. This parameter is typically set to between 5 and 8. The default is 8. **Notes:** ▪ This parameter is applicable only to FXO interfaces (for Tel-to-IP calls). ▪ This timeout is calculated from the end of the ring until the start of the next ring. For example, if the ring cycle is two seconds on and four seconds off, the timeout value should be configured to five seconds (i.e., greater than the off time, e.g., four).
Web: Rings before Detecting Caller ID	Determines the number of rings before the device starts

Parameter	Description
EMS: Rings Before Caller ID CLI: rings-b4-det-callerid **[RingsBeforeCallerID]**	detecting Caller ID. ▪ **[0]** 0 = Before first ring. ▪ **[1]** 1 = After first ring (default). ▪ **[2]** 2 = After second ring. **Note:** This parameter is applicable only to FXO interfaces.
Web/EMS: Guard Time Between Calls CLI: guard-time-btwn-calls **[GuardTimeBetweenCalls]**	Defines the time interval (in seconds) after a call has ended and a new call can be accepted for IP-to-Tel (FXO) calls. The valid range is 0 to 10. The default value is 1. **Notes:** ▪ Occasionally, after a call ends and on-hook is applied, a delay is required before placing a new call (and performing off-hook). This is necessary to prevent incorrect hook-flash detection or other glare phenomena. ▪ This parameter is applicable only to FXO interfaces.
Web: FXO Double Answer CLI: fxo-dbl-ans **[EnableFXODoubleAnswer]**	Enables the FXO Double Answer feature, which rejects (disconnects) incoming Tel (FXO)-to-IP collect calls and signals (informs) this call denial to the PSTN. ▪ **[0]** Disable (default) ▪ **[1]** Enable

A.12.12FXS Parameters

The general FXS parameters are described in the table below.

Table A-63: General FXS Parameters

Parameter	Description
Web: FXS Coefficient Type EMS: Country Coefficients CLI: fxs-country-coefficients **[FXSCountryCoefficients]**	Determines the FXS line characteristics (AC and DC) according to USA or Europe (TBR21) standards. ▪ **[66]** Europe = TBR21 ▪ **[70]** USA = United States (default) **Note:** For this parameter to take effect, a device reset is required.

A.12.13Trunk Groups and Routing Parameters

The routing parameters are described in the table below.

Table A-64: Routing Parameters

Parameter	Description
Trunk Group Table	
Web: Trunk Group Table EMS: SIP Endpoints > Phones CLI: configure voip > gw hunt-or-trunk-group TrunkGroup **[TrunkGroup]**	This *para*meter table is used to define and activate the device's endpoints/Trunk channels, by defining telephone numbers and assigning them to Trunk Groups. The format of this parameter is shown below: [TrunkGroup] FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum, TrunkGroup_FirstTrunkId, TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel, TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId, TrunkGroup_LastTrunkId, TrunkGroup_Module; [\TrunkGroup] For example, the configuration below assigns Trunk 1 (channels 1 to 30) of Module 1 to Trunk Group ID 2: TrunkGroup 0 = 2, 0, 1, 30, 50000, 0, 0, 1; the configuration below assigns BRI channels 1 through 4 of Module 2 to Trunk Group ID 2 with phone numbers 208 to 211: TrunkGroup 1 = 2, 0, 1, 4, 208, 0, 0 ,2; **Notes:** ▪ The first entry in this table starts at index 0. ▪ Trunk Group ID 1 is denoted as 0 in the table. ▪ This parameter can appear up to four times per module. ▪ For configuring this table in the Web interface, see Configuring Trunk Group Table on page 271. ▪ For a description of *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Trunk Group Settings	
Web: Trunk Group Settings EMS: SIP Routing > Hunt Group CLI: configure voip > gw hunt-or-trunk-group trunk-group-setting **[TrunkGroupSettings]**	This *para*meter table defines rules for channel allocation per Trunk Group. If no rule exists, the rule defined by the global parameter ChannelSelectMode takes effect. The format of this parameter is as follows: [TrunkGroupSettings] FORMAT TrunkGroupSettings_Index = TrunkGroupSettings_TrunkGroupId, TrunkGroupSettings_ChannelSelectMode, TrunkGroupSettings_RegistrationMode, TrunkGroupSettings_GatewayName,TrunkGroupSettings_ContactUser, TrunkGroupSettings_ServingIPGroup, TrunkGroupSettings_MWIInterrogationType; [\TrunkGroupSettings] Where, ▪ MWIInterrogationType = defines QSIG MWI to IP interworking for interrogating MWI supplementary services: ✓ **[255]** Not Configured ✓ **[0]** None = disables the feature.

Parameter	Description
	✓ **[1]** Use Activate Only = don't send any MWI Interrogation messages and only "passively" respond to MWI Activate requests from the PBX. ✓ **[2]** Result Not Used = send MWI Interrogation message, but don't use its result. Instead, wait for MWI Activate requests from the PBX. ✓ **[3]** Use Result = send MWI Interrogation messages, use its results, and use the MWI Activate requests. MWI Activate requests are interworked to SIP NOTIFY MWI messages. The SIP NOTIFY messages are sent to the IP Group defined by the NotificationIPGroupID parameter. For example: TrunkGroupSettings 0 = 1, 0, 5, branch-hq, user, 1, 255; TrunkGroupSettings 1 = 2, 1, 0, localname, user1, 2, 255; **Notes:** ▪ This parameter can include up to 120 indices. ▪ For configuring Trunk Group Settings using the Web interface, see 'Configuring Trunk Group Settings' on page 273. ▪ For a description on using *ini* file table parameters, see to 'Configuring ini File Table Parameters' on page 98.
Web: Channel Select Mode EMS: Channel Selection Mode CLI: ch-select-mode **[ChannelSelectMode]**	Method for allocating incoming IP-to-Tel calls to a channel. ▪ **[0]** By Dest Phone Number = Selects the channel according to the called (destination) number (default). If the number is not located, the call is released. If the channel is unavailable (e.g., busy), the call is put on call waiting (if call waiting is enabled and no other call is on call waiting); otherwise, the call is released. ▪ **[1]** Cyclic Ascending = Selects the next available channel (in the Trunk Group) in an ascending cyclic order. When the device reaches the highest channel number in the Trunk Group, it selects the lowest channel number in the Trunk Group and then starts ascending again. ▪ **[2]** Ascending = Selects the lowest available channel in the Trunk Group and if unavailable, selects the next higher channel. ▪ **[3]** Cyclic Descending = Selects the next available channel in descending cyclic order. It always selects the next lower channel number in the Trunk Group. When the device reaches the lowest channel number in the Trunk Group, it selects the highest channel number in the Trunk Group and then starts descending again. ▪ **[4]** Descending = Selects the highest available channel in the Trunk Group and if unavailable, selects the next lower channel. ▪ **[5]** Dest Number + Cyclic Ascending = The device first selects the channel according to the called number. If the called number isn't found, it then selects the next available channel in ascending cyclic order. **Note:** If the called number is located but the port associated with the number is busy, the call is released. ▪ **[6]** By Source Phone Number = The device selects the channel according to the calling number.

Parameter	Description
	▪ **[7]** Trunk Cyclic Ascending = The device selects the channel from the first channel of the next trunk (adjacent to the trunk from which the previous channel was allocated). This option is applicable only to digital interfaces. ▪ **[8]** Trunk & Channel Cyclic Ascending = The device implements the Trunk Cyclic Ascending and Cyclic Ascending methods to select the channel. This method selects the next physical trunk (pertaining to the Trunk Group) and then selects the B-channel of this trunk according to the cyclic ascending method (i.e., selects the channel after the last allocated channel). This option is applicable only to digital interfaces. For example, if the Trunk Group includes two physical trunks, 0 and 1: ✓ For the first incoming call, the first channel of Trunk 0 is allocated. ✓ For the second incoming call, the first channel of Trunk 1 is allocated. ✓ For the third incoming call, the second channel of Trunk 0 is allocated. ▪ **[9]** Ring to Hunt Group = The device allocates IP-to-Tel calls to all the FXS ports (channels) pertaining to a specific Hunt Group. When a call is received for a specific Hunt Group, all telephones connected to the FXS ports belonging to the Hunt Group start ringing. The call is eventually received by whichever telephone answers the call first (afterwhich the other phones stop ringing). This option is applicable only to FXS interfaces. ▪ **[10]** Select Trunk by ISDN SuppServ Table = The device selects the BRI port/module according to the settings in the ISDN Supplementary Services table (defined by the ISDNSuppServ parameter), allowing the routing of IP-to-Tel calls to specific BRI endpoints. ▪ **[11]** Dest Number + Ascending = The device allocates a channels to incoming IP-to-Tel calls as follows: **a.** The device attempts to route the call to the channel that is associated with the destination (called) number. If located, the call is sent to that channel. **b.** If the number is not located or the channel is unavailable (e.g., busy), the device searches, in ascending order, for the next available channel in the Trunk Group. If located, the call is sent to that channel. **c.** If the device all the channels are unavailable, the call is released. **Notes:** ▪ For defining the channel select mode per Trunk Group, see 'Configuring Trunk Group Settings' on page 273. ▪ The logical (for digital interfaces) phone numbers of the device's B-channels are defined by the TrunkGroup parameter.
Web: Default Destination Number CLI: dflt-dest-nb **[DefaultNumber]**	Defines the default destination phone number, which is used if the received message doesn't contain a called party number and no phone number is configured in the Trunk Group Table' (see Configuring the Trunk Group Table on page 271). This

Parameter	Description
	parameter is used as a starting number for the list of channels comprising all the device's Trunk Groups. The default value is 1000.
Web: Source IP Address Input CLI: src-ip-addr-input **[SourceIPAddressInput]**	Determines which IP address the device uses to determine the source of incoming INVITE messages for IP-to-Tel routing. ▪ **[-1]** = Auto Decision - if the IP-to-IP feature is enabled, this parameter is automatically set to Layer 3 Source IP. If the IP-to-IP feature is disabled, this parameter is automatically set to SIP Contact Header (1). (default) ▪ **[0]** SIP Contact Header = The IP address in the Contact header of the incoming INVITE message is used. ▪ **[1]** Layer 3 Source IP = The actual IP address (Layer 3) from where the SIP packet was received is used.
Web: Use Source Number As Display Name EMS: Display Name CLI: src-nb-as-disp-name **[UseSourceNumberAsDisplayName]**	Determines the use of Tel Source Number and Display Name for Tel-to-IP calls. ▪ **[0]** No = If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the IP Display Name remains empty (default). ▪ **[1]** Yes = If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the Tel Source Number is used as the IP Source Number and also as the IP Display Name. ▪ **[2]** Overwrite = The Tel Source Number is used as the IP Source Number and also as the IP Display Name (even if the received Tel Display Name is not empty). ▪ **[3]** Original = Similar to option [2], except that the operation is done before regular calling number manipulation.
Web/EMS: Use Display Name as Source Number CLI: disp-name-as-src-nb **[UseDisplayNameAsSourceNumber]**	Determines the use of Source Number and Display Name for IP-to-Tel calls. ▪ **[0]** No = If IP Display Name is received, the IP Source Number is used as the Tel Source Number and the IP Display Name is used as the Tel Display Name. If no Display Name is received from IP, the Tel Display Name remains empty (default). ▪ **[1]** Yes = If an IP Display Name is received, it is used as the Tel Source Number and also as the Tel Display Name, and Presentation is set to Allowed (0). If no Display Name is received from IP, the IP Source Number is used as the Tel Source Number and Presentation is set to Restricted (1). For example: When 'From: 100 <sip:200@201.202.203.204>' is received, the outgoing Source Number and Display Name are set to '100' and the Presentation is set to Allowed (0). When 'From: <sip:100@101.102.103.104>' is received, the outgoing Source Number is set to '100' and the Presentation is set to Restricted (1).

Parameter	Description
Web: Use Routing Table for Host Names and Profiles EMS: Use Routing Table For Host Names CLI: rte-tbl-4-host-names **[AlwaysUseRouteTable]**	Determines whether to use the device's routing table to obtain the URI host name and optionally, an IP profile (per call) even if a Proxy server is used. ▪ **[0]** Disable = Don't use internal routing table (default). ▪ **[1]** Enable = Use the Outbound IP Routing Table'. **Notes:** ▪ This parameter appears only if the 'Use Default Proxy' parameter is enabled. ▪ The domain name is used instead of a Proxy name or IP address in the INVITE SIP URI.
Web/EMS: Tel to IP Routing Mode CLI: tel2ip-rte-mode **[RouteModeTel2IP]**	For a description of this parameter, see 'Configuring Outbound IP Routing Table' on page 291.
Outbound IP Routing Table	
Web: Outbound IP Routing Table EMS: SIP Routing > Tel to IP CLI: configure voip > gw routing outbound-ip-routing **[Prefix]**	This *para*meter table configures the Outbound IP Routing Table for routing Tel-to-IP and IP-to-IP calls. The format of this parameter is as follows: [PREFIX] FORMAT PREFIX_Index = PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix, PREFIX_ProfileId, PREFIX_MeteringCode, PREFIX_DestPort, PREFIX_SrcIPGroupID, PREFIX_DestHostPrefix, PREFIX_DestIPGroupID, PREFIX_SrcHostPrefix, PREFIX_TransportType, PREFIX_SrcTrunkGroupID, PREFIX_DestSRD, PREFIX_CostGroup, PREFIX_ForkingGroup; [\PREFIX] For example: PREFIX 0 = *, domain.com, *, 0, 255, $$, -1, , 1, , -1, -1, -1,,; PREFIX 1 = 20, 10.33.37.77, *, 0, 255, $$, -1, , 2, , 0, -1,,; **Notes:** ▪ This parameter can include up to 200 indices. ▪ For a detailed description of the table's parameters and for configuring this table using the Web interface, see 'Configuring Outbound IP Routing Table' on page 291. ▪ For a description on using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Inbound IP Routing Table	
Web: Inbound IP Routing Table EMS: SIP Routing > IP to Hunt CLI: configure voip > gw routing inbound-ip-routing **[PSTNPrefix]**	This *para*meter table configures the routing of IP calls to Trunk Groups (or inbound IP Groups). The format of this parameter is as follows: [PSTNPrefix] FORMAT PstnPrefix_Index = PstnPrefix_DestPrefix, PstnPrefix_TrunkGroupId, PstnPrefix_SourcePrefix, PstnPrefix_SourceAddress, PstnPrefix_ProfileId, PstnPrefix_SrcIPGroupID, PstnPrefix_DestHostPrefix, PstnPrefix_SrcHostPrefix; [\PSTNPrefix] For example: PstnPrefix 0 = 100, 1, 200, *, 0, 2, , ;

Parameter	Description
	PstnPrefix 1 = *, 2, *, , 1, 3, acl, joe; **Notes:** ▪ This parameter can include up to 24 indices. ▪ For a description of the table's parameters, refer to the corresponding Web parameters in 'Configuring Inbound IP Routing Table' on page 299. ▪ To support the In-Call Alternative Routing feature, you can use two entries that support the same call but assigned with a different Trunk Group. The second entry functions as an alternative route if the first rule fails as a result of one of the release reasons configured in the AltRouteCauseIP2Tel table. ▪ Selection of Trunk Groups (for IP-to-Tel calls) is according to destination number, source number,and source IP address. ▪ The source IP address (SourceAddress) can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 and 10.8.8.99. ▪ The source IP address (SourceAddress) can include the asterisk ('*') wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255. ▪ If the source IP address (SourceAddress) includes an FQDN, DNS resolution is performed according to the parameter DNSQueryType. ▪ For available notations for denoting a range of multiple numbers, see 'Dialing Plan Notation for Routing and Manipulation' on page 374. ▪ For a description on using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Web/EMS: IP to Tel Routing Mode CLI: ip2tel-rte-mode **[RouteModeIP2Tel]**	Determines whether to route IP calls to the Trunk Group (or IP Group) before or after manipulation of the destination number (configured in 'Configuring Number Manipulation Tables' on page 277). ▪ **[0]** Route calls before manipulation = Calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied.
Web: IP Security EMS: Secure Call From IP CLI: ip-security **[SecureCallsFromIP]**	Determines the device's policy on accepting or blocking SIP calls (IP-to-Tel calls). This is useful in preventing unwanted SIP calls, SIP messages, and/or VoIP spam. ▪ **[0]** Disable = The device accepts all SIP calls (default). ▪ **[1]** Secure Incoming calls = The device accepts SIP calls (i.e., calls from the IP side) only from IP addresses that are defined in the Outbound IP Routing Table or Proxy Set table, or IP addresses resolved from DNS servers from FQDN values defined in the Proxy Set table. All other incoming calls are rejected. ▪ **[2]** Secure All calls = The device accepts SIP calls only from IP addresses (in dotted-decimal notation format) that are defined in the Outbound IP Routing Table table or Proxy Set table, and rejects all other incoming calls. In addition, if an FQDN is defined in the routing table or Proxy Set table, the

Parameter	Description
	call is allowed to be sent only if the resolved DNS IP address appears in one of these tables; otherwise, the call is rejected. Therefore, the difference between this option and option **[1]** is that this option is concerned only about numerical IP addresses that are defined in the tables. **Note:** If this parameter is set to **[1]** or **[2],** when using Proxies or Proxy Sets, it is unnecessary to configure the Proxy IP addresses in the routing table. The device allows SIP calls received from the Proxy IP addresses even if these addresses are not configured in the routing table.
Web/EMS: Filter Calls to IP CLI: filter-calls-to-ip **[FilterCalls2IP]**	Enables filtering of Tel-to-IP calls when a Proxy is used (i.e., IsProxyUsed parameter is set to 1 - see 'Configuring Proxy and Registration Parameters' on page 239). ▪ **[0]** Don't Filter = device doesn't filter calls when using a Proxy (default). ▪ **[1]** Filter = Filtering is enabled. When this parameter is enabled and a Proxy is used, the device first checks the Outbound IP Routing Table before making a call through the Proxy. If the number is not allowed (i.e., number isn't listed in the table or a call restriction routing rule of IP address 0.0.0.0 is applied), the call is released. **Note:** When no Proxy is used, this parameter must be disabled and filtering is according to the Outbound IP Routing Table'.
CLI: ip2tel-tagging-dst **[IP2TelTaggingDestDialPlanIndex]**	Determines the Dial Plan index in the external Dial Plan file (.dat) in which string labels ("tags") are defined for tagging incoming IP-to-Tel calls. The special “tag” is added as a prefix to the called party number, and then the Inbound IP Routing Table uses this “tag” instead of the original prefix. Manipulation is then performed (after routing) in the Manipulation table which strips the “tag” characters before sending the call to the endpoint. The valid values are 0 to 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The default is -1 (i.e., no dial plan file used). The routing label can be up to 9 (text) characters. **Notes:** ▪ This parameter is applicable only to digital interfaces. ▪ The routing must be configured to be performed before manipulation. ▪ For more information on this feature, see Dial Plan Prefix Tags for IP-to-Tel Routing on page 379.
[EnableETSIDiversion]	Determines the method in which the Redirect Number is sent to the Tel side. ▪ **[0]** = Q.931 Redirecting Number Information Element (IE) (default) ▪ **[1]** = ETSI DivertingLegInformation2 in a Facility IE
Web: Add CIC CLI: add-cic **[AddCicAsPrefix]**	Determines whether to add the Carrier Identification Code (CIC) as a prefix to the destination phone number for IP-to-Tel calls. ▪ **[0]** No (default) ▪ **[1]** Yes When this parameter is enabled, the cic parameter in the incoming SIP INVITE can be used for IP-to-Tel routing decisions. It routes the call to the appropriate Trunk Group

Parameter	Description
	based on this parameter's value. The SIP cic parameter enables the transmission of the cic parameter from the SIP network to the ISDN. The cic parameter is a three- or four-digit code used in routing tables to identify the network that serves the remote user when a call is routed over many different networks. The cic parameter is carried in the SIP INVITE and maps to the ISDN Transit Network Selection Information Element (TNS IE) in the outgoing ISDN Setup message (if the EnableCIC parameter is set to 1). The TNS IE identifies the requested transportation networks and allows different providers equal access support, based on customer choice. For example, as a result of receiving the below INVITE, the destination number after number manipulation is cic+167895550001: INVITE sip:5550001;cic=+16789@172.18.202.60:5060;user=phone SIP/2.0 **Note:** After the cic prefix is added, the Inbound IP Routing Table can be used to route this call to a specific Trunk Group. The Destination Number IP to Tel Manipulation table must be used to remove this prefix before placing the call to the ISDN.

A.12.14IP Connectivity Parameters

The IP connectivity parameters are described in the table below.

Table A-65: IP Connectivity Parameters

Parameter	Description
Web: Alt Routing Tel to IP Mode EMS: Alternative Routing Mode CLI: alt-rte-tel2ip-mode **[AltRoutingTel2IPMode]**	Determines the IP connectivity event(s) reason for triggering alternative routing. ▪ **[0]** None = Alternative routing is not used. ▪ **[1]** Connectivity = Alternative routing is performed if a ping or SIP OPTIONS message to the initial destination fails (determined according to the AltRoutingTel2IPConnMethod parameter). ▪ **[2]** QoS = Alternative routing is performed if poor QoS is detected. ▪ **[3]** Both = Alternative routing is performed if either ping to initial destination fails, poor QoS is detected, or the DNS host name is not resolved (default). **Notes:** ▪ QoS is quantified according to delay and packet loss calculated according to previous calls. QoS statistics are reset if no new data is received within two minutes. ▪ To receive quality information (displayed in the 'Quality Status' and 'Quality Info.' fields in "Viewing IP Connectivity" on page 725) per destination, this parameter must be set to 2 or 3.
Web: Alt Routing Tel to IP Connectivity Method EMS: Alternative Routing Telephone to IP Connection Method CLI: alt-rte-tel2ip-method **[AltRoutingTel2IPConnMethod]**	Determines the method used by the device for periodically querying the connectivity status of a destination IP address. ▪ **[0]** ICMP Ping (default) = Internet Control Message Protocol (ICMP) ping messages. ▪ **[1]** SIP OPTIONS = The remote destination is considered offline if the latest OPTIONS transaction timed out. Any response to an OPTIONS request, even if indicating an error, brings the connectivity status to online.
Web: Alt Routing Tel to IP Keep Alive Time EMS: Alternative Routing Keep Alive Time CLI: alt-rte-tel2ip-keep-alive **[AltRoutingTel2IPKeepAliveTime]**	Defines the time interval (in seconds) between SIP OPTIONS Keep-Alive messages used for the IP Connectivity application. The valid range is 5 to 2,000,000. The default value is 60.
Web: Max Allowed Packet Loss for Alt Routing **[%]** CLI: mx-pkt-loss-4-alt-rte **[IPConnQoSMaxAllowedPL]**	Defines the packet loss (in percentage) at which the IP connection is considered a failure and Alternative Routing mechanism is activated. The default value is 20%.
Web: Max Allowed Delay for Alt Routing **[msec]** CLI: mx-all-dly-4-alt-rte **[IPConnQoSMaxAllowedDelay]**	Defines the transmission delay (in msec) at which the IP connection is considered a failure and the Alternative Routing mechanism is activated. The range is 100 to 10,000. The default value is 250.

A.12.15 Alternative Routing Parameters

The alternative routing parameters are described in the table below.

Table A-66: Alternative Routing Parameters

Parameter	Description
Web/EMS: Redundant Routing Mode CLI: redundant-routing-m **[RedundantRoutingMode]**	Determines the type of redundant routing mechanism when a call can't be completed using the main route. ▪ **[0]** Disable = No redundant routing is used. If the call can't be completed using the main route (using the active Proxy or the first matching rule in the Routing table), the call is disconnected. ▪ **[1]** Routing Table = Internal routing table is used to locate a redundant route (default). ▪ **[2]** Proxy = Proxy list is used to locate a redundant route. **Note:** To implement the Redundant Routing Mode mechansim, you first need to configure the parameter AltRouteCauseTEL2IP (Reasons for Alternative Routing table).
Web: Enable Alt Routing Tel to IP EMS: Enable Alternative Routing CLI: alt-routing-tel2ip **[AltRoutingTel2IPEnable]**	Enables the Alternative Routing feature for Tel-to-IP calls. ▪ **[0]** Disable = Disables the Alternative Routing feature (default). ▪ **[1]** Enable = Enables the Alternative Routing feature. ▪ **[2]** Status Only = The Alternative Routing feature is disabled, but read-only information on the QoS of the destination IP addresses is provided.
[EnableAltMapTel2IP]	Enables different Tel-to-IP destination number manipulation rules per routing rule when several (up to three) Tel-to-IP routing rules are defined and if alternative routing using release causes is used. For example, if an INVITE message for a Tel-to-IP call is returned with a SIP 404 Not Found response, the call can be re-sent to a different destination number (as defined using the parameter NumberMapTel2IP). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Web/EMS: Alternative Routing Tone Duration **[ms]** CLI: alt-rte-tone-duration **[AltRoutingToneDuration]**	Defines the duration (in milliseconds) for which the device plays a tone to the endpoint on each Alternative Routing attempt. When the device finishes playing the tone, a new SIP INVITE message is sent to the new destination. The tone played is the Call Forward Tone (Tone Type #25 in the CPT file). The valid range is 0 to 20,000. The default is 0 (i.e., no tone is played).
Reasons for Alternative Tel-to-IP Routing Table	
Web: Reasons for Alternative Routing EMS: Alt Route Cause Tel to IP CLI: configure voip > gw manipulations general-setting alt-route-cause-tel2ip **[AltRouteCauseTel2IP]**	This *para*meter table configures SIP call failure reason values received from the IP side. If an IP call is released as a result of one of these reasons, the device attempts to locate an alternative IP route (address) for the call in the Outbound IP Routing Table (if a Proxy is not used) or used as a redundant Proxy (you need to set the parameter RedundantRoutingMode to 2). The release reason for Tel-to-IP calls is provided in SIP 4xx, 5xx, and 6xx response codes. The format of this parameter is as follows: [AltRouteCauseTel2IP]

Parameter	Description
	FORMAT AltRouteCauseTel2IP_Index = AltRouteCauseTel2IP_ReleaseCause; [\AltRouteCauseTel2IP] For example: AltRouteCauseTel2IP 0 = 486; (Busy Here) AltRouteCauseTel2IP 1 = 480; (Temporarily Unavailable) AltRouteCauseTel2IP 2 = 408; (No Response) **Notes:** ▪ This parameter can include up to 5 indices. ▪ The reasons for alternative routing for Tel-to-IP calls apply only when a Proxy is not used. ▪ When there is no response to an INVITE message (after INVITE retransmissions), the device issues an internal 408 'No Response' implicit release reason. ▪ The device sends the call to an alternative IP route only after the call has failed and the device has subsequently attempted twice to establish the call unsuccessfully. ▪ The device also plays a tone to the endpoint whenever an alternative route is used. This tone is played for a user-defined time (configured by the parameter AltRoutingToneDuration). ▪ For configuring *ini* file table parameters, see "Configuring ini File Table Parameters" on page 98
Reasons for Alternative IP-to-Tel Routing Table	
Web: Reasons for Alternative IP-to-Tel Routing EMS: Alt Route Cause IP to Tel CLI: configure voip > gw manipulations general-setting alt-route-cause-ip2tel **[AltRouteCauseIP2Tel]**	This *para*meter table configures call failure reason values received from the PSTN side (in Q.931 presentation). If a call is released as a result of one of these reasons, the device attempts to locate an alternative Hunt Group for the call in the Inbound IP Routing Table. The format of this parameter is as follows: [AltRouteCauseIP2Tel] FORMAT AltRouteCauseIP2Tel_Index = AltRouteCauseIP2Tel_ReleaseCause; [\AltRouteCauseIP2Tel] For example: AltRouteCauseIP2Tel 0 = 3 (No Route to Destination) AltRouteCauseIP2Tel 1 = 1 (Unallocated Number) AltRouteCauseIP2Tel 2 = 17 (Busy Here) **Notes:** ▪ This parameter can include up to 5 indices. ▪ If the device fails to establish a call to the PSTN because it has no available channels in a specific Hunt Group (e.g., all the channels are occupied, or the spans are disconnected or out-of-sync), it uses the Internal Release Cause '3' (No Route to Destination). This cause can be used in the AltRouteCauseIP2Tel table to define routing to an alternative Hunt Group. ▪ This table can be used for example, in scenarios where the destination is busy and the Release Reason #17 is issued or for other call releases that issue the default Release Reason (#3). ▪ The device also plays a tone to the endpoint whenever an

Parameter	Description
	alternative route is used. This tone is played for a user-defined time (configured by the parameter AltRoutingToneDuration). ▪ For configuring *ini* file table parameters, see "Configuring ini File Table Parameters" on page 98.
Forward On Busy Trunk Destination Table	
Web/EMS: Forward On Busy Trunk Destination CLI: configure voip > gw routing fwd-on-bsy-trk-dest **[ForwardOnBusyTrunkDest]**	This parameter table configures the Forward On Busy Trunk Destination table. This table allows you to define an alternative IP destination if a trunk is busy, for IP-to-Tel calls. The destination can be an IP address or a SIP Request-URI user name and host part (i.e., user@host). The format of this parameter is as follows: [ForwardOnBusyTrunkDest] FORMAT ForwardOnBusyTrunkDest_Index = ForwardOnBusyTrunkDest_TrunkGroupId, ForwardOnBusyTrunkDest_ForwardDestination; [\ForwardOnBusyTrunkDest] For example, the below configuration forwards IP-to-Tel calls to destination user “112” at host IP address 10.13.4.12, port 5060, using transport protocol TCP, if Trunk Group ID 2 is unavailable: ForwardOnBusyTrunkDest 1 = 2, 112@10.13.4.12:5060;transport=tcp; When configured with user@host, the original destination number is replaced by the user part. **Notes:** ▪ The maximum number of indices (starting from 1) depends on the maximum number of Hunt Groups. ▪ For the destination, instead of a dotted-decimal IP address, FQDN can be used. In addition, the following syntax can be used: "host:port;transport=xxx"(i.e., IP address, port and transport type).

A.12.16Number Manipulation Parameters

The number manipulation parameters are described in the table below.

Table A-67: Number Manipulation Parameters

Parameter	Description
[ManipulateIP2PSTNReferTo]	Enables the manipulation of the called party (destination) number according to the SIP Refer-To header received by the device for TDM (PSTN) blind transfer. The number in the SIP Refer-To header is manipulated for all types of blind transfers to the PSTN (TBCT, ECT, RLT, QSIG, FXO, and CAS). ▪ **[0]** Disable (default) ▪ **[1]** Enable During the blind transfer, the device initiates a new call to the PSTN and the destination number of this call can be manipulated

Parameter	Description
	if this parameter is enabled. When enabled, the manipulation is done as follows: 1 If you configure a value for the xferPrefix parameter, then this value (string) is added as a prefix to the number in the Refer-To header. 2 This called party number is then manipulated using the IP-to-Tel Destination Phone Number Manipulation table. The source number of the transferred call is taken from the original call, according to its initial direction: ✓ Source number of the original call if it is a Tel-to-IP call ✓ Destination number of the original call if it is an IP-to-Tel call This source number can also be used as the value for the 'Source Prefix' field in the IP-to-Tel Destination Phone Number Manipulation table. The local IP address is used as the value for the 'Source IP Address' field. **Note:** This manipulation does not affect IP-to-Trunk Group routing rules.
Use EndPoint Number As Calling Number Tel2IP CLI: epn-as-cpn-tel2ip **[UseEPNumAsCallingNumTel2IP]**	Enables the use of the B-channel number as the calling number (sent in the From field of the INVITE) instead of the number received in the Q.931 Setup message, for Tel-to-IP calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable For example, if the incoming calling party number in the Q.931 Setup message is "12345" and the B-channel number is 17, then the outgoing INVITE From header is set to "17" instead of "12345". **Note:** When enabled, this feature is applied before routing and manipulation on the source number.
Use EndPoint Number As Calling Number IP2Tel CLI: epn-as-cpn-ip2tel **[UseEPNumAsCallingNumIP2Tel]**	Enables the use of the B-channel number as the calling party number (sent in the Q.931 Setup message) instead of the number received in the From header of the INVITE, for IP-to-Tel calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable For example, if the incoming INVITE From header contains "12345" and the destined B-channel number is 17, then the outgoing calling party number in the Q.931 Setup message is set to "17" instead of "12345". **Note:** When enabled, this feature is applied after routing and manipulation on the source number (i.e., just before sending to the Tel side).

Parameter	Description
Web: Tel2IP Default Redirect Reason **[Tel2IPDefaultRedirectReason]**	Determines the default redirect reason for Tel-to-IP calls when no redirect reason (or "unknown") exists in the received Q931 ISDN Setup message. The device includes this default redirect reason in the SIP History-Info header of the outgoing INVITE. If a redirect reason exists in the received Setup message, this parameter is ignored and the device sends the INVITE message with the reason according to the received Setup message. If this parameter is not configured (-1), the outgoing INVITE is sent with the redirect reason as received in the Setup message (if none or "unknown" reason, then without a reason). ▪ **[-1**] Not Configured (default) = Received redirect reason is not changed ▪ **[1]** Busy = Call forwarding busy ▪ **[2]** No Reply = Call forwarding no reply ▪ **[9]** DTE Out of Order = Call forwarding DTE out of order ▪ **[10]** Deflection = Call deflection ▪ **[15]** Systematic/Unconditional = Call forward unconditional
Web: Set Redirect number Screening Indicator to TEL EMS: Set IP To Tel Redirect Screening Indicator CLI: redir-nb-si-2tel **[SetIp2TelRedirectScreeningInd]**	Determines the value of the Redirect Number screening indicator in ISDN Setup messages. ▪ **[-1**] Not Configured (default) ▪ **[0]** User Provided ▪ **[1]** User Passed ▪ **[2]** User Failed ▪ **[3]** Network Provided **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).
Web: Set IP-to-TEL Redirect Reason CLI: ip2tel-redir-reason **[SetIp2TelRedirectReason]**	Defines the redirect reason for IP-to-Tel calls. If redirect (diversion) information is received from the IP, the redirect reason is set to the value of this parameter before the device sends it on to the Tel. ▪ **[-1**] Not Configured (default) ▪ **[0]** Unkown ▪ **[1]** Busy ▪ **[2]** No Reply ▪ **[3]** Network Busy ▪ **[4]** Deflection ▪ **[9]** DTE out of Order ▪ **[10]** Forwarding DTE ▪ **[13]** Transfer ▪ **[14]** PickUp ▪ **[15]** Systematic/Unconditional **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).

Parameter	Description
Web: Set TEL-to-IP Redirect Reason CLI: tel2ip-redir-reason **[SetTel2IpRedirectReason]**	Defines the redirect reason for Tel-to-IP calls. If redirect (diversion) information is received from the Tel, the redirect reason is set to the value of this parameter before the device sends it on to the IP. ▪ **[-1**] Not Configured (default) ▪ **[0]** Unkown ▪ **[1]** Busy ▪ **[2]** No Reply ▪ **[3]** Network Busy ▪ **[4]** Deflection ▪ **[9]** DTE out of Order ▪ **[10]** Forwarding DTE ▪ **[13]** Transfer ▪ **[14]** PickUp ▪ **[15]** Systematic/Unconditional **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).
Web: Send Screening Indicator to IP EMS: Screening Indicator To IP **[ScreeningInd2IP]**	Overrides the calling party's number (CPN) screening indication in the received ISDN SETUP message for Tel-to-IP calls. ▪ **[-1]** Not Configured = (Default) Not configured (interworking from ISDN to IP) or set to 0 for CAS. ▪ **[0]** User Provided = CPN set by user, but not screened (verified). ▪ **[1]** User Passed = CPN set by user, verified and passed. ▪ **[2]** User Failed = CPN set by user, and verification failed. ▪ **[3]** Network Provided = CPN set by network. **Note:** This parameter is applicable only if the Remote Party ID (RPID) header is enabled.
Web: Send Screening Indicator to ISDN EMS: Screening Indicator To ISDN **[ScreeningInd2ISDN]**	Overrides the screening indicator of the calling party's number for IP-to-Tel ISDN calls. ▪ **[-1]** Not Configured = (Default) Not configured (interworking from IP to ISDN). ▪ **[0]** User Provided = user provided, not screened. ▪ **[1]** User Passed = user provided, verified and passed. ▪ **[2]** User Failed = user provided, verified and failed. ▪ **[3]** Network Provided = network provided **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).
Web: Copy Destination Number to Redirect Number EMS: Copy Dest to Redirect Number CLI: cp-dst-nb-2-redir-nb **[CopyDest2RedirectNumber]**	Determines whether the device copies the received ISDN (digital interfaces) called number to the outgoing SIP Diversion header for Tel-to-IP calls (even if a Redirecting Number IE is not received in the ISDN Setup message, for digital interfaces). Therefore, the called number is used as a redirect number. Call redirection information is typically used for Unified Messaging and voice mail services to identify the recipient of a message. ▪ **[0]** Don't copy = Disable (default). ▪ **[1]** Copy after phone number manipulation = Copies the called number after manipulation. The device first performs Tel-to-IP destination phone number manipulation (i.e., on the SIP To

Parameter	Description
	header), and only then copies the manipulated called number to the SIP Diversion header for the Tel-to-IP call. Therefore, with this option, the called and redirect numbers are identical. ▪ **[2]** Copy before phone number manipulation = Copies the called number before manipulation. The device first copies the original called number to the SIP Diversion header, and then performs Tel-to-IP destination phone number manipulation. Therefore, this allows you to have different numbers for the called (i.e., SIP To header) and redirect (i.e., SIP Diversion header) numbers. **Notes:** ▪ For digital interfaces: If the incoming ISDN-to-IP call includes a Redirect Number, this number is overridden by the new called number if this parameter is set to **[1]** or **[2].** ▪ When configured in an IP Profile, this parameter can also be used for IP-to-Tel calls. The device can overwrite the redirect number with the destination number from the received SIP INVITE message in the outgoing ISDN call. This is achieved by assigning an IP Profile (IPProfile parameter) defined with the CopyDest2RedirectNumber parameter set to 1, to the IP-to-Tel Routing table (PSTNPrefix parameter). Even if there is no SIP Diversion or History header in the incoming INVITE message, the outgoing Q.931 Setup message will contain a redirect number. ▪ This parameter can also be configured per IP Profile (using the IPProfile parameter).
CLI: rep-calling-w-redir disc-on-bsy-tone-i **[ReplaceCallingWithRedirectNumber]**	Enables the replacement of the calling number with the redirect number for ISDN-to-IP calls. ▪ **[0]** = Disable (default). ▪ **[1]** = The calling name is removed and left blank. The outgoing INVITE message excludes the redirect number that was used to replace the calling number. The replacement is done only if a redirect number is present in the incoming Tel call. ▪ **[2]** = Manipulation is done on the new calling party number (after manipulation of the original calling party number, using the Tel2IPSourceNumberMappingDialPlanIndex parameter), but before the regular calling or redirect number manipulation: ✓ If a redirect number exists, it replaces the calling party number. If there is no redirect number, the calling number is left unchanged. ✓ If there is a calling “display” name, it remains unchanged. ✓ The redirect number remains unchanged and is included in the SIP Diversion header.
Web/EMS: Add Trunk Group ID as Prefix CLI: trkgrpid-prefix **[AddTrunkGroupAsPrefix]**	Determines whether the Trunk Group ID is added as a prefix to the destination phone number (i.e., called number) for Tel-to-IP calls. ▪ **[0]** No = Don't add Trunk Group ID as prefix (default). ▪ **[1]** Yes = Add Trunk Group ID as prefix to called number. **Notes:** ▪ This option can be used to define various routing rules. ▪ To use this feature, you must configure the Trunk Group IDs (see Configuring Trunk Group Table on page 271).

Parameter	Description
Web: Add Trunk ID as Prefix EMS: Add Port ID As Prefix CLI: trk-id-as-prefix **[AddPortAsPrefix]**	Determines whether the slot number / port number / Trunk ID is added as a prefix to the called number for Tel-to-IP calls. ▪ **[0]** No = slot number / port number / Trunk ID not added as prefix (default). ▪ **[1]** Yes = slot number / port number / Trunk ID added as prefix If enabled, the slot number (a single digit in the range of 1 to 6) and port number/Trunk ID (single digit in the range 1 to 8) are added as a prefix to the called (destination) phone number. For example, for the first trunk/channel located in the first slot, the number "11" is added as the prefix. This option can be used to define various routing rules.
Web/EMS: Add Trunk Group ID as Prefix to Source CLI: trkgrpid-pref2source **[AddTrunkGroupAsPrefixToSource]**	Determines whether the device adds the Trunk Group ID (from where the call originated) as the prefix to the calling number (i.e. source number). ▪ **[0]** No (default) ▪ **[1]** Yes
Web: Replace Empty Destination with B-channel Phone Number EMS: Replace Empty Dst With Port Number CLI: empty-dst-w-bch-nb **[ReplaceEmptyDstWithPortNumber]**	Determines whether the internal channel number is used as the destination number if the called number is missing. ▪ **[0]** No (default) ▪ **[1]** Yes **Note:** This parameter is applicable only to Tel-to-IP calls and if the called number is missing.
[CopyDestOnEmptySource]	▪ **[0]** = Leave Source Number empty (default). ▪ **[1]** = If the Source Number of a Tel-to-IP call is empty, the Destination Number is copied to the Source Number.
Web: Add NPI and TON to Calling Number EMS: Add NPI And TON As Prefix To Calling Number CLI: npi-n-ton-to-cng-nb **[AddNPIandTON2CallingNumber]**	Determines whether the Numbering Plan Indicator (NPI) and Type of Numbering (TON) are added to the Calling Number for Tel-to-IP calls. ▪ **[0]** No = Do not change the Calling Number (default). ▪ **[1]** Yes = Add NPI and TON to the Calling Number ISDN Tel-to-IP call. For example: After receiving a Calling Number of 555, NPI of 1, and TON of 3, the modified number becomes 13555. This number can later be used for manipulation and routing.
Web: Add NPI and TON to Called Number EMS: Add NPI And TON As Prefix To Called Number CLI: npi-n-ton-to-cld-nb **[AddNPIandTON2CalledNumber]**	Determines whether NPI and TON are added to the Called Number for Tel-to-IP calls. ▪ **[0]** No = Do not change the Called Number (default). ▪ **[1]** Yes = Add NPI and TON to the Called Number of ISDN Tel-to-IP call. For example: After receiving a Called Number of 555, NPI of 1 and TON of 3, the modified number becomes 13555. This number can later be used for manipulation and routing.
Web: IP to Tel Remove Routing Table Prefix EMS: Remove Prefix CLI: ip2tel-rmv-rte-tbl **[RemovePrefix]**	Determines whether the device removes the prefix from the destination number for IP-to-Tel calls. ▪ **[0]** No = Don't remove prefix (default) ▪ **[1]** Yes = Remove the prefix (defined in the Inbound IP Routing Table - see 'Configuring Inbound IP Routing Table' on page 299) from a telephone number for an IP-to-Tel call

Parameter	Description
	before forwarding it to Tel. For example: To route an incoming IP-to-Tel call with destination number 21100, the Inbound IP Routing Table is scanned for a matching prefix. If such a prefix is found (e.g., 21), then before the call is routed to the corresponding Trunk Group, the prefix (21) is removed from the original number, and therefore, only 100 remains. **Notes:** ▪ This parameter is applicable only if number manipulation is performed after call routing for IP-to-Tel calls (i.e., RouteModeIP2Tel parameter is set to 0). ▪ Similar operation (of removing the prefix) is also achieved by using the usual number manipulation rules.
[UseReferredByForCallingNumber]	Determines whether the device uses the number from the URI in the SIP Referred-By header as the calling number in the outgoing Q.931 Setup message, when SIP REFER messages are received. ▪ **[0]** = (Default) No ▪ **[1]** = Yes **Notes:** ▪ This parameter is applicable to all ISDN (TBCT, RLT, ECT) and CAS blind call transfers (except for in-band) and when the device receives SIP REFER messages with a Referred-By header. ▪ This manipulation is done before regular IP-to-Tel source number manipulation.
Web/EMS: Swap Redirect and Called Numbers CLI: swap-rdr-n-called-nb **[SwapRedirectNumber]**	▪ **[0]** No = Don't change numbers (default). ▪ **[1]** Yes = Incoming ISDN call that includes a redirect number (sometimes referred to as 'original called number') uses the redirect number instead of the called number.
[SwapTel2IPCalled&CallingNumbers]	Determines whether the device swaps the calling and called numbers received from the Tel side (for Tel-to-IP calls). The SIP INVITE message contains the swapped numbers. ▪ **[0]** = Disabled (default) ▪ **[1]** = Swap calling and called numbers **Note:** This parameter can also be configured per Tel Profile, using the TelProfile parameter.
Web: Add NPI and TON to redirect Number CLI: np-n-ton-2-redirnb **[AddNPIandTON2RedirectNumber]**	Determines whether the NPI and TON values are added as the prefix to the Redirect number in the INVITE message's Diversion or History-Info headers, for ISDN Tel-to-IP calls. ▪ **[0]** Yes (Default) ▪ **[1]** No
Web/EMS: Add Prefix to Redirect Number CLI: add-pref-to-redir-nb **[Prefix2RedirectNumber]**	Defines a string prefix that is added to the Redirect number received from the Tel side. This prefix is added to the Redirect Number in the SIP Diversion header. The valid range is an 8-character string. The default is an empty string.
Web: Add Number Plan and Type to RPI Header EMS: Add Ton 2 RPI CLI: np-n-type-to-rpi-hdr	Determines whether the TON/PLAN parameters are included in the Remote-Party-ID (RPID) header. ▪ **[0]** No

Parameter	Description
[AddTON2RPI]	▪ **[1]** Yes (default) If the Remote-Party-ID header is enabled (EnableRPIHeader = 1) and AddTON2RPI = 1, it's possible to configure the calling and called number type and number plan using the Number Manipulation tables for Tel-to-IP calls.
Web/EMS: Source Manipulation Mode CLI: src-manipulation **[SourceManipulationMode]**	Determines the SIP headers containing the source number after manipulation: ▪ **[0]** = The SIP From and P-Asserted-Identity headers contain the source number after manipulation (default). ▪ **[1]** = Only SIP From header contains the source number after manipulation, while the P-Asserted-Identity header contains the source number before manipulation.
Calling Name Manipulations IP-to-Tel Table	
CLI: config voip > gw manipulations calling-name-map-ip2tel **[CallingNameMapIp2Tel]**	Configures rules for manipulating the calling name (caller ID) in the received SIP message for IP-to-Tel calls. This can include modifying or removing the calling name. The format of this ini file parameter table is as follows: [CallingNameMapIp2Tel] FORMAT CallingNameMapIp2Tel_Index = CallingNameMapIp2Tel_DestinationPrefix, CallingNameMapIp2Tel_SourcePrefix, CallingNameMapIp2Tel_CallingNamePrefix, CallingNameMapIp2Tel_SourceAddress, CallingNameMapIp2Tel_RemoveFromLeft, CallingNameMapIp2Tel_RemoveFromRight, CallingNameMapIp2Tel_LeaveFromRight, CallingNameMapIp2Tel_Prefix2Add, CallingNameMapIp2Tel_Suffix2Add; [\CallingNameMapIp2Tel]
Calling Name Manipulations Tel-to-IP Table	
CLI: config voip > gw manipulations calling-name-map-tel2ip **[CallingNameMapTel2Ip]**	Configures rules for manipulating the calling name (caller ID) for Tel-to-IP calls. This can include modifying or removing the calling name. [CallingNameMapTel2Ip] FORMAT CallingNameMapTel2Ip_Index = CallingNameMapTel2Ip_DestinationPrefix, CallingNameMapTel2Ip_SourcePrefix, CallingNameMapTel2Ip_CallingNamePrefix, CallingNameMapTel2Ip_SrcTrunkGroupID, CallingNameMapTel2Ip_SrcIPGroupID, CallingNameMapTel2Ip_RemoveFromLeft, CallingNameMapTel2Ip_RemoveFromRight, CallingNameMapTel2Ip_LeaveFromRight, CallingNameMapTel2Ip_Prefix2Add, CallingNameMapTel2Ip_Suffix2Add; **[** \CallingNameMapTel2Ip]
Destination Phone Number Manipulation for IP-to-Tel Calls Table	
Web: Destination Phone Number Manipulation Table for IP > Tel Calls EMS: EMS: SIP Manipulations >	This *para*meter table manipulates the destination number of IP-to-Tel calls. The format of this parameter is as follows: [NumberMapIp2Tel] FORMAT NumberMapIp2Tel_Index =

Parameter	Description
Destination IP to Telcom CLI: config voip > gw manipulations NumberMapIp2Tel2 **[NumberMapIP2Tel]**	NumberMapIp2Tel_DestinationPrefix, NumberMapIp2Tel_SourcePrefix, NumberMapIp2Tel_SourceAddress, NumberMapIp2Tel_NumberType, NumberMapIp2Tel_NumberPlan, NumberMapIp2Tel_RemoveFromLeft, NumberMapIp2Tel_RemoveFromRight, NumberMapIp2Tel_LeaveFromRight, NumberMapIp2Tel_Prefix2Add, NumberMapIp2Tel_Suffix2Add, NumberMapIp2Tel_IsPresentationRestricted; [\NumberMapIp2Tel] For example: NumberMapIp2Tel 0 = 01,034,10.13.77.8,$$,0,$$,2,$$,667,$$; NumberMapIp2Tel 1 = 10,10,1.1.1.1,255,255,3,0,5,100,$$,255; **Notes:** ▪ This table parameter can include up to 100 indices. ▪ The manipulation rules are done in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ If the called and calling numbers match the DestinationPrefix, SourcePrefix, and/or SourceAddress conditions, then the RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, and/or NumberPlan are applied. ▪ The Source IP address can include the following wildcards: ✓ 'x': represents single digits. For example: 10.8.8.xx represents addresses between 10.8.8.10 and 10.8.8.99. ✓ '*' (asterisk): represents any number between 0 and 255. For example, 10.8.8.* represents addresses between 10.8.8.0 and 10.8.8.255. ▪ The following parameteris not applicable: IsPresentationRestricted. ▪ To configure manipulation of destination numbers for IP-to-Tel calls using the Web interface, see 'Configuring Number Manipulation Tables' on page 277). ▪ For a description on using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
[PerformAdditionalIP2TELDestinationManipulation]	Enables additional destination number manipulation for IP-to-Tel calls. The additional manipulation is done on the initially manipulated destination number, and this additional rule is also configured in the manipulation table (NumberMapIP2Tel parameter). This enables you to configure only a few manipulation rules for complex number manipulation requirements (that generally require many rules). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Destination Phone Number Manipulation for Tel-to-IP Calls Table	
Web: Destination Phone Number Manipulation Table for Tel > IP Calls EMS: SIP Manipulations > Destination Telcom to IPs CLI: config voip > gw	This *para*meter table manipulates the destination number of Tel-to-IP calls. The format of this parameter is as follows: [NumberMapTel2Ip] FORMAT NumberMapTel2Ip_Index = NumberMapTel2Ip_DestinationPrefix, NumberMapTel2Ip_SourcePrefix,

Parameter	Description
manipulations NumberMapTel2Ip **[NumberMapTel2IP]**	NumberMapTel2Ip_SourceAddress, NumberMapTel2Ip_NumberType, NumberMapTel2Ip_NumberPlan, NumberMapTel2Ip_RemoveFromLeft, NumberMapTel2Ip_RemoveFromRight, NumberMapTel2Ip_LeaveFromRight, NumberMapTel2Ip_Prefix2Add, NumberMapTel2Ip_Suffix2Add, NumberMapTel2Ip_IsPresentationRestricted, NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_ SrcIPGroupID; [\NumberMapTel2Ip] For example: NumberMapTel2Ip 0 = 01,$$,*,0,0,2,$$,$$,971,$$,$$,$$,$$; NumberMapTel2Ip 1 = 10,10,*,255,255,3,0,5,100,$$,255,$$,$$; **Notes:** ▪ This table parameter can include up to 120 indices (0-119). ▪ The manipulation rules are done in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ If the called and calling numbers match the DestinationPrefix and/or SourcePrefix conditions, then the parameters NumberType, NumberPlan, RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, and/or LeaveFromRight are applied. ▪ Number Plan and Type can be used in the Remote-Party-ID header by configuring the EnableRPIHeader and AddTON2RPI parameters. ▪ The following parameters are not applicable: SourceAddress and IsPresentationRestricted. ▪ To configure manipulation of destination numbers for Tel-to-IP calls using the Web interface, see 'Configuring the Number Manipulation Tables' on page 277). ▪ For a description on using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Source Phone Number Manipulation for IP-to-Tel Calls Table	
Web: Source Phone Number Manipulation Table for IP > Tel Calls EMS: EMS: SIP Manipulations > Source IP to Telcom CLI: config voip > gw manipulations SourceNumberMapIp2Tel **[SourceNumberMapIP2Tel]**	This *para*meter table manipulates the source number for IP-to-Tel calls. The format of this parameter is as follows: [SourceNumberMapIp2Tel] FORMAT SourceNumberMapIp2Tel_Index = SourceNumberMapIp2Tel_DestinationPrefix, SourceNumberMapIp2Tel_SourcePrefix, SourceNumberMapIp2Tel_SourceAddress, SourceNumberMapIp2Tel_NumberType, SourceNumberMapIp2Tel_NumberPlan, SourceNumberMapIp2Tel_RemoveFromLeft, SourceNumberMapIp2Tel_RemoveFromRight, SourceNumberMapIp2Tel_LeaveFromRight, SourceNumberMapIp2Tel_Prefix2Add, SourceNumberMapIp2Tel_Suffix2Add, SourceNumberMapIp2Tel_IsPresentationRestricted; [\SourceNumberMapIp2Tel] For example: SourceNumberMapIp2Tel 0 = 22,03,$$,$$,$$,$$,2,667,$$,$$;

Parameter	Description
	SourceNumberMapIp2Tel 1 = 034,01,1.1.1.1,$$,0,2,$$,$$,972,$$,10; **Notes:** ▪ This table parameter can include up to 120 indices. ▪ The manipulation rules are done in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ If the called and calling numbers match the DestinationPrefix, SourcePrefix, and/or SourceAddress conditions, then the RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, and/or NumberPlan are applied. ✓ 'x': represents single digits. For example: 10.8.8.xx represents addresses between 10.8.8.10 and 10.8.8.99. ✓ '*' (asterisk): represents any number between 0 and 255. For example, 10.8.8.* represents addresses between 10.8.8.0 and 10.8.8.255. ▪ To configure manipulation of source numbers for IP-to-Tel calls using the Web interface, see 'Configuring Number Manipulation Tables' on page 277). ▪ For a description on using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
[PerformAdditionalIP2TELSourceManipulation]	Enables additional source number manipulation for IP-to-Tel calls. The additional manipulation is done on the initially manipulated source number, and this additional rule is also configured in the manipulation table (SourceNumberMapIP2Tel parameter). This enables you to configure only a few manipulation rules for complex number manipulation requirements (that generally require many rules). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Source Phone Number Manipulation for Tel-to-IP Calls Table	
Web: Source Phone Number Manipulation Table for Tel > IP Calls EMS: SIP Manipulations > Source Telcom to IP CLI: config voip > gw manipulations SourceNumberMapTel2Ip **[SourceNumberMapTel2IP]**	This *para*meter table manipulates the source phone number for Tel-to-IP calls. The format of this parameter is as follows: [SourceNumberMapTel2Ip] FORMAT SourceNumberMapTel2Ip_Index = SourceNumberMapTel2Ip_DestinationPrefix, SourceNumberMapTel2Ip_SourcePrefix, SourceNumberMapTel2Ip_SourceAddress, SourceNumberMapTel2Ip_NumberType, SourceNumberMapTel2Ip_NumberPlan, SourceNumberMapTel2Ip_RemoveFromLeft, SourceNumberMapTel2Ip_RemoveFromRight, SourceNumberMapTel2Ip_LeaveFromRight, SourceNumberMapTel2Ip_Prefix2Add, SourceNumberMapTel2Ip_Suffix2Add, SourceNumberMapTel2Ip_IsPresentationRestricted, NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_SrcIPGroupID; [\SourceNumberMapTel2Ip] For example: SourceNumberMapTel2Ip 0 = 22,03,$$,0,0,$$,2,$$,667,$$,0,$$,$$;

Parameter	Description
	SourceNumberMapTel2Ip 0 = 10,10,*,255,255,3,0,5,100,$$,255,$$,$$; **Notes:** ▪ This table parameter can include up to 120 indices. ▪ The manipulation rules are done in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ If the called and calling numbers match the DestinationPrefix and/or SourcePrefix conditions, then the RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, NumberPlan, and/or IsPresentationRestricted are applied. ▪ An asterisk ('*') represents all IP addresses. ▪ IsPresentationRestricted is set to 'Restricted' only if 'Asserted Identity Mode' is set to 'P-Asserted'. ▪ Number Plan and Type can optionally be used in the Remote Party ID header by configuring the EnableRPIHeader and AddTON2RPI parameters. ▪ To configure manipulation of source numbers for Tel-to-IP calls using the Web interface, see 'Configuring Number Manipulation Tables' on page 277). ▪ For a description on using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
For the ETSI ISDN variant, the following Number Plan and Type combinations (Plan/Type) are supported in the Destination and Source Manipulation tables: ▪ 0,0 = Unknown, Unknown ▪ 9,0 = Private, Unknown ▪ 9,1 = Private, Level 2 Regional ▪ 9,2 = Private, Level 1 Regional ▪ 9,3 = Private, PISN Specific ▪ 9,4 = Private, Level 0 Regional (local) ▪ 1,0 = Public(ISDN/E.164), Unknown ▪ 1,1 = Public(ISDN/E.164), International ▪ 1,2 = Public(ISDN/E.164), National ▪ 1,3 = Public(ISDN/E.164), Network Specific ▪ 1,4 = Public(ISDN/E.164), Subscriber ▪ 1,6 = Public(ISDN/E.164), Abbreviated For the NI-2 and DMS-100 ISDN variants, the valid combinations of TON and NPI for calling and called numbers are (Plan/Type): ▪ 0/0 - Unknown/Unknown ▪ 1/1 - International number in ISDN/Telephony numbering plan ▪ 1/2 - National number in ISDN/Telephony numbering plan ▪ 1/4 - Subscriber (local) number in ISDN/Telephony numbering plan ▪ 9/4 - Subscriber (local) number in Private numbering plan	
Redirect Number IP -to-Tel Table	
Web: Redirect Number IP -> Tel EMS: Redirect Number Map IP to Tel CLI: config voip > gw manipulations redirect-number-	This parameter table manipulates the redirect number for IP-to-Tel calls. This manipulates the value of the SIP Diversion, History-Info, or Resource-Priority headers (including the reason the call was redirected).

Parameter	Description
map-ip2tel **[RedirectNumberMapIp2Tel]**	The format of this parameter is as follows: [RedirectNumberMapIp2Tel] FORMAT RedirectNumberMapIp2Tel_Index = RedirectNumberMapIp2Tel_DestinationPrefix, RedirectNumberMapIp2Tel_RedirectPrefix, RedirectNumberMapIp2Tel_SourceAddress, RedirectNumberMapIp2Tel_NumberType, RedirectNumberMapIp2Tel_NumberPlan, RedirectNumberMapIp2Tel_RemoveFromLeft, RedirectNumberMapIp2Tel_RemoveFromRight, RedirectNumberMapIp2Tel_LeaveFromRight, RedirectNumberMapIp2Tel_Prefix2Add, RedirectNumberMapIp2Tel_Suffix2Add, RedirectNumberMapIp2Tel_IsPresentationRestricted; [\RedirectNumberMapIp2Tel] For example: RedirectNumberMapIp2Tel 1 = *, 88, *, 1, 1, 2, 0, 255, 9, , 255; **Notes:** ▪ This parameter table can include up to 20 indices (1-20). ▪ If the table's characteristics rule (i.e., DestinationPrefix, RedirectPrefix, and SourceAddress) matches the IP-to-Tel call, then the redirect number manipulation rule (defined by the other parameters) is applied to the call. ▪ The manipulation rules are done in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ The RedirectPrefix parameter is used before any manipulation has been performed on it.
Redirect Number Tel-to-IP Table	
Web: Redirect Number Tel -> IP EMS: Redirect Number Map Tel to IP CLI: config voip > gw manipulations redirect-number-map-tel2ip **[RedirectNumberMapTel2IP]**	This parameter table manipulates the redirect number for Tel-to-IP calls. The manipulated Redirect Number is sent in the SIP Diversion, History-Info, or Resource-Priority headers. The format of this parameter is as follows: [RedirectNumberMapTel2Ip] FORMAT RedirectNumberMapTel2Ip_Index = RedirectNumberMapTel2Ip_DestinationPrefix, RedirectNumberMapTel2Ip_RedirectPrefix, RedirectNumberMapTel2Ip_NumberType, RedirectNumberMapTel2Ip_NumberPlan, RedirectNumberMapTel2Ip_RemoveFromLeft, RedirectNumberMapTel2Ip_RemoveFromRight, RedirectNumberMapTel2Ip_LeaveFromRight, RedirectNumberMapTel2Ip_Prefix2Add, RedirectNumberMapTel2Ip_Suffix2Add, RedirectNumberMapTel2Ip_IsPresentationRestricted, RedirectNumberMapTel2Ip_SrcTrunkGroupID, RedirectNumberMapTel2Ip_SrcIPGroupID; [\RedirectNumberMapTel2Ip] For example: RedirectNumberMapTel2Ip 1 = *, 4, 255, 255, 0, 0, 255, , 972, 255, 1, 2; **Notes:** ▪ This parameter table can include up to 20 indices (1-20).

Parameter	Description
	▪ The manipulation rules are done in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ If the table's matching characteristics rule (i.e., DestinationPrefix, RedirectPrefix, SrcTrunkGroupID, and SrcIPGroupID) is located for the Tel-to-IP call, then the redirect number manipulation rule (defined by the other parameters) is applied to the call. ▪ Redirect number manipulation for Tel-to-IP calls is not performed if the CopyDest2RedirectNumber parameter is enabled. This parameter copies the received destination number to the outgoing redirect number. ▪ The parameters NumberType and NumberPlan are applicable only to the SIP Resource-Priority header.
Phone Context Table	
Web: Phone Context Table EMS: SIP Manipulations > Phone Context CLI: config voip > gw manipulations phone-context-table **[PhoneContext]**	This *para*meter table defines the Phone Context table. This parameter maps NPI and TON to the SIP Phone-Context parameter. When a call is received from the ISDN/Tel, the NPI and TON are compared against the table and the corresponding Phone-Context value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a Phone-Context attribute is received. The Phone-Context parameter appears in the standard SIP headers (Request-URI, To, From, Diversion) where a phone number is used. The format for this parameter is as follows: [PhoneContext] FORMAT PhoneContext_Index = PhoneContext_Npi, PhoneContext_Ton, PhoneContext_Context; [\PhoneContext] For example: PhoneContext 0 = 0,0,unknown.com PhoneContext 1 = 1,1,host.com PhoneContext 2 = 9,1,na.e164.host.com **Notes:** ▪ This parameter can include up to 20 indices. ▪ Several entries with the same NPI-TON or Phone-Context are allowed. In this scenario, a Tel-to-IP call uses the first match. ▪ To configure the Phone Context table using the Web interface, see 'Mapping NPI/TON to SIP Phone-Context' on page 285. ▪ For a description on using *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Web/EMS: Add Phone Context As Prefix CLI: add-ph-cntxt-as-pref **[AddPhoneContextAsPrefix]**	Determines whether the received Phone-Context parameter is added as a prefix to the outgoing ISDN Setup message with (for digital interfaces) Called and Calling numbers. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable.

A.12.17LDAP Parameters

The Lightweight Directory Access Protocol (LDAP) parameters are described in the table below. For more information on routing based on LDAP, refer to 'Routing Based on LDAP Active Directory Queries' on page 189.

Table A-68: LDAP Parameters

Parameter	Description
Web: LDAP Service CLI: enable **[LDAPServiceEnable]**	Enables the LDAP feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: LDAP Server IP **[LDAPServerIP]**	Defines the LDAP server's IP address in dotted-decimal notation (e.g., 192.10.1.255). The default is 0.0.0.0.
Web: LDAP Server Port CLI: server-port **[LDAPServerPort]**	Defines the LDAP server's port number. The valid value range is 0 to 65535. The default port number is 389.
Web: LDAP Server Domain Name CLI: server-domain-name **[LDAPServerDomainName]**	Defines the host name of the LDAP server.
Web: LDAP Password **[LDAPPassword]**	Defines the LDAP server's user password.
Web: LDAP Bind DN CLI: password **[LDAPBindDN]**	Defines the LDAP server's bind DN. This is used as the username during connection and binding to the server. For example: LDAPBindDN = "CN=Search user,OU=Labs,DC=OCSR2,DC=local"
Web: LDAP Search Dn CLI: search-dn **[LDAPSearchDN]**	Defines the search DN for LDAP search requests. This is the top DN of the subtree where the search is performed. This parameter is mandatory for the search. For example: LDAPSearchHDN = "CN=Search user,OU=Labs,DC=OCSR2,DC=local"
Web: LDAP Server Max Respond Time CLI: server-max-respond-time **[LDAPServerMaxRespondTime]**	Defines the time (in seconds) that the device waits for LDAP server responses. The valid value range is 0 to 86400. The default is 3000.
[LDAPDebugMode]	Determines whether to enable the LDAP task debug messages. This is used for providing debug information regarding LDAP tasks. The valid value range is 0 to 3. The default is 0.
Web: MS LDAP OCS Number attribute name CLI: ldap-ocs-nm-attr **[MSLDAPOCSNumAttributeName]**	Defines the name of the attribute that represents the user OCS number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "msRTCSIP-PrimaryUserAddress".
Web: MS LDAP PBX Number attribute name CLI: ldap-pbx-nm-attr **[MSLDAPPBXNumAttributeName]**	Defines the name of the attribute that represents the user PBX number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "telephoneNumber".

Parameter	Description
Web: MS LDAP MOBILE Number attribute name CLI: ldap-mobile-nm-attr **[MSLDAPMobileNumAttributeName]**	Defines the name of the attribute that represents the user Mobile number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "mobile".
CLI: ldap-private-nm-attr **[MSLDAPPrivateNumAttributeName]**	Defines the name of the attribute that represents the user's private number in the AD. If this value equals the value of the MSLDAPPrimaryKey or MSLDAPSecondaryKey parameter, then the <device> queries the AD for the destination number in this private attribute name; otherwise, this parameter is not used as a search key. The default value is "msRTCSIP-PrivateLine".
CLI: ldap-primary-key **[MSLDAPPrimaryKey]**	Defines the name of the attribute used as a query search key for the destination number in the AD. This is used instead of the "PBX" attribute name (configured by the MSLDAPPBXNumAttributeName parameter). The default value is not configured.
CLI: ldap-secondary-key **[MSLDAPSecondaryKey]**	Defines the name of the attribute used as the second query search key for the destination number in the AD, if the primary search key or PBX search is not found.

A.12.18Least Cost Routing Parameters

The Least Cost Routing parameters are described in the table below.

Table A-69: LCR Parameters

Parameter	Description
Web: Routing Rule Groups Table **[RoutingRuleGroups]**	This parameter table enables the LCR feature and configures the average call duration and default call cost. The default call cost determines whether routing rules that are not configured with a Cost Group are considered as a higher or lower cost route compared to other matching routing rules that are assigned Cost Groups. [RoutingRuleGroups] FORMAT RoutingRuleGroups_Index = RoutingRuleGroups_LCREnable, RoutingRuleGroups_LCRAverageCallLength, RoutingRuleGroups_LCRDefaultCost; [\RoutingRuleGroups]
Web: Cost Group Table **[CostGroupTable]**	This parameter table configures the Cost Groups for LCR, where each Cost Group is configured with a name, fixed call connection charge, and a call rate (charge per minute). [CostGroupTable] FORMAT CostGroupTable_Index = CostGroupTable_CostGroupName, CostGroupTable_DefaultConnectionCost, CostGroupTable_DefaultMinuteCost; [\CostGroupTable] For example: CostGroupTable 2 = "Local Calls", 2, 1;

Parameter	Description
Web: Cost Group > Time Band Table **[CostGroupTimebands]**	This parameter table configures time bands and associates them with Cost Groups [CostGroupTimebands] FORMAT CostGroupTimebands_TimebandIndex = CostGroupTimebands_StartTime, CostGroupTimebands_EndTime, CostGroupTimebands_ConnectionCost, CostGroupTimebands_MinuteCost; [\CostGroupTimebands]

A.13 SBC Parameters

The SBC parameters are described in the table below.

Table A-70: SBC Parameters

Parameter	Description
Web: Enable SBC EMS: Enable SBC CLI: enable-sbc **[EnableSBCApplication]**	Enables the Session Border Control (SBC) application. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ In addition to enabling this parameter, the number of maximum SBC/IP-to-IP sessions must be defined in the Software Upgrade Key.
WAN Interface Name **[WanInterfaceName]**	Defines the WAN interface for the VoIP interface. The available interface options depends on the hardware configuration (e.g., Ethernet, T1, or SHDSL) and/or whether VLANs are defined for the WAN interface (see Virtual LAN Interface (VLAN) on page 646). The value must be enclosed in single quotation marks ('...'), for example, WanInterfaceName = 'GigabitEthernet 0/0'. This WAN interface can be assigned to SIP signaling and/or media interfaces, in the SIP Interface table, where it is represented as "WAN" (see Configuring SIP Interface Table on page 210). If VLANs are configured, for example, for the Ethernet WAN interface (see Data Settings on page 565), then you can select the WAN VLAN on which you want to run these SIP signaling and/or media interfaces. Therefore, for each outgoing SIP packet, the device sends it on the defined outgoing WAN interface; for each incoming SIP packet, the device identifies the packet according to the WAN interface from where it is received. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only if the data-routing functionality is supported (i.e., relevant Software Upgrade Feature Key is installed on the device).
Web: Allow Unclassified Calls CLI: unclassified-calls **[AllowUnclassifiedCalls]**	Determines whether calls (incoming packets) that cannot be classified (i.e. classification process fails) into a Source IP Group (in the Classification table) are either rejected or processed. ▪ **[0]** Reject = the call is rejected if classification fails. ▪ **[1]** Allow = if classification fails, the incoming packet is assigned to the default IP Group of the default SRD (and the call is subsequently processed). (Default.)
Web: SBC No Answer Timeout CLI: sbc-no-arelt-timeout **[SBCAlertTimeout]**	Defines the timeout (in seconds) for SBC outgoing (outbound IP routing) SIP INVITE messages. If the called IP party does not answer the call within this user-defined interval, the device disconnects the session. The device starts the timeout count upon receipt of a SIP 180 Ringing response from the called party. If no other SIP response (for example, 200 OK) is received thereafter within this timeout, the call is released. The valid range is 0 to 3600 seconds. the default is 600.

Parameter	Description
Web: SBC Max Forwards Limit **[SBCMaxForwardsLimit]**	Defines the Max-Forwards SIP header value. The Max-Forwards header is used to limit the number of servers (such as proxies) that can forward the SIP request. The Max-Forwards value indicates the remaining number of times this request message is allowed to be forwarded. This count is decremented by each server that forwards the request. This parameter affects the Max-Forwards header in the received message as follows: ▪ If the received header's original value is 0, the message is not passed on and is rejected. ▪ If the received header's original value is less than this parameter's value, the header's value is decremented before being sent on. ▪ If the received header's original value is greater than the parameter's value, the header's value is replaced by the user-defined parameter's value. The valid value range is 1-70. The default is 10.
Web: Minimum Session-Expires CLI: min-session-expires **[SBCMinSE]**	Defines the minimum amount of time (in seconds) between session refresh requests in a dialog before the session is considered timed out. This value is conveyed in the SIP Min-SE header. The valid range is 0 (default) to 1,000,000 (where 0 means that the device does not limit Session-Expires).
Web/EMS: Handle P-Asserted-Identity CLI: p-assert-id **[SBCAssertIdentity]**	Determines the device's privacy handling of the P-Asserted-Identity header. This indicates how the outgoing SIP message asserts identity. ▪ **[0]** Don't Care (default) = P-Asserted Identity header is not affected. ▪ **[1]** Add P-Asserted-Identity Header = Adds a P-Asserted-Identity header. The header's values are taken from the source URL. ▪ **[2]** Remove P-Asserted-Identity Header = Removes the P-Asserted-Identity header. **Notes:** ▪ This parameter affects only the initial INVITE request. ▪ The configuration of privacy handling in the IP Group table takes precedence over the settings of this global parameter. ✓ If in the IP Group this parameter is set to 'Don't care', then the settings of this global parameter is used. ✓ If this global parameter and the IP Group are set to 'Don't care', the device uses the same P-Asserted-Identity header (if present) in the incoming message for the outgoing message. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: Keep original user in Register **[SBCKeepContactUserinRegister]**	Determines whether the device replaces the Contact user with a unique Contact user in the outgoing message in response to a REGISTER request. ▪ **[0]** Disable = (default) The device replaces the original Contact user with a unique Contact user, for example: ✓ Received Contact: <sip:123@domain.com> ✓ Outgoing (unique) Contact: <sip:FEU1_7_1@SBC> ▪ **[1]** Enable = The original Contact user is retained and used in

Parameter	Description
	the outgoing REGISTER request. **Note:** This parameter is applicable only to REGISTER messages received from USER IP Groups and that are sent to SERVER IP Groups.
CLI: sbc-refer-bhvr **[SBCReferBehavior]**	Determines the device's handling of REFER requests. ▪ **[0]** = Refer-To header is unchanged (default). ▪ **[1]** = Uses the database for Refer-To as described below. When enabled, the device handles REFERs as follows: 1 Before passing on the REFER request, the device changes the host part to the device's IP address and adds a special prefix ("T~&R_") to the Contact user part. 2 The incoming INVITE is identified as a REFER-resultant INVITE according to the special prefix. 3 The device replaces the host part (in the Request-URI) with the host from the REFER contact. The prefix ("T~&R_") remains in the user part for regular classification, manipulation, and routing. The special prefix can be used for specific routing rules for REFER-resultant INVITES. 4 The prefix is removed before the resultant INVITE is sent to the destination.
CLI: sbc-xfer-prefix **[SBCXferPrefix]**	When the SBCReferBehavior is set to 1, the device, while interworking the SIP REFER message, adds the prefix "T~&R-" to the user part of the URI in the Refer-To header. After this, the device can receive an INVITE with such a prefix (the INVITE is sent by the UA that receives the REFER message or 302 response). If the device receives an INVITE with such a prefix, it replaces the prefix with the value defined for the SBCXferPrefix parameter. The default value is empty (""). **Note:** This feature is also applicable to 3xx redirect responses. The device adds the prefix "T~&R-" to the URI user part in the Contact header if the SBC3xxBehavior parameter is set to 1.
CLI: sbc-3xx-bhvt **[SBC3xxBehavior]**	Determines the device's handling of SIP 3xx responses. When enabled, the device handles SIP redirections between different subnets (e.g., between LAN and WAN sides). This is required where the new address provided by the redirector (Redirect sever) may not be reachable by the far-end user (FEU) located in another subnet. For example, a far-end user (FEU) in the WAN sends a SIP request via the device to a Redirect server in the LAN, and the Redirect server replies with a SIP 3xx response to a PBX in the LAN in the Contact header. If the device sends this response as is (i.e., with the original Contact header), the FEU is unable to reach the new destination. ▪ **[0]** (default) = The device sends the received SIP 3xx response without changing the Contact header (transparent handling). ▪ **[1]** = The device changes the URI in the Contact header of the received SIP 3xx response to its own URI and adds a special user prefix ("T~&R_"), which is then sent to the FEU. The FEU then sends a new INVITE to the device, which the device then sends to the correct destination. **Notes:** ▪ When this parameter is changed from 1 to 0, new 3xx Contact

Parameter	Description
	headers remain unchanged. However, requests with the special prefix continue using the device's database to locate the new destination. ▪ Only one database entry is supported for the same host, port, and transport combination. For example, the following URLs cannot be distinguished by the device: ✓ sip:10.10.10.10:5060;transport=tcp;param=a ✓ sip:10.10.10.10:5060;transport=tcp;param=b ▪ The database entry expires two hours after the last use. ▪ The maximum number of destinations (i.e., database entries) is 50. ▪ For more information on SIP 3xx Redirect response handling, see 'Handling SIP 3xx Redirect Responses' on page 435.
Web: Registration Mode **[IPGroup_RegistrationMode]**	Defines the registration mode for an IP Group: ▪ **[0]** User initiates registrations (default) ▪ **[1]** SBC initiate registrations (works only with User Info file) = Used when the device serves as a client (e.g., with an IP PBX) ▪ **[2]** No registrations needed = The device adds users to its database in active state. **Note:** This is an IP Group table parameter.
Web: Authentication Mode **[PGroup_AuthenticationMode]**	Defines the authentication mode: ▪ **[0]** User Authenticates (default) = The device does not handle the authentication, but simply passes the authentication messages between the SIP user agents. ▪ **[1]** SBC Authenticates (as client) = The device authenticates as a client. It receives the 401/407 response from the proxy requesting for authentication. The device sends the proxy the authorization credentials (i.e., user name and password) according to one of the following: 1) account defined in the Account table (only if authenticating SERVER-type IP Group), 2) global username and password parameters (only if authenticating SERVER-type IP Group), 3) User Information file, or 4) sends request to users requesting credentials (only if authenticating USER-type IP Group). ▪ **[2]** SBC Authenticates (as server) = The device authenticates as a server (using the User Information file). **Note:** This is an IP Group table parameter.
Web: Authentication Method List **[IPGroup_MethodList]**	Defines the SIP methods that the device must challenge. Multiple entries are separated by the forward slash "/". If none are defined (default), no methods are challenged. **Note:** This is an IP Group table parameter.
Web: Lifetime of the nonce in seconds CLI: lifetime-of-nonce **[AuthNonceDuration]**	Defines the lifetime (in seconds) that the current nonce is valid for server-based authentication. The device challenges a message that attempts to use a server nonce beyond this period. This parameter is used to provide replay protection (i.e., ensures that old communication streams are not used in replay attacks). The valid value range is 30 to 600. The default value is 300.

Parameter	Description
Web: Authentication Challenge Method CLI: auth-chlng-mthd **[AuthChallengeMethod]**	Defines the type of server-based authentication challenge. ▪ **[0]** 0 = Send SIP 401 "Unauthorized" with a WWW-Authenticate header as the authentication challenge response. (default) ▪ **[1]** 1 = Send SIP 407 "Proxy Authentication Required" with a Proxy-Authenticate header as the authentication challenge response.
Web: Authentication Quality of Protection CLI: auth-qop **[AuthQOP]**	Defines the authentication and integrity level of quality of protection (QOP) for digest authentication offered to the client. When the device challenges a SIP request (e.g., INVITE), it sends a SIP 401 response with the Authorization header containing the 'qop' parameter indicating the QoP level of the message to be authenticated. In response, the SBC client needs to send the device another INVITE with the MD5 hash of the INVITE message and indicates its auth or auth-int support. ▪ **[0]** Auth = The device sends 'qop=auth' in the SIP response, requesting authentication (i.e., validates user by checking user name and password). This option does not authenticate the message body (i.e., SDP). ▪ **[1]** auth-int = = The device sends 'qop=auth-int' in the SIP response, indicating required authentication and authentication with integrity (e.g., checksum). This option restricts the client to authenticating the entire SIP message, including the body, if present. ▪ **[2]** Auth-Int and Auth = The device sends 'qop=auth, auth-int' in the SIP response, indicating either authentication or integrity (default). This enables the client to choose auth or auth-int. If the client chooses auth-int, the body is included in the authentication. If the client chooses auth, then the body is not authenticated. (default)
Web: SBC User Registration Time CLI: sbc-usr-reg-time **[SBCUserRegistrationTime]**	Defines the duration (in seconds) of the periodic registrations between the user and the device (the device responds with this value to the user). When set to 0, the device does not change the Expires header's value received in the user's REGISTER request. If no Expires header is received in the REGISTER message and the SBCUserRegistrationTime parameter is set to 0, then by default, the Expires header's value is set to 180 seconds. The valid range is 0 to 2,000,000 seconds. The default is 0. **Note:** For this parameter to take effect, a device reset is required.
Web: SBC Proxy Registration Time CLI: sbc-prxy-reg-time **[SBCProxyRegistrationTime]**	Defines the duration (in seconds) for which the user is registered in the proxy database (after the device forwards the REGISTER message). When set to 0, the device sends the Expires header's value as received from the user to the proxy. The valid range is 0 to 2,000,000 seconds. The default is 0.
Web: SBC Survivability Registration Time CLI: sbc-surv-reg-time **[SBCSurvivabilityRegistrationTime]**	Defines the duration of the periodic registrations between the user and the device, when the device is in survivability state (i.e., when REGISTER requests cannot be forwarded to the proxy and are terminated by the device). When set to 0, the device uses the value set by the SBCUserRegistrationTime parameter for the device's response. The valid range is 0 to 2,000,000 seconds. The default is 0.
[SBCEnableAASTRASurvivab	Enables the device to notify Aastra IP phones that it is currently

Parameter	Description

ilityNotice]	operating in Survivability mode. When this occurs, the Aastra IP phones display the message, "Stand Alone Mode" on their LCD screens. Survivability mode occurs when connectivity with the WAN fails and as a result, the device enables communication between IP phone users within the LAN enterprise.
	▪ **[0]** = Disable
	▪ **[1]** = Enable
	When this feature is enabled and the SBC device is in Survivability mode, it responds to SIP REGISTER messages from the IP phones with a SIP 200 OK containing the following XML body:
	`Content-Type: application/xml`
	`<?xml version="1.0" encoding="utf-8"?>`
	`<LMIDocument version="1.0">`
	`<LocalModeStatus>`
	`<LocalModeActive>true</LocalModeActive>`
	`<LocalModeDisplay>StandAlone Mode</LocalModeDisplay>`
	`</LocalModeStatus>`
	`</LMIDocument>`

Web: SBC GRUU Mode CLI: sbc-gruu-mode **[SBCGruuMode]**	Determines the Globally Routable User Agent (UA) URI (GRUU) support, according to RFC 5627. ▪ **[0]** None = No GRUU is supplied to users. ▪ **[1]** As Proxy = The device provides same GRUU types as the proxy provided the device's GRUU clients. (default) ▪ **[2]** Temporary only = Supply only temporary GRUU to users. (Currently not supported.) ▪ **[3]** Public only = The device provides only public GRUU to users. ▪ **[4]** Both = The device provides temporary and public GRUU to users. (Currently not supported.) This parameter allows the device to act as a GRUU server for its SIP UA clients, providing them with public GRUU's, according to RFC 5627. The public GRUU provided to the client is denoted in the SIP Contact header parameters, "pub-gruu". Public GRUU remains the same over registration expirations. On the other SBC leg communicating with the Proxy/Registrar, the device acts as a GRUU client. The device creates a GRUU value for each of its registered clients, which is mapped to the GRUU value received from the Proxy server. In other words, the created GRUU value is only used between the device and its clients (endpoints). `Public-GRUU: sip:userA@domain.com;gr=unique-id`
Enable Bye Authentication **[SBCEnableByeAuthentication]**	Enables authenticating a SIP BYE request before disconnecting the call. This feature prevents, for example, a scenario in which the SBC SIP client receives a BYE request from a third-party imposer assuming the identity of a participant in the call and as a consequence, the call between the first and second parties is inappropriately disconnected. ▪ **[0]** Disable (default) ▪ **[1]** Enable = The device forwards the SIP authentication response (for the BYE request) to the request sender and waits for the user to authenticate it. The call is disconnected only if the

Parameter	Description
	authenticating server responds with a 200 OK.
[SBCExtensionsProvisioningMode]	Enables SBC user registration for interoperability with BroadSoft's BroadWorks server, to provide call survivability in case of connectivity failure with the BroadWorks server. ▪ **[0]** = Normal processing of REGISTER messages (default). ▪ **[1]** = Registration method for BroadWorks server. In a failure scenario with BroadWorks, the device acts as a backup SIP proxy server, maintaining call continuity between the enterprise LAN users (subscribers) and between the subscribers and the PSTN (if provided). **Note:** For a detailed description of this feature, see 'Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server' on page 438.
Web: SBC Direct Media **[SBCDirectMedia]**	Enables the No Media Anchoring feature (i.e., direct media) for all SBC calls. No Media Anchoring uses SIP signaling capabilities without handling the RTP/SRTP (media) flow between remote SIP user agents (UA). The RTP packets do not traverse the device, instead, the two SIP UAs establish a direct RTP/SRTP flow between one another. Signaling continues to traverse the device with minimal intermediation and involvement to enable certain SBC abilities such as routing ▪ **[0]** Disable = All SRD calls via SBC are not direct media - internal SRD calls are according to SRD configuration (default). ▪ **[1]** Enable = All SBC calls use the No Media Anchoring feature (i.e., direct media). **Notes:** ▪ For more information on No Media Anchoring, see 'No Media Anchoring' on page 428. ▪ When No Media Anchoring is enabled: ✓ Manipulation is not done on SDP data (offer/answer transaction) such as ports and IP addresses. ✓ Opening voice channels and allocation of IP media ports are not required. ✓ Forced Transcoding and Extension Coders features are disabled for No Media Anchoring calls. ✓ The Coder Restriction feature (Allowed Coders List) operates simultaneously with No Media Anchoring calls. Restricted coders are removed from the SDP offer message. ▪ No Media Anchoring is typically implemented in the following scenarios: ✓ SBC device is located in the LAN. ✓ Calls between two SIP UAs in the same LAN and signals are sent to a SIP proxy server that is located in the WAN. ✓ SBC device does not do NAT traversal (for media) and all the users are in the same domain. ▪ The benefits of implementing the No Media Anchoring feature includes the following: saves network bandwidth, reduces CPU usage (no RTP/SRTP handling), and avoids interference in SDP negotiation and header manipulation on RTP/SRTP. ▪ The process for handling the No Media Anchoring feature is as follows:

Parameter	Description
	✓ Identifying a No Media Anchoring call according to configuration and the call's properties (such as source, destination, IP Group, and SRD). ✓ Handing the identified No Media Anchoring call. ▪ You can enable No Media Anchoring per SRD (using the IntraSRDMediaAnchoring parameter), whereby calls between two UAs that pertain to the same SRD (source and destination) are handled as a No Media Anchoring (direct media) call. ▪ Chosen configuration can't handle call from any UA to a foreign UA (vice versa) but both UAs belong to the same SRD and the parameter IntraSRDMediaAnchoring for that specific SRD is > 0. ▪ When this parameter is disabled, No Media Anchoring calls between two UAs that belong to separate SRDs cannot be configured. No Media Anchoring calls between two UAs that belong to the same SRD is configurable only (in this case).
Web: Transcoding Mode CLI: transcoding-mode **[TranscodingMode]**	Defines the voice transcoding mode (media negotiation) between the two SBC legs for the SBC application. ▪ **[0]** Only if Required = Do not force transcoding. Many of the media settings (such as gain control) are not implemented on the voice stream. The SBC application passes packets RTP to RTP without any processing (default). ▪ **[1]** Force = Forces transcoding on the outgoing SBC leg. The device's SBC application interworks the media by implementing DSP transcoding. **Note:** This parameter can also be defined per IP Profile, using the IPProfile parameter (see Configuring IP Profiles on page 248).
CLI: media-sec-bhvior **[IpProfile_SBCMediaSecurityBehaviour]**	Determines the transcoding method between SRTP and RTP. The device can also enforce SBC legs to use SRTP\RTP, using the IP Profile parameter SBCMediaSecurityBehaviour: ▪ **[0]** As is (default): no special handling for RTP\SRTP is done. ▪ **[1]** SRTP: SBC legs negotiate only SRTP media lines, and RTP media lines are removed from the incoming SDP offer\answer. ▪ **[2]** RTP: SBC legs negotiate only RTP media lines, and SRTP media lines are removed from the incoming offer\answer. ▪ **[3]** Both: each offer\answer is extended (if not already) to two media lines - one RTP and the other SRTP. If two SBC legs (after offer\answer negotiation) use different security types (i.e., one RTP and the other SRTP), then the device performs RTP-SRTP transcoding. To transcode between RTP and SRTP, the following prerequisites must be met: ▪ At least one supported SDP "crypto" attribute and parameters ▪ EnableMediaSecurity must be set to 1 If one of the above transcoding prerequisites is not met, then: ▪ any value other than "As is" is discarded. ▪ if the incoming offer is SRTP, force transcoding, coder transcoding, and DTMF extensions are not applied. Transcoding between RTP and SRTP requires alloctaion of one DSP. SRTP-to-SRTP transcoding does not require any DSP allocation. **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page

Parameter	Description
	248).
[IpProfile_SBCRFC2833Behavior]	Determines the RFC 2833 SDP offer\answer negotiation. ▪ **[0]** As is = The device does not intervene in the RFC 2833 negotiation. (default) ▪ **[1]** Extend = Each outgoing offer\answer includes RFC 2833 in the offered SDP (the device adds RFC 2833 only if the incoming offer does not include RFC 2833). ▪ **[2]** Disallow = The device removes RFC 2833 from the incoming offer. **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see Configuring IP Profiles on page 248).
[IpProfile_SBCAlternativeDTMFMethod]	The device's first priority for DTMF method at each leg is RFC 2833. Therefore, if a specific leg negotiates RFC 2833 successfully, then the chosen DTMF method for this leg is RFC 2833. For legs where RFC 2833 is not negotiated successfully, the device uses this parameter to determine the chosen DTMF method for the leg. ▪ **[0]** = Don't care - the device does not attempt to interwork any special DTMF method. (default) ▪ **[1]** = In Band ▪ **[2]** = INFO, Cisco ▪ **[3]** = INFO, Nortel ▪ **[4]** = INFO, Korea **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see Configuring IP Profiles on page 248).
Web: Diversion Mode **[IpProfile_SBCDiversionMode]**	Determines the device's handling of the SIP Diversion header. For more information on interworking of the History-Info and Diversion headers, see 'Interworking SIP Diversion and History-Info Headers' on page 436. ▪ **[0]** Don't Care = Diversion header is not handled. (default) ▪ **[1]** Add = History-Info header converted to a Diversion header. ▪ **[2]** Remove = Removes the Diversion header and the conversion to the History-Info header depends on the settings of the SBCHistoryInfoMode parameter. **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: History Info Mode **[IpProfile_SBCHistoryInfoMode]**	Determines the device's handling of the History-Info header. For more information on interworking of the History-Info and Diversion headers, see 'Interworking SIP Diversion and History-Info Headers' on page 436. ▪ **[0]** Don't Care = History-Info header is not handled. (default) ▪ **[1]** Add = Diversion header converted to a History-Info header. ▪ **[2]** Remove = History-Info header removed from the SIP dialog and the conversion to the Diversion header depends on the settings of the SBCDiversionMode parameter. **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page

Parameter	Description
	248).
Web: Extension Coders Group ID **[IpProfile_SBCExtensionCodersGroupID]**	Defines the Coder Group ID used for extended (additional) coders per IP Profile. Extended coders are added only on the outgoing leg. This is used when transcoding is required between two user agents (i.e., the SDP answer from one user agent doesn't include any coder included in the offer previously sent by the other user agent). Therefore, to allow user agents of different IP Groups to communicate with each other (regardless of their capabilities), an extended coders table with at least one coder that is supported by each IP Groups' user agents needs to be assigned to each IP Group. Therefore, each offer destined to specific IP Groups includes this coder. **Notes:** ▪ This parameter can be configured only as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248). ▪ To configure Coders Groups, see 'Configuring Coder Groups' on page 244.
Web: Allowed Coders Mode **[IpProfile_SBCAllowedCodersMode]**	Determines the mode of the Allowed Coders feature for both SBC legs. ▪ **[0]** Restriction = In the incoming SDP offer, the device uses only coders that are also listed in the Allowed Coders Group; the rest are removed from the SDP offer (i.e., only coders common between SDP offered coders and Allowed Coders Group are used). If an Extension Coders Group is also selected (using the IP Profile's SBCExtensionCodersGroupID parameter), then these coders are added to the SDP offer. ▪ **[1]** Preference = The device re-arranges the priority (order) of the coders in the incoming SDP offer according to their order of appearance in the Allowed Coders Group list. (This option also retains all the coders received in the SDP offer.) ▪ **[2]** Restriction and Preference = Performs both Restriction and Preference. **Notes:** ▪ If the AllowedCodersGroup parameter is set to None, then this parameter is not applicable. ▪ This parameter can only be configured as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248). ▪ To select the Allowed Coders Group ID, use the AllowedCodersGroup parameter. ▪ To select the Extension Coders Group ID, use the CodersGroups parameter. ▪ For more information on the Allowed Coders feature, see 'Coder Restrictions Control' on page 430.
SBC Preferences Mode **[SBCPreferencesMode]**	Determines the order of the Extension coders (coders added if there are no common coders between SDP offered coders and Allowed coders) and Allowed coders (defined in the Allowed Coders Group table) in the outgoing SIP message (in the SDP). ▪ **[0]** Doesn't Include Extensions = Extension coders are added at the end of the coder list (default). ▪ **[1]** Include Extensions = Extension coders and Allowed coders are arranged according to their order of appearance in the

Parameter	Description
	Allowed Coders Group table. **Note:** If the SBCExtensionCodersGroupID parameter of the IP Profile table is set to **None**, then this parameter is not applicable.
SBC Fax Coders Group ID **[IpProfile_SBCFaxCodersGroupID]**	Selects the supported fax coders (Coders Group ID) for fax negotiation. Coders Groups are configured in the Coders Group Settings table. **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
SBC Fax Behavior **[IpProfile_SBCFaxBehavior]**	Defines the negotiation method for fax offer: ▪ **[0]** = Pass fax transparently, without interference (default). ▪ **[1]** = Handle fax according to fax settings in the IP Profile for all offer-answer transactions (including the initial INVITE). ▪ **[2]** = Handle fax according to fax settings in the IP Profile for all re-INVITE offer-answer transactions (except for initial INVITE). **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
SBC Fax Offer Mode **[IpProfile_SBCFaxOfferMode]**	Defines the coders included in the outgoing SDP offer (sent to the called "fax"). ▪ **[0]** All = Use only (and all) the coders of the selected Coders Group ID configured using the SBCFaxCodersGroupID parameter. (Default) ▪ **[1]** Single = Use only one coder. If a coder in the incoming offer (from the calling "fax") matches a coder in the SBCFaxCodersGroupID, then the device uses this coder. If no match exists, then the device uses the first coder listed in the Coders Group ID (SBCFaxCodersGroupID). **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
SBC Fax Answer Mode **[IpProfile_SBCFaxAnswerMode]**	Defines the coders included in the outgoing SDP answer (sent to the calling "fax"). ▪ **[0]** All = Use matched coders between the incoming offer coders (from the calling "fax") and the coders of the selected Coders Group ID (configured using the SBCFaxCodersGroupID parameter). ▪ **[1]** Single = Use only one coder. If the incoming answer (from the called "fax") includes a coder that matches a coder match between the incoming offer coders (from the calling "fax") and the coders of the selected Coders Group ID (SBCFaxCodersGroupID, then the device uses this coder. If no match exists, the device uses the first listed coder of the matched coders between the incoming offer coders (from the calling "fax") and the coders of the selected Coders Group ID. (Default) **Note:** This parameter can only be configured as an IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 248).
Web: Enable SBC Client Forking	Enables call forking for USER-type IP Groups.

Parameter	Description
[IPGroup_EnableSBCClientForking]	▪ **[0]** No (default) ▪ **[1]** Yes = The device forks INVITE messages (to up to five separate SIP outgoing legs). **Note:** This parameter can only be configured for an IP Group, using the IPGroup parameter.
[EnableSBCMediaSync]	Enables SBC media synchronization process for calls established from SIP forking that is initiated by external proxy servers. It is possible that a call is established with the media not synchronized between the SBC legs. Media synchronization resolves this issue. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Admission Control Table	
Web: Admission Control EMS: Call Admission Control CLI: configure voip > sbc sbc-admission-control **[SBCAdmissionControl]**	This *parameter* table defines limitations on the number of allowed concurrent calls (SIP dialogs). This is useful for controlling bandwidth utilization between Voice and Data traffic. The format of this parameter is as follows: [SBCAdmissionControl] FORMAT SBCAdmissionControl_Index = SBCAdmissionControl_LimitType, SBCAdmissionControl_IPGroupID, SBCAdmissionControl_SRDID, SBCAdmissionControl_RequestType, SBCAdmissionControl_RequestDirection, SBCAdmissionControl_Limit, SBCAdmissionControl_LimitPerUser, SBCAdmissionControl_Rate, SBCAdmissionControl_MaxBurst; [\SBCAdmissionControl] For example, the below configuration allows a maximum of 10 concurrent SIP INVITEs for IP Group 1: SBCAdmissionControl 1 = 0, 1, -1, 1, 0, 10, -1, 0, 0; **Notes:** ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, see 'Configuring Admission Control' on page 446. ▪ For a description on configuring ini file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Allowed Audio Coders Table	
Web: Allowed Audio Coders CLI: configure voip > sbc allowed-coders-group AllowedCodersGroup0 **[AllowedCodersGroup0]** **[AllowedCodersGroup1]** **[AllowedCodersGroup2]** **[AllowedCodersGroup3]** **[AllowedCodersGroup4]**	This *parameter* table allows you to define up to 5 Allowed Coders Groups, each with up to 10 coders. The Allowed Coders Group determines the coders that can be used for a specific SBC leg. Coders excluded from the Allowed Coders Group are removed from the SDP offer (only coders common between SDP offered coders and Allowed Coders are used). In addition, coders defined in top entries in the Allowed Coders Group are assigned higher priority than those entered in lower entries. [AllowedCodersGroupx] FORMAT AllowedCodersGroup_Index = AllowedCodersGroup_Name; [\AllowedCodersGroup] Where, ▪ AllowedCodersGroupx = Allowed Coders Group index (0-4). ▪ Index = Coder index number per group (0-9). ▪ Name = Coder name. For supported coders, see the

Parameter	Description
	CodersGroup parameter. For example, below represents two configured Allowed Coders Groups, 0 and 1. Group 0 has two coders; Group 1 has one coder. The highest priority coder is G.723.1. [AllowedCodersGroup0] FORMAT AllowedCodersGroup0_Index = AllowedCodersGroup0_Name; AllowedCodersGroup0 0 = g7231; AllowedCodersGroup0 1 = g711Alaw64k; [\AllowedCodersGroup0] [AllowedCodersGroup1] FORMAT AllowedCodersGroup1_Index = AllowedCodersGroup0_Name; AllowedCodersGroup1 0 = g711Ulaw64k; [\AllowedCodersGroup1] **Notes:** ▪ The Allowed Coders table is performed on audio media only. ▪ Allowed Coder Groups can be assigned to IP Profiles (see 'Configuring IP Profiles' on page 248). ▪ For configuring the table using the Web interface, see 'Configuring Allowed Coder Groups' on page 448. ▪ For a description on configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Message Policy Table	
Web: Message Policy Table CLI: configure voip > sbc message-policy **[MessagePolicy]**	This *para*meter table configures SIP message policy rules. The format of this parameter is as follows: [MessagePolicy] FORMAT MessagePolicy_Index = MessagePolicy_Policy, MessagePolicy_MaxMessageLength, MessagePolicy_MaxHeaderLength, MessagePolicy_MaxBodyLength, MessagePolicy_MaxNumHeaders, MessagePolicy_MaxNumBodies, MessagePolicy_SendRejection, MessagePolicy_MethodListType, MessagePolicy_MethodList, MessagePolicy_BodyListType, MessagePolicy_BodyList; [/MessagePolicy]
Classification Table	
Web: Classification Table EMS: SBC Classification CLI: configure voip > sbc routing classification **[Classification]**	This *para*meter table configures the Classification table. This table classifies the incoming SIP INVITE to a Source IP Group. The format of this parameter is as follows: [Classification] FORMAT Classification_Index = Classification_MessageCondition, Classification_SrcSRDID, Classification_SrcAddress, Classification_SrcPort, Classification_SrcTransportType, Classification_SrcUsernamePrefix, Classification_SrcHost, Classification_DestUsernamePrefix, Classification_DestHost, Classification_ActionType, Classification_SrcIPGroupID; [\Classification] For example: Classification 1 = 1, , 10.8.6.15, 5060, 2, *, *, *, *, 1, 4;

Parameter	Description
	Notes: ▪ This table can include up to 20 indices (where 0 is the first index). ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, see 'Configuring Classification Table' on page 452. ▪ For a description on configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
Condition Table	
Web: Condition Table CLI: configure voip > sbc routing condition-table **[ConditionTable]**	Configures Conditions for SIP messages and supports the same syntax used in the SIP Message Manipulation table. These Condition rules are later assigned to Classification rules in the Classification table for enhancing the process for classifying an incoming SIP dialog to an IP Group. [ConditionTable] FORMAT ConditionTable_Index = ConditionTable_Condition, ConditionTable_Description; [\ConditionTable]
SBC IP-to-IP Routing Table	
Web: IP2IP Routing Table EMS: IP to IP Routing CLI: configure voip > sbc routing ip2ip-routing **[IP2IPRouting]**	This *para*meter table configures the SBC IP-to-IP Routing table for routing received SIP messages such as INVITE messages to an IP destination. The format of this parameter is as follows: [IP2IPRouting] FORMAT IP2IPRouting_Index = IP2IPRouting_SrcIPGroupID, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_RequestType, IP2IPRouting_MessageCondition, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupID, IP2IPRouting_DestSRDID, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions, IP2IPRouting_CostGroup; [\IP2IPRouting] For example: IP2IPRouting 1 = 1, *, *, *, *, 3, 0, -1, -1, , 0, -1, 0,,; **Notes:** ▪ This table can include up to 120 indices (where 0 is the first index). ▪ For a specific routing rule to be effective, the matching characteristics must match. If no matching rule is located, the call is rejected. ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, see 'Configuring SBC IP2IP Routing' on page 457. ▪ For a description on configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
SBC Alternative Routing Reasons Table	
Web: SBC Alternative Routing Reasons EMS: Alternative Routing Reasons CLI: configure voip > sbc	This *para*meter table configures the SBC Alternative Routing Reasons table. This table is used for alternative IP-to-IP routing (defined in the IP2IP Routing table). If 4xx, 5xx, or 6xx SIP responses are received as a result of outgoing SIP dialog-initiating methods (e.g., INVITE, OPTIONS, and SUBSCRIBE messages),

Parameter	Description
routing sbc-alternative-routing-reasons **[SBCAlternativeRoutingReasons]**	the device re-sends the messages (to an alternative route) if the response is defined in this table and if there are alternative routes configured in the IP2IP Routing table. The format of this parameter is as follows: [SBCAlternativeRoutingReasons] FORMAT SBCAlternativeRoutingReasons_Index = SBCAlternativeRoutingReasons_ReleaseCause; [\SBCAlternativeRoutingReasons] For example: SBCAlternaiveRoutingReasons 0 = 403; SBCAlternativeRoutingReasons 1 = 404; **Notes:** ▪ This table can include up to five indices (where 0 is the first index). ▪ For a description on configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.
IP to IP Inbound Manipulation Table	
Web: IP to IP Inbound Manipulation EMS: IP to IP Inbound Manipulation CLI: configure voip > sbc manipulations ip-inbound-manipulation **[IPInboundManipulation]**	This *para*meter table configures the IP to IP Inbound Manipulation table. This table allows you to manipulate the SIP URI user part (source and/or destination) of the inbound SIP dialog message. The format of this parameter is as follows: **[IPInboundManipulation]** FORMAT IPInboundManipulation_Index = IPInboundManipulation_IsAdditionalManipulation, IPInboundManipulation_ManipulatedURI, IPInboundManipulation_ManipulationPurpose, IPInboundManipulation_SrcIPGroupID, IPInboundManipulation_SrcUsernamePrefix, IPInboundManipulation_SrcHost, IPInboundManipulation_DestUsernamePrefix, IPInboundManipulation_DestHost, IPInboundManipulation_RequestType, IPInboundManipulation_RemoveFromLeft, IPInboundManipulation_RemoveFromRight, IPInboundManipulation_LeaveFromRight, IPInboundManipulation_Prefix2Add, IPInboundManipulation_Suffix2Add; **[\IPInboundManipulation**] For example: IPInboundManipulation 1 = 0, 0, 0, -1, *, abc, *, *, 0, 0, 0, 255, , ; **Notes:** ▪ This table can include up to 100 indices. ▪ For SIP URI host name (source and destination) manipulations, you can also use the IP Group table. These host names are simply replaced with the names configured for the Source and Destination IP Groups respectively. ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, see 'Configuring IP-to-IP Inbound Manipulations' on page 467. ▪ For a description on configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

Parameter	Description
IP to IP Outbound Manipulation Table	
Web: IP to IP Outbound Manipulation EMS: IP to IP Outbound Manipulation CLI: configure voip > sbc manipulations ip-outbound-manipulation **[IPOutboundManipulation]**	This *parameter* table configures the IP to IP Outbound Manipulation table. This table allows you to manipulate the SIP URI user part (source and/or destination) of the outbound SIP dialog message. The format of this parameter is as follows: [IPOutboundManipulation] FORMAT IPOutboundManipulation_Index = IPOutboundManipulation_IsAdditionalManipulation, IPOutboundManipulation_ManipulatedURI, IPOutboundManipulation_SrcIPGroupID, IPOutboundManipulation_DestIPGroupID, IPOutboundManipulation_SrcUsernamePrefix, IPOutboundManipulation_SrcHost, IPOutboundManipulation_DestUsernamePrefix, IPOutboundManipulation_DestHost, IPOutboundManipulation_RequestType, IPOutboundManipulation_RemoveFromLeft, IPOutboundManipulation_RemoveFromRight, IPOutboundManipulation_LeaveFromRight, IPOutboundManipulation_Prefix2Add, IPOutboundManipulation_Suffix2Add, IPOutboundManipulation_PrivacyRestrictionMode; [\IPOutboundManipulation] For example: IPOutboundManipulation 1 = 0, 0, 2, -1, *, *, *, *, 1, 3, 0, 255, , , 0; **Notes:** ▪ This table can include up to 100 indices (where 0 is the first index). ▪ Manipulated destination SIP URI user names are done on the following SIP headers: Request URI, To, and Remote-Party-ID (if exists). Manipulated source SIP URI user names are done on the following SIP headers: From, P-Asserted (if exists), P-Preferred (if exists), and Remote-Party-ID (if exists). ▪ For SIP URI host name (source and destination) manipulations, you can also use the IP Group'table. These host names are simply replaced with the names configured for the Source and Destination IP Groups respectively. ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, see 'Configuring IP-to-IP Outbound Manipulations' on page 469. ▪ For a description on configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.14 Standalone Survivability Parameters

The Stand-alone Survivability (SAS) parameters are described in the table below.

Table A-71: SAS Parameters

Parameter	Description
Web: Enable SAS EMS: Enable CLI: enable-sas **[EnableSAS]**	Enables the Stand-Alone Survivability (SAS) feature. ▪ **[0]** Disable Disabled (default) ▪ **[1]** Enable = SAS is enabled When enabled, the device receives the registration requests from different SIP entities in the local network and then forwards them to the defined proxy. If the connection to the proxy fails ('Emergency Mode'), the device serves as a proxy by allowing calls internal to the local network or outgoing to PSTN. **Note:** For this parameter to take effect, a device reset is required.
Web: SAS Local SIP UDP Port EMS: Local SIP UDP CLI: sas-local-sip-udp-port **[SASLocalSIPUDPPort]**	Defines the local UDP port for sending and receiving SIP messages for SAS. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5080.
Web: SAS Default Gateway IP EMS: Default Gateway IP CLI: sas-default-gw-ip **[SASDefaultGatewayIP]**	Defines the Default Gateway used in SAS 'Emergency Mode'. When an incoming SIP INVITE is received and the destination Address-Of-Record is not included in the SAS database, the request is immediately sent to this default gateway. The address can be configured as an IP address (dotted-decimal notation) or as a domain name (up to 49 characters). You can also configure the IP address with a destination port, e.g., "10.1.2.3:5060". The default is a null string, i.e., the local IP address of the gateway.
Web: SAS Registration Time EMS: Registration Time CLI: sas-registration-time **[SASRegistrationTime]**	Defines the value of the SIP Expires header that is sent in a 200 OK response to an incoming REGISTER message when in SAS 'Emergency Mode'. The valid range is 0 (Analog) or 10 (Digital) to 2,000,000. The default value is 20.
Web: SAS Local SIP TCP Port EMS: Local SIP TCP Port CLI: sas-local-sip-tcp-port **[SASLocalSIPTCPPort]**	Defines the local TCP port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5080.
Web: SAS Local SIP TLS Port EMS: Local SIP TLS Port CLI: sas-local-sip-tls-port **[SASLocalSIPTLSPort]**	Defines the local TLS port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5081.
Web/EMS: Enable Record-Route CLI: record-route	Determines whether the device's SAS application adds the SIP Record-Route header to SIP requests. This ensures that SIP messages traverse the device's SAS agent by including the SAS

Parameter	Description
[SASEnableRecordRoute]	IP address in the Record-Route header. ▪ **[0]** Disable (default) ▪ **[1]** Enable The Record-Route header is inserted in a request by a SAS proxy to force future requests in the dialog session to be routed through the SAS agent. Each traversed proxy in the path can insert this header, causing all future dialogs in the session to pass through it as well. When this feature is enabled, the SIP Record-Route header includes the URI "lr" parameter, indicating loose routing, for example: `Record-Route: <sip:server10.biloxi.com;lr>`
Web: SAS Proxy Set EMS: Proxy Set CLI: sas-proxy-set **[SASProxySet]**	Defines the Proxy Set (index number) used in SAS Normal mode to forward REGISTER and INVITE requests from users that are served by the SAS application. The valid range is 0 to 5. The default value is 0 (i.e., default Proxy Set).
Web: Redundant SAS Proxy Set EMS: Redundant Proxy Set CLI: rdcy-sas-proxy-set **[RedundantSASProxySet]**	Defines the Proxy Set (index number) used in SAS Emergency mode for fallback when the user is not found in the Registered Users database. Each time a new SIP request arrives, the SAS application checks whether the user is listed in the registration database. If the user is located in the database, the request is sent to the user. If the user is not found, the request is forwarded to the next redundant SAS defined in the Redundant SAS Proxy Set. If that SAS Proxy IP appears in the Via header of the request, it is not forwarded (thereby, preventing loops in the request's course). If no such redundant SAS exists, the SAS sends the request to its default gateway (configured by the parameter SASDefaultGatewayIP). The valid range is -1 to 5. The default value is -1 (i.e., no redundant Proxy Set).
Web/EMS: SAS Block Unregistered Users CLI: sas-block-unreg-usrs **[SASBlockUnRegUsers]**	Determines whether the device rejects SIP INVITE requests received from unregistered SAS users. This applies to SAS Normal and Emergency modes. ▪ **[0]** Un-Block = Allow INVITE from unregistered SAS users (default). ▪ **[1]** Block = Reject dialog-establishment requests from un-registered SAS users.
CLI: sas-contact-replace **[SASEnableContactReplace]**	Enables the device to change the SIP Contact header so that it points to the SAS host and therefore, the top-most SIP Via header and the Contact header point to the same host. ▪ **[0]** (default) = Disable - when relaying requests, the SAS agent adds a new Via header (with the SAS IP address) as the top-most Via header and retains the original Contact header. Thus, the top-most Via header and the Contact header point to different hosts. ▪ **[1]** = Enable - the device changes the Contact header so that it points to the SAS host and therefore, the top-most Via header and the Contact header point to the same host. **Note:** Operating in this mode causes all incoming dialog requests to traverse the SAS, which may cause load problems.
Web: SAS Survivability Mode	Determines the Survivability mode used by the SAS application.

Parameter	Description
EMS: Survivability Mode CLI: sas-survivability **[SASSurvivabilityMode]**	▪ **[0]** Standard = Incoming INVITE and REGISTER requests are forwarded to the defined Proxy list of SASProxySet in Normal mode and handled by the SAS application in Emergency mode (default). ▪ **[1]** Always Emergency = The SAS application does not use Keep-Alive messages towards the SASProxySet, instead it always operates in Emergency mode (as if no Proxy in the SASProxySet is available). ▪ **[2]** Ignore Register = Use regular SAS Normal/Emergency logic (same as option **[0]),** but when in Normal mode incoming REGISTER requests are ignored. ▪ **[3]** Auto-answer REGISTER = When in Normal mode, the device responds to received REGISTER requests by sending a SIP 200 OK (instead of relaying the registration requests to a Proxy), and enters the registrations in its SAS database. ▪ **[4]** Use Routing Table only in Normal mode = The device uses the IP-to-IP Routing table to route IP-to-IP SAS calls only when in SAS Normal mode (and is unavailable when SAS is in Emergency mode). This allows routing of SAS IP-to-IP calls to different destinations (and not only to the SAS Proxy Set).
Web: Enable ENUM CLI: enable-enum **[SASEnableENUM]**	Enables SAS to perform ENUM (E.164 number to URI mapping) queries when receiving INVITE messages in SAS emergency mode. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: SAS Binding Mode EMS: Binding Mode CLI: sasbindingmode **[SASBindingMode]**	Determines the SAS application database binding mode. ▪ **[0]** URI = If the incoming AoR in the INVITE requests is using a 'tel:' URI or 'user=phone' is defined, the binding is performed according to the user part of the URI only. Otherwise, the binding is according to the entire URI, i.e., User@Host (default). ▪ **[1]** User Part only = The binding is always performed according to the User Part only.
Web: SAS Emergency Numbers CLI: sas-emerg-nb **[SASEmergencyNumbers]**	Defines emergency numbers for the device's SAS application. When the device's SAS agent receives a SIP INVITE (from an IP phone) that includes one of the emergency numbers (in the SIP user part), it forwards the INVITE to the default gateway (configured by the parameter SASDefaultGatewayIP), i.e., the device itself, which sends the call directly to the PSTN. This is important for routing emergency numbers such as 911 (in North America) directly to the PSTN. This is applicable to SAS operating in Normal and Emergency modes. Up to four emergency numbers can be defined, where each number can be up to four digits.
CLI: sas-emerg-prefix **[SASEmergencyPrefix]**	Defines a prefix that is added to the Request-URI user part of the INVITE message that is sent by the device's SAS agent when in Emergency mode to the default gateway or to any other destination (using the IP2IP Routing table). This parameter is required to differentiate between normal SAS calls routed to the default gateway and emergency SAS calls. Therefore, this allows you to define different manipulation rules for normal and

Parameter	Description
	emergency calls. This valid value is a character string. The default is an empty string "".
Web: SAS Inbound Manipulation Mode CLI: sas-inb-manipul-md **[SASInboundManipulationMode]**	Enables destination number manipulation in incoming INVITE messages when SAS is in Emergency the state. The manipulation rule is done in the IP to IP Inbound Manipulation table. ▪ **[0]** None (default) ▪ **[1]** Emergency only **Notes:** ▪ Inbound manipulation applies only to INVITE requests. ▪ For more information on SAS inbound manipulation, see 'Manipulating Destination Number of Incoming INVITE' on page 489.
SAS Registration Manipulation Table	
Web: SAS Registration Manipulation EMS: Stand-Alone Survivability CLI: config voip > sas SASRegistrationManipulation **[SASRegistrationManipulation]**	This *para*meter table configures the SAS Registration Manipulation table. This table is used by the SAS application to manipulate the SIP Request-URI user part of incoming INVITE messages and of incoming REGISTER request AoR (To header), before saving it to the registered users database. The format of this table parameter is as follows: [SASRegistrationManipulation] FORMAT SASRegistrationManipulation_Index = SASRegistrationManipulation_RemoveFromRight, SASRegistrationManipulation_LeaveFromRight; [\SASRegistrationManipulation] ▪ RemoveFromRight = number of digits removed from the right side of the user part before saving to the registered user database. ▪ LeaveFromRight = number of digits to keep from the right side. If both RemoveFromRight and LeaveFromRight are defined, the RemoveFromRight is applied first. The registered database contains the AoR before and after manipulation. The range of both RemoveFromRight and LeaveFromRight is 0 to 30. For example, the manipulation rule below routes an INVITE with Request-URI header "sip:7184002@10.33.4.226" to user "4002@10.33.4.226" (i.e., keep only four digits from right of user part): `SASRegistrationManipulation 0 = 0, 4;` **Notes:** ▪ You can only configure one index entry. ▪ For a detailed description of the individual parameters in this table and for configuring this table using the Web interface, see 'Manipulating Destination Number of Incoming INVITE' on page 489.
Web: SAS IP-to-IP Routing Table	

Parameter	Description
[IP2IPRouting]	This *para*meter table configures the IP-to-IP Routing table for SAS routing rules. The format of this parameter is as follows: [IP2IPRouting] FORMAT IP2IPRouting_Index = IP2IPRouting_SrcIPGroupID, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupID, IP2IPRouting_DestSRDID, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions; [\IP2IPRouting] For example: IP2IPRouting 1 = -1, *, *, *, *, 0, -1, -1, , 0, -1, 0; **Notes:** ▪ This table can include up to 120 indices (where 0 is the first index). ▪ For a detailed description of the individual parameters in this table and for configuring this table using the Web interface, see 'Configuring IP2IP Routing Table (SAS)' on page 492. ▪ For a description on configuring *ini* file table parameters, see 'Configuring ini File Table Parameters' on page 98.

A.15 IP Media Parameters

The IP media parameters are described in the table below.

Table A-72: IP Media Parameters

Parameter	Description
Web: Number of Media Channels EMS: Media Channels CLI: media-channels **[MediaChannels]**	Defines the number of DSP channels that are allocated for various functionality (IP streaming, IP conferencing, IP transcoding, IP-to-IP sessions). The RTP streams for IP-to-IP calls always transverse the device and two DSP channels are allocated per IP-to-IP session. Therefore, the maximum number of media channels for IP-to-IP calls is 120, corresponding to 60 IP-to-IP calls. The maximum number of media channels depends on the number of installed Media Processing (MPM) modules: ▪ One module = 40 channels (assuming it is not inserted in Slot #6) ▪ Two modules = 80 channels ▪ Three modules = 120 channels The default value is 0. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The SBC application does not require DSP channels. The SBC application uses DSP channels only if media transcoding is needed, where two DSP channels are used per transcoding session. ▪ For IP conferencing, the MPM module must be installed in Slot

Parameter	Description
	#6. ▪ If the MPM module is installed in Slot #6, the first DSP channel is used for a conference and the second DSP, supporting up to 20 channels, is used for transcoding. ▪ Other DSP channels can be used for PSTN interfaces. ▪ For DSP resource options, see "DSP Channel Resources for Transcoding" on page 180.
CLI: IP-media-channels-enable **[EnableIPMediaChannels]**	Enables IP media channel support. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
[IPmediaChannels]	This ini file parameter table defines the number of DSP channels that are "borrowed" from each of the device's digital modules for IP media functionality. The format of this parameter is as follows: **[IPMediaChannels**] FORMAT IPMediaChannels_Index = IPMediaChannels_ModuleID, IPMediaChannels_DSPChannelsReserved; **[**IPMediaChannels] For example, the below settings use 15 and 10 DSP channels from modules 1 and 2, respectively: IPMediaChannels 1 = 1, 15; IPMediaChannels 2 = 2, 10; **Notes:** ▪ The value of DSPChannelsReserved must be in multiples of 5 (since the reservation is done per DSP device and not per DSP channel). ▪ By default, the MPM module is set to the maximum value of IPM channels, therefore, there is no need to define it. ▪ By default, a digital module (i.e., TRUNKS module) is set to 0 IPM channels. ▪ For a description on DSP utilization, see DSP Channel on page 180.
Web: Enable Voice Streaming CLI: enable **[EnableVoiceStreaming]**	Enables the HTTP Voice Streaming application (play/record). ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** For this parameter to take effect, a device reset is required.
CLI: upload-method **[VoiceStreamUploadMethod]**	Defines the HTTP request type for loading the voice stream to the file server. ▪ **[0]** = POST (default). ▪ **[1]** = PUT. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to MSCML recording.
CLI: post-uri **[VoiceStreamUploadPostURI]**	Defines the URI used on the POST request to upload voice data from the media server to a Web server. **Note:** For this parameter to take effect, a device reset is required.
CLI: use-aps-bundle **[APSEnabled]**	Determines whether Voice Prompt index references refer to audio provided by the Audio Provisioning Server (APS) or by the local

Parameter	Description
	Voice Prompts file. ▪ **[0]** = APS disabled. Local Voice Prompts file is used. An audio reference in a play request (such as http://localhost/0) indicates that the Voice Prompt at index 0 in the Voice Prompts file is played. ▪ **[1]** = APS enabled (default). An audio reference (such as http://localhost/99) indicates that the audio segment provisioned on the APS with segment ID 99 is played. **Note:** For this parameter to take effect, a device reset is required.
Web: Calling Number Playback ID CLI: call-num-plybck-id **[CallingNumberPlayBackID]**	Defines the Calling Number identification string for local, audio playing of the calling number. When the device receives from the Application Server (or SIP user Agent) a regular SIP INVITE message with a SIP URI that includes this user-defined Calling Number identification string, the device plays the calling number to the phone. For example, upon the receipt of the below INVITE message, the device plays the numbers 1, 0, and then 1: INVITE sip:callingnumber@domain.com; From: <sip:101@10.132.11.245>; The valid value can be up to 16 characters. The default is "callingnumber". **Note:** The APS server support must be enabled to support this feature. Below are the relevant ini file parameter settings: ▪ CallingNumberPlayBackID = callingnumber ▪ VpFileUrl = 'http://10.132.10.46/vp.dat' ▪ APSSegmentsFileURL = 'http://10.132.10.46/segments.xml' ▪ APSEnabled = 1 ▪ AMSProfile = 1 ▪ AASPackagesProfile = 3 ▪ EnableVoiceStreaming = 1
Web: NetAnn Announcement ID **[NetAnnAnncID]**	Defines the NetAnn identification string (up to 16 characters) for playing an announcement using the NetAnn interface. The application server sends a regular SIP INVITE message with a SIP URI that includes this identifier string and a “play=” parameter that identifies the necessary announcement. The default value is ‘annc’. Example 1: INVITE sip: annc@10.2.3.4;play=http://localhost/1. Example 2: INVITE sip: annc@10.2.3.4;play=http://10.2.3.4/Annc/hello.wav.
Web: MSCML ID CLI: mscml-id **[MSCMLID]**	Defines the Media Server Control Markup Language (MSCML) identification string (up to 16 characters). To start an MSCML session, the application server sends a regular SIP INVITE message with a SIP URI that includes this string. The default value is ‘ivr’. For example: INVITE sip:ivr@10.2.3.4 Subsequent INFO messages carry the requests and responses.
Web: Transcoding ID **[TranscodingID]**	Defines the Transcoding identification string (up to 16 characters) used for identifying an incoming Transcoding call. The default value is ‘trans’. For more information on Transcoding, see NetAnn Interface on page 521.

Parameter	Description
AMS Parameters	
CLI: enable **[AmsProfile]**	Enables advanced audio. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
CLI: profile **[AASPackagesProfile]**	Must be set to 3 to use advanced audio. **Note:** For this parameter to take effect, a device reset is required.
[AmsPrimaryLanguage]	Determines the primary language used in the advanced audio package. The default value is “eng”. The languages are according to ISO standard 639-2 language codes.
[AmsSecondaryLanguage]	Determines the secondary language used in the advanced audio package. The default value is “heb”. The languages are according to ISO standard 639-2 language codes.
CLI: allow-url-as-alias **[AMSAllowUrlAsAlias]**	Determines whether or not play requests for remote URLs are first verified with local audio segments to determine if any have an alias matching for the URL. If a match is found, the corresponding local audio segment is played. ▪ **[0]** = Always use remote storage (default).\ ▪ **[1]** = Check local storage first. One application for this capability is that of a 'provisioned' cache within the device. For details on provisioning an alias and other audio provisioning capabilities, refer to the Audio Provisioning Server (APS) User's Manual.
Conferencing Parameters	
Web/EMS: Conference ID CLI: conf-id **[ConferenceID]**	Defines the Conference Identification string (up to 16 characters). The default value is ‘conf’. For example: ConferenceID = MyConference **Note:** To join a conference, the INVITE URI must include the Conference ID string, preceded by the number of the participants in the conference, and terminated by a unique number. For example: Invite sip:4MyConference1234@10.1.10.10. INVITE messages with the same URI join the same conference.
Web: Beep on Conference CLI: beep-on-conf **[BipOnConference]**	Determines whether or not a beep is played when a participant joins or leaves a conference (in the latter case, a beep of a different pitch is heard). ▪ **[0]** Disable = Beep is disabled. ▪ **[1]** Enable = Beep is enabled (default).
Web: Enable Conference DTMF Clamping CLI: conf-dtmf-clamping **[EnableConferenceDTMFClamp]**	Determines the device logic once a DTMF is received on any conference participant. If enabled, the DTMF is not regenerated toward the other conference participants. This logic is only relevant for simple conferencing (NetAnn). ▪ **[0]** Disable = Disable ▪ **[1]** Enable = Enable (default)

Parameter	Description
Web: Enable Conference DTMF Reporting CLI: conf-dtmf-reporting **[EnableConferenceDTMFReporting]**	Determines the device logic once a DTMF is received on any conference participant. If enabled, the device reports this DTMF in an out-of-band SIP message (according to TxDTMFOptions). This logic is only relevant for simple conferencing (NetAnn). ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enable
Web: Active Speakers Min. Interval CLI: active-speaker-notification-minimum-interval **[ActiveSpeakersNotificationMinInterval]**	Defines the minimum interval (in 100 msec units) between each Active Speaker Notification (ASN) events report. These events report on the active speakers in a conference. The event is issued whenever the active speakers change. Minimum configurable interval between events is 500 msec (5 units). The range is 5 to 2147483647 units. The default is 20 (i.e., 100 msec).
Web: Playback Audio Format **[cpPlayCoder]**	Determines the coder when playing a RAW file. ▪ **[1]** G711 Mulaw ▪ **[2]** G711 Alaw (default)
Web: Record Audio Format **[cpRecordCoder]**	Determines the coder for recording all supported file types. ▪ **[1]** G711 Mulaw ▪ **[2]** G711 Alaw (default) **Note:** For this parameter to take effect, a device reset is required.
Web: End of Record Trim CLI: end-of-record-cut-time **[cpEndOfRecordCutTime]**	Defines the maximum amount (in milliseconds) of audio to remove from the end of a recording. This is used to remove the DTMF signals generated by the end user for terminating the record. The valid range is 0 to 65,535. The default is 0.
CLI: client-max-retransmission **[NFSClientMaxRetransmission]**	Since NFS is carried over UDP, retransmission is performed for messages without a response. This parameter enables the user to define the maximum number of retransmissions performed for such a command. By default, the parameter is not used and the number of retransmissions is derived from the parameter ServerRespondTimeout. The range is 1 to 100. The default is 0 (derived from ServerRespondTimeout).
[StreamingPlayingUnderRunTimeout]	Defines the maximum time (in msec) that the device waits for the streaming server to acknowledge data sent to it. The range is 100 to 10,000. The default is 5,000.
[StreamingRecordingOverRunTimeout]	Defines the maximum time (in msec) that the streaming server waits to acknowledge a data request sent from the device. The range is 100 to 10,000. The default is 5,000.
CLI:server-resp-timeout **[ServerRespondTimeout]**	Defines the maximum time (in msec) that the device must wait for a response when operating with a remote server. The valid range is 1,000 to 90,000. The default is 5,000.
Automatic Gain Control (AGC) Parameters	
Web: Enable AGC EMS: AGC Enable CLI: AGC-enable **[EnableAGC]**	Enables the AGC mechanism. The AGC mechanism adjusts the level of the received signal to maintain a steady (configurable) volume level. ▪ **[0]** Disable (default).

Parameter	Description
	▪ **[1]** Enable. **Notes:** ▪ This parameter can also be configured per Tel Profile, using the TelProfile parameter. ▪ For a description of AGC, see Automatic Gain Control (AGC) on page 177.
Web: AGC Slope EMS: Gain Slope CLI: AGC-gain-slope **[AGCGainSlope]**	Determines the AGC convergence rate: ▪ **[0]** 0 = 0.25 dB/sec ▪ **[1]** 1 = 0.50 dB/sec ▪ **[2]** 2 = 0.75 dB/sec ▪ **[3]** 3 = 1.00 dB/sec (default) ▪ **[4]** 4 = 1.25 dB/sec ▪ **[5]** 5 = 1.50 dB/sec ▪ **[6]** 6 = 1.75 dB/sec ▪ **[7]** 7 = 2.00 dB/sec ▪ **[8]** 8 = 2.50 dB/sec ▪ **[9]** 9 = 3.00 dB/sec ▪ **[10]** 10 = 3.50 dB/sec ▪ **[11]** 11 = 4.00 dB/sec ▪ **[12]** 12 = 4.50 dB/sec ▪ **[13]** 13 = 5.00 dB/sec ▪ **[14]** 14 = 5.50 dB/sec ▪ **[15]** 15 = 6.00 dB/sec ▪ **[16]** 16 = 7.00 dB/sec ▪ **[17]** 17 = 8.00 dB/sec ▪ **[18]** 18 = 9.00 dB/sec ▪ **[19]** 19 = 10.00 dB/sec ▪ **[20]** 20 = 11.00 dB/sec ▪ **[21]** 21 = 12.00 dB/sec ▪ **[22]** 22 = 13.00 dB/sec ▪ **[23]** 23 = 14.00 dB/sec ▪ **[24]** 24 = 15.00 dB/sec ▪ **[25]** 25 = 20.00 dB/sec ▪ **[26]** 26 = 25.00 dB/sec ▪ **[27]** 27 = 30.00 dB/sec ▪ **[28]** 28 = 35.00 dB/sec ▪ **[29]** 29 = 40.00 dB/sec ▪ **[30]** 30 = 50.00 dB/sec ▪ **[31]** 31 = 70.00 dB/sec
Web: AGC Redirection EMS: Redirection CLI: AGC-redirection **[AGCRedirection]**	Determines the AGC direction. ▪ **[0]** 0 = AGC works on signals from the TDM side (default). ▪ **[1]** 1 = AGC works on signals from the IP side.
Web: AGC Target Energy EMS: Target Energy CLI: AGC-target-energy **[AGCTargetEnergy]**	Defines the signal energy value (dBm) that the AGC attempts to attain. The valid range is 0 to -63 dBm. The default value is -19 dBm.
EMS: Minimal Gain CLI: AGC-min-gain	Defines the minimum gain (in dB) by the AGC when activated.

Parameter	Description
[AGCMinGain]	The range is 0 to -31. The default is -20. **Note:** For this parameter to take effect, a device reset is required.
EMS: Maximal Gain CLI: AGC-max-gain **[AGCMaxGain]**	Defines the maximum gain (in dB) by the AGC when activated. The range is 0 to 18. The default is 15. **Note:** For this parameter to take effect, a device reset is required.
EMS: Disable Fast Adaptation CLI: AGC-disable-fast-adaptation **[AGCDisableFastAdaptation]**	Enables the AGC Fast Adaptation mode. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
Answer Machine Detector (AMD) Parameters	
Web: Web: Answer Machine Detector Sensitivity Parameter Suit CLI: AMD-sensitivity-parameter-suit **[AMDSensitivityParameterSuit]**	Determines the AMD Parameter Suite that you want the device to use. ▪ **[0]** = USA Parameter Suite with 8 detection sensitivity levels (from 0 to 7). (default) ▪ **[1]** = USA Parameter Suite with high detection sensitivity resolution (16 sensitivity levels, from 0 to 15). ▪ **[2]-[**3] = Other countries parameter suites with up to 16 sensitivity levels. **Notes:** ▪ The sensitivity level is selected by the AMDSensitivityLevel parameter. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see Configuring IP Profiles on page 248).
Web: Answer Machine Detector Sensitivity Level CLI: AMD-sensitivity-level **[AMDSensitivityLevel]**	Defines the AMD detection sensitivity level of the selected AMD Parameter Suite. The valid value range is 0 (for best detection of an answering machine) to 15 (for best detection of a live call). The default value is 8. **Notes:** ▪ This parameter is applicable only if the AMDSensitivityParameterSuit parameter is set to any option other than 0. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see Configuring IP Profiles on page 248).
Web: Answer Machine Detector Sensitivity EMS: Sensitivity **[AMDDetectionSensitivity]**	Defines the AMD detection sensitivity level of the selected Parameter Suite. AMD can be useful in automatic dialing applications. In some of these applications, it is important to detect if a human voice or an answering machine is answering the call. AMD can be activated and de-activated only after a channel is already open. The valid value range is 0 to 7, where 0 is the best detection for answering machines and 7 is the best detection for live calls (i.e., voice detection). The default is 3. **Notes:** ▪ This parameter is applicable only if the AMDSensitivityParameterSuit parameter is set to 0. ▪ To enable the AMD feature, set the EnableDSPIPMDetectors parameter to 1.

Parameter	Description
	▪ For more information on AMD, see Answer Machine Detector (AMD) on page 172.
Web: AMD Sensitivity File **[AMDSensitivityFileName]**	Defines the name of the AMD Sensitivity file that contains the AMD Parameter Suites. **Notes:** ▪ This file must be in binary format (.dat). You can use the DConvert utility to convert the original file format from XML to .dat. ▪ You can load this file using the Web interface (see Loading Auxiliary Files on page 681).
[AMDSensitivityFileUrl]	Defines the URL path to the AMD Sensitivity file for downloading from a remote server.
[AMDMinimumVoiceLength]	Defines the AMD minimum voice activity detection duration (in 5-ms units). Voice activity duration below this threshold is ignored and considered as non-voice. The valid value range is 10 to 100. The default is 42 (i.e., 210 ms).
[AMDMaxGreetingTime]	Defines the maximum duration to detect greeting message. **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see Configuring IP Profiles on page 248).
[AMDMaxPostGreetingSilenceTime]	Defines the maximum duration of silence from after the greeting time is over (defined by AMDMaxGreetingTime) until the AMD decision. **Note:** This parameter can also be configured per IP Profile, using the IPProfile parameter (see Configuring IP Profiles on page 248).
EMS: Time Out **[AMDTimeout]**	Defines the timeout (in msec) between receiving Connect messages from the ISDN and sending AMD results. The valid range is 1 to 30,000. The default is 2,000 (i.e., 2 seconds).
Web/EMS: AMD Beep Detection Mode CLI: amd-beep-detection **[AMDBeepDetectionMode]**	Determines the AMD beep detection mode. This mode detects the beeps played at the end of an answering machine message, by using the X-Detect header extension. The device sends a SIP INFO message containing the field values Type=AMD and SubType=Beep. This feature allows users of certain third-party, Application server to leave a voice message after an answering machine plays the “beep”. ▪ **[0]** Disabled (default) ▪ **[1]** Start After AMD ▪ **[2]** Start Immediately
Web: Answer Machine Detector Beep Detection Timeout EMS: Beep Detection Timeout CLI: AMD-beep-detection-timeout **[AMDBeepDetectionTimeout]**	Defines the AMD beep detection timeout (i.e., the duration that the beep detector functions from when detection is initiated). This is used for detecting beeps at the end of an answering machine message. The valid value is in units of 100 milliseconds, from 0 to 1638. The default value is 200 (i.e., 20 seconds).
Web: Answer Machine Detector Beep Detection Sensitivity	Defines the AMD beep detection sensitivity for detecting beeps at the end of an answering machine message.

Parameter	Description
EMS: Beep Detection Sensitivity CLI: AMD-beep-detection-sensitivity **[AMDBeepDetectionSensitivity]**	The valid value is 0 to 3, where 0 (default) is the least sensitive.
Energy Detector Parameters **Note:** Currently, this feature is not supported.	
Enable Energy Detector CLI: energy-detector-enable **[EnableEnergyDetector]**	Enables the Energy Detector feature. This feature generates events (notifications) when the signal received from the PSTN is higher or lower than a user-defined threshold (defined by the EnergyDetectorThreshold parameter). ▪ **[0]** Disable (default) ▪ **[1]** Enable
Energy Detector Quality Factor CLI: energy-detector-sensitivity **[EnergyDetectorQualityFactor]**	Defines the Energy Detector's sensitivity level. The valid range is 0 to 10, where 0 is the lowest sensitivity and 10 the highest sensitivity. The default is 4.
Energy Detector Threshold CLI: energy-detector-threshold **[EnergyDetectorThreshold]**	Defines the Energy Detector's threshold. A signal below or above this threshold invokes an 'Above' or 'Below' event. The threshold is calculated as follows: Actual Threshold = -44 dBm + (EnergyDetectorThreshold * 6) The valid value range is 0 to 7. The default is 3 (i.e., -26 dBm).
Pattern Detection Parameters **Note:** For an overview on the pattern detector feature for TDM tunneling, see DSP Pattern Detector on page 261.	
Web: Enable Pattern Detector **[EnablePatternDetector]**	Enables the Pattern Detector (PD) feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable
[PDPattern]	Defines the patterns that can be detected by the Pattern Detector. The valid range is 0 to 0xFF. **Note:** For this parameter to take effect, a device reset is required.
[PDThreshold]	Defines the number of consecutive patterns to trigger the pattern detection event. The valid range is 0 to 31. The default is 5. **Note:** For this parameter to take effect, a device reset is required.
VXML Parameters	
Web/EMS: Enable VXML **[EnableVXML]**	Enables the VXML stack. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: VXML ID CLI: vxml-id **[VXMLID]**	Defines the VoiceXML identification string (up to 16 characters) for identifying an incoming VXML call. The default value is 'dialog'.
[VxmlBargeInAllowed]	Determines whether the VXML property indicates if prompts can be interrupted. ▪ **[0]** = prompts cannot be interrupted ▪ **[1]** = prompts can be interrupted (default)

Parameter	Description
	Note: For this parameter to take effect, a device reset is required.
[VxmlBuiltinGrammarPath]	Defines the path on the remote Automatic Speech Recognition (ASR) / text-to-speech (TTS) server to access the built-in grammars. The path must not end in a forward slash ('/') as this is added as needed during runtime. The default value is NULL. **Note:** For this parameter to take effect, a device reset is required.
[VxmlCompleteTimeout]	Optional parameter that defines the amount of silence (in msec) to wait after speech grammar has been matched before reporting the match. The default value is 0 (i.e., don't set this parameter on recognition attempt). **Note:** For this parameter to take effect, a device reset is required.
[VxmlConfidenceLevel]	Defines the default speech recognition confidence threshold for VXML. The range is from 0 to 100. The default value is 50. **Note:** For this parameter to take effect, a device reset is required.
[VxmlDefaultLanguage]	Defines the default language for speech recognition, if speech recognition has been enabled. If the root document doesn't specify a language and a field or menu element generates speech recognition requests using the GRXML MIME type, the default language is used in the request. The default value is 'en_us'. **Note:** For this parameter to take effect, a device reset is required.
[VxmlIncompleteTimeout]	Optional parameter that defines the amount of silence (in msec) to wait after speech grammar has not matched a voice grammar. The default value is 0 (i.e., don't set this parameter on recognition attempt). **Note:** For this parameter to take effect, a device reset is required.
[VxmlInterDigitTimeout]	Defines the inter-digit timeout value (in msec) used when DTMF is received. The valid range for this parameter is 0 to 7,000 msec. The default value is 3,000. **Note:** For this parameter to take effect, a device reset is required.
[VxmlMaxActiveFiles]	Defines the maximum number of static VXML scripts that can be loaded to the system at any one time. The valid range for this parameter is 0 to 30. The default value is 10. **Note:** This parameter does not affect the number of dynamic scripts that can be simultaneously active. **Note:** For this parameter to take effect, a device reset is required.
[VxmlMaxPorts]	Defines the number of channels in the system that can simultaneously run VXML scripts. The range is from 0 to the maximum number of channels in the system. This value can be used to ensure there are sufficient VXML resources for each call. For example, if the system is running dynamic scripts that each requires many resources, the VxmlMaxPorts value can be lowered to help ensure that each individual call has adequate resources. The default value is 0. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
[VxmlMaxSpeechTimeout]	Defines the maximum time the caller can speak (in msec) in an attempt to match a speech grammar before a no match event is thrown. The range is 0 - 7,000. The default value is 0 (i.e., no time limit in the speech recognition attempt). **Note:** For this parameter to take effect, a device reset is required.
[VxmlNoInputTimeout]	Defines the no input timeout for digit (DTMF) collection or speech recognition (in msec). The range is 0 - 7,000. The default value is 3,000. **Note:** For this parameter to take effect, a device reset is required.
[VxmlSensitivityLevel]	Defines the default speech recognition sensitivity level for VXML. The valid range for this parameter is 0 to 100. The default value is 50. **Note:** For this parameter to take effect, a device reset is required.
[VxmlSpeedVsAccuracy]	Defines the hint to the speech recognition engine for the balance of speed vs. accuracy. The valid range is from 0 to 100. The default value is 50. A low number means the speech recognition engine must perform recognition rapidly, at the cost (i.e., trade off) of accuracy. A high number, such as 100, means the speech recognition engine must perform the speech recognition accurately, at the cost of speed. **Note:** For this parameter to take effect, a device reset is required.
[VxmlSystemInputModes]	Determines which inputs are valid for grammars. ▪ **[0]** = DTMF is valid (default) ▪ **[1]** = Voice is valid ▪ **[2]** = Both are valid **Note:** For this parameter to take effect, a device reset is required.
[VxmlTermChar]	Defines the default terminating digit for received DTMF. The default value is 35 (equivalent to ASCII '#'). **Note:** For this parameter to take effect, a device reset is required.
[VxmlTermTimeout]	Defines the time to wait before terminating received DTMF (in msec). The range is 0 - 7,000. The default value is 3,000. **Note:** For this parameter to take effect, a device reset is required.
Media Resource Control Protocol (MRCP) / Real Time Streaming Protocol (RTSP) Parameters	
[MRCPDefaultMIMEType]	Determines the default format for speech recognition for inline grammars. ▪ **[0]** = indicates GRXML (default) ▪ **[1]** = indicates GL (Nuance format)
[MRCPEnabled]	Activates the Media Resource Control Protocol (MRCP) functionality. ▪ **[0]** = Disable (default) ▪ **[1]** = Activate
[MRCPMaxPorts]	Defines the number of ports that are allocated to running MRCP-related activities such as speech recognition and text-to-speech.

Parameter	Description
	A port is considered duplex, so that speech recognition and text-to-speech can run on the same port. The value should not exceed the number of channels in the system. The range is 0 - 120. The default value is 10.
[MRCPServerName]	Defines the hostname of the MRCP server. This is used to build a URI for the server. The default value is NULL.
[MRCPServerIp]	Defines the IP address of the MRCP speech server. The default value is 0.0.0.0.
[MRCPServerPort]	Defines the control port on the MRCP speech server. The range is 0 - 65,535. The default value is 554.
[RTSPConnectionRetryInterval]	Defines the time (in seconds) that the system must wait before trying to create a socket for the RTSP speech server if the socket was never created or was created and then brought down. The range is 0 - 65,535. The default value is 10. **Note:** For this parameter to take effect, a device reset is required.
[RTSPEnabled]	Activates the RTSP functionality. ▪ **[0]** = Disable (default) ▪ **[1]** = Activate **Note:** For this parameter to take effect, a device reset is required.
[RTSPMaxPorts]	Defines the number of channels that can be simultaneously active in RTSP sessions. The range is 0 - 20.The default value is 10. **Note:** For this parameter to take effect, a device reset is required.

A.16 Auxiliary and Configuration Files Parameters

This subsection describes the device's auxiliary and configuration files parameters.

A.16.1 Auxiliary/Configuration File Name Parameters

The configuration files (i.e., auxiliary files) can be loaded to the device using the Web interface (see 'Loading Auxiliary Files' on page 681). For loading these files using the *ini* file, you need to configure these files in the *ini* file and configured whether they must be stored in the non-volatile memory. The table below lists the *ini* file parameters associated with these auxiliary files. For more information on the auxiliary files, see 'Auxiliary Configuration Files' on page 684.

Table A-73: Auxiliary and Configuration File Parameters

Parameter	Description
General Parameters	
[SetDefaultOnIniFileProcess]	Determines if all the device's parameters are set to their defaults before processing the updated *ini* file. ▪ **[0]** Disable - parameters not included in the downloaded *ini* file are not returned to default settings (i.e., retain their current settings). ▪ **[1]** Enable (default) **Note:** This parameter is applicable only for automatic HTTP update or Web *ini* file upload (not applicable if the *ini* file is loaded using BootP).
[SaveConfiguration]	Determines if the device's configuration (parameters and files) is saved to flash (non-volatile memory). ▪ **[0]** = Configuration isn't saved to flash memory. ▪ **[1]** = Configuration is saved to flash memory (default).
Auxiliary and Configuration File Name Parameters	
Web/EMS: Call Progress Tones File **[CallProgressTonesFilename]**	Defines the name of the file containing the Call Progress Tones definitions. For more information on how to create and load this file, refer ro the *Product Reference Manual*. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Voice Prompts File **[VoicePromptsFileName]**	Defines the name (and path) of the file containing the Voice Prompts. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For more information on this file, see Voice Prompts File on page 689.
Web/EMS: Prerecorded Tones File **[PrerecordedTonesFileName]**	Defines the name (and path) of the file containing the Prerecorded Tones. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: CAS File EMS: Trunk Cas Table Index **[CASFileName**_x]	Defines the CAS file name (e.g., 'E_M_WinkTable.dat'), which defines the CAS protocol (where x denotes the CAS file ID 0 to 7). It is possible to define up to eight different CAS files by repeating this parameter. Each CAS file can be associated with one or more of the device's trunks, using the parameter CASTableIndex or it can be associated per B-channel using the parameter CASChannelIndex. **Note:** For this parameter to take effect, a device reset is required.
Web: Dial Plan EMS: Dial Plan Name **[CasTrunkDialPlanName**_x]	Defines the Dial Plan name (up to 11-character strings) that is used on a specific trunk (denoted by x).
Web: Dial Plan File EMS: Dial Plan File Name **[DialPlanFileName]**	Defines the name (and path) of the Dial Plan file (defining dial plans). This file should be constructed using the DConvert utility (refer to the Product Reference Manual).
[UserInfoFileName]	Defines the name (and path) of the file containing the User Information data.

A.16.2 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

Table A-74: Automatic Update of Software and Configuration Files Parameters

Parameter	Description
General Automatic Update Parameters	
[AutoUpdateCmpFile]	Enables the Automatic Update mechanism for the cmp file. ▪ **[0]** = The Automatic Update mechanism doesn't apply to the cmp file (default). ▪ **[1]** = The Automatic Update mechanism includes the cmp file. **Note:** For this parameter to take effect, a device reset is required.
[AutoUpdateFrequency]	Defines the number of minutes that the device waits between automatic updates. The default value is 0 (i.e., the update at fixed intervals mechanism is disabled). **Note:** For this parameter to take effect, a device reset is required.
[AutoUpdatePredefinedTime]	Defines schedules (time of day) for automatic updates. The format of this parameter is: 'HH:MM', where *HH* denotes the hour and *MM* the minutes, for example, 20:18. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The actual update time is randomized by five minutes to reduce the load on the Web servers.
EMS: AUPD Verify Certificates **[AUPDVerifyCertificates]**	Determines whether the Automatic Update mechanism verifies server certificates when using HTTPS. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
[AUPDCheckIfIniChanged]	Determines whether the Automatic Update mechanism performs CRC checking to determine if the *ini* file has changed prior to

Parameter	Description
	processing. ▪ **[0]** = Do not check CRC. The *ini* file is loaded whenever the server provides it. (default) ▪ **[1]** = Check CRC for the entire file. Any change, including line order, causes the *ini* file to be re-processed. ▪ **[2]** = Check CRC for individual lines. Use this option when the HTTP server scrambles the order of lines in the provided *ini* file.
[ResetNow]	Invokes an immediate device reset. This option can be used to activate offline (i.e., not on-the-fly) parameters that are loaded using the parameter IniFileUrl. ▪ **[0]** = The immediate restart mechanism is disabled (default). ▪ **[1]** = The device immediately resets after an *ini* file with this parameter set to 1 is loaded.
Software/Configuration File URL Path for Automatic Update Parameters	
[CmpFileURL]	Defines the name of the *cmp* file and the path to the server (IP address or FQDN) from where the device can load the *cmp* file and update itself. The *cmp* file can be loaded using HTTP/HTTPS. For example: http://192.168.0.1/filename **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When this parameter is configured, the device always loads the *cmp* file after it is reset. ▪ The *cmp* file is validated before it's burned to flash. The checksum of the *cmp* file is also compared to the previously burnt checksum to avoid unnecessary resets. ▪ The maximum length of the URL address is 255 characters.
[IniFileURL]	Defines the name of the *ini* file and the path to the server (IP address or FQDN) on which it is located. The *ini* file can be loaded using HTTP/HTTPS. For example: http://192.168.0.1/filename http://192.8.77.13/config<MAC> https://<username>:<password>@<IP address>/<file name> **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When using HTTP or HTTPS, the date and time of the *ini* file are validated. Only more recently dated *ini* files are loaded. ▪ The optional string <MAC>' is replaced with the device's MAC address. Therefore, the device requests an *ini* file name that contains its MAC address. This option allows the loading of specific configurations for specific devices. ▪ The maximum length of the URL address is 99 characters.
[PrtFileURL]	Defines the name of the Prerecorded Tones (PRT) file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
[CptFileURL]	Defines the name of the CPT file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file.

Parameter	Description
	Note: The maximum length of the URL address is 99 characters.
[VpFileURL]	Defines the name of the Voice Prompts file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
[CasFileURL]	Defines the name of the CAS file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
[TLSRootFileUrl]	Defines the name of the TLS trusted root certificate file and the URL from where it can be downloaded. **Note:** For this parameter to take effect, a device reset is required.
[TLSCertFileUrl]	Defines the name of the TLS certificate file and the URL from where it can be downloaded. **Note:** For this parameter to take effect, a device reset is required.
[TLSPkeyFileUrl]	Defines the URL for downloading a TLS private key file using the Automatic Update facility.
[UserInfoFileURL]	Defines the name of the User Information file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file **Note:** The maximum length of the URL address is 99 characters.

B SIP Message Manipulation Syntax

This appendix provides a detailed description on the support and syntax for configuring SIP message manipulation rules, using the SIP Message Manipulation table, as described in 'Configuring Message Manipulations' on page 464.

B.1 Actions

The actions that can be done on SIP message manipulation in the Message Manipulations table are listed in the table below.

Table B-1: Message Manipulation Actions

Action	Value
Add	0
Remove	1
Modify	2
Add Prefix	3
Add Suffix	4
Remove Suffix	5
Remove Prefix	6

The maximum length of the value for a manipulation is 299 characters.

B.2 Header Types

B.2.1 Accept

An example of the header is shown below:

```
Accept:  application/sdp
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	No	N/A
Keyword	**Sub Types**			**Attributes**
N/A	N/A			N/A

Below is a header manipulation example:

Rule:	If the supported header does not contain 'mm,100rel,timer,replaces', then in all INVITE messages add an Accept header: `MessageManipulations 8 = 1, invite, header.supported != 'mm,100rel,timer,replaces', header.accept, 0, ' application/x-private ', 0;`
Result:	`Accept: application/x-private`

B.2.2 Accept-Language

An example of the header is shown below:

```
Accept-Language: da, en-gb;q=0.8, en;q=0.7
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	No	N/A
Keyword	**Sub Types**		**Attributes**	
N/A	N/A		N/A	

Below is a header manipulation example:

Rule:	Add a new Language header to all INVITE messages: `MessageManipulations 0 = 1, invite, , header.accept-language, 0, 'en, il, cz, it', 0;`
Result:	`Accept-Language: en, il, cz, it`

B.2.3 Allow

An example of the header is shown below:

```
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	No	N/A

Keyword	Sub Types	Attributes
N/A	N/A	Read/Write

Below is a header manipulation example:

Rule:	Add an Allow header to all INVITE messages: `MessageManipulations 0 = 1, invite, , header.allow, 0, 'REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO ,SUBSCRIBE, XMESSAGE', 0;`
Result:	`Allow:` `REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO, SUBSCRIBE, XMESSAGE`

B.2.4 Call-Id

An example of the header is shown below:

```
Call-ID: JNIYXOLCAIWTRHWOINNR@10.132.10.128
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	No	No	No	NA

Keyword	Sub Types	Attributes
ID	String	Read Only

Below is a header manipulation example:

Rule:	Add a proprietary header to all INVITE messages using the data in the Call-id header: `MessageManipulations 0 = 1, invite, , header.Xitsp-abc, 0, header.call-id, 0;`
Result:	`Xitsp-abc: GIAPOFWRBQKJVAETIODI@10.132.10.128`

B.2.5 Contact

An example of the header is shown below:

```
Contact: <sip:555@10.132.10.128:5080>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	No	No	No	8

Keyword	Sub Types	Attributes
Expires	Integer	Read/Write
GruuContact	String	Read/Write
IsGRUU	Boolean	Read/Write
Name	String	Read/Write
Param	Param	Read/Write
URL	'URL' on page 1041	Read/Write*

* Host name cannot be modified in the URL structure for a contact header.

Below is a header manipulation example:

Rule:	Change the user part in the Contact header in all INVITE messages to fred: `MessageManipulations 0 = 1, Invite, ,header.contact.url.user, 2, 'fred', 0;`
Result:	`Contact: <sip:fred@10.132.10.128:5070>`

B.2.6 Cseq

An example of the header is shown below:

```
CSeq: 1 INVITE
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	No	No	No	N/A

Keyword	Sub Types	Attributes
Num	Integer	Read Only
Type	String	Read Only

Below is a header manipulation example:

Rule:	If the Cseq number is 1, then modify the user in the Contact header to fred. `MessageManipulations 0 = 1, Invite, header.cseq.num=='1',header.contact.url.user, 2, 'fred', 0;`
Result:	`Contact: <sip:fred@10.132.10.128:5070>`

B.2.7 Diversion

An example of the header is shown below:

```
Diversion: <sip:654@IPG2Host;user=phone>;reason=user-
busy;screen=no;privacy=off;counter=1
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	3

Keyword	Sub Types	Attributes
Name	String	Read/Write
Param	Param	Read/Write
Privacy	Enum Privacy (see 'Privacy' on page 1047)	Read/Write
Reason	Enum Reason (see 'Reason (Diversion)' on page 1047)	Read/Write
Screen	Enum Screen (see 'Screen' on page 1050)	Read/Write
URL	URL Structure (see 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Diversion header to all INVITE messages: `MessageManipulations 0 = 1, invite, , header.Diversion, 0, '<tel:+101>;reason=unknown; counter=1;screen=no; privacy=off', 0;`
	Result:	`Diversion: <tel:+101>;reason=user-busy;screen=no;privacy=off;counter=1`
Example 2	**Rule:**	Modify the Reason parameter in the header to 1, see 'Reason (Diversion)' on page 1047 for possible values: `MessageManipulations 1 = 1, invite, , header.Diversion.reason, 2, '1', 0;`
	Result:	`Diversion: <tel:+101>;reason=user-busy;screen=no;privacy=off;counter=1`
Example 3	**Rule:**	The URL in the Diversion header is modified to that which is contained in the header URL: `MessageManipulations 2 = 1, invite, , header.Diversion.URL, 2, header.from.url, 0;`
	Result:	`Diversion:<sip:555@IPG2Host;user=phone>;reason=user-busy;screen=no;privacy=off;counter=1`

B.2.8 Event

An example of the header is shown below:

```
Event: foo; id=1234
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
EventKey	Event Structure (see 'Event Structure' on page 1039)	Read/Write
Param	Param	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add parameter itsp-abc=voip to the Event header: `MessageManipulations 0 = 1, invite, , header.event.param.itsp-abc, 0, 'voip' , 0;`
	Result:	`Event: foo;id=1234;itsp-abc=voip`
Example 2	**Rule:**	Modify the Event ID string: `MessageManipulations 1 = 1, invite, , header.event.EVENTKEY.id, 2, '5678', 0;`
	Result:	`Event: foo;id=5678;`
Example 3	**Rule:**	Modify the Event package enum: `MessageManipulations 2 = 1, invite, , header.event.EVENTKEY.EVENTPACKAGE, 2, '2', 0;`

	Result:	`Event: refer;id=5678`

B.2.9 From

An example of the header is shown below:

```
From: <sip:555@10.132.10.128;user=phone>;tag=YQLQHCAAYBWKKRVIMWEQ
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	No	No	No	NA

Keyword	Sub Types	Attributes
Name	String	Read/Write
Param	Param	Read/Write
tag	String	Read Only
URL	URL Structure (refer to 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Change the user part of the From header if the user is not 654: `MessageManipulations 8 = 1, invite, header.from.url.user != '654', header.from.url.user, 2, 'fred', 0;`
	Result:	`From: <sip:fred@IPG2Host;user=phone>;tag=1c20161`
Example 2	**Rule:**	Add a new parameter to the From header called p1 and set its value to myParameter: `MessageManipulations 1 = 1, Invite.request, ,header.from.param.p1, 0, 'myParameter', 0;`
	Result:	`From: <sip:fred@IPG2Host;user=phone>;p1=myParameter;tag=1c5891`
Example 3	**Rule:**	Modify the URL in the From header: `MessageManipulations 0 = 1, any, , header.from.url, 2, 'sip:3200@110.18.5.41;tusunami=0', 0;`
	Result:	`From: <sip:3200@110.18.5.41;user=phone;tusunami=0>;tag=1c23750`

B.2.10 History-Info

An example of the header is shown below:

```
History-Info: <sip:UserA@ims.example.com;index=1>
History-Info: <sip:UserA@audc.example.com;index=2>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	20

Keyword	Sub Types	Attributes
HistoryInfo	String	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a new History-Info header to the message: `MessageManipulations 0 = 1, any, , header.History-Info, 0, '<sip:UserA@audc.mydomain.com;index=3>', 0`

	Result:	`History-Info:sip:UserA@ims.example.com;index=1`
		`History-Info:sip:UserA@audc.example.com;index=2`
		`History-Info: <sip:UserA@audc.mydomain.com;index=3>`

Example 2	**Rule:**	Delete an unwanted History-Info header from the message: `MessageManipulations 0 = 1, any, , header.History-Info.1, 1, , 0;`
	Result:	`History-Info: <sip:UserA@ims.example.com;index=1>`
Example 3	**Rule:**	Delete all History-Info from the message: `MessageManipulations 0 = 1, any, , header.History-Info, 1, , 0;`
	Result:	`All history-info headers are removed.`

B.2.11 Min-Se and Min-Expires

An example of the header is shown below:

```
Min-SE: 3600
Min-Expires: 60
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Param	Param	Read/Write
Time	Integer	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Min-Se header to the message using a value of 50: `MessageManipulations 1 = 1, any, , header.min-se, 0, '50', 0;`
	Result:	`Min-SE: 50`
Example 2	**Rule:**	Modify a Min-Expires header with the min-expires value and add an additional 0: `MessageManipulations 0 = 1, Invite, , header.Min-Expires.param, 2, header.Min-Expires.time + '0', 0;`

	Result:	`Min-Expires: 340;3400`
Example 3	**Rule:**	Modify a Min-Expires header changing the time to 700: `MessageManipulations 0 = 1, Invite, , header.Min-Expires.time, 2, '700', 0;`
	Result:	`Min-Expires: 700`

B.2.12 P-Asserted-Identity

An example of the header is shown below:

```
P-Asserted-Identity: Jane Doe <sip:567@itsp.com>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	1

Keyword	Sub Types	Attributes
URL	URL Structure (see 'URL' on page 1041)	Read/Write
Name	String	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a P-Asserted-Id header to all INVITE messages: `MessageManipulations 2 = 1, invite, , header.p-asserted-identity, 0, '<sip:567@itsp.com>', 0;`
	Result:	`P-Asserted-Identity: <sip:567@itsp.com>`
Example 2	**Rule:**	Modify the P-Asserted-Identity host name to be the same as the host name in the To header: `MessageManipulations 2 = 1, invite, , header.p-asserted-identity.URL.host, 2, header.to.url.host, 0;`
	Result:	`P-Asserted-Identity: <sip:567@10.132.10.128>`

B.2.13 P-Associated-Uri

An example of the header is shown below:

```
P-Associated-URI: <sip:12345678@itsp.com>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	1

Keyword	Sub Types	Attributes
Name	String	Read/Write
Param	Param	Read/Write

Keyword	Sub Types	Attributes
URL	URL Structure (see 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a P-Associated-Uri header to all INVITE response messages: `MessageManipulations 5 = 1, register.response, ,header.P-Associated-URI, 0, '<sip:admin@10.132.10.108>', 0;`
	Result:	`P-Associated-URI:<sip:admin@10.132.10.108>`
Example 2	**Rule:**	Modify the user portion of the URL in the header to 'alice': `MessageManipulations 5 = 1, register.response, ,header.P-Associated-URI.url.user, 2, 'alice', 0;`
	Result:	`P-Associated-URI:<sip:alice@10.132.10.108>`

B.2.14 P-Called-Party-Id

An example of the header is shown below:

```
P-Called-Party-ID: <sip:2000@gw.itsp.com>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Name	String	Read/Write
URL	URL Structure (see 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a P-Called-Party-Id header to all messages: `MessageManipulations 8 = 1, any, , header.p-called-party-id, 0, 'sip:2000@MSBG.ITSP.COM', 0;`
	Result:	`P-Called-Party-ID: <sip:2000@gw.itsp.com>`
Example 2	**Rule:**	Append a parameter (p1) to all P-Called-Party-Id headers: `MessageManipulations 9 = 1, invite, , header.p-called-party-id.param.p1, 0, 'red', 0;`
	Result:	`P-Called-Party-ID: <sip:2000@gw.itsp.com>;p1=red`
Example 3	**Rule:**	Add a display name to the P-Called-Party-Id header: `MessageManipulations 3 = 1, any, , header.p-called-party-id.name, 2, 'Secretary', 0;`
	Result:	`P-Called-Party-ID: Secretary <sip:2000@gw.itsp.com>;p1=red`

B.2.15 P-Charging-Vector

An example of the header is shown below:

```
P-Charging-Vector: icid-value=1234bc9876e; icid-generated-
at=192.0.6.8; orig-ioi=home1.net
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	No	N/A

Keyword	Sub Types	Attributes
N/A	N/A	N/A

Below are header manipulation examples:

Rule:	Add a P-Charging-Vector header to all messages: `MessageManipulations 1 = 1, any, , header.P-Charging-Vector, 0, 'icid-value=1234bc9876e; icid-generated-at=192.0.6.8; orig-ioi=home1.net', 0;`
Result:	`P-Charging-Vector: icid-value=1234bc9876e; icid-generated-at=192.0.6.8; orig-ioi=home1.net`

B.2.16 P-Preferred-Identity

An example of the header is shown below:

```
P-Preferred-Identity: "Cullen Jennings" <sip:fluffy@abc.com>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Name	String	Read/Write
URL	URL Structure (see 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a P-Preferred-Identity header to all messages: `MessageManipulations 1 = 1, any, , header.P-Preferred-Identity, 0, 'Cullen Jennings <sip:fluffy@abc.com>', 0;`
	Result:	`P-Preferred-Identity: "Cullen Jennings" <sip:fluffy@abc.com>`
Example 2	**Rule:**	Modify the display name in the P-Preferred-Identity header: `MessageManipulations 2 = 1, any, , header.P-Preferred-Identity.name, 2, 'Alice Biloxi', 0;`
	Result:	`P-Preferred-Identity: "Alice Biloxi" <sip:fluffy@abc.com>`

B.2.17 Privacy

An example of the header is shown below:

```
Privacy: none
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	No	N/A

Keyword	Sub Types	Attributes
privacy	'Privacy Struct' on page 1040	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a privacy header and set it to ''session': `MessageManipulations 1 = 1, any, , header.Privacy, 0, 'session', 0;`
	Result:	`Privacy: session`
Example 2	**Rule:**	Add 'user' to the list: `MessageManipulations 1 = 3, , , header.privacy.privacy.user, 2, '1', 0;`
	Result:	`Privacy: session;user`

B.2.18 Proxy-Require

An example of the header is shown below:

```
Proxy-Require: sec-agree
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Capabilities	SIPCapabilities Struct	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Proxy-Require header to the message: `MessageManipulations 1 = 1, any, , header.Proxy-Require, 0, 'sec-agree', 0;`
	Result:	`Proxy-Require: sec-agree`
Example 2	**Rule:**	Modify the Proxy-Require header to itsp.com: `MessageManipulations 2 = 1, any, , header.Proxy-Require, 2, 'itsp.com' , 0;`

	Result:	`Proxy-Require: itsp.com`
Example 3	**Rule:**	Set the privacy options tag in the Proxy-Require header: `MessageManipulations 0 = 0, invite, , header. Proxy-Require.privacy, 0, 1 , 0;`
	Result:	`Proxy-Require: itsp.com, privacy`

B.2.19 Reason

An example of the header is shown below:

```
Reason: SIP ;cause=200 ;text="Call completed elsewhere"
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
MLPP	MLPP Structure (see 'MLPP' on page 1039)	Read/Write
Reason	Reason Structure (see 'Reason Structure' on page 1040)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Reason header: `MessageManipulations 0 = 1, any, ,header.reason, 0, 'SIP;cause=200;text="Call completed elsewhere"', 0;`
	Result:	`Reason: SIP ;cause=200 ;text="Call completed elsewhere"`
Example 2	**Rule:**	Modify the reason cause number: `MessageManipulations 0 = 1, any, ,header.reason.reason.cause, 0, '200', 0;`
	Result:	`Reason: Q.850 ;cause=180 ;text="Call completed elsewhere"`
Example 3	**Rule:**	Modify the cause number: `MessageManipulations 0 = 1, any, ,header.reason.reason.reason, 0, '483', 0;`
	Result:	`Reason: SIP ;cause=483 ;text="483 Too Many Hops"`

Note: The protocol (SIP or Q.850) is controlled by setting the cause number to be greater than 0. If the cause is 0, then the text string (see Example 3) is generated from the reason number.

B.2.20 Referred-By

An example of the header is shown below:

```
Referred-By: <sip:referrer@referrer.example>;
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
param	param	Read/Write
URL	URL Structure (see 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Referred-By header: `MessageManipulations 0 = 1, any, ,header.Referred-By, 0, '<sip:refer@refer.com>', 0;`
	Result:	`Referred-By: <sip: sip:refer@refer.com>`
Example 2	**Rule:**	Modify the host: `MessageManipulations 0 = 1, any, ,header.Referred-By.url.host, 0, 'yahoo.com', 0;`
	Result:	`Referred-By: <sip:refer@yahoo.com>`
Example 3	**Rule:**	Add a new parameter to the header: `MessageManipulations 0 = 1, any, ,header.Referred-By.param.p1, 0, 'fxs', 0`
	Result:	`Referred-By: <sip:referrer@yahoo.com>;p1=fxs`

B.2.21 Refer-To

An example of the header is shown below:

```
Refer-To: sip:conference1@example.com
Refer-To:
<sips:a8342043f@atlanta.example.com?Replaces=12345601%40atlanta.example.com%3bfrom-tag%3d314159%3bto-tag%3d1234567>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	No	N/A

Keyword	Sub Types	Attributes
N/A	N/A	N/A

Below are header manipulation examples:

Example 1	**Rule:**	Add a basic header: `MessageManipulations 0 = 1, any, ,header.Refer-to, 0, '<sip:referto@referto.com>', 0;`
	Result:	`Refer-To: <sip:referto@referto.com>`
Example 2	**Rule:**	Add a Refer-To header with URI headers: `MessageManipulations 0 = 1, any, ,header.Refer-to, 0, '<sips:a8342043f@atlanta.example.com?Replaces=12345601%40atlanta.example.com%3bfrom-tag%3d314159%3bto-tag%3d1234567>', 0;`
	Result:	`Refer-To: <sips:a8342043f@atlanta.example.com?Replaces=12345601%40atlanta.example.com%3bfrom-tag%3d314159%3bto-tag%3d1234567>`

B.2.22 Remote-Party-Id

An example of the header is shown below:

```
Remote-Party-ID: "John Smith"
<sip:john.smith@itsp.com>;party=calling; privacy=full;screen=yes
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	3

Keyword	Sub Types	Attributes
Counter	Integer	Read/Write
Name	String	Read/Write
NumberPlan	Enum Number Plan (see 'Number Plan' on page 1046)	Read/Write
NumberType	Enum Number Type (see 'NumberType' on page 1046)	Read/Write
Param	Param	Read/Write
Privacy	Enum Privacy (see 'Privacy' on page 1047)	Read/Write
Reason	Enum Reason (RPI) (see 'Reason (Remote-Party-Id)' on page 1050)	Read/Write
Screen	Enum Screen (see 'Screen' on page 1050)	Read/Write
ScreenInd	Enum ScreenInd (see 'ScreenInd' on page 1050)	Read/Write
URL	URL Structure (see 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Remote-Party-Id header to the message: `MessageManipulations 0 = 1, invite, ,header.REMOTE-PARTY-ID, 0, '<sip:999@10.132.10.108>;party=calling', 0;`

	Result:	`Remote-Party-ID: <sip:999@10.132.10.108>;party=calling;npi=0;ton=0`
Example 2	**Rule:**	Create a Remote-Party-Id header using the url in the From header using the + operator to concatenate strings: `MessageManipulations 0 = 1, Invite, ,header.REMOTE-PARTY-ID, 0, '<'+header.from.url +'>' + ';party=calling', 0;`
	Result:	`Remote-Party-ID: <sip:555@10.132.10.128;user=phone>;party=calling;npi=0;ton=0`
Example 3	**Rule:**	Modify the number plan to 1 (ISDN): `MessageManipulations 1 = 1, invite, , header.Remote-Party-ID.numberplan, 2, '1', 0;`
	Result:	`Remote-Party-ID: <sip:555@10.132.10.128;user=phone>;party=calling;npi=1;ton=0`
Example 4	**Rule:**	Modify the Remote-Party-Id header to set the privacy parameter to 1 (Full): `MessageManipulations 1 = 1, invite, , header.Remote-Party-ID.privacy, 2, '1', 0;`
	Result:	`Remote-Party-ID: <sip:555@10.132.10.128;user=phone>;party=calling;privacy=full;npi=0;ton=0`

B.2.23 Request-Uri

An example of the header is shown below:

```
sip:alice:secretword@atlanta.com;transport=tcp
SIP/2.0 486 Busy Here
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	No	No	Yes	NA

Keyword	Sub Types	Attributes
Method	String	Read/Write
MethodType	Enum	Read/Write
URI	String	Read/Write
URL	URL Structure (see 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Test the Request-URI transport type. If 1 (TCP), then modify the URL portion of the From header: `MessageManipulations 1 = 1, Invite.request, header.REQUEST-URI.url.user == '101', header.REMOTE-PARTY-ID.url, 2, 'sip:3200@110.18.5.41;tusunami=0', 0;`

	Result:	`Remote-Party-ID: <sip:3200@110.18.5.41;tusunami=0>;party=calling;npi=0; ton=0`
Example 2	**Rule:**	If the method type is 5 (INVITE), then modify the Remote-Party-Id header: `MessageManipulations 2 = 1, Invite.request, header.REQUEST-URI.methodtype == '5', header.REMOTE-PARTY-ID.url, 2, 'sip:3200@110.18.5.41;tusunami=0', 0;`
	Result:	`Remote-Party-ID: <sip:3200@110.18.5.41;tusunami=0>;party=calling;npi=0; ton=0`
Example 3	**Rule:**	For all request URI's whose method types are 488, modify the message type to a 486: `MessageManipulations 1 = 1, , header.request-uri.methodtype=='488', header.request-uri.methodtype, 2, '486', 0;`
	Result:	`SIP/2.0 486 Busy Here`

B.2.24 Require

An example of the header is shown below:

```
Require: 100rel
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Capabilities	SIPCapabilities Struct	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Require header to all messages: `MessageManipulations 1 = 1, , ,header.require, 0, 'early-session,em,replaces', 0;`
	Result:	`Require: em,replaces,early-session`
Example 2	**Rule:**	If a Require header exists, then delete it: `MessageManipulations 2 = 1, Invite, header.require exists ,header.require, 1, '', 0;`
	Result:	The Require header is deleted.
Example 3	**Rule:**	Set the early media options tag in the header: `MessageManipulations 0 = 0, invite, , header.require.earlymedia, 0, 1 , 0;`
	Result:	`Require: em,replaces,early-session, early-media`
Example 4	**Rule:**	Set the privacy options tag in the Require header: `MessageManipulations 0 = 0, invite, , header.require.privacy, 0, 1 , 0;`
	Result:	`Require: em,replaces,early-session, privacy`

B.2.25 Resource-Priority

An example of the header is shown below:

```
Resource-Priority: wps.3
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	2

Keyword	Sub Types	Attributes
Namespace	String	Read/Write
RPriority	String	Read/Write

B.2.26 Retry-After

An example of the header is shown below:

```
Retry-After: 18000
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Time	Integer	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Retry-After header: `MessageManipulations 2 = 1, Invite,  ,header.Retry-After, 0, '3600', 0;`
	Result:	`Retry-After: 3600`
Example 2	**Rule:**	Modify the Retry-Time in the header to 1800: `MessageManipulations 3 = 1, Invite,  ,header.Retry-After.time, 2, '1800', 0;`
	Result:	`Retry-After: 1800`

B.2.27 Server or User-Agent

An example of the header is shown below:

```
User-Agent: Sip Message Generator V1.0.0.5
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
N/A	N/A	N/A

Below are header manipulation examples:

Example 1	**Rule:**	Remove the User-Agent header: `MessageManipulations 2 = 1, Invite,  ,header.user-agent, 1, '', 0;`
	Result:	The header is removed.
Example 2	**Rule:**	Change the user agent name in the header: `MessageManipulations 3 = 1, Invite,  ,header.user-agent, 2, 'itsp analogue gateway', 0;`
	Result:	`User-Agent: itsp analog gateway`

B.2.28 Service-Route

An example of the header is shown below:

```
Service-Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
<sip:HSP.HOME.EXAMPLE.COM;lr>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	7

Keyword	Sub Types	Attributes
ServiceRoute	String	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add two Service-Route headers:
		`MessageManipulations 1 = 1, Invite,  ,header.service-route, 0, '<P2.HOME.EXAMPLE.COM;lr>', 0;`
		`MessageManipulations 2 = 1, Invite,  ,header.service-route, 0, '<sip:HSP.HOME.EXAMPLE.COM;lr>', 0;`

	Result:	`Service-Route:<P2.HOME.EXAMPLE.COM;lr>` `Service-Route: <sip:HSP.HOME.EXAMPLE.COM;lr>`
Example 2	**Rule:**	Modify the Service-Route header in list entry 1: `MessageManipulations 3 = 1, Invite,  ,header.service-`

		`route.1.serviceroute, 2, '<sip:itsp.com;lr>', 0;`
	Result:	`Service-Route:sip:itsp.com;lr` `Service-Route: <sip:HSP.HOME.EXAMPLE.COM;lr>`
Example 3	**Rule:**	Modify the Service-Route header in list entry 0: `MessageManipulations 4 = 1, Invite, ,header.service-route.0.serviceroute, 2, '<sip:home.itsp.com;lr>', 0;`
	Result:	`Service-Route:sip:home.itsp.com;lr` `Service-Route: <sip:itsp.com;lr>`

B.2.29 Session-Expires

An example of the header is shown below:

```
Session-Expires: 480
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Param	Param	Read/Write
Refresher	Enum Refresher (see 'Refresher' on page 1050)	Read/Write
Time	Integer	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add a Session-Expires header: `MessageManipulations 0 = 1, any, , header.Session-Expires, 0, '48' + '0', 0;`
	Result:	`Session-Expires: 480`
Example 2	**Rule:**	Modify the Session-Expires header to 300: `MessageManipulations 1 = 1, any, , header.Session-Expires.time, 2, '300', 0;`
	Result:	`Session-Expires: 300`
Example 3	**Rule:**	Add a param called longtimer to the header: `MessageManipulations 1 = 1, any, , header.Session-Expires.param.longtimer, 0, '5', 0;`
	Result:	`Session-Expires: 480;longtimer=5`
Example 4	**Rule:**	Set the refresher to 1 (UAC): `MessageManipulations 3 = 1, any, , header.session-expires.refresher, 2, '1', 0;`
	Result:	`Session-Expires: 300;refresher=uac;longtimer=5`

B.2.30 Subject

An example of the header is shown below:

```
Subject: A tornado is heading our way!
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Subject	String	Read/Write

Below is a header manipulation example:

Rule:	Add a Subject header: `MessageManipulations 0 = 1, any, , header.Subject, 0, 'A tornado is heading our way!', 0;`
Result:	`Subject: A tornado is heading our way!`

B.2.31 Supported

An example of the header is shown below:

```
Supported: early-session
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Capabilities	SIPCapabilities Struct	Read/Write

Below is a header manipulation example:

Example 1	**Rule:**	Add a Supported header: `MessageManipulations 1 = 1, Invite, ,header.supported, 0, 'early-session, 0;`
	Result:	`Supported: early-session`
Example 2	**Rule:**	Set path in the Supported headers options tag: `MessageManipulations 0 = 0, invite, , header.supported.path, 0, true, 0;`
	Result:	`Supported: early-session, path`

B.2.32 To

An example of the header is shown below:

```
To: <sip:101@10.132.10.128;user=phone>
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	No	No	No	NA

Keyword	Sub Types	Attributes
Name	String	Read/Write
Param	Param	Read/Write
tag	String	Read Only
URL	URL Structure (refer to 'URL' on page 1041)	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Set the user phone Boolean to be false in the To header's URL: `MessageManipulations 4 = 1, invite.request, , header.to.url.UserPhone, 2, '0', 0;`
	Result:	`To: <sip:101@10.132.10.128>`
Example 2	**Rule:**	Change the URL in the To header: `MessageManipulations 4 = 1, invite.request, , header.to.url.UserPhone, 2, '0', 0;`
	Result:	`To: <sip:101@10.20.30.60:65100>`
Example 3	**Rule:**	Set the display name to 'Bob': `MessageManipulations 5 = 1, invite.request, , header.to.name, 2, 'Bob', 0;`
	Result:	`To: "Bob Dylan" sip:101@10.20.30.60:65100`
Example 4	**Rule:**	Add a proprietary parameter to all To headers: `MessageManipulations 6 = 1, invite.request, , header.to.param.artist, 0, 'singer', 0;`
	Result:	`To: "Bob Dylan" <sip:101@10.20.30.60:65100>;artist=singer`

B.2.33 Unsupported

An example of the header is shown below:

```
Unsupported: 100rel
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	N/A

Keyword	Sub Types	Attributes
Capabilities	SIPCapabilities Struct	Read/Write

Below are header manipulation examples:

Example 1	**Rule:**	Add an Unsupported header to the message: `MessageManipulations 0 = 1, Invite.response, ,header.unsupported, 0, 'early-session, myUnsupportedHeader', 0;`
	Result:	`Unsupported: early-session`
Example 2	**Rule:**	Modify the Unsupported header to 'replaces': `MessageManipulations 1 = 1, Invite, ,header.unsupported, 2, 'replaces', 0;`
	Result:	`Unsupported: replaces`
Example 3	**Rule:**	Set the path in the Unsupported headers options tag: `MessageManipulations 0 = 0, invite, , header.unsupported.path, 0, true, 0;`
	Result:	`Unsupported: replaces, path`

B.2.34 Via

An example of the header is shown below:

```
Via: SIP/2.0/UDP 10.132.10.128;branch=z9hG4bKUGOKMQPAVFKTAVYDQPTB
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	No	No	No	10

Keyword	Sub Types	Attributes
Alias	Boolean	Read Only
Branch	String	Read Only
Host	Host Structure (see 'Host' on page 1039)	Read Only
MAddrIp	gnTIPAddress	Read Only
Param	Param	Read/Write

Keyword	Sub Types	Attributes
Port	Integer	Read Only
TransportType	Enum TransportType (see 'TransportType' on page 1051)	Read Only

Below is a header manipulation example:

Rule:	Check the transport type in the first Via header and if it's set to UDP, then modify the From header's URL: `MessageManipulations 0 = 1, Invite.request, header.VIA.0.transporttype == '0', header.from.url, 2, 'sip:3200@110.18.5.41;tusunami=0', 0;`
Result:	`From: <sip:3200@110.18.5.41;user=phone;tusunami=0>;tag=1c7874`

B.2.35 Warning

An example of the header is shown below:

```
Warning: 307 isi.edu "Session parameter 'foo' not understood"
Warning: 301 isi.edu "Incompatible network address type 'E.164'"
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	1

Keyword	Sub Types	Attributes
N/A	N/A	N/A

Below is a header manipulation example:

Rule:	Add a Warning header to the message: `MessageManipulations 0 = 1, Invite.response.180, ,header.warning, 0, 'Incompatible 380', 0;`
Result:	`Warning: Incompatible 380`

B.2.36 Unknown Header

An Unknown header is a SIP header that is not included in this list of supported headers. An example of the header is shown below:

```
MYEXP: scooby, doo, goo, foo
```

The header properties are shown in the table below:

Header Level Action	Add	Delete	Modify	List Entries
Operations Supported	Yes	Yes	Yes	3

Keyword	Sub Types	Attributes
N/A	N/A	N/A

Below are header manipulation examples:

Example 1	**Rule:**	Add a custom header to all messages: `MessageManipulations 0 = 1, , , header.myExp, 0, 'scooby, doo, goo, foo', 0;`
	Result:	`MYEXP: scooby, doo, goo, foo`
Example 2	**Rule:**	Take the value from the Expires parameter in the Contact header, append 00 to the value and create a new myExp header: `MessageManipulations 0 = 1, any, , header.media, 0, header.Session-Expires.time + 'ooo' + ';refresher=' + header.Session-Expires.Refresher, 0;`
	Result:	`MEDIA: 3600ooo;refresher=1`

Example 3	**Rule:**	Create lists of Unknown headers:
		`MessageManipulations 1 = 1, Invite, , header.myExp.1, 0, 'scooby, doo, goo, foo1', 0;`
		`MessageManipulations 2 = 1, Invite, , header.myExp.2, 0, 'scooby, doo, goo, foo2', 0;`

	Result:	`MYEXP: scooby, doo, goo, foo1` `MYEXP: scooby, doo, goo, foo2`
Example 4	**Rule:**	Remove the SIP header 'colour' from INVITE messages: `MessageManipulations 1 = 1, Invite, , header.colour, 1, '', 0;`
	Result:	The colour header is removed.

B.3 Structure Definitions

B.3.1 Event Structure

The Event structure is used in the Event header (see 'Event' on page 1019).

Table B-2: Event Structure

Keyword	Sub Types	Attributes
EventPackage	Enum Event Package (see 'Event Package' on page 1045)	Read/Write
EventPackageString*	String	Read/Write
Id	String	Read/Write

Event package string is used for packages that are not listed in the Enum Event Package table (see 'Event Package' on page 1045).

B.3.2 Host

The host structure is applicable to the URL structure (see 'URL' on page 1041) and the Via header (see 'Via' on page 1036).

Table B-3: Host Structure

Keyword	Sub Types
Port	Short
Name	String

B.3.3 MLPP

This structure is applicable to the Reason header (see 'Reason' on page 1026).

Table B-4: MLPP Structure

Keyword	Sub Types
Type	Enum MLPP Reason (see 'MLPP Reason Type' on page 1046)
Cause	Int

B.3.4 Privacy Struct

This structure is applicable to the Privacy header (see 'Privacy' on page 1025).

Table B-5: Privacy Structure

Keyword	Sub Types
NONE	Boolean
HEADER	Boolean
SESSION	Boolean
USER	Boolean
CRITICAL	Boolean
IDENTITY	Boolean
HISTORY	Boolean

B.3.5 Reason Structure

This structure is applicable to the Reason header (see 'Reason' on page 1026).

Table B-6: Reason Structure

Keyword	Sub Types
Reason	Enum Reason (see 'Reason (Reason Structure)' on page 1047)
Cause	Int
Text	String

B.3.6 SIPCapabilities

This structure is applicable to the following headers:

- Supported (see 'Supported' on page 1034)
- Require (see 'Require' on page 1030)
- Proxy-Require (see 'Proxy-Require' on page 1025)
- Unsupported (see 'Unsupported' on page 1036)

Table B-7: SIPCapabilities Structure

Keyword	Sub Types
EarlyMedia	Boolean
ReliableResponse	Boolean
Timer	Boolean
EarlySession	Boolean
Privacy	Boolean
Replaces	Boolean

Keyword	Sub Types
History	Boolean
Unknown	Boolean
GRUU	Boolean
ResourcePriority	Boolean
TargetDialog	Boolean
SdpAnat	Boolean

B.3.7 URL

This structure is applicable to the following headers:

- Contact (see 'Contact' on page 1017)
- Diversion (see 'Diversion' on page 1018)
- From (see 'From' on page 1020)
- P-Asserted-Identity (see 'P-Asserted-Identity' on page 1022)
- P-Associated-Uri (see 'P-Associated-Uri' on page 1022)
- P-Called-Party-Id (see 'P-Called-Party-Id' on page 1023)
- P-Preferred-Identity (see 'P-Preferred-Identity' on page 1024)
- Referred-By (see 'Referred-By' on page 1027)
- Refer-To (see 'Refer-To' on page 1027)
- Remote-Party-Id (see 'Remote-Party-Id' on page 1028)
- Request-Uri (see 'Request-Uri' on page 1029)
- To (see 'To' on page 1035)

Table B-8: URL Structure

Keyword	Sub Types
Type	Enum Type (see 'Type' on page 1051)
Host	Host Structure (see 'Host' on page 1039)
MHost	Structure
UserPhone	Boolean
LooseRoute	Boolean
User	String
TransportType	Enum Transport (see 'TransportType' on page 1051)
Param	Param

B.4 Random Type

Manipulation rules can include random strings and integers. An example of a manipulation rule using random values is shown below:

```
MessageManipulations 4 = 1, Invite.Request, , Header.john, 0,
rand.string.56.A.Z, 0;
```

In this example, a header called "john" is added to all INVITE messages received by the device and a random string of 56 characters containing characters A through Z is added to the header.

For a description of using random values, see the subsequent subsections.

B.4.1 Random Strings

The device can generate random strings in header manipulation rules that may be substituted where the type 'String' is required. The random string can include up to 298 characters and include a range of, for example, from a to z or 1 to 10. This string is used in the table's 'Action Value' field.

The syntax for using random strings is:

```
Rand.string.<number of characters in string>.<low character>.<high
character>
```

Examples:

- Rand.string.5.a.z: This generates a 5-character string using characters a through z.
- Rand.string.8.0.z: This generates an 8-character string using characters and digits.

B.4.2 Random Integers

The device can generate a random numeric value that may be substituted where the type 'Int' is required. The syntax for random numeric values is:

```
Rand.number.<low number>.<high number>
```

Examples:

- Rand.number.5.32: This generates an integer between 5 and 32

B.5 Wildcarding for Header Removal

The device supports the use of the "*" wildcard character to remove headers. The "*" character may only appear at the end of a string. For example, "X-*" is a valid wildcard request, but "X-*ID" is not.

Below are examples of using the wildcard:

- header.p-* - removes all headers that have the prefix "p-"
- header.via* - removes all Via headers
- header.x-vendor* - removes all headers that start with "x-vendor"
- header.* - removes all non-critical headers
- header.to* - removes all headers that start with "to", except the To header, which is protected

Note: The wildcard does not remove the following headers: Request-Uri, Via, From, To, Callid, Cseq, and Contact.

B.6 Message Manipulation using SDP Conditions

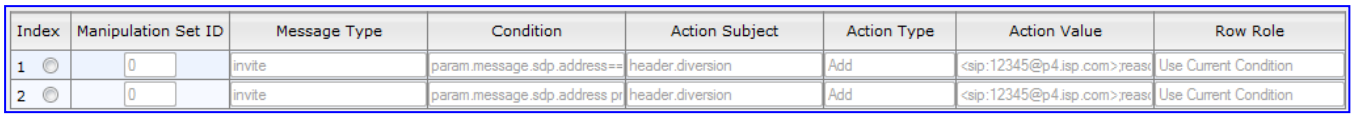
You can configure message manipulation rules based on user-defined SDP conditions. For example, you can configure a manipulation rule to add a Diversion header to incoming INVITE messages if the SDP contains a specific IP address, or a prefix or suffix of this IP address.

The device supports the following SDP condition syntax:

- **param.message.sdp.address suffix** '10.10'
- **param.message.sdp.address prefix** '10.132'
- **param.message.sdp.address==**'10.33.37.78'

The example below shows a manipulation rule that adds a Diversion header ("Diversion: <sip:12345@p4.isp.com>;reason=no-answer") to incoming INVITE messages, if the SDP contains the IP address 10.33.37.78 or the prefix of this IP address, i.e., 10.33. The IP address is contained in the "c=" line of the SDP (e.g., "c=IN IP4 10.33.37.75").

Figure B-1: Message Manipulation using SDP Conditions



Index	Manipulation Set ID	Message Type	Condition	Action Subject	Action Type	Action Value	Row Role
1	0	invite	param.message.sdp.address==	header.diversion	Add	<sip:12345@p4.isp.com>;reas	Use Current Condition
2	0	invite	param.message.sdp.address pr	header.diversion	Add	<sip:12345@p4.isp.com>;reas	Use Current Condition

The table below shows the example configuration:

Parameter	Rule Index 1	Rule Index 2
Message Type	invite	invite
Condition	param.message.sdp.address=='10.33.37.78'	param.message.sdp.address=='10.33'
Action Subject	header.diversion	header.diversion
Action Type	Add	Add
Action Value	<sip:12345@p4.isp.com>;reason=no-answer	<sip:12345@p4.isp.com>;reason=no-answer

You can configure several such manipulation rules and then apply them per IP Group using the 'Inbound Message Manipulation Set' parameter.

Note: This feature is applicable only to the SBC application.

B.7 Copying Information between Messages using Variables

You can use variables in SIP message manipulation rules to copy specific information from one message to another. Information from one message is copied to a variable and then information from that variable is copied to any subsequent message. The device can store information in local or global variables. Local variables are stored on a per call basis and change when a new call is made. Up to two local variables can be used per call. Global variables do not change as new calls are made. Up to 10 global variables can be used.

The syntax for using variables is as follows:

- Var.call.<src || dst><local index>

  where *local index* is an integer between 1 and 2 inclusive
- Var.global.<global index>

  where *global index* is an integer between 1 and 10 inclusive

To store data in a variable, add the name of the variable in the Action Subject field and set the Action Type to Modify. To retrieve data from a variable, add it in the Action Value field and it can be used in any manipulation where a ManStringElement is valid as an Action Subject.

Below are examples of manipulation rules implementing variables:

- Example 1:
  - Store a value in a call variable: Stores the subject URI parameter from the To header:

    ```
    MessageManipulations 0 = 0, Invite.Request, ,
    var.call.dst.1, 2, header.to.url.param.subject, 0;
    ```
  - Use the stored value: Allocates a Subject header for the 200 OK response for the same call and assigns it the stored value:

    ```
    MessageManipulations 0 = 0, Invite.response.200, ,
    header.subject, 0, var.call.dst.1, 0;
    ```
- Example 2:
  - Store a value in a global variable: Stores the Priority header of the INVITE with 'company' in the host part of the From header:

    ```
    MessageManipulations 0 = 0, Invite.Request,
    header.from.url.host == 'company', var.global.1, 2,
    header.priority, 0;
    ```
  - Use the stored value: Assigns the same priority as the INVITE request to SUBSCRIBE requests arriving with 'company' in the host part of the From header:

    ```
    MessageManipulations 0 = 0, Subscribe.request,
    header.from.url.host == 'company', header.priority, 0,
    var.global.1, 0;
    ```

B.8 Enum Definitions

B.8.1 AgentRole

These ENUMs are applicable to the Server or User-Agent headers (see 'Server or User-Agent' on page 1032).

Table B-9: Enum Agent Role

AgentRole	Value
Client	1
Server	2

B.8.2 Event Package

These ENUMs are applicable to the Server or User-Agent (see 'Server or User-Agent' on page 1032) and Event (see 'Event' on page 1019) headers.

Table B-10: Enum Event Package

Package	Value
TELEPHONY	1
REFER	2
REFRESH	3
LINE_STATUS	4
MESSAGE_SUMMARY	5
RTCPXR	6
SOFT_SYNC	7
CHECK_SYNC	8
PSTN	9
DIALOG_PACKAGE	10
REGISTRATION	11
START_CWT	12
STOP_CWT	13
UA_PROFILE	14
LINE_SEIZE	15

B.8.3 MLPP Reason Type

These ENUMs are applicable to the MLPP Structure (see 'MLPP' on page 1039).

Table B-11: Enum MLPP Reason Type

Type	Value
PreEmption Reason	0
MLPP Reason	1

B.8.4 Number Plan

These ENUMs are applicable to the Remote-Party-Id header (see 'Remote-Party-Id' on page 1028).

Table B-12: Enum Number Plan

Plan	Value
ISDN	1
Data	3
Telex	4
National	8
Private	9
Reserved	15

B.8.5 NumberType

These ENUMs are applicable to the Remote-Party-Id header (see 'Remote-Party-Id' on page 1028).

Table B-13: Enum Number Type

Number Type	Value
INTERNATIONAL LEVEL2 REGIONAL	1
NATIONAL LEVEL1 REGIONAL	2
NETWORK PISN SPECIFIC NUMBER	3
SUBSCRIBE LOCAL	4
ABBREVIATED	6
RESERVED EXTENSION	7

B.8.6 Privacy

These ENUMs are applicable to the Remote-Party-Id (see 'Remote-Party-Id' on page 1028) and Diversion (see 'Diversion' on page 1018) headers.

Table B-14: Enum Privacy

Privacy Role	Value
Full	1
Off	2

B.8.7 Reason (Diversion)

These ENUMs are applicable to the Diversion header (see 'Diversion' on page 1018).

Table B-15: Enum Reason

Reason	Value
Busy	1
No Answer	2
Unconditional	3
Deflection	4
Unavailable	5
No Reason	6
Out of service	7

B.8.8 Reason (Reason Structure)

These ENUMs are used in the Reason Structure (see 'Reason Structure' on page 1040).

Table B-16: Enum Reason (Reason Structure)

Reason	Value
INVITE	5
REINVITE	6
BYE	7
OPTIONS	8
ACK	9
CANCEL	10
REGISTER	11
INFO	12
MESSAGE	13
NOTIFY	14

Reason	Value
REFER	15
SUBSCRIBE	16
PRACK	17
UPDATE	18
PUBLISH	19
LAST_REQUEST	20
TRYING_100	100
RINGING_180	180
CALL_FORWARD_181	181
QUEUED_182	182
SESSION_PROGRESS_183	183
OK_200	200
ACCEPTED_202	202
MULTIPLE_CHOICE_300	300
MOVED_PERMANENTLY_301	301
MOVED_TEMPORARILY_302	302
SEE_OTHER_303	303
USE_PROXY_305	305
ALTERNATIVE_SERVICE_380	380
BAD_REQUEST_400	400
UNAUTHORIZED_401	401
PAYMENT_REQUIRED_402	402
FORBIDDEN_403	403
NOT_FOUND_404	404
METHOD_NOT_ALLOWED_405	405
NOT_ACCEPTABLE_406	406
AUTHENTICATION_REQUIRED_407	407
REQUEST_TIMEOUT_408	408
CONFLICT_409	409
GONE_410	410
LENGTH_REQUIRED_411	411
CONDITIONAL_REQUEST_FAILED_412	412
REQUEST_TOO_LARGE_413	413
REQUEST_URI_TOO_LONG_414	414
UNSUPPORTED_MEDIA_415	415
UNSUPPORTED_URI_SCHEME_416	416
UNKNOWN_RESOURCE_PRIORITY_417	417

Reason	Value
BAD_EXTENSION_420	420
EXTENSION_REQUIRED_421	421
SESSION_INTERVAL_TOO_SMALL_422	422
SESSION_INTERVAL_TOO_SMALL_423	423
ANONYMITY_DISALLOWED_433	433
UNAVAILABLE_480	480
TRANSACTION_NOT_EXIST_481	481
LOOP_DETECTED_482	482
TOO_MANY_HOPS_483	483
ADDRESS_INCOMPLETE_484	484
AMBIGUOUS_485	485
BUSY_486	486
REQUEST_TERMINATED_487	
NOT_ACCEPTABLE_HERE_488	488
BAD_EVENT_489	489
REQUEST_PENDING_491	491
UNDECIPHERABLE_493	493
SECURITY_AGREEMENT_NEEDED_494	494
SERVER_INTERNAL_ERROR_500	500
NOT_IMPLEMENTED_501	501
BAD_GATEWAY_502	502
SERVICE_UNAVAILABLE_503	503
SERVER_TIME_OUT_504	504
VERSION_NOT_SUPPORTED_505	505
MESSAGE_TOO_LARGE_513	513
PRECONDITION_FAILURE_580	580
BUSY_EVERYWHERE_600	600
DECLINE_603	603
DOES_NOT_EXIST_ANYWHERE_604	604
NOT_ACCEPTABLE_606	606

B.8.9 Reason (Remote-Party-Id)

These ENUMs are applicable to the Remote-Party-Id header (see 'Remote-Party-Id' on page 1028).

Table B-17: Enum Reason (RPI)

Reason	Value
Busy	1
Immediate	2
No Answer	3

B.8.10 Refresher

These ENUMs are used in the Session-Expires header (see 'Session-Expires' on page 1033).

Table B-18: Enum Refresher

Refresher String	Value
UAC	1
UAS	2

B.8.11 Screen

These ENUMs are applicable to the Remote-Party-Id (see 'Remote-Party-Id' on page 1028) and Diversion (see 'Diversion' on page 1018) headers.

Table B-19: Enum Screen

Screen	Value
Yes	1
No	2

B.8.12 ScreenInd

These ENUMs are applicable to the Remote-Party-Id header (see 'Remote-Party-Id' on page 1028).

Table B-20: Enum ScreenInd

Screen	Value
User Provided	0
User Passed	1
User Failed	2
Network Provided	3

B.8.13 TransportType

These ENUMs are applicable to the URL Structure (see 'URL' on page 1041) and the Via header (see 'Via' on page 1036).

Table B-21: Enum TransportType

TransportType	Value
UDP	0
TCP	1
TLS	2
SCTP	3

B.8.14 Type

These ENUMs are applicable to the URL Structure (see 'URL' on page 1041).

Table B-22: Enum Type

Type	Value
SIP	1
Tel	2
Fax	3
SIPS	4

B.9 Actions and Types

Table 43-23: Action and Types

Element Type	Command Type	Command	Value Type	Remarks
IPGroup	Match	==	String	Returns true if the parameter equals to the value.
		!=	String	Returns true if the parameter not equals to the value.
		contains	String	Returns true if the string given is found in the parameter value.
		!contains	String	Returns true if the string given is not found in the parameter value.
Call-Parameter	Match	==	String	Returns true if the parameter equals to the value.
		!=	String	Returns true if the parameter not equals to the value.
		contains	String	Returns true if the string given is found in the parameter value.
		!contains	String	Returns true if the string given is not found in the parameter value.
Body	Match	==	String	Returns true if the body's content equals to the value.
		!=	String	Returns true if the body's content not equals to the value.
		contains	String	Returns true if the string given is found in the body's content.
		!contains	String	Returns true if the string given is not found in the body's content.
		exists		Returns true if this body type exists in the message.
		!exists		Returns true if this body type does not exist in the message.
	Action	Modify	String	Modifies the body content to the new value.
		Add	String	Adds a new body to the message. If such body exists the body content will be modified.
		Remove		Removes the body type from the message.
Header-List	Match	==	String *Header-list	Returns true if the header's list equals to the string.
		!=	String	Returns true if the header's list not equals to the string.

Element Type	Command Type	Command	Value Type	Remarks
			*Header-list	
		contains	String	Returns true if the header's list contains the string.
		!contains	String	Returns true if the header's list does not contain the string.
		exists		Returns true if at least one header exists in the list.
		!exists		Returns true if no headers exist in the list.
	Action	Modify	String *Header	Removes all the headers from the list and allocates a new header with the given value.
		Add	String *Header	Adds a new header to the end of the list.
		Remove		Removes the whole list from the message.
Header	Match	==	String *Header	Returns true if a header equals to the value. The header element must not be a list.
		!=	String *Header	Returns true if a header not equals to the value. The header element must not be a list.
		contains	String	Returns true if the header contains the string.
		!contains	String	Returns true if the header does not contain the string.
		exists		Returns true if the header exists.
		!exists		Returns true if the header does not exist.
	Action	Modify	String *Header	Replaces the entire header with the new value.
		Remove		Removes the header from the message, if the header is part of a list only that header will be removed.
		Add	String *Header	Adds a new header to the end of the list.
Parameter-List	Match	==	String Parameter-list	Returns true if the header's list equals to the string.
		!=	String Parameter-list	Returns true if the header's list not equals to the string.

Element Type	Command Type	Command	Value Type	Remarks
		contains	String	Returns true if the header's list contains the string.
		!contains	String	Returns true if the header's list does not contain the string.
		exists		Returns true if at least one parameter exists in the list.
		!exists		Returns true if the header's parameter list is empty.
	Action	Modify	String Parameter-list	Replaces the current parameters with the new value.
		Add	String Parameter	Adds a new parameter to the parameter's list.
		Remove		Removes all the unknown parameters from the list.
Parameter	Match	==	String Parameter	Returns true if the header's parameter's value equals to the value.
		!=	String Parameter	Returns true if the header's parameter's value not equals to the value.
		contains	String	Returns true if the header's parameter contains the string.
		!contains	String	Returns true if the header's parameter does not contain the string.
		exists		Returns true if the header's parameter exists.
		!exists		Returns true if the header's parameter does not exist.
	Action	Modify	String Parameter	Sets the header's parameter to the value.
		Remove		Removes the header's parameter from the parameter list.
Structure	Match	==	String *Structure	Returns true if the header's structure's value equals to the value. The string given must be able to be parsed to the structure.
		!=	String *Structure	Returns true if the header's structure's value not equals to the value. The string given must be able to be parsed to the structure.
	Action	Modify	String *Structure	Sets the header's structure to the value. The string given must be able to be parsed to the structure.

Element Type	Command Type	Command	Value Type	Remarks
Integer	Match	==	Integer	Returns true if value equals to the integer element
		!=	Integer	Returns true if value not equals to the integer element
		>	Integer	Returns true if value is greater than the value.
		>=	Integer	Returns true if value is greater than or equals to the value.
		<	Integer	Returns true if value is less than the value.
		<=	Integer	Returns true if value is less than or equals to the value.
	Action	Modify	Integer	Sets the integer element to the value. A string value must be a representation of an integer.
String	Match	==	String	Returns true if the string element equals to the value.
		!=	String	Returns true if the string element not equals to the value.
		contains	String	Returns true if the value is found in the string element.
		!contains	String	Returns true if the value is not found in the string element.
		>	String	Performs a character by character compare. Returns true if the ASCII value of the character is greater than that in the value
		>=	String	Performs a character by character compare. Returns true if the ASCII value of the character is greater than or equal to that in the value
		<	String	Performs a character by character compare. Returns true if the ASCII value of the character is less than that in the value
		<=	String	Performs a character by character compare. Returns true if the ASCII value of the character is less than or equal to that in the value
	Action	Modify	String	Sets the string element to the value.
		Add prefix	String	Adds the value to the beginning of the string element.
		Remove prefix	String	Removes the value from the beginning of the string element.
		Add suffix	String	Adds the value to the end of the string element.

Element Type	Command Type	Command	Value Type	Remarks
		Remove suffix	String	Removes the value from the end of the string element.
Boolean	Match	==	Boolean	Returns true if the Boolean element equals to the value. Boolean – can be either 0 or 1.
		!=	Boolean	Returns true if the Boolean element not equals to the value. Boolean – can be either 0 or 1.
		>	Boolean	Returns true if the Boolean element not equals to the value. Boolean – can be either 0 or 1.
		<	Boolean	Returns true if the Boolean element not equals to the value. Boolean – can be either 0 or 1.
	Action	Modify	Boolean	Sets the Boolean element to the value. Boolean – can be either 0 or 1.
Attribute	Match	==	Integer *Attribute	Returns true if the attribute element equals to the value. An attribute element value must be of the same type of the attribute element.
		!=	Integer *Attribute	Returns true if the attribute element not equals to the value. An attribute element value must be of the same type of the attribute element.
	Action	Modify	Integer *Attribute	Sets the attribute element to the value. An attribute element value must be of the same type of the attribute element.

B.10 Syntax

Rules table:

Man Set ID	Message Type	Condition	Action Element	Action Type	Action Value	Row Rule
ID	<message-type>	<match-condition>	<message-element>	<action-type>	<value>	ID

1. **message-type:**

   Description: Rule is applied only if this is the message's type

   Syntax: <method>.<message-role>

   Examples:

   - invite.request
   - invite.response.200
   - subscribe.response.2xx

   a. **method:**

      Description: Rule is applied only if this is the message's method

      Syntax: token / any

      Examples:

      - Invite, subscribe — rule applies only to INVITE messages
      - Unknown — unknown methods are also allowed
      - Any — no limitation on the method type

   b. **message-role**

      Description: Rule is applied only if this is the message's role

      Syntax: request / response.response-code / any

      Examples:

      - Request — rule applies only on requests
      - Response.200 — rule applies only on 200 OK messages
      - Any — no limitations on the type of the message

   c. **response-code**

      Description: Response code of the message

      Syntax: 1xx / 2xx / 3xx / 4xx / 5xx / 6xx / 3digit / any

      Examples:

      - 3xx — any redirection response
      - 200 — only 200 OK response
      - Any — any response

2. **match-condition:**

   Description: Matching criteria for the rule

   Syntax: (message-element / param) SWS match-type [SWS value] * [SWS logical-expression SWS match-condition]

   Examples:

   - header.from.user == '100'
   - header.contact.header-param.expires > '3600'
   - header.to.host contains 'itsp'

- param.call.dst.user != '100'
- header.john exists
- header.john exists AND header.to.host !contains 'john'
- header.from.user == '100' OR header.from.user == '102' OR header.from.user == '300'

a. match-type

Description: Comparison to be made

Syntax:

- == equals
- != not equals
- > greater than
- < less than
- >= greater than or equal to
- <= less than or equal to
- contains does a string contain a value (relevant only to string fields)
- exists does a certain header exists
- !exists does a certain header not exists
- !contains does a string exclude a value. Relevant only to string fields

3. logical-expression:

Description: Condition for the logical expression

Syntax:

- AND logical And
- OR logical Or

Note: "A AND B OR C" is calculated as A AND (B OR C).

4. message-element:

Description: Element in the message

Syntax: (header / body).message-element-name [.header-index] * [.(sub-element / sub-element-param)]

Examples:

- header.from
- header.via.2.host
- header.contact.header-param.expires
- header.to.uri-param.user-param
- body.application/dtmf-relay

a. message-element-name

Description: Name of the message's element - "/" only used for body types

Syntax: 1 * (token / "/")

Examples:

- from (header's name)
- to (header's name)
- application/dtmf-relay (body's name)

b. header-index

Description: Header's index in the list of headers

Syntax: Integer

Examples: If five Via headers arrive:

- 0 (default) refers to first Via header in message
- 1 second Via header

- 4 fifth Via header

c. sub-element

Description: Header's element

Syntax: sub-element-name

Examples:

- user
- host

d. sub-element-param

Description: Header's element

Syntax: sub-element-name [.sub-element-param-name]

Example:

- header.from.param.expires

e. sub-element-param-name

Description: Header's parameter name - relevant only to parameter sub-elements

Syntax: token

Examples:

- expires (contact's header's param)
- duration (retry-after header's param)
- unknown-param (any unknown param can be added/removed from the header)

f. param

Description: Params can be as values for match and action

Syntax: param.param-sub-element.param-dir-element.(call-param-entity / ipg-param-entity)

Examples:

- param.ipg. src.user
- param.ipg.dst.host
- param.ipg.src.type
- param.call.src.user

g. param-sub-element

Description: Determines whether the param being accessed is a call or an IP Group

Syntax:

- call relates to source or destination URI for the call
- ipg relates to source or destination IP Group

h. param-dir-element

Description: Direction relating to the classification

Syntax:

- src refers to source
- ds refers to destination

i. call-param-entity

Description: Parameters that can be accessed on the call

Syntax:

- user refers to username in request-URI for call

j. ipg-param-entity

Description: Name of the parameter

Syntax:

- user refers to Contact user in IP Group
- host refers to Group Name in IP Group table
- type refers to Type field in IP Group table
- id refers to IP Group ID (used to identify source or destination IP Group)

k. string

Description: String

Syntax: string enclosed in single apostrophe

Examples:

- 'username
- '123'
- 'user@host'

l. integer

Description: A number

Syntax: 1 * digit

Example:

- 123

5. action-type:

Description: Action to be performed on the element

Syntax:

- modify sets element to new value (all element types)
- add-prefix adds value at beginning of string (string element only)
- remove-prefix removes value from beginning of string (string element only)
- add-suffix adds value at end of string (string element only)
- remove-suffix removes value from end of string (string element only)
- add adds a new header/param/body (header or parameter elements)
- remove removes a header/param/body (header or parameter elements)

6. value:

Description: Value for action and match

Syntax: ('string' / message-element / param) * (+ ('string' / message-element / param))

Examples:

- 'itsp.com'
- header.from.user
- param.ipg.src.user
- param.ipg.dst.host + '.com'
- param.call.src.user + '<' + header.from.user + '@' + header.p-asserted-id.host + '>'

C DSP Templates

This section lists the DSP templates supported by the device. Each DSP template provides support for specific voice coders (as well as channel capacity and various features). You can use the following parameters to select the required DSP template:

- DSP Version Template Number (DSPVersionTemplateNumber) - allows you to select a specific DSP template.
- DSP Templates table (DSPTemplates) - allows you to select and use two DSP templates for the device and determine the percentage of DSP resources allocated per DSP template.

Notes:

- Installation and use of voice coders is subject to obtaining the appropriate license and royalty payments.
- The number of channels refers to the maximum channel capacity of the device.
- The maximum number of channels on any form of analog, digital and MPM modules assembly is 120.
- For additional DSP templates, contact your AudioCodes representative.

C.1 Analog Interfaces

The DSP templates for analog interfaces are shown in the table below.

Table C-1: DSP Firmware Templates for Analog (FXS/FXO) Interfaces

	DSP Template	
	0, 1, 2, 4, 5, 6	**10, 11, 12, 14,15, 16**
	Number of Channels	
Default Settings	4	3
With SRTP	3	3
Voice Coder		
G.711 A/Mu-law PCM	Yes	Yes
G.726 ADPCM	Yes	Yes
G.723.1	Yes	Yes
G.729 A, B	Yes	Yes
G.722	-	Yes

C.2 Digital Interfaces

The DSP templates for digital interfaces are shown in the table below.

Table C-2: DSP Firmware Templates for Digital Interfaces

	DSP Template														
	0 or 10			1 or 11			2 or 12			5 or 15			6 or 16		
	Number of Spans														
	1	2	4	1	2	4	1	2	4	1	2	4	1	2	4
	Number of Channels														
Default settings	31	62	120	31	48	80	24	36	60	24	36	60	31	60	100
With 128 ms EC	31	60	100	31	48	80	24	36	60	24	36	60	31	60	100
With SRTP	31	60	100	NA	NA	NA	24	36	60	24	36	60	31	48	80
With IPM Features (*)	31	60	100	NA	NA	NA	NA	NA	NA	NA	NA	NA	31	60	100
With IPM Features & SRTP	31	48	80	NA	NA	NA	NA	NA	NA	NA	NA	NA	31	48	80
Voice Coder															
G.711 A-law/Mm-law PCM	Yes			Yes			Yes			Yes			Yes		
G.726 ADPCM	Yes			Yes			Yes			Yes			-		
G.723.1	Yes			-			-			-			-		
G.729 A, B	Yes			Yes			Yes			Yes			Yes		
GSM FR	Yes			Yes			-			-			-		
MS GSM	Yes			Yes			-			-			-		
iLBC	-			-			-			Yes			-		
EVRC	-			-			Yes			-			-		
QCELP	-			-			Yes			-			-		
AMR	-			Yes			-			-			-		
GSM EFR	-			Yes			-			-			-		
G.722	-			-			-			-			Yes		
Transparent	Yes			Yes			Yes			Yes			Yes		

Notes: IPM Features refers to the configuration that includes at least one of the following:

- Mounted MPM module in Slot #6 for conference applications.
- IPM detectors (e.g., Answer Detector) are enabled.
- The IP Media Channels featured is enabled.

C.3 Media Processing Interfaces

The DSP templates for the media processing interfaces (i.e., MPM module) are shown in the following table.

Notes:

- Assembly of the MPM module in Slot #6 enables DSP conferencing capabilities.
- To use the MPM module, the device must be installed with the IP Media Channels Feature Key.

Table C-3: DSP Firmware Templates for MPM Module

			DSP Template									
			0 or 10		1 or 11		2 or 12		5 or 15		6 or 16	
			Assembly Slot no.									
			1-5	6	1-5	6	1-5	6	1-5	6	1-5	6
Supplementary Capabilities			Number of Channels									
SRTP	IPM Detectors	Conference										
-	-	-	40	20	32	16	24	12	24	12	40	20
Yes	-	-	40	20	NA	NA	24	12	24	12	40	20
-	Yes	-	40	20	NA	NA	NA	NA	NA	NA	40	20
Yes	Yes	-	32	16	NA	NA	NA	NA	NA	NA	32	16
-	-	Yes	40	20	32	16	24	12	24	12	40	20
Yes	-	Yes	32	16	NA	NA	24	12	24	12	32	16
Yes	Yes	Yes	32	16	NA	NA	NA	NA	NA	NA	32	16
Voice Coder												
G.711 A-law/Mm-law PCM			Yes		Yes		Yes		Yes		Yes	
G.726 ADPCM			Yes		Yes		Yes		Yes		-	
G.723.1			Yes		-		-		-		-	
G.729 A, B			Yes		Yes		Yes		Yes		Yes	
GSM FR			Yes		Yes		-		-		-	
MS GSM			Yes		Yes		-		-		-	
iLBC			-		-		-		Yes		-	
EVRC			-		-		Yes		-		-	
QCELP			-		-		Yes		-		-	
AMR			-		Yes		-		-		-	
GSM EFR			-		Yes		-		-		-	
G.722			-		-		-		-		Yes	
Transparent			Yes		Yes		Yes		Yes		Yes	

D Technical Specifications

The device's technical specifications are listed in the table below.

Table D-1: Technical Specifications

Function	Specification
Interfaces	
PSTN Modularity and Capacity	Voice interface: 6 Slots for hosting voice processing and PSTN termination modules (up to 120 TDM-VoIP channels)
Digital Module	1, 2, or 4 E1/T1/J1 spans per module (up to 4) using RJ-48c connectors with an option of PSTN Fallback
Analog Module	4 ports FXO or FXS per module (up to 6) using RJ-11 connectors, ground start and loop start
BRI Module	4 BRI ports (8 calls) per module (up to 5), network S/T interfaces. NT or TE termination
Media Processing Module	Media processing options of up to 60 Conferencing legs (3 way or N-way), play, record to IP or PSTN
Networking Interfaces	
WAN	10/100/1000Base-T, 1000Base-SX/LX, T1 DSU/CSU, SHDSL, ADSL2+*, E1 DSU/CSU*
LAN	3 ports 10/100/1000Base-T
OSN Server Platform (Optional)	
Single Chassis Integration	Embedded, Partner Application Platform for third-party services
CPU	Pentium M 1.4 GHz
Memory	1G RAM or 2G RAM
Storage	Single/Dual hard disk drives
Interfaces	10/100/1000Base-T, USB, VGA, RS-232
Media Processing	
Voice Coders	G.711, G.722, G.723.1, G.729A/B, G.726, GSM FR, MS GSM, iLBC, EVRC, QCELP, AMR, GSM EFR. Independent dynamic vocoder selection per channel
Echo Cancellation	G.165 and G.168-2002, with 32, 64 or 128 tail length
Quality Enhancement	Dynamic programmable jitter buffer, VAD, CNG
DTMF/MF Tones	Packetside or PSTN side detection and generation, RFC 2833 compliant DTMF relay Call Progress tones Detection and Generation
IP Transport	VoIP (RTP/RTCP) per IETF RFC 3550 and 3551
Fax and Modem	Transport T.38 compliant (real time fax), Automatic bypass to PCM or ADPCM, V.34

Function	Specification
Signaling	
Digital – PSTN Protocols	▪ CAS: MF-R1: T1 CAS (E&M, loop start, Feature Group-D, E911CAMA), ▪ E1 CAS (R2 MFC) ▪ ISDN PRI: ETSI/EURO ISDN, ANSI NI2 and other variants (DMS-100, 5ESS) QSIG, VN3, VN4, VN6 ▪ ISDN BRI: Support Euro ISDN, VN4/6 or QSIG
Analog Signaling	FXS, Caller ID, polarity reversal, metering tones, distinctive ringing, visual message waiting indication, loop start, ground start
Data Routing (Optional)	
	▪ DHCP/PPPoE/L2TP/PPTP client towards WAN ▪ DHCP server towards LAN ▪ VLAN ▪ Layer 3 routing ▪ Internal layer 2 switching ▪ Static and dynamic routing (RIP, OSPF, BGP) ▪ PPP, HDLC ▪ IPv6*
Control and Management	
Control Protocols	▪ SIP-TCP, UDP, TLS and MSCML ▪ Stand Alone Survivability for service continuity
Operations & Management	▪ Element Management System (EMS) management tool ▪ Embedded HTTP Web Server, Telnet, SNMP V2/V3 ▪ Remote configuration and software download via TFTP, HTTP, HTTPS, DHCP and ▪ BootP, RADIUS, Syslog (for events, alarms and CDRs) ▪ Voice Menu using touch-tone phone (FXS interface) for basic configuration
IP/VoIP Quality of Service	
	▪ IEEE 802.1p, TOS, DiffServ ▪ IEEE 802.1Q VLAN tagging ▪ Shaping, Policing, Queuing, Bandwidth Reservation ▪ RTCP-XR Report Publish (RFC 3611)
Security	
Session Border Controller	▪ SIP Header conversion: IP to IP Routing translations of SIP, UDP, TCP, TLS ▪ Translation of RTP, SRTP; Support SIP trunk with multi-ITSP (Registrations to ITSPs is invoked independently); Topology hiding; Call Admission Control; Call Black/White list ▪ Intrusion detection/prevention (NIDS); Anti SPIT & SPAM mechanisms

Function	Specification
Data Security	▪ IPsec, up to 8 links: ESF – Tunnel mode; Encryption; Authentication; IKE mode – IPsec VPN ▪ DoS Protection of: Fragmented traffic; Malformed Request; Ping of Death; DoS attack; Properly formed request from unauthenticated source; DDoS attack; SYN flood ▪ Stateful packet inspection firewall ▪ Bad fragment, spoofed connection ▪ DMZ Host ▪ Port Triggering ▪ Packet Filtering ▪ Application Layer Gateway ▪ SRTP
Data-Routing Features	
LAN Switching	▪ 802.1Q for VLAN tagging. ▪ Rapid Spanning Tree Protocol / RSTP (802.1D-2004 and 802.1w). ▪ LAN switch ports with non-blocking switching performance. ▪ Up to four RJ-45 10/100/1000Base-T (Gigabit Ethernet) LAN ports. ▪ Eight RJ-45 10/100Base-TX (Fast Ethernet) LAN ports. ▪ Power-over-Ethernet (PoE) supported on all LAN ports, complying with IEEE 802.3af-2003 (15.4W maximum wattage per port; 120 total wattage over all ports). ▪ LAN ports supporting half- and full-duplex modes, auto-negotiation, and straight or crossover cable detection. ▪ CRMX module supporting three on-board 10/100/1000Base-T ports, auto-negotiated LAN switch ports with non-blocking switching performance. ▪ High performance lookup engine with support for up to 1,024 MAC addresses.
Routing	▪ Dynamic Host Configuration Protocol (DHCP): ✓ DHCP server, DHCP relay and DHCP client ✓ DHCP server supports fixed binding of IP-to-MAC address ✓ DHCP server supports user-defined DNS server allocation ▪ Multiple IP interfaces for LAN routing: IP interfaces assignment to different VLANs. ▪ Assignment of different VLAN IDs to VoIP and Data traffic. ▪ Symmetric High-Speed Digital Subscriber Line (SHDSL): ✓ ATM: RFC 2684 over AAL5 with LLC-SNAP and VC-MUX over AAL5; ATM UNI 4.1; UBR, CBR, VBR-RT and VBR-NRT; IP QoS; Multiple IP interfaces; RFC 2684 Routed Mode; RFC 2364 PPPoA; RFC 2516 PPPoE over ATM; Up to 8 PVCs ✓ EFM: ITU G.991.2 Annex E for Ethernet, also known as EFM or 2Base-TL, as defined in IEEE 802.3ah; Up to 8 IP interfaces for EFM/2Base-TL; IP QoS classification, Marking, scheduling and shaping; 802.1Q VLANs ▪ Routing protocols: ✓ Static routing ✓ RIPv1 - RFC 1058 ✓ RIPv2 - RFC 2453

Function	Specification
	✓ OSPFv2 - RFC 2328 ✓ BGPv4 - RFC 1771 and RFC 2858 ✓ BGP Extended Community Attribute for BGP/MPLS VPNs ✓ Policy-based routing (e.g. DSCP-based and BGP policy routing) ▪ Network Address Translation (NAT/NAPT): ✓ ACL-like classification with ALG support ✓ Source and destination-based IP addresses ACLs ✓ Multiple NAT and NAPT WAN addresses ▪ WAN access via PPPoE, PPTP, L2TP, DHCP. ▪ Quality of service (QoS): ✓ Traffic Classification and Marking: Connection-based (with SPI) or packet-based classification; VoIP classification for SIP signaling and media traffic (tracking UDP ports selected by SDP offer-answer negotiations); DSCP and 802.1p marking; Explicit classification criteria (source/destination MAC and IP addresses; Protocol ALG-based; L4 port numbers; DSCP/802.1p; Length of packets or their data portion only) ✓ Traffic scheduling and shaping: Ingress traffic policing; Traffic reservation - when not utilized, other classes are served with extra bandwidth; Maximum egress traffic shaping; Scheduling (Strict Priority, Fair, Weighted Round-Robin Queuing, and Class-Based WRR Queuing); Queue management (RED; TCP Serialization Reduction to minimize jitter in VoIP environments)
Data Security	▪ Zero Configuration Firewall wizard with three security levels: ✓ Minimum (Inbound and Outbound policies set to 'Accept') ✓ Typical (Inbound policy set to 'Reject'; Outbound Policy to 'Accept') ✓ Maximum (only selected applications are allowed in Outbound policy) ▪ Access Control for pinpoint security policy. ▪ Extensive list of ALG-modules combined with SPI for error-free configuration and maximum security. ▪ Port-forwarding and DMZ support for local applications and hosts. ▪ Website Restriction allows static URL-based blocking of public/extranet websites. ▪ Advanced Filtering allows full control on Inbound/Outbound Rules per interface/device. ▪ Site-to-Site VPN: ✓ Supports two IPSec use-cases: Site-to-Site Gateway-to-Gateway) VPN; Teleworker (User-to-Gateway) VPN ✓ Fully compliant with IPSec RFCs: RFC 2401 - Security Architecture for IP; RFC 2402 - IP Authentication Header; RFC 2406 – ESP; RFC 2403 and RFC 2404 for Authentication ▪ PPTP/L2TP Client-Server VPN: ✓ Supports two VPN use-cases: Server support for remote Teleworker VPN access; Client-to-Gateway support with PPTP/L2TP ✓ Point-to-Point Tunneling Protocol - RFC 2637 ✓ Layer Two Tunneling Protocol - RFC 2661 (with L2TP/IPSec) ✓ Support all WiN OS versions as well as Linux ▪ DoS and IDS/IPS:

Function	Specification
	✓ Denial of Service (DoS) protection: TCP RST, Ping Flood, ICMP Echo storm, UDP snork attack, ICMP Smurf, UDP fraggle and more ✓ IP spoofing attacks: FTP bounce, Broadcast/multicast source IP attack ✓ Intrusion and scanning attacks: IP source route, ICMP Echo reply without request, ICMP Ping sweep, TCP Stealth; Scan (FIN, XMAS, NULL), UDP port, FTP passive attack, loopback / Echo chargen, Block security hazard ICMP messages ✓ IP fragment overlap, Ping of Death, Fragmentation attacks and more
Supported RFCs	▪ IP/Routing: ✓ RFC 768 UDP ✓ RFC 791 IP ✓ RFC 792 ICMP ✓ RFC 793 TCP ✓ RFC 959 FTP ✓ RFC 2822 Internet Message Format ✓ RFC 1305 NTP ✓ RFC 826 ARP ✓ RFC 2131 DHCP ✓ RFC 1918 IP Addressing ✓ RFC 2516 PPPoE ✓ RFC 2663 NAT ▪ QoS: ✓ IEEE 802.1p Priority Tagging ✓ RFC 1349 ✓ RFC 2475 ✓ RFC 2597 ✓ RFC 3246 ▪ IPSec: ✓ RFC 2401 Security Architecture for IP ✓ RFC 2402 AH - IP Authentication Header ✓ RFC 2403 IPsec Authentication - MD5 ✓ RFC 2404 IPsec Authentication - SHA-1 ✓ RFC 2405 IPsec Encryption - DES ✓ RFC 2406 ESP - IPsec Encryption ✓ RFC 2407 IPsec DOI ✓ RFC 2408 ISAKMP ✓ RFC 2409 IKE ✓ RFC 2410 IPsec Encryption – NULL ✓ RFC 2411 ✓ RFC 2412 OAKLEY

Function	Specification
Hardware Specifications	
Power Supply	Single universal power supply 100-240V, 50-60 Hz, 1.5A max. or optional redundant power supply
Physical	1U high, 19-inch wide
Regulatory Compliance	
Safety and EMC Standards	▪ UL60950-1, FCC 47 CFR part 15 Class B ▪ CE Mark (EN55022 Class B, EN60950-1, EN55024, EN300 386, EN61000-3-2/3-3)
Telecommunication Standards	TIA/EIA-IS-968, TBR-4, TBR-13 and TBR-21
Environmental Specifications	ETS 300019-2-1 Storage T1.2, ETS 300019-2-2 Transportation T2.3, ETS 300019-2-3 Operating T3.2

*Future support.

Reader's Notes

AudioCodes

User's Manual Ver. 6.4

www.audiocodes.com