Configuration Guide

Multi-Service Business Routers Product Series

Mediant MSBR

Basic System Setup through CLI

Version 6.8

HD VoIP Sounds Better

AudioCodes

Table of Contents

This page is intentionally left blank.

Notice

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.

Date Published: October-18-2017

Trademarks

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 services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our Web site at www.audiocodes.com/support.

Abbreviations and Terminology

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

Document Revision Record

LTRT	Description
31606	Initial document release.
31607	Added Chapter 8 - Upgrading the MSBR.
31608	Updates to Copy method, NTP command, RADIUS configuration, Syslog configuration and Debug packet capturing.
31612	Added RADIUS and TACACS+ for console bypass commands. Added TACACS+ configuration flags.
31614	Added Chapter for BroadSoft's BroadCloud PacketSmart.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our Web site at http://www.audiocodes.com/downloads.

1 Introduction

This document describes the configuration of the system functionality of AudioCodes Mediant Multi-Service Business Routers (MSBR), using the command-line interface (CLI).

The document describes many of the administration aspects of the MSBR such as CLI management, SNMP management, uploading and downloading of software files to and from remote servers (such as HTTPS and an attached USB device), clock features, management access authorization, authentication and accounting, password recovery process, configuration reload, packet capturing and many others.

The document describes the CLI commands required for configuring each aspect including typical configuration examples. The document also describes the configuration of third-party applications (such as RADIUS server) where necessary.

This page is intentionally left blank.

2 CLI Management Interface

Management through the CLI allows the administrator to configure every feature of the MSBR. The CLI administration is easy, efficient and intuitive.

The CLI is divided into *Basic* configuration mode and *Enabled* configuration mode. The Basic configuration mode is marked by the chevron ">". The Enabled mode is marked by the hash sign "#". The Basic mode provides a limited set of commands and options. The Enabled mode allows the use of all the commands including access to configuration of the system, data and voice functionalities, as well as show and debug commands and access to the maintenance actions such as copy, write and reload.

Use the following commands to access or exit the Enabled mode:

Command	Description
`Enable`	Enters the Enabled mode
`Exit`	When in the Basic or Enabled mode, the command exits the CLI and the CLI awaits for the username to be entered again. Leaves the current command-set and returns one level up.
`Quit`	While in the Basic or Enabled mode, the command exits the CLI and the CLI awaits for the username to be entered again.

To improve the work of the network administrator, the CLI allows the use of the following keyboard shortcuts:

Keyboard shortcut	Description
Up arrow	Re-displays the previously entered command. If you continue pressing the up arrow key, it will cycle through all the previously entered commands, starting with the most recent.
Tab	Pressing the Tab key after entering a partial (but unique) command completes the command, displays it on the command prompt line, and waits for further input. Pressing the Tab key after entering a partial and non-unique command displays all completing options.
?	▪ Displays a list of all subcommands in the current mode. ▪ Displays a list of available commands beginning with certain letter(s). ▪ Obtains syntax help for the commands. ▪ Displays the range of values and a brief description of the next parameter expected for that particular command. ▪ If there is a command that can be invoked (all its arguments are inserted), using the question mark at its end displays "<cr>".
CTRL + A	Jumps to the beginning of the displayed command line.
CTRL + E	Jumps to the end of the displayed command line.
CTRL + U	Clears the current displayed command line.
CTRL + Z	Returns to the Enabled mode prompt "#".

If sufficient letters are entered to identify a command, the auto-finish function of the CLI identifies the command and there is no need to write the entire command. For example, instead of typing the entire command "enable", you can simply type "en".

To access the MSBR, Use the default username and password, as listed in the following table:

Access	Default Value
Username	Admin
Password	Admin
Enable password	Admin

CLI management of the MSBR is available using SSH, Telnet or the console. To access the console port, use the following RS-232 terminal emulation configuration for any terminal client (e.g., PuTTY, Tera Term, and HyperTerminal):

- 115200 Baud rate
- 8 Data bits
- No parity
- 1 Stop bits
- No flow control

By default, Telnet access to the management interface is allowed. Use any Telnet client (such as Telnet or PuTTY) to access the MSBR. The default MSBR address is 192.168.0.1.

By default, SSH access to the management interface is disabled. Use the following commands to enable or disable SSH or Telnet access to the MSBR:

Command	Description
`# configure system`	Enters system configuration level.
`(config-system)# cli-terminal`	Enters cli-terminal configuration level.
`(config-system)# set telnet enable`	Enables Telnet to the MSBR.
`(config-system)# set telnet disable`	Disables Telnet to the MSBR.
`(config-system)# set ssh on`	Enables SSH to the MSBR.
`(config-system)# set ssh off`	Disables SSH to the MSBR.

By default, the device administration through the WAN port is disabled. Use the following command to enable device administration through the WAN port:

Command	Description
`set wan-telnet-allow`	Enables Telnet to the MSBR through the WAN port.

The following are common commands used in the CLI:

Command	Description
do	Executes commands in the Enable mode without the need to exit the current command set.
no	Undoes an issued command or disables a feature.
list	Displays a list of the available command(s) of the current command set.
history	Displays a list of previously run commands.
exit	Leaves the current command set and returns one level up.

The configuration of the device is divided into three configuration set levels:

- System: Contains the general and system oriented configuration command of the MSBR
- VoIP: Contains VoIP-oriented configuration commands.
- Data: Contains all configuration tasks relating to the data entity of the MSBR.

The following commands enter these different configuration levels:

Command	Description
`configure system`	Enters the System configuration level.
`configure voip`	Enters the VoIP configuration level.
`configure data`	Enters the Data configuration level.

Note: It is important to understand some of the MSBR's architecture design qualities. One of the important qualities is the existence of two CPUs. One CPU handles the voice traffic while the other handles the data traffic. The management services such as SSH is handled by the voice CPU. This becomes important when commands such as `set wan-telnet-allow` are issued. On the LAN side, there is no impact - the switch is connected to both CPUs and it knows to which CPU to deliver the Telnet session. However, on the WAN side, the WAN ports are connected only to the data CPU and the Telnet session needs to be delivered to the voice CPU. The voice CPU and data CPUs are connected. When the `set wan-telnet-allow` command is issued, the data CPU acts as a proxy for the Telnet protocol, and connection to the WAN interface is delivered to the voice CPU through the connection between the two CPUs. In addition, if for protocols such as RADIUS, TACACS+, SNMP and others, no source IP is configured, the default source IP is the voice CPU address.

2.1 Examples

2.1.1 Accessing the MSBR

```
Welcome to AudioCodes CLI


Username: Admin
Password: *****

MSBR> ena
Password: *****
MSBR#
```

2.1.2 Using the "do" Command

```
MSBR(config-data)# do sh run

# Running Configuration MSBR

  configure voip
   coders-and-profiles coders-group-0set p-time 20
    activate
   exit
   interface network-if 0
```

2.1.3 Accessing the Data Configuration Mode

```
MSBR# conf data

MSBR(config-data)#
```

2.1.4 Exiting the Data Configuration Mode

```
MSBR(config-data)# exit

MSBR#
```

2.1.5 Accessing the MSBR through WAN Port

The following procedure describes how to enable access to the MSBR through its WAN port:

1. Enter the System configuration level:
```
MSBR# configure system
```
2. Access the cli-terminal commands:
```
MSBR(config-system)# cli-terminal
```
3. Enable telnet on the MSBR:
```
MSBR(cli-terminal)# set telnet enable
```
4. Enable telnet through the WAN interface:
```
MSBR(cli-terminal)# set wan-telnet-allow on
Note: Setting this parameter requires a reset.
```
5. Exit to the previous level:
```
MSBR(cli-terminal)*# exit
```
6. Exit to the previous level:
```
MSBR(config-system)*# exit
```
7. Reset the MSBR:
```
MSBR*# reload now
Writing configuration and restarting...

MSBR*#
```

This page is intentionally left blank.

3 Port Naming Convention

The port naming convention is a method for assigning names to ports. Each port name consists of a port name, module slot number, and port number.

The port name is typically the type of interface. The port name depends in the MSBR assembly. The following table describes the port naming conventions for different port types:

Port Type	Port Name
Fast Ethernet 100Mbps	`FastEthernet`
Giga Ethernet 1Gbps	`GigabitEthernet`
Fiber 1 GIG SFP Ethernet	`Fiber`
PSTN ports, including FXS, FXO, BRI, PRI	`Port`

The module slot number also depends on the MSBR assembly; however some of the slot numbers are always fixed for the same module types. The following table describes the module types and the numbers assigned to the ports:

Module Type	Module Number
WAN	0
LAN	1,4,5
VOICE	2,3

The port numbers are assigned to ports according to the number of ports in each module.

To view the modules installed in the MSBR, use the following command:

Command	Description
`show system assembly`	Displays installed modules and port types.

3.1 Examples

3.1.1 Displaying System Assembly

Output of show system assembly command:

```
MSBR# show system assembly

Board Assembly Info:

| Slot No.            | Ports      | Module Type              |
| 0/0                 | 1          | WAN-Copper               |
| 0/1                 | 1          | WAN-Fiber                |
| 0/2                 | 1          | WAN-A/VDSL               |
| 1                   | 1-4        | LAN-GE                   |
| 2                   | 1-4        | FXS                      |

USB Port 1: Empty
USB Port 2: Empty

MSBR#
```

The output of the show system assembly command displays every slot, port and module type of the ports installed on the MSBR. The "Slot No." column displays the slot number of a port; the "Ports" column displays the port number; the "Module Type" displays the port type.

3.1.2 Port Naming

The following table describes the port names of each interface in different module types.

Port Description	Port Name
WAN port, 1 Gbps	`GigabitEthernet 0/0`
LAN port number 2, 1 Gbps port, module slot 1	`GigabitEthernet 1/2`
LAN port number 1, 100 Mbps, on module slot 5	`FastEthernet 5/1`
WAN port, DSL at port number 3	`DSL 0/3`

4 SNMP Management

The MSBR supports Simple Network Management Protocol (SNMP) for configuration and management. The MSBR supports SNMPv2c and SNMPv3 for access and for sending traps.

The SNMP engine – the process which responds to SNMP requests and sends SNMP traps – runs on the VoIP CPU. Therefore, SNMP requests need to be sent to the VoIP CPU.

4.1 SNMPV2C

To configure SNMPv2 read-only access to the MSBR, use the following commands:

Command	Description
`# configure system`	Enters the System configuration level.
`(config-system)# snmp`	Enters the SNMP configuration level.
`(snmp)# set ro-community-string 0 P@ssw0rd`	Sets the read-only community string with the index 0 to "P@ssw0rd". The index can be a value from 0 to 4 and therefore, there can be only five read-only community strings.
`(snmp)# activate`	Changes to parameters will take effect when applying the `activate` or `exit` command.

To configure read-write community string, use the following command:

Command	Description
`(snmp)# set rw-community-string 0 rw-P@ssw0rd`	Sets the read-write community string with the index 0 to "rw-P@ssw0rd". The index can be a value from 0 to 4 and therefore, there can be only five read-write community strings.

4.2 SNMPV3

To configure SNMPv3, use the following commands:

Command	Description
`MSBR# configure system`	Enters the System configuration level.
`(config-system)# snmp v3-users 0`	Enters the configuration level of an SNMPv3 user with the index 0. If a user with index 0 does not exist, a new user at index 0 will be created. If a user with index 0 does exist, this user configuration will be modified. Use `new` instead of an index number, and a new user will be created at the first available index. Use `display` instead of the index number and users configuration will be displayed.
`(v3-users-0)# set username Tim`	Sets the SNMPv3 username to "Tim".

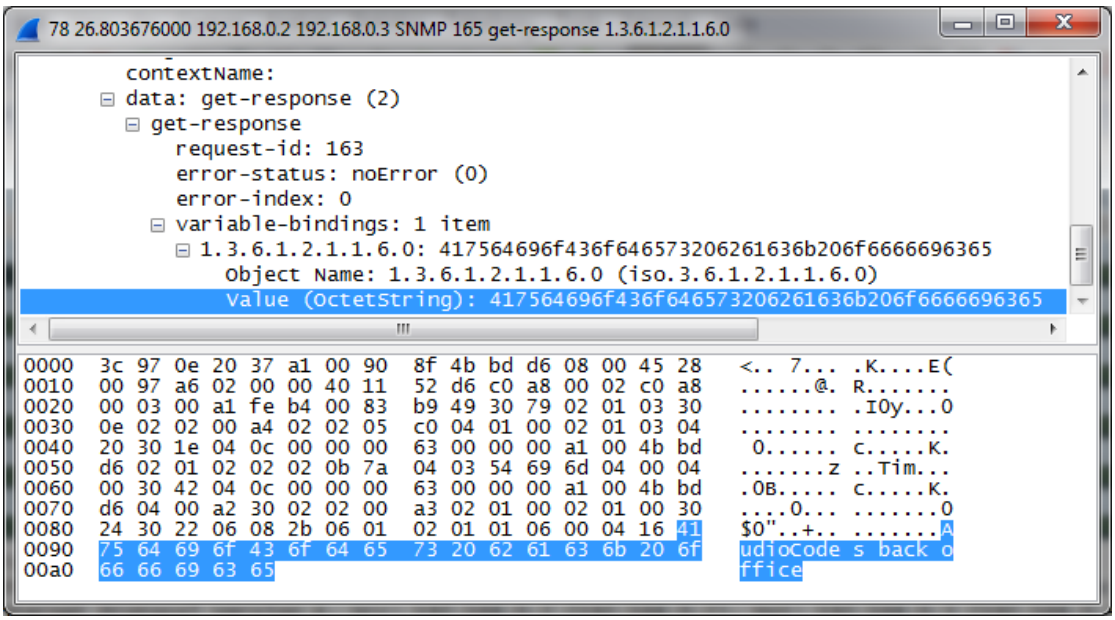
Command	Description
`(v3-users-0)# set auth-protocol sha-1`	Sets the authentication protocol for the user to sha-1. Other options include md-5 and none for not using authentication.
`(v3-users-0)# set auth-key P@ssw0rd`	Sets the authentication key to "P@ssw0rd".
`(v3-users-0)# set group read-write`	Assigns the user to the read-write group. Other options are to assign the user to the read-only group and to the trap group. Assignment of the user to the trap group is described in the SNMPv3 traps section.

The SNMPv3 can be configured in three modes of the security level:

Security Level	Description
`NoAuth, NoPriv`	No authentication and no privacy. No authentication means that the username is not authenticated. No privacy means that the data of the MIB is not encrypted.
`Auth,NoPriv`	Authentication; however no privacy. The user is authenticated, but the MIB data is sent without encryption. The key encryption algorithms available for authentication are MD-5 and SHA-1.
`Auth,Priv`	Authentication and privacy. The user is authenticated and the MIB data is encrypted. The key encryption algorithms available for privacy are DES, 3des, AES-128, AES-192, and AES-256.

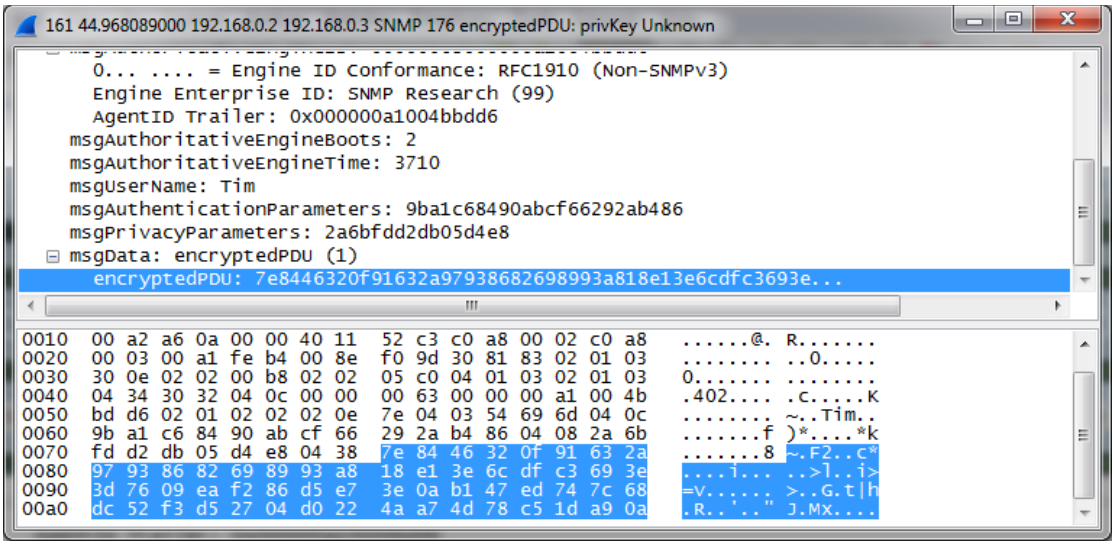
To emphases the encryption of the SNMPv3 packets, see the below captured SNMPv3 packet. The packet is an MSBR response to SNMPv3 Get of the system location MIB value. The next captured packet shows the NoAuth-NoPriv operation mode. The MIB value is sent unencrypted.

Figure 4-1: SNMP Packet in NoAuth-NoPriv mode



The screenshot below displays a captured packed using the AuthPriv mode. The MIB value is sent encrypted.

Figure 4-2: SNMP Packet in AuthPriv mode



4.3 SNMPV2C and SNMPV3 General Commands

The following commands are applicable to both SNMP versions for accessing the MSBR:

Command	Description
`(snmp)# set sys-name "AudioCodes"`	Sets system name.
`(snmp)# set sys-location "AudioCodes main office"`	Sets system location. The brackets are required if spaces are used.
`(snmp)# set sys-contact "AudioCodes Inc"`	Sets the system contact.
`(snmp)# set wan-snmp-allow on`	Allows SNMP access on the WAN interface.
`(snmp)# set port 2162`	Sets the MSBR to use port 2162 for SNMP.
`(snmp)# set snmp-acl community-string P@ssw0rd ro snmp-acl`	Sets ACL called snmp-acl for RO community P@ssw0rd. It is recommended to use either the `snmp-acl` command or `trusted-managers` command, but not both.
`(snmp)# set trusted-managers 0 192.168.0.3`	Allows the IP address of 192.168.0.3 to access the SNMP. It is recommended to use either the `snmp-acl` command or `trusted-managers` command, not both of them.
`(snmp)# set sys-oid <string>`	Changes the system OID value.
`(snmp)# set engine-id <Engine ID>`	Changes the engine ID value for SNMPv3.

4.4 SNMP Traps

To send SNMP traps, use the following commands:

Command	Description
`MSBR# configure system`	Enters the System configuration level.
`(config-system)# snmp trap`	Accesses the SNMP trap configuration level.
`(snmp-trap)# set community-string P@ssw0rd`	Sets the community string for traps to "P@ssw0rd".
`(config-system)# snmp trap destination 0`	Sets the number of SNMP trap destinations. The 0 represents the index, meaning the number of the SNMP trap destination to edit. The index can be between 0 and 4 and therefore, there can be only five destinations for sending traps. Use the `display` keyword instead of the index number to display IP destinations configuration.
`(trap-destination 0)# set ip-address 192.168.0.3`	Sets the IP address 192.168.1.3 as the trap destination.
`(trap-destination 0)# set trap-user Tim`	Enables SNMPv3 traps, assuming an SNMPv3 user called "Tim" was configured. Traps will be sent using this user. For SNMPv2C traps, do not configure any user. The traps are sent using the community string configured above.
`(trap-destination 0)# set send-trap enable`	Enables the sending traps from the MSBR device.

4.5 Examples

4.5.1 SNMPV2C Access

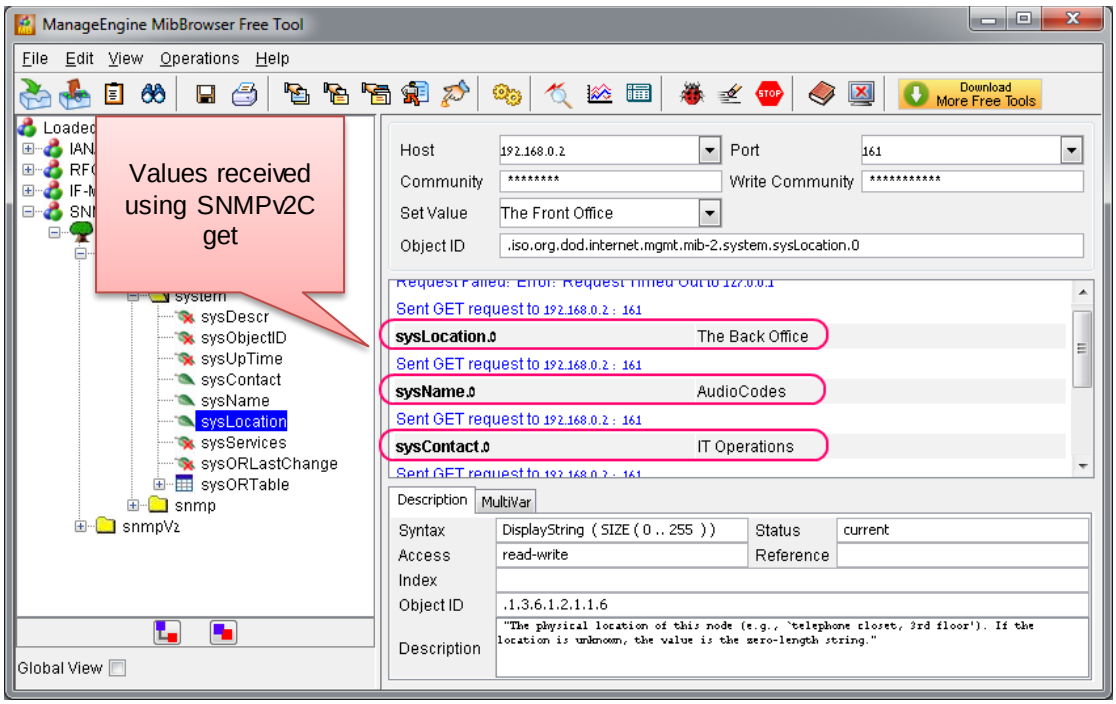
This example uses a free MIB browser to get and set MIB values using SNMP:

```
MSBR# configure system

MSBR(config-system)# snmp
Note: Changes to parameters will take effect when applying the
'activate' or 'exit' command

# Configure SNMPv2C RO connection string to "P@ssw0rd"
MSBR(snmp)#  set ro-community-string 0 P@ssw0rd

# Configure SNMPv2C RW connection string to rw-P@ssw0rd
MSBR(snmp)#  set rw-community-string 0 rw-P@ssw0rd

# Configure system name to "Audio Codes"
MSBR(snmp)# set sys-name AudioCodes

# Configure system location name to
MSBR(snmp)# set sys-location "The Back Office"
```

```
# Configure system contact to IT Operations
MSBR(snmp)# set sys-contact "IT Operations"

MSBR(snmp)# exit

MSBR(config-system)# exit

MSBR#
```

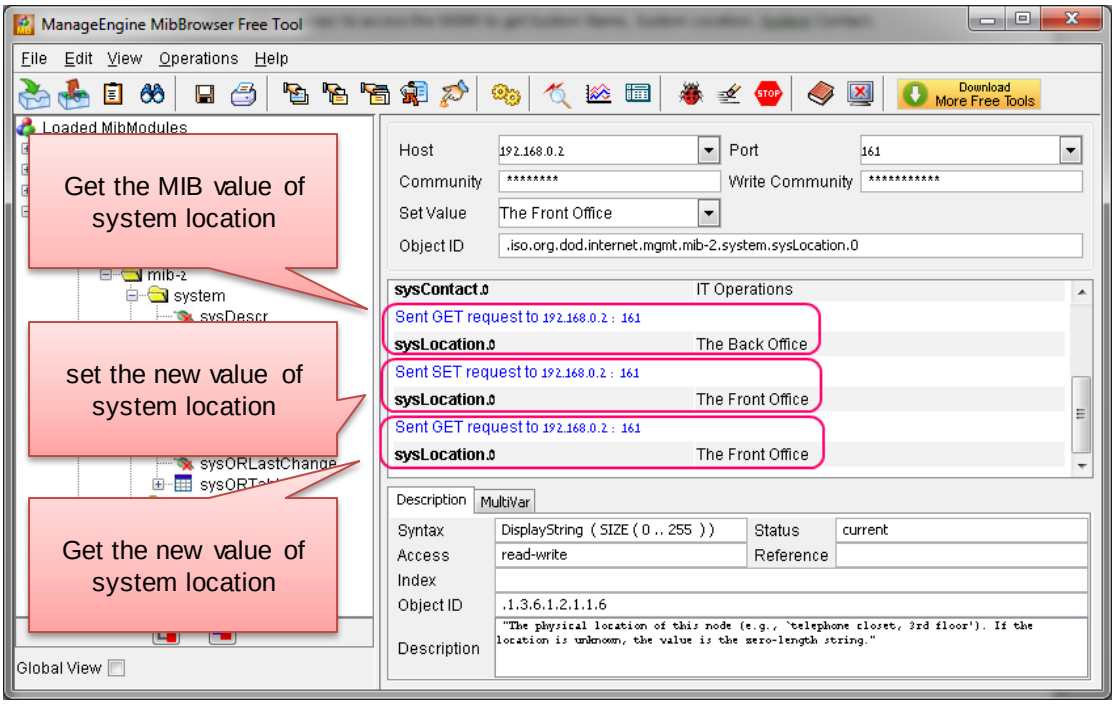
Use an SNMP MIB browser to access the MSBR to get System Name, System Location, System Contact:

Figure 4-3: SNMPv2C get



Use the SNMP MIB browser to set the System Location to The Front Office:

Figure 4-4: SNMPv2C Get and Set



5 NetFlow

NetFlow is a feature that provides the ability to collect IP network traffic. The NetFlow records are generated from the firewall statistics. Since the NetFlow information is taken from the firewall, you must activate firewall capabilities on the monitored interface.

5.1 CLI Commands

The commands used to configure the MSBR NetFlow parameters are listed below:

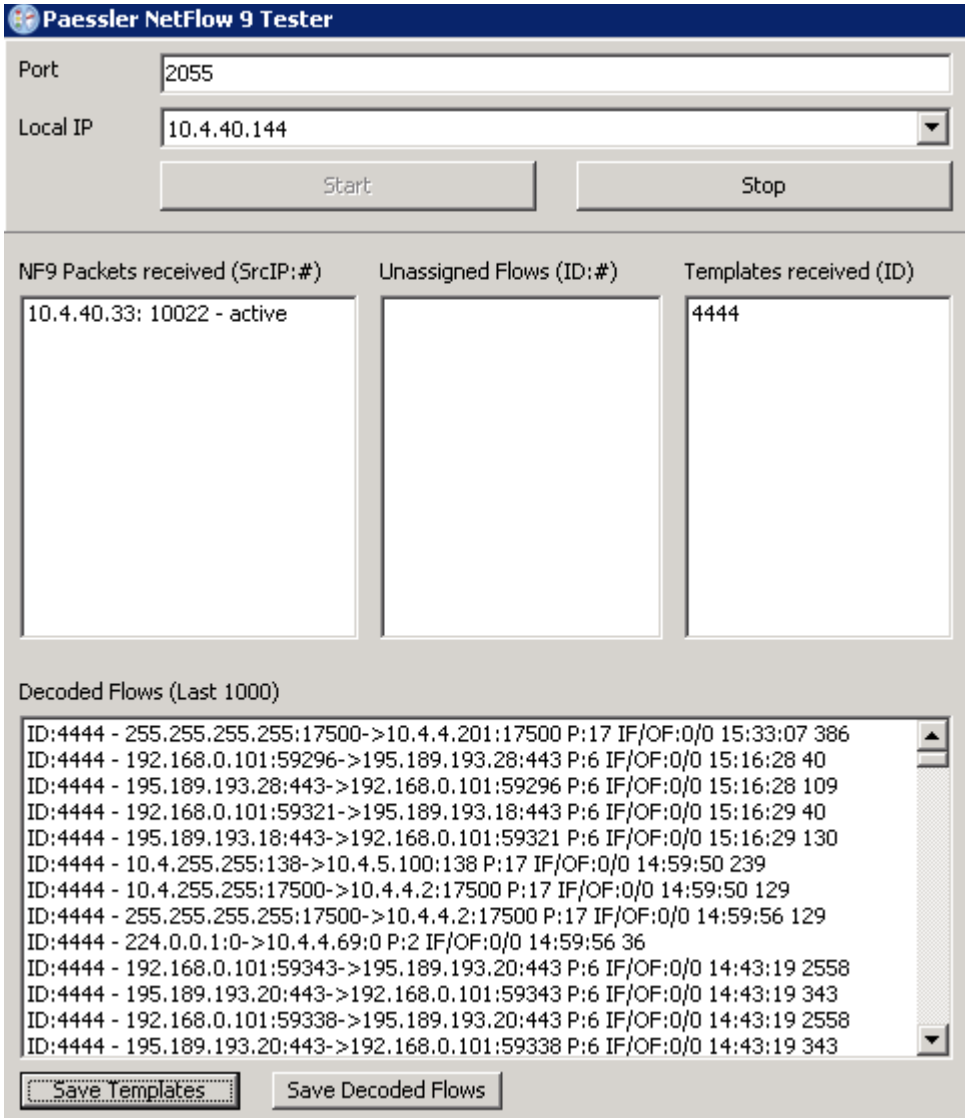
Command	Description
`ip flow-export enable`	Enables NetFlow.
`ip flow-export destination <Netflow Server Address> <Netflow Server Port >`	Sets the NetFlow destination server and server port (default port 2055).
`ip flow-export version 5 enable`	Enables NetFlow version 5.
`ip flow-export version 9 enable`	Enables NetFlow version 9.
`ip flow-export source-address interface < bvi | cellular | gigabitethernet | gre | ipip | l2tp | loopback | pppoe | pptp | vlan >`	Sets the source of the NetFlow packets. If not specified, the source will be set according to the routing table interface.

5.2 Examples

This example activates the firewall and NAT. The MSBR WAN IP address is obtained from a DHCP server located on the WAN subnet.

```
configure data
   ip flow-export enable
   ip flow-export destination 10.4.40.144 2055
   ip flow-export version 5 enable
   ip flow-export version 9 enable
   ip flow-export source-address interface GigabitEthernet 0/0
   interface GigabitEthernet 0/0
    ip address dhcp
    mtu auto
    desc "WAN Ethernet"
    speed auto
    duplex auto
    no service dhcp
    ip dns server auto
    napt
    firewall enable
    no shutdown
   exit
```

Figure 5-1: NetFlow Displayed in Simple Grabber



6 Copy Methods

The MSBR allows you to copy files using HTTP, HTTPS, TFTP and NFS.

6.1 CLI Commands

The commands for copying files from a server to the MSBR are listed below:

Command	Description
`Copy <file> from <URL> source [data | voip] [[interface | source-address | vrf] | voip]`	Copies a file from a server using HTTP, HTTPS, TFTP or NFS.

The following files can be copied from the server using the `copy` command:

File	Description
adsl-firmware	ADSL firmware file
call-progress-tones	Call Progress Tones file
cas-table	CAS configuration table file
cli-script	CLI configuration file
coder-table	Coder table file
data-configuration	Data configuration file
dial-plan	Dial Plan file
firmware	Firmware, burn and reload
nqm-history	Export Network Quality Monitoring history file
prerecorded-tones	Prerecorded tones file
startup-script	CLI configuration file
tls-cert	TLS certificate file
tls-private-key	TLS private key file
tls-root-cert	TLS trusted root certificate file
user-info	User Info file
voice-configuration	Voice configuration file (ini file)
voice-prompts	Voice Prompts file
voice-xml	Voice XML file
web-logo	Web logo file
source [data | voip]	Copy using source : data or voip

The voice configuration and cli-script can also be exported using the following command:

Command	Description
copy <voice-configuration or cli-script>from <URL>	Copies the voice configuration or CLI-script to HTTP, HTTPS, TFTP or NFS server

The `cli-script` is the complete configuration of the MSBR. Therefore, to export the cli-script means to export the entire device configuration.

When `cli-config` is copied to the MSBR, the configuration is appended to the current device configuration. When the startup-script is copied to the device, the device configuration is cleared, and the device resets. After the reset, the new configuration from the startup script is applied and the device resets again.

When using the `copy` command, please note that the HTTP server timeout is greater than the TFTP server timeout. Therefore, it is recommended to use a TFTP server to copy from or to the LAN and an HTTP server to copy to or from the WAN.

To upload a file to an HTTP server, the Web-based Distributed Authoring and Versioning (WebDAV) extensions to HTTP protocols must be used. WebDAV is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote Web servers. Basically, it allows the MSBR to upload files to an HTTP server. The MSBR does not send a username and password. The WebDAV server should be configured without username and password.

6.2 Examples

6.2.1 Copying Firmware from TFTP Server

In this example, the MSBR copies the firmware file from the TFTP server, burns it to memory, and then reboots.

```
MSBR# copy firmware from
http://192.169.11.11:80/M5XX_SIP_F6.60A.260.002.cmp
Copying file...
done.
Restarting...
```

6.2.2 Copying Configuration from HTTP Server

In this example, configuration is downloaded from a text file on an HTTP server.

```
MSBR# copy cli-script from http://192.168.0.199:80/runcfg.txt
Copying file...
MSBR# # Running Config voip

MSBR(config-voip)# coders-and-profiles coders-group-0 0

MSBR(coders-group-0-0)# set name "g711Alaw64k"

... output omitted
```

If the HTTP port is 80, it is not necessary to add the port number. However, if the port number is different, then the port number should be added to the syntax.

6.2.3 Using Startup-Script

This example shows how to use the `startup-script` keyword with the `copy` command. The configuration from a text file on an HTTP server is downloaded to the MSBR. The MSBR configuration is then cleared and the MSBR resets. The configuration from the downloaded file is applied to the device, after which it resets again.

```
MSBR# copy startup-script from http://192.168.0.199/runcfg.txt
Copying file...
done.
Restarting system...
MSBR# [4788750.760000] Restarting system.
**AUDC*** end of serial init


U-Boot 1.1.1 (Development build) (Build time: Dec  2 2012 -
17:18:21)

AudioCodes uKernel U-Boot Version: MP500 K6

…output omitted
```

6.2.4 Export Device Configuration

The example below shows how to export the MSBR configuration to a text file.

```
MSBR# copy cli-script to tftp://192.168.0.3/sci-scr.txt
Sending file...done
```

This page is intentionally left blank.

7 USB Functionality

7.1 USB Commands

Command	Description
`usb list`	Prints files to a USB. This behaves similar to the "dir" command in Windows or Linux.
`copy <file> from/to <URL>`	Copies files to or from a USB.
`MSBR# usb remove`	Safely removes attached USB device.

7.2 USB Auto-Run

You can run commands by simply connecting a USB flash drive to the MSBR. Once connected, the MSBR runs commands located in the file, "ac_autorun.txt", line-by-line similar to a Telnet connection. The MSBR treats the commands in the "ac_autorun.txt" file as a regular console input and therefore, the username, password and enable password need to be included in the "ac_autorun.txt" file. The output of the commands is written in the file "ac_output.txt".

While reading and executing commands from the USB flash drive, the "Status" LED is lit red. After finishing the command execution, the LED flashes green.

7.3 Examples of USB Commands

The following is an example of using USB commands:

```
# Message that appears on USB insertion
MSBR# [4297251.615000] sda: assuming d[4297251.621000] sda:
assuming drive cache: write through
[4297251.628000]  sda: p1 exceeds device capacity

# Backup configuration
MSBR# copy cli-script to usb:///config_back_up_27apr2014.cfg
Sending file...done

# Show files on the USB
MSBR# usb list
-rwxrwxrwx    1 root     0        34330640 Apr 24  2014
MP500_MSBG_SIP_F6.80A.025.cmp
drwxrwxrwx    2 root     0            4096 Feb 25 20:58 System
Volume Information
-rwxrwxrwx    1 root     0        31759825 Apr  9  2014
YairE_CFM_FIX_MSBR_LAB_UB.cmp
-rwxrwxrwx    1 root     0            3559 Apr  4 23:29
config_back_up_27apr2014.cfg
-rwxrwxrwx    1 root     0            3559 Apr  4 22:54 runcfg.txt
```

```
#Remove the USB drive
MSBR# usb remove
You may now remove the USB drive safely.

MSBR#
```

7.4 Examples of USB Auto-Run

In this example, the "USB auto-run" function used to deliver basic configuration to the MSBR for the administrator to log in remotely. The configuration sent to the MSBR sets the WAN interface Gig0/0 IP address to 100.0.10.10 and allows an SSH connection from the WAN interface.

```
Admin
Admin
en
Admin
configure data
    interface GigabitEthernet 0/0
    ip address 100.0.10.10 255.255.0.0
    exit
exit
configure system
    cli-terminal
          set ssh on
          set wan-ssh-allow on
    exit
exit
reload now
```

The output from the MSBR to the "ac_output" file:

```
Welcome to AudioCodes CLI


Username: Admin
Password:

MSBR> en
Password:
MSBR# configure data

MSBR(config-data)# interface GigabitEthernet 0/0

MSBR(conf-if-GE 0/0)# ip address 100.0.10.10 255.255.0.0

MSBR(conf-if-GE 0/0)# exit

MSBR(config-data)# exit

MSBR# configure system

MSBR(config-system)# cli-terminal
```

```
MSBR(cli-terminal)#
activate defaults exit help
history list pwd quit
set

MSBR(cli-terminal)# set ssh on

MSBR(cli-terminal)#
activate defaults exit help
history list pwd quit
set

MSBR(cli-terminal)# set wan-ssh-allow on
Note: Setting this parameter requires a reset.

MSBR(cli-terminal)*# exit

MSBR(config-system)*# exit

MSBR*# write
Writing configuration...done

MSBR*#
```

This page is intentionally left blank.

8 Upgrading the MSBR

8.1 Upgrading the MSBR via CLI

Upgrading the device from the network is possible using HTTP, HTTPS or TFTP servers.

Command	Description
MSBR# copy firmware from **http://10.31.2.7/MP500_MSBG_SIP_F6.80.264.cmp**	Copies software from http server
MSBR# copy firmware from http://10.31.2.7/MP500_MSBG_SIP_F6.80.264.cmp	Copies software from https server
MSBR# copy firmware from tftp://10.31.2.7/MP500_MSBG_SIP_F6.80.264.cmp	Copies software from tftp server

After issuing the *copy* command, the device will load the software version and reboot.

8.2 Example

```
MSBR # copy firmware from http://10.180.1.215
/MP500_MSBG_SIP_F6.80A.281.004.cmp


  % Total    % Received % Xferd  Average Speed   Time    Time
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
100 33.9M  100 33.9M    0     0   936k      0  0:00:37  0:00:37 --
:--:--  936k

Processing firmware file. The system will reboot when done...
```

8.3 Upgrading from Version 6.6

In Version 6.8, the image file, in addition to the MSBR software, contains an image for the ADSL component. In Version 6.6, the ADSL component is not in the MSBR software image file, so it needs to be updated manually. When upgrading from a sub-version of 6.8 to another sub-version of 6.8 (e.g., 6.8.120 to 6.8.121), the ADSL component is automatically installed. However, while upgrading from Version 6.6 to Version 6.8, the ADSL component is not installed.

➢ **To upgrade from Version 6.6 to Version 6.8 in order to upgrade the ADSL component:**

1. Upgrade from Version 6.6 to Version 6.8 using the steps described in Section 8.1 on page 35. The device will reboot.
2. Perform the upgrade again to the same image as described in Section 8.1 on page 35. The image will be loaded and the ADSL component will be upgraded. A reboot of the MSBR is not required, and the MSBR will not reboot by itself.

It is also possible to upgrade the A/VDSL image before upgrading to Version 6.8. This is sometimes useful, when the upgrade is performed via the DSL link itself(*). In this case, the upgrade of Version 6.8 is required to be done only once. The command for uploading the A/VDSL image is *copy adsl-firmware from http://adress/file.* As with MSBR software, the URL can be HTTP, HTTPS or TFTP server.

Note: The exact upgrade technique, especially between major versions, has to be carefully planned and verified at the customer lab, before applied to the field.

Command	Description
`MSBR# copy adsl-firmware from http://10.31.2.7/ADSL_A_F6.80.281.004.img`	Copy ADSL software from http server

This command is only available in Version 6.6.

8.4 Example

This example describes the output of upgrading from image *MP500_MSBG_SIP_F6.80A.281.004.cmp* to the same image *MP500_MSBG_SIP_F6.80A.281.004.cmp*.

```
MSBR# copy firmware from http://
10.180.1.215/MP500_MSBG_SIP_F6.80A.281.004.cmp

  % Total    % Received % Xferd  Average Speed   Time    Time
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
100 33.9M  100 33.9M    0     0   938k      0  0:00:37  0:00:37 --
:--:--  943k

Processing firmware file. The system will reboot when done...

Firmware file was not modified. Update skipped.
```

9 Automatic Update

The Automatic Update feature allows you to download a configuration file or an image file from a server. If the file is different from the file currently on the MSBR, it will be applied using the same rules as the `copy` command. In other words, configuration of the "cli-script" is added to the current configuration, and the "startup-script" will then rewrite the configuration and the MSBR will reset twice.

To configure Automatic Update, use the following commands:

Command	Description
`MSBR# configure system`	Accesses the System configuration level.
`(config-system)# automatic-update`	Accesses the Automatic-Update configuration level.
`(automatic-update)# set <file> from <URL>`	Sets file to check for update. This file is checked at the URL and will be applied if it is different than the file on the MSBR.
`(automatic-update)# set update-frequency <minutes>`	Sets the frequency for checking for an update.

The <file> for the Automatic Update can be one of the following:

File	Description
`adsl-firmware`	ADSL firmware file
`call-progress-tones`	Call progress tones file
`cas-table`	CAS configuration table file
`cli-script`	CLI configuration file
`coder-table`	Code table file
`data-configuration`	Data configuration file
`dial-plan`	Dial plan file
`firmware`	Firmware, burn and reload
`nqm-history`	Export Network Quality Monitoring history file
`prerecorded-tones`	Prerecorded tones file
`startup-script`	CLI configuration file
`tls-cert`	TLS certificate file
`tls-private-key`	TLS private key file
`tls-root-cert`	TLS trusted root certificate file
`user-info`	User info file
`voice-configuration`	Voice configuration file (ini file)
`voice-prompts`	Voice prompts file
`voice-xml`	Voice xml file
`web-logo`	WEB logo file

9.1 Example

In this example, Auto-Update will be configured to get the cli-script file from HTTP server, with a frequency of one minute. Later on, the hostname in the fetched configuration file will be changed.

```
tim@Server:~$ ssh Admin@192.168.0.1
Welcome to AudioCodes CLI
Admin@192.168.0.1's password:
Last login: Wed Mar 26 2014 at 10:52:14

MSBR> en
Password:
MSBR#

MSBR# configure system

MSBR(config-system)# automatic-update

MSBR(automatic-update)#
MSBR(automatic-update)# set update-frequency 1
Note: Changes to this parameter will take effect when applying the
'activate' or 'exit' command

MSBR(automatic-update)# set cli-script "http://192.168.0.199/cli-
conf.txt"
Note: Changes to this parameter will take effect when applying the
'activate' or 'exit' command

MSBR(automatic-update)# activate

MSBR(automatic-update)# exit

MSBR(config-system)# exit

MSBR#
```

Now the hostname in the file cli-conf.txt at the HTTP server is changed to "MSBR-2". After one minute, the hostname will be changed.

```
tim@Server:~$ ssh Admin@192.168.0.1
Welcome to AudioCodes CLI
Admin@192.168.0.1's password:
Last login: Wed Mar 26 2014 at 10:52:14

MSBR-2> en
```

The hostname changed to "MSBR-2".

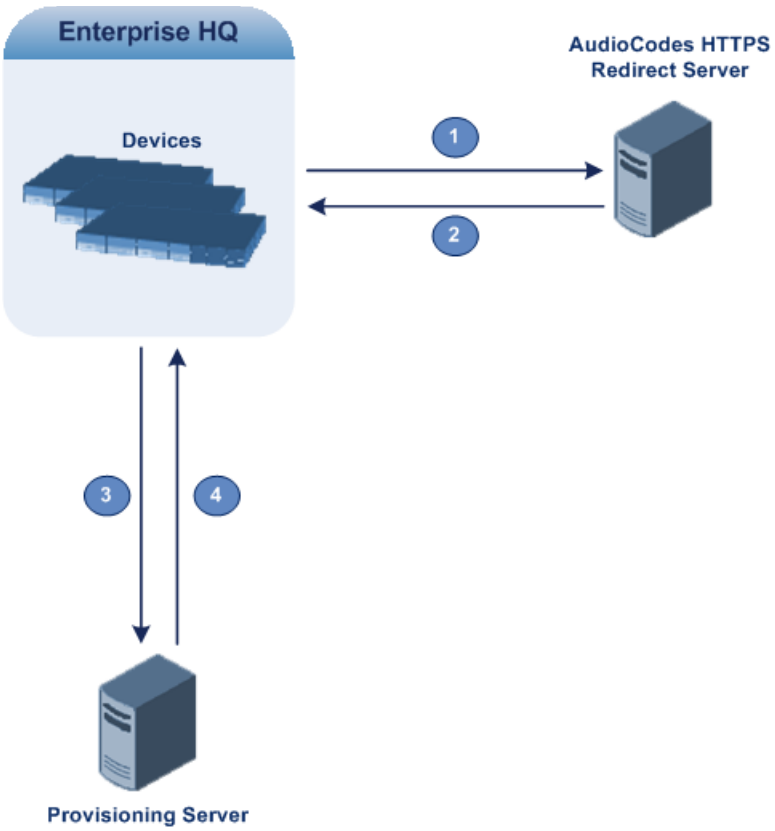
9.2 Zero Configuration

The Zero Configuration feature enables automatic, remote configuration of newly deployed, non-configured devices, using AudioCodes HTTPS Redirect Server. This feature offers an almost plug-and-play experience for quick-and-easy initial deployment of multiple devices at the end-customer premises. Zero Configuration requires only minimal pre-configuration of the device for WAN connectivity. Once an Internet connection is established, all that is needed is a device reset to activate the Zero Configuration mechanism.

Zero Configuration operates in combination with the Automatic Update feature. It redirects the device to an HTTP/S provisioning server from where the configuration file, configured with Automatic Update settings, can be downloaded and applied to the device. The device then performs the regular Automatic Update process according to these Automatic Update settings.

Once the device is powered up and connectivity to the WAN is established, it automatically sends an HTTP request to AudioCodes HTTPS Redirect server. If the device's MAC address is listed on the server, the server responds to the device with an HTTP Redirect response containing the URL of the HTTP/S server (typically, a provisioning server maintained by the Service Provider) where the configuration file is located. The device then downloads the configuration file from this provisioning server and updates its configuration. Typically, this configuration file only enables the Automatic Update mechanism and therefore, once downloaded, the device executes the Automatic Update mechanism accordingly.

Figure 9-1: Zero Configuration Process



The following describes the process that is described in Figure 9-1:

1. Device sends HTTPS request to AudioCodes HTTPS Redirect server.
2. Redirect server sends HTTPS response with redirect URL.
3. Device sends HTTPS request to redirected URL (i.e., provisioning server).
4. Device downloads configuration file for enabling the regular Automatic Update feature.

The configuration file contains only CLI commands for configuration, which its settings are applied to the device, in addition to the device's current configuration. The device resets only if the configuration file contains an explicit command instructing it to reset.

To enable Zero Configuration, the customer needs to define the devices on the HTTPS Redirect server by entering their MAC addresses and the configuration file URL. This may be done either through the corresponding Web interface or through SOAP/XML interface (that may be integrated with the Service Provider's provisioning system). For more information, contact AudioCodes support.

If the regular Automatic Update process succeeds, the device repeats the Zero Configuration process only if it undergoes a reset to factory defaults. If the Automatic Update process fails, the device repeats the Zero Configuration process at the next device reset or power up.

For security reasons, communication between the device and the HTTPS Redirect server is encrypted (HTTPS) and setup with mutual authentication. The device uses a special factory-set certificate to authenticate itself with the HTTPS Redirect server and to verify authenticity of the latter. If the redirect URL (where the configuration file is stored) also uses the HTTPS protocol, the device can use a regular certificate or the Zero Configuration certificate to authenticate itself and validate the server's certificate if a trusted root certificate (regular) is configured. This is determined by the AupdUseZeroConfCerts parameter.

If the Automatic Update feature has been configured, the Zero Configuration process is performed first. Only after Zero Configuration completes (successfully or not), does the Automatic Update process begin.

If the device is configured with multiple WAN interfaces, Zero Configuration is attempted on all configured WAN interfaces, sequentially.

The recommended method for using both Zero Configuration and Automatic Update is as follows:

- Zero Configuration is done to redirect the non-configured device to the URL of the provisioning server which contains only the configuration for the Automatic Update feature (e.g., CLI script URL and timeout for periodic update check).
- Once the Zero Configuration process completes (i.e., the device has downloaded the configuration file and applied the Automatic Update settings) without undergoing a reset, the Automatic Update mechanism begins.

To configure Zero Configuration, use the following commands:

Command	Description
`MSBR# configure system`	Access the System configuration level.
`(config-system)# automatic-update`	Access the Automatic-Update configuration level.
`(automatic-update)# set zero-conf on`	Activate zero configuration. **Note:** This configuration change requires a reset.
`(automatic-update)*# set zero-conf-server http://192.168.0.199/`	Configure the server IP address from which the router downloads the Zero Configuration. Note: This configuration change requires a reset.

10 NTP

The MSBR supports NTP clock synchronization. To configure NTP, use the following commands:

Command	Description
`MSBR# configure system`	Accesses the System configuration level.
`(config-system)# ntp`	Accesses the NTP configuration level.
`(ntp)# set primary-server 192.168.0.199`	Configures the primary NTP server.
`(ntp)# set secondary-server 192.168.0.3`	Configures the secondary NTP server.
`(ntp)# set utc-offset 120`	Adds two hours to the clock that is received from the NTP server. Use this command to calibrate the clock according to the MSBR's time zone.
`(ntp)# set auth-key-id 1`	Sets the authentication key ID. If 0, the authentication is off.
`(ntp)# set auth-key-md5 <key>`	Sets the key for the authentication.
`(ntp)# activate`	Activates the NTP configuration.

It is possible to send NTP requests from a specific interface.

Command	Description
`(ntp)# source data int g 0/0`	Select interface g 0/0 as source for NTP requests
`(ntp)# source voip`	Select voice as a source for NTP requests

To view NTP status, use the following command:

Command	Description
`MSBR# show system ntp-status`	Displays the NTP status.

10.1 Examples

The following example configures NTP:

```
ntp
    set secondary-server "192.168.0.3"
    set primary-server "192.168.0.199"
    activate
```

Output of the "show system ntp-status" command

```
MSBR# show system ntp-status
Configured NTP server #1 is 192.168.0.199
Configured NTP server #2 is 192.168.0.3
NTP is synchronized, stratum 0, reference is INIT
** Precision        0.00000 seconds
** Root delay       0.00000 seconds
** Root dispersion  0.01824 seconds
** Reference time   00000000.00000000 (2036-02-07 06:28:16 UTC)
** UTC offset       0 seconds
Current local time: 2014-03-16 10:49:03
```

The output contains synchronization status, synchronization data, and a synchronized clock.

11 Banner Message

The banner message appears when the administrator connects to the MSBR. To configure the banner message, use the following commands:

Command	Description
`MSBR# configure system`	Accesses the System configuration level.
`(config-system)# welcome-msg [index | new | display]`	You can configure 20 banner messages, index counting from 0 to 19. The `new` keyword configures the first banner message with an empty configuration. The `display` keyword displays the banner configuration.
`(welcome-msg-0)# set text "banner text"`	Enters the message and enclose it in double apostrophes.

11.1 Example

This example below configures a short banner message:

```
MSBR# configure system

MSBR(config-system)# welcome-msg 0

MSBR(welcome-msg-0)# set text "Property of AudioCodes"
Note: Changes to this parameter will take effect when applying the
'activate' or 'exit' command

MSBR(welcome-msg-0)# exit
MSBR(config-system)#
MSBR# exitConnection closed by foreign host.
tim@Server:~$
```

The message will appear when connecting to the MSBR:

```
tim@Server:~$ telnet 192.168.2.1
Trying 192.168.2.1...
Connected to 192.168.2.1.
Escape character is '^]'.

Property of AudioCodes


Username: Admin
Password:

MSBR>
```

This page is intentionally left blank.

12 RADIUS Configuration

MSBR supports the RADIUS protocol. Use the following configuration steps to configure the MSBR to authenticate using RADIUS with an external RADIUS server.

Command	Description
`MSBR# configure system`	Accesses the System configuration level.
`(config-system)# radius`	Accesses the RADIUS configuration level.
`(radius)# set auth-server-ip 192.168.0.199`	Configures the RADIUS server IP address. **Note:** This configuration requires a reset.
`(radius)# source data interface vlan 1`	**Optional:** set source interface for communicating with RADIUS server
`(radius)*# set auth-server-port 1812`	Configures the RADIUS server port number. **Note:** This configuration requires a reset.
`(radius)*# set enable-mgmt-login on`	Enables RADIUS for access to the MSBR's management interface.
`(radius)*# set allow-console-bypass on`	Enables option to bypass RADIUS authentication when user is connected directly via serial interface.
`(radius)*# exit` `(config-system)*# exit`	Exits to the main configuration level.
`MSBR*# reload now` `Writing configuration and restarting...`	Resets the MSBR.

You can also use an internal RADIUS server in the MSBR. To configure an internal RADIUS server, use the following configuration step:

Command	Description
`MSBR# configure system`	Enters system configuration level

12.1 Example

12.1.1 FreeRADIUS Configuration

In this example, a program called FreeRADIUS acts as a RADIUS server and is used to authenticate administrators connecting to the MSBR. The program is installed on the Ubuntu Linux platform.

1. Use the following commands in Ubuntu to install FreeRADIUS:

```
sudo apt-get install mysql-client mysql-server
sudo apt-get install freeradius freeradius-utils freeradius-mysql
sudo apt-get install php5 php-pear php5-gd php-DB
```

2. After the installation is complete, use the following configuration to configure the MSBR in the FreeRADIUS server. Edit the file "clients.conf", and add the IP address of the VoIP CPU as the RADIUS client:

```
' open the "clients.conf" file for edit
edit sudo nano /etc/freeradius/clients.conf

' add the clients at the bottom of the file
client 192.168.0.2 {
secret=P@ssw0rd
shortname=audiocodes
}
```

3. Edit the "client" file to add the users:

```
' open the "users" file for edit
sudo nano /etc/freeradius/users

' add the users at the bottom
tim Cleartext-password := "P@ssw0rd"
```

4. Use the next set of commands to configure the MSBR to work with radius server

```
radius
    set enable on
    set auth-server-port 1812
    set auth-server-ip 192.168.0.199
    set enable-mgmt-login on
    activate
   exit
```

12.1.2 Internal RADIUS Configuration

When the RADIUS server is internally activated for the MSBR, wireless security (WPA2-Enterprise) and LAN security (802.1x) can work with the internal server, allowing easier deployment.

This supports both password-based authentication and certificates.

```
configure data
  dot1x local-user AUDIOCODES-USER password 1234
  dot1x local-user dot1x password PASSWORD
```

- **Wireless:**

```
  interface dot11radio 1
    security 802.1x radius local
    security wpa mode 802.1x
    security mode wpa2
    no shutdown
```

- **Wired:**

```
  dot1x lan-authentication enable
  dot1x radius-server local
  interface GigabitEthernet 4/4
    authentication dot1x
```

12.1.2.1 Testing Password-based Authentication on Windows

➢ **To test password-based authentication on Windows:**

1. On the Windows task bar, click the **wireless** icon to open the Wireless Network Connection window.
2. In the list of wireless connections, right-click the MSBR wireless connection and then from the shortcut menu, choose **Properties**; the Wireless Network Properties dialog box appears.
3. Click the **Security** tab, and then click the **Settings** button, located alongside the PEAP authentication method; the Protected EAP Properties dialog box appears.
4. Clear the **Validate server certificate** check box.
5. Click the **Configure** button; the EAP MSCHAPv2 Properties dialog box appears.
6. Clear the **Automatically use my Windows logon name and** check box, and then click **OK**.

12.1.2.2 Testing Certificates

➢ **To test certificates:**

1. Load a signed certificate to the MSBR.
2. Reset the MSBR.
3. Load a client certificate to your PC, and then install the CA certificate.
4. On the Windows task bar, click the wireless icon to open the Wireless Network Connection window.
5. In the list of wireless connections, right-click the MSBR wireless connection and then from the shortcut menu, choose **Properties**; the Wireless Network Properties dialog box appears.
6. Click the **Security** tab.
7. From the 'Choose a network authentication method' drop-down list, select **Smart card or other certificate**, and then click **OK**.

This page is intentionally left blank.

13 TACACS+ Configuration

MSBR supports the TACACS+ protocol. Use the following configuration steps to configure the MSBR to authenticate using TACACS+.

Command	Description
`MSBR# configure data`	Accesses the Data configuration level. TACACS+ configuration needs to be done in the data level.
`(config-data)# tacacs-server host 192.168.0.199`	Configures the TACACS+ server IP address.
`(config-data)# aaa authentication login tacacs+ local`	Configures authentication using the TACACS+ server. The local keyword means that if the TACACS+ server is unavailable, the local user configuration is used to authenticate.
`(config-data)# aaa authentication login tacacs+ allow-console-bypass authentication`	Allow bypassing TACACS+ authentication when user is connected via serial interface. After login, non-privileged commands will be allowed without negotiating with the TACACS+ Server. This will not affect TACACS+ users.
`(config-data)# aaa authentication login tacacs+ allow-console-bypass authentication authorization`	Allow bypassing TACACS+ enable authorization (privileged mode) when the user is connected via serial interface. After login, privileged commands will be allowed without negotiating with the TACACS+ server. This will not affect TACACS+ users.
`(config-data)# tacacs-server key <key>`	Assigns the shared <key> to the TACACS+ server.
`(config-data)# aaa authorization login tacacs+`	Configures authorization for the login using the TACACS+ server.
`(config-data)# aaa authorization command tacacs+`	Configures authorization for commands using the TACACS+ server.
`(config-data)# aaa accounting command start-stop tacacs+`	Configures accounting for commands using the TACACS+ server.
`(config-data)# aaa accounting exec start-stop tacacs+`	Configures accounting for execution using the TACACS+ server.

The MSBR sends packets to the TACACS+ server from its VoIP CPU. If the TACACS+ server is installed on the LAN side, no problems are experienced, because the VoIP CPU IP address is local. However, if the TACACS+ server is on the WAN side, the packets, originating from the VoIP CPU's local IP address, need to be NAT'ed. Use the `NATP enable` command or preferably, a NAT rule to make sure that the packets that are arriving to the TACACS+ server come from the same IP address. In this case, the NAT IP address needs to be configured as the host address. From version 6.8, the source address for the TACACS+ server can be configured using CLI.

13.1 Example for TACACS+ Authentication

In this example, simple authentication using a TACACS+ server is configured. The TACACS+ server is installed on an Ubuntu Linux server.

To install a TACACS+ server, use the following command on Ubuntu server:

```
apt-get install tacacs+
```

Tacacs_plus server configuration can be found in the "/etc/tacacs+/tac_plus.conf" file. Edit this file using a text editor such as vi or nano, and make sure that the following configuration line is in this file:

```
# This is the shared key that MSBR uses to access Tacacs+
key = P@ssw0rd

# Tacacs host ip address. In our case it the NATed VOIP CPU
address
host = 180.1.100.151 {
        key = P@ssw0rd
}
# Username configuration
user = AudioCodes {
        name = "AudioCodes"
        member = staff
        login = cleartext P@ssw0rd
}
# user $enab15$. This is a user that configured for the MSBR's
enable command
user = $enab15$ {
    login = cleartext P@ssw0rd
}
# AudioCodes's username group configuration permits all commands
group = staff {
        cmd = conf {
                permit .*
        }
}
```

Remember to restart the TACACS+ service on the server, using the following command:

```
root@server-VirtualBox:~# sudo service tacacs_plus restart
 * Restarting TACACS+ authentication daemon tacacs+
[ OK ]
root@server-VirtualBox:~#
```

MSBR configuration:

```
Conf data
MSBR2# conf data

MSBR2(config-data)# aaa authentication login tacacs+

MSBR2(config-data)# tacacs-server host 192.162.0.199

MSBR2(config-data)# tacacs-server key P@ssw0rd
```

```
#Configure NAT for the WAN side
MSBR(config-data)# access-list tacacs_ACL permit ip 192.168.0.2
0.0.0.0 any


MSBR(config-data)# ip nat pool tacacs_srv 180.1.100.151
180.1.100.151


MSBR(config-data)#  ip nat inside source list tacacs_ACL interface
GigabitEthernet 0/0 pool tacacs_srv
```

13.2 Example for TACACS+ Authorization

In this example, the TACACS+ server is used to authenticate two types of administrators -- voice administrator and data administrator. The voice administrator will have access to voice configuration, and the data administrator will have access to data administration.

1. Configure authorization and authentication in the MSBR to work with TACACS+:

```
Conf data
MSBR2# conf data
MSBR2(config-data)# aaa authentication login tacacs+
MSBR2(config-data)# aaa authorization command tacacs+
MSBR2(config-data)# tacacs-server host 192.162.0.199
MSBR2(config-data)# tacacs-server key P@ssw0rd
```

2. Configure the TACACS server to authenticate two different user types. The following is the voice user configuration on TACACS+ server on Ubuntu Linux, in the "/etc/tacacs+/tac_plus.conf" file:

```
user = voice-user {
name = "Voice administrator"
member = voice-admin
login = cleartext P@ssw0rd
}
```

3. The data user configuration:

```
user = data-user {
        name = "Data administrator"
        member = data-admin
        login = cleartext P@ssw0rd
}
```

The user names are "voice-admin" and "data-admin". The voice-user is a member of the "voice-admin" group. The data-user is a member of the "data-admin" group. The password for both is "P@ssw0rd".

4. Configure the "voice-admin" and "data-admin" groups, and the commands each group is allowed to use:

```
# voice group
group = voice-admin {
cmd = configure {
      permit voip
}
cmd = enable {
      permit .*
}
      cmd = access-list {
      permit .*
}
cmd = appli-enabling {
      permit .*
}
cmd = coders-and-profiles {
      permit .*
}
cmd = control-network {
      permit .*
}
```

```
cmd = dns {
      permit .*
}
cmd = ether-group {
      permit .*
}
cmd = exit {
      permit .*
}
cmd = gw {
      permit .*
}
cmd = help {
      permit .*
}
cmd = history {
      permit .*
}
cmd = interface {
      permit .*
}
cmd = ip-media {
      permit .*
}
cmd = ldap {
      permit .*
}
cmd = list {
      permit .*
}
cmd = media {
      permit .*
}
cmd = physical-port {
      permit .*
}
cmd = pwd {
      permit .*
}
cmd = qos {
      permit .*
}
cmd = quit {
      permit .*
}
cmd = rba {
      permit .*
}
cmd = routing {
      permit .*
}
```

```
cmd = sas {
      permit .*
}
cmd = sbc {
      permit .*
}
cmd = services {
      permit .*
}
cmd = sip-definition {
      permit .*
}
cmd = tdm {
      permit .*
}
cmd = do {
      permit .*
}
cmd = no {
      permit .*
}
}
#data group
group = data-admin {
cmd = configure {
      permit data
}
        cmd = enable {
                permit .*
}
cmd = aaa {
      permit .*
}
cmd = access-list {
      permit .*
}
cmd = backup-group {
      permit .*
}
cmd = crypto {
      permit .*
}
cmd = exit {
      permit .*
}
cmd = help {
      permit .*
}
cmd = history {
      permit .*
}
```

```
cmd = interface {
      permit .*
}
cmd = ip {
      permit .*
}
cmd = key {
      permit .*
}
cmd = l2tp-server {
      permit .*
}
cmd = list {
      permit .*
}
cmd = lldp {
      permit .*
}
cmd = pptp-server {
      permit .*
}
cmd = pwd {
      permit .*
}
cmd = qos {
      permit .*
}
cmd = quit {
      permit .*
}
cmd = route-map {
      permit .*
}
cmd = router {
      permit .*
}
cmd = router-id {
      permit .*
}
cmd = service {
      permit .*
}
cmd = spanning-tree {
      permit .*
}
cmd = tacacs-server {
      permit .*
}
cmd = track {
      permit .*
}
```

```
cmd = vpn-users {
      permit .*
}
cmd = web-restrict {
      permit .*
}
cmd = do {
      permit .*
}
cmd = no {
      permit .*
}
}
```

13.3 TACACS+ Flags and Flow Chart

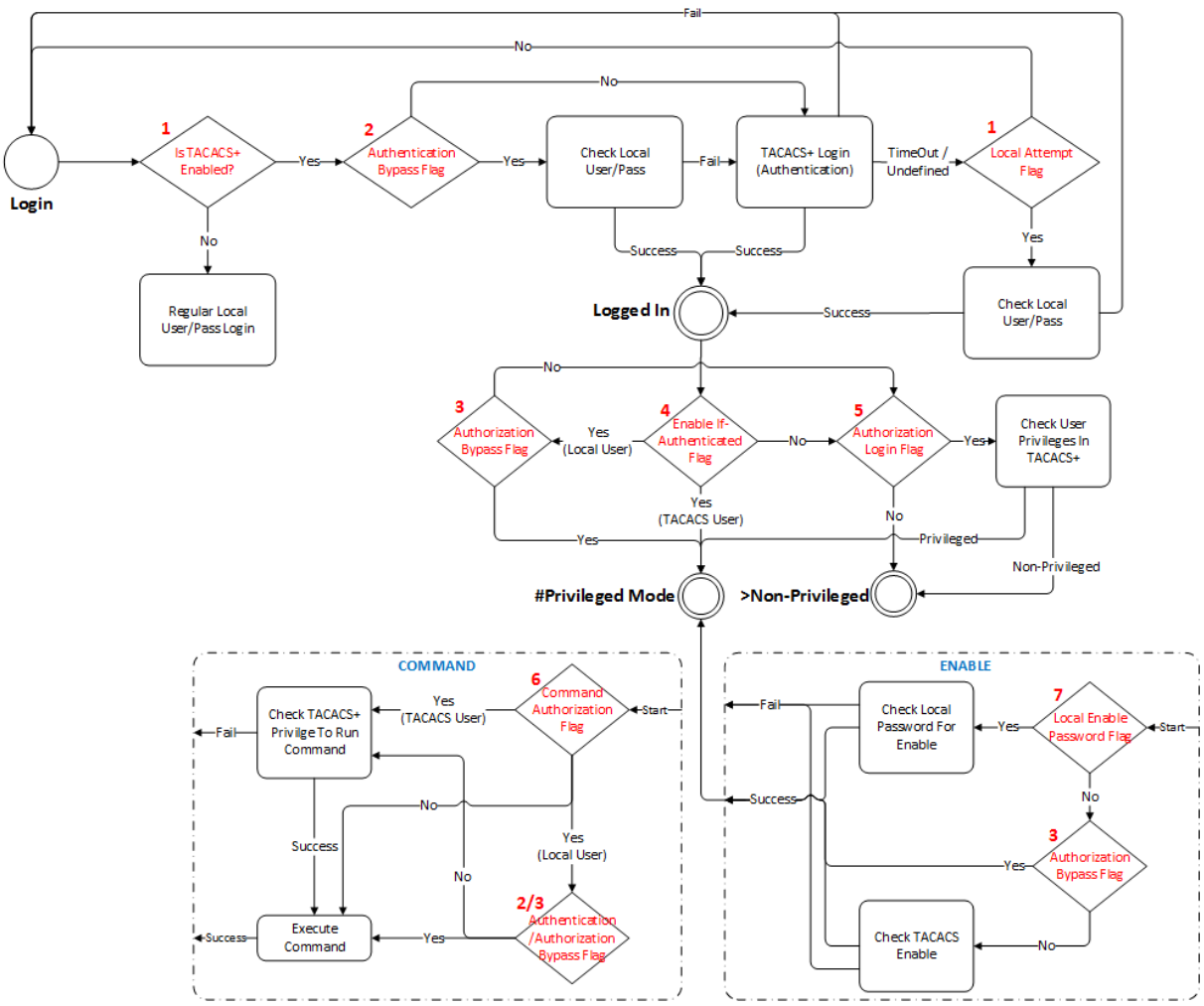
This section describes the TACACS+ flags and flow chart.

13.3.1 TACACS+ Configuration Flags

- aaa authentication login tacacs+ <local>
- aaa authentication login tacacs+ allow-console-bypass authentication
- aaa authentication login tacacs+ allow-console-bypass authentication authorization
- aaa authorization enable if-authenticated tacacs+
- aaa authorization login tacacs+
- aaa authorization command tacacs+
- aaa authorization enable local tacacs+

13.3.2 TACACS+ Flow Chart

Figure 13-1: TACACS+ Flow Chart



This page is intentionally left blank.

14 Recovery Procedures

14.1 Password Recovery Procedure

If the login password for accessing the device's management interface has been forgotten, the Password Recovery procedure can be used to gain access to the MSBR. Press the MSBR's reset button for 15 to 30 seconds. The MSBR's configuration is deleted and the username and password are set to "Admin". The enable password is also set to "Admin".

14.2 Rescue Process

If the MSBR's operation system file has been corrupted, follow the Rescue process to rescue the MSBR. Press the MSBR's reset button for more than thirty seconds. The MSBR resets and uses the BootP protocol to boot itself from the first LAN port. All other ports enter shutdown mode. This is called the *Rescue Mode*. The MSBR also enters rescue mode if it resets as a result of crashing three times while booting, or if a software upgrade fails.

The following is the description of the rescue procedure:

1. Attach a computer to the first LAN port of the MSBR.
2. Configure the IP address 192.168.0.3/24 on the attached computer.
3. Verify the MTU size. The MTU mustn't be greater than 1500. To set the MTU size in Windows 7:
   a. Start CMD.
   b. Type "netsh", and then press Enter.
   c. Type "interface ipv4", and then press Enter.
   d. Type "set interface "Local Area Connection" mtu=1500", and then press Enter.
4. Create a BootP client in the BootP/TFTP utility.
5. Assign the IP address of 192.168.0.2/24 to the MAC address of the MSBR.
6. Select the .cmp file to upload to the MSBR.
7. Boot the MSBR to rescue mode by pressing the reset button for 30 seconds; the MSBR downloads the .cmp image file.

This page is intentionally left blank.

15 Factory Setting

To delete the MSBR's configuration, use the following command:

Command	Description
`MSBR# write factory`	Clears configuration and resets the MSBR.

The MSBR's configuration can also be cleared be pressing the reset button for a period of 15 to 30 seconds.

This page is intentionally left blank.

16 MSBR Reload

To reload the MSBR, enter the following command:

Command	Description
`MSBR# reload now`	Saves configuration and resets the MSBR.

An alternative method to reload the MSBR is by pressing the reset button for a period of one to fifteen seconds.

This page is intentionally left blank.

17 Certificates

To import certificates, use the following command:

Command	Description
`MSBR# copy <cert file> from <server>`	Copies the certificate file from the server.

The certificate file can be one of the following:

File	Description
`tls-cert`	TLS Certificate file.
`tls-private-key`	TLS Private Key file.
`tls-root-cert`	TLS Trusted-Root Certificate file.

17.1 Example

This example uses the `copy` command to download the certificate from the TFTP server to the MSBR.

```
MSBR# copy tls-cert from tftp://192.168.0.3/cert.pem
Copying file... 0 bytes
done.
use 'write' command in order to burn to NV memory

MSBR# copy tls-root-cert from tftp://192.168.0.3/caroot.pem
Copying file... 0 bytes
done.
use 'write' command in order to burn to NV memory

MSBR# copy tls-private-key from tftp://192.168.0.3/pkey.pem
Copying file... 0 bytes
done.
use 'write' command in order to burn to NV memory

MSBR# write
Writing configuration...done

MSBR#
```

This page is intentionally left blank.

18 Syslog

The MSBR supports remote logging. To configure the remote Syslog server, use the following commands:

Command	Description
`MSBR# configure system`	Accesses the Data system configuration level.
`(config-system)# logging`	Accesses the logging configuration level.
`(logging)# set syslog-ip 192.168.0.3`	Configures the Syslog server's IP address. **Note:** Changes to this parameter will take effect when applying the `activate` or `exit` command.
`(logging)# source data interface vlan 1`	**Optional:** Set source interface for sending syslog messages
`(logging)# set debug-level 0`	Sets the debug level, where 0 is the lowest debug level and 7 is the highest.
`(logging)# activate`	Activates the configuration.

The configurable debug levels are from 0 to 7. The most common option is level 1, where the VoIP debug is enabled. At level 0, the VoIP debug is disabled, however at level 1, VoIP debugging is enabled.

18.1 Examples

The following is an example of the Syslog configuration:

```
MSBR# conf syst
MSBR(config-system)# logging
MSBR(logging)# set syslog-ip 192.168.0.3
Note: Changes to this parameter will take effect when applying the
'activate' or 'exit' command
MSBR(logging)# set debug-level 0
MSBR(logging)# activate
```

Log messages received at the Syslog server for state changes in interface Gig0/0:

```
Mar 16 13:10:31 192.168.0.2 [S=354] RAISE-
ALARM:acDataInterfaceStatus; Textual Description: Data interface
GigabitEthernet 0/0 is DOWN; Severity:indeterminate; Source:;
Unique ID:6;


Mar 16 13:10:40 192.168.0.2 [S=357] RAISE-
ALARM:acDataInterfaceStatus; Textual Description: Data interface
GigabitEthernet 0/0 is UP; Severity:indeterminate; Source:; Unique
ID:7;
```

This page is intentionally left blank.

19 Network Quality Monitor

This chapter describes the Network Quality Monitoring (NQM) feature.

19.1 Overview

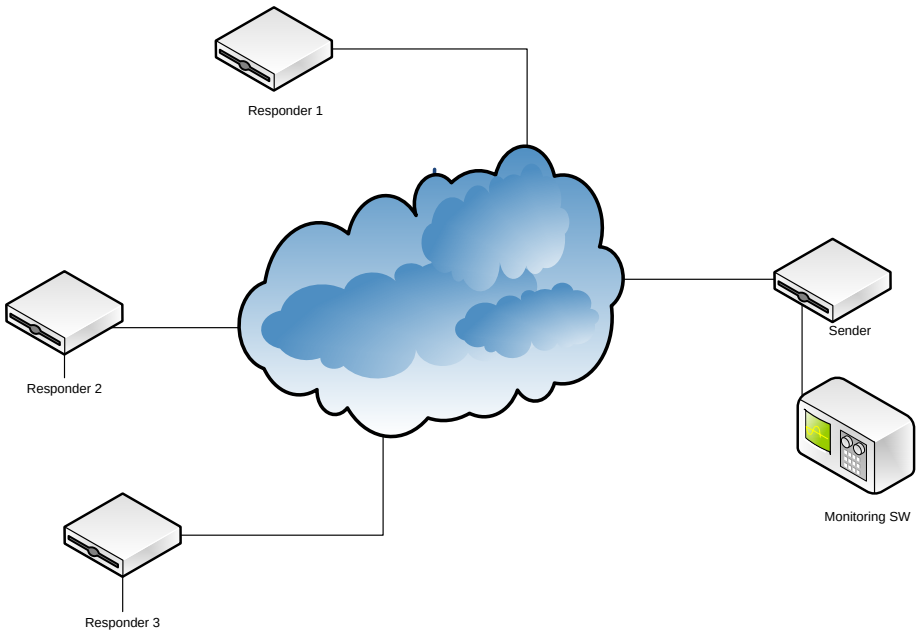
The NQM feature is designed for monitoring the quality of a current network path between two network NQM terminations, a 'Sender termination' and a 'Responder termination'.

The quality is measured according to the following criteria:

- Round trip time
- Packet jitter
- Packet loss rate
- Listener quality MOS as per ITU-T spec.[1].
- Conversation quality MOS as per ITU-T spec[2].

The figure below illustrates the network paths between the Responder and the Sender termination points.

Figure 19-1: Network Quality Monitor Overview



[1] Available only when packets sent are a valid g711 stream in terms of payload size and packet interval. – see table in Section 19.1.1 for valid g711 parameter values.
[2] See note 1 above.

19.1.1 MOS Results

The table below shows the legal pair values for valid MOS results.

Sender table parameter → packet-interval [msec]	Sender table parameter → Payload-size [bytes]
5	60
10	100
20	180
40	340
60	500
80	660
100	820
120	980

19.2 How to Setup NQM

Enter the 'configure system' sub menu in the CLI as follows:

```
MSBR#> enable
Password:
MSBR# configure system
MSBR(config-system)#
```

19.3 Configuring the 'Sender Termination' Side

This section describes how to configure the Sender Termination side.

19.3.1 Step 1: Bind a WAN Interface to the NQM Service

Bind a WAN interface to the NQM service:

```
MSBR(config-system)# bind GigabitEthernet 0/0 nqm
MSBR(config-system)#
```

Note: The chosen WAN interface should be the interface on which the NQM packets are planned to flow bi-directionally and binding is necessary to create the corresponding static NAT rules.
If the NQM session is planned to flow within the LAN, then no binding is needed and this step can be skipped.

19.3.2 Step 2: Configure a Line in the Probing Table

Configure a line in the Probing table:

```
MSBR(config-system)# nqm probing-table 0
MSBR(probing-table-0)#
```

Configure a Probe name – name tag to identify this line:

```
MSBR(probing-table-0)# set probe-name voip_probe_1
```

Activate the probe line:

```
MSBR(probing-table-0)# exit
MSBR(config-system)#
```

19.3.3 Step 3: Configure a Line in the Sender Table to Define a Sender Termination

Configure a line in the Sender table to define a Sender termination:

```
MSBR(config-system)# nqm sender-table 0
MSBR(sender-table-0)#
```

Configure a Sender name – name tag to identify this specific sender:

```
MSBR(sender-table-0)# set sender-name main_office_voip_checker_1
```

Configure a Target IP address – set IP address of Responder termination:

```
MSBR(sender-table-0)# set target-ip 10.4.3.98
```

Configure a Target port – set port number on which the Responder termination listens:

```
MSBR(sender-table-0)# set target-port 3900
Activate this
MSBR(sender-table-0)# set start-time now
```

Note: A Responder termination defined by the pair <target IP address, target port> can be defined only once for a single sender line. Two or more senders can't be defined to send packets to the same Responder termination.

Configure a Probe name – name of probing line previously configured to be used by this sender:

```
MSBR(sender-table-0)# set probe-name voip_probe_1
```

Note: A single probe line in the probing table may be shared by several senders thereby sharing and simplifying common attributes configuration.

Configure a Source network interface name – name of network interface to send packets:

```
MSBR(sender-table-0)# set source-interface-name OAM_IF
```

Note: If you wish to output packets to the WAN interface, simply set NQM_WAN as the source interface name, otherwise set the interface name to be a specific interface name found in the network interface table.

Activate the sender line:

```
MSBR(sender-table-0)# exit
MSBR(config-system)#
```

19.4 Configuring the 'Responder Termination' Side

Enter the 'configure system' sub menu in the CLI:

```
MSBR> enable
Password:
MSBR# configure system
MSBR(config-system)#
```

19.4.1 Step 1: Bind a WAN interface to the NQM service

Bind a WAN interface to the NQM service:

```
MSBR(config-system)# bind GigabitEthernet 0/0 nqm
MSBR(config-system)#
```

Note: The chosen WAN interface should be the interface on which the NQM packets are planned to flow bi-directionally and binding is necessary to create the corresponding static NAT and port forwarding rules.
If the NQM session is planned to flow within the LAN, then no binding is required and therefore this step can be skipped.

19.4.2 Step 2: Configure a Line in the Responder Table

Configure a line in the Responder table as follows:

```
MSBR(config-system)# nqm responder-table 0
MSBR(responder-table-0)#
```

Configure a Responder name – name tag to identify this line:

```
MSBR(responder-table-0)# set responder-name
main_office_voip_responder_1
```

Configure a Target port – set port number on which the Responder termination listens:

```
MSBR(responder-table-0)# set local-port 3900
```

Note: Make sure the local-port value is in-sync with the target-port value set for the corresponding Sender termination.

Configure the source network interface name – name of network interface to listen for incoming packets:

```
MSBR(sender-table-0)# set source-interface-name OAM_IF
```

Note: If you wish to listen to the WAN interface, simply set NQM_WAN as the source interface name, otherwise set the interface name to be a specific interface name found in the Network Interface table

Note: Make sure the network interface that the Responder termination is listening upon is in-sync with the target-ip value set for the corresponding Sender termination.

Activate the responder line:

```
MSBR(responder-table-0)# exit
MSBR(config-system)#
```

19.5 Viewing Results

This section describes how to view the results of the Responder termination.

19.5.1 CLI interface

On the Sender termination device, in the CLI, configure the following:

```
MSBR> enable
Password:
MSBR# show system nqm 0 8

|     Probe Time    | Valid | RTT |  PL  |  PL  | Total | Jit. | Jit. | Total | MOS | MOS |
|                   |       |     |  Tx  |  Rx  |   PL  |  Tx  |  Rx  | Jit.  | CQ  | LQ  |
|-------------------|-------|-----|------|------|-------|------|------|-------|-----|-----|
|01-01-2010@02:46:24|  yes  |   7|     0|     0|      0|     0|    17|     17|  0.0|  0.0|
|01-01-2010@02:47:24|  yes  |  10|     0|     0|      0|    30|     1|     31|  0.0|  0.0|
|01-01-2010@02:48:25|  yes  |   9|     0|     0|      0|    31|    20|     51|  0.0|  0.0|
|01-01-2010@02:49:25|  yes  |   6|     0|     0|      0|    32|     4|     36|  0.0|  0.0|
|01-01-2010@02:50:25|  yes  |   5|     0|     0|      0|     0|     5|      5|  0.0|  0.0|
|01-01-2010@02:51:25|  yes  |   5|     0|     0|      0|    15|    15|     30|  0.0|  0.0|
|01-01-2010@02:52:25|  yes  |   6|     0|     0|      0|    32|     7|     39|  0.0|  0.0|
|01-01-2010@02:53:25|  yes  |   6|     0|     0|      0|    30|     5|     35|  0.0|  0.0|

there are 10 entries in the log, displaying last 8 entries
```

```
MSBR#
```

19.5.2 SNMP Interface

Access the acSysNqmHistoryTable object.

SNMP OID info

Name: acSysNqmHistoryTable

Type: OBJECT-TYPE

OID: 1.3.6.1.4.1.5003.9.10.10.2.12.1

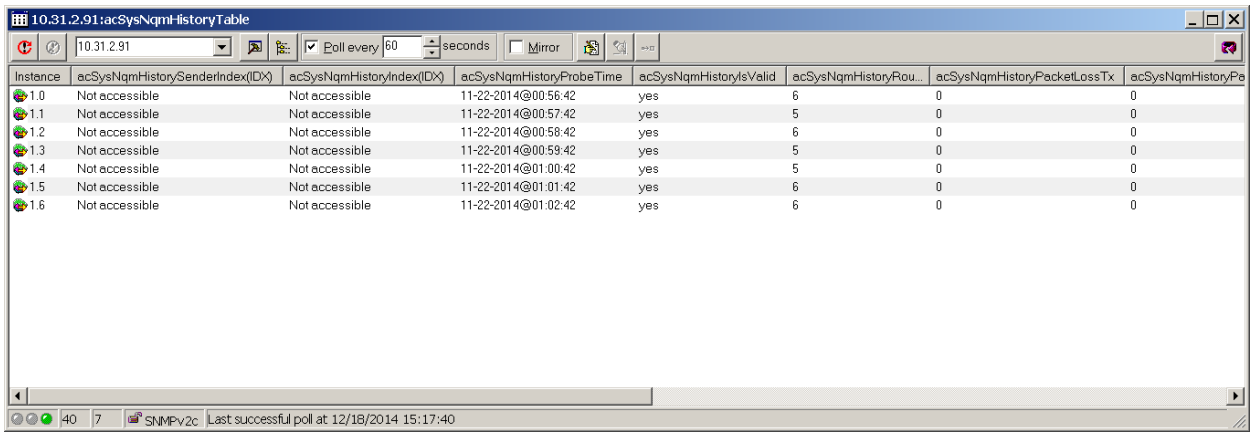
Full path: iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).audioCodes(5003).acProducts(9).acBoardMibs(10).acSystem(10).acSystemStatus(2).acSysNqmStatus(12).acSysNqmHistoryTable(1)

Module: AC-SYSTEM-MIB

Parent: acSysNqmStatus

First child: acSysNqmHistoryEntry

Figure 19-2: SNMP Interface



19.5.3 Examples

Sender Show run

```
## System Configuration

  configure system
   cli-terminal
    idle-timeout 0
    activate
   exit
   logging
    syslog on
    syslog-ip 10.31.2.44
    activate
   exit
   nqm probing-table 0
    probe-name "M500_Sender"
    start-time "now"
    activate
   exit
   nqm sender-table 0
    sender-name "NQM_Sender1"
    target-ip-address 10.31.2.86
    target-port 3910
    probe-name "M500_Sender"
    source-interface-name "NQM_WAN"
    activate
   exit
   exit
   nqm sender-table 1
    sender-name "NQM_Sender1"
    target-ip-address 10.31.2.84
    target-port 3910
    probe-name "M500_Sender"
    source-interface-name "NQM_WAN"
    activate
   exit
   ntp
    set primary-server "0.0.0.0"
    activate
   exit
   snmp
    no activate-keep-alive-trap
    activate
   exit
   web
    set https-cipher-string "RC4:EXP"
    activate
   exit
  hostname MSBR_NQM_Sender
  configuration-version 0
  bind interface gigabitethernet 0/0 nqm
  exit
```

Responder Show run

```
## System Configuration

  configure system
   cli-terminal
    wan-ssh-allow on
    wan-telnet-allow on
    ssh on
    idle-timeout 0
    activate
   exit
   nqm responder-table 0
    responder-name "NQM_84"
    local-port 3910
    source-interface-name "NQM_WAN"
    activate
   exit
   ntp
    set primary-server "0.0.0.0"
    activate
   exit
   snmp
    no activate-keep-alive-trap
    activate
   exit
   web
    wan-http-allow on
    set https-cipher-string "RC4:EXP"
    activate
   exit
  hostname NQM_84
  configuration-version 0
```

20 Debugging - Packet Capturing

The MSBR supports advanced debugging using packet capturing. The captured files are saved to a PCAP file. You can also send the file to an FTP or a TFTP server or save the file to a USB device connected to the MSBR. You can also save the file locally on the MSBR, where in this case, the file size is limited to 20 MB.

To capture traffic on a physical interface, use the following commands:

Command	Description
`MSBR# debug capture data physical eth-lan` `Interface eth-lan was added to the debug capture rules`	Sets the Ethernet interface as a source for capturing packets.
`MSBR# debug capture data physical target tftp`	Sets the destination for the captured packet file as a TFTP server.

`MSBR# debug capture data physical start`
`NOTE: Debug capture data will be collected locally, and later`
`sent to a PC via TFTP/FTP.`
`Please make sure that`
`VLAN 1 is defined and the PC is accessible through it.`
	Starts capturing files.
	Note: The capture data is collected locally, and only then sent to the PC later on.

`MSBR# debug capture data physical stop 192.168.0.3`
`Trying to send capture to TFTP/FTP server , filename debug-capture-data-16032014-154400`
`Finished`
 	
`MSBR#`
	The command stops capturing files and then uploads the file to a TFTP server with IP address 192.168.0.3.

`MSBR# debug capture data physical stop 192.168.0.3 VRF MGMT`
`Trying to send capture to TFTP/FTP server , filename debug-capture-data-16032014-154400`
`Finished`
 	
`MSBR#`
	There is an ability to stop the capture and send the captured traffic from a specific VRF, in this example, the VRF is called MGMT.

The available sources for file captures are listed below:

Source	Description
`cellular-wan`	Cellular WAN interface.
`eth-lan`	LAN Ethernet interfaces.
`eth-wan`	WAN Ethernet interfaces.
`fiber-wan`	WAN fiber interface.
`xdsl-wan`	Any DSL interface (ADSL, VDSL) that is installed on the MSBR.

Use the following commands to capture traffic on a logical interface:

Command	Description
MSBR# debug capture data interface <interface> <proto | ipsec> <all | arp | icmp | ip | ipv6 | tcp | udp> host <IP | IPv6 | all> <cr | port> <any | 1-65535 <cr | ftp | tftp> IP	▪ <interface>: interface to capture the data on. ▪ <proto | ipsec>: if IPSec is selected, it is decrypted and captured. ▪ <all | arp | icmp | ip | ipv6 | tcp | udp>: selects protocol for capturing. ▪ host <IP | IPv6 | all>: select traffic to capture using the IP or IPv6 address as a filter. ▪ <cr | port> <any | 1 – 65535>: select the port to capture or press Enter. If you press Enter, the packets are displayed in the console. ▪ <cr | ftp | tftp> IP: press Enter to display the captured packets on screen, or send captured packets to TFTP or FTP server.

To view the currently configured capture, use the following command:

Command	Description
MSBR# debug capture data physical show	Displays currently configured capture.

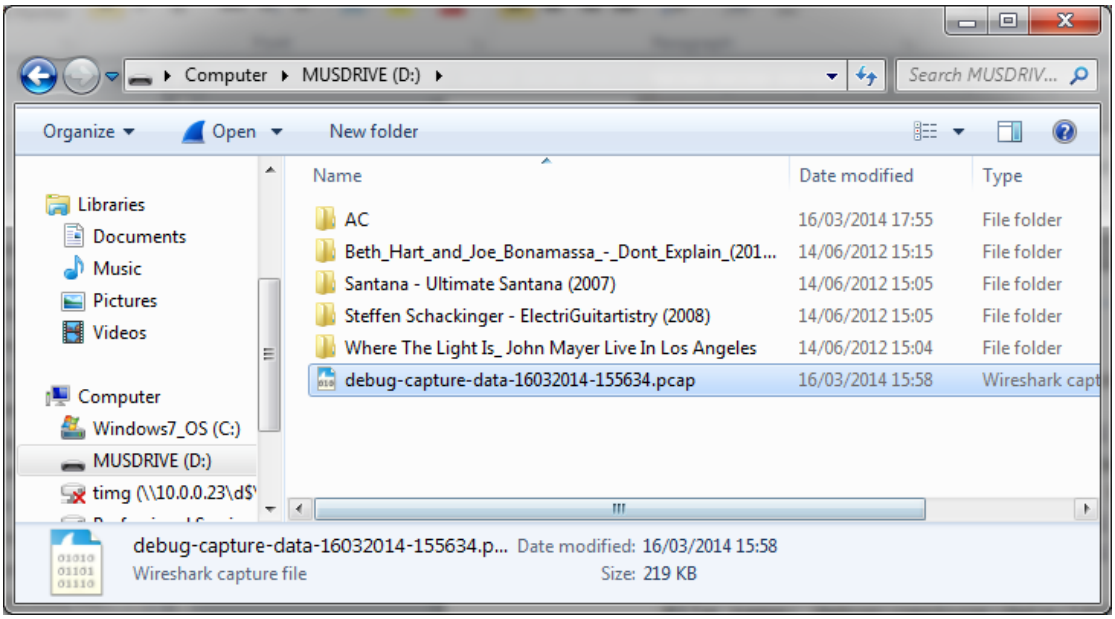
20.1 Example of Capturing Data on Physical Interface

This example captures data from the Ethernet interface on the LAN side and sends it to a USB device:

```
MSBR# debug capture data physical eth-lan
Interface eth-lan was added to the debug capture rules
      Use start command in order to start the debug capture
MSBR# debug capture data physical target usb
MSBR# debug capture data physical start
Saving capture to USB storage.
File name: debug-capture-data-16032014-155634.pcap
MSBR# debug capture data physical stop
Finished. Type "usb remove" to safely remove the drive.

MSBR# usb remove
You may now remove the USB drive
```

The captured file is written to the root directory of the USB drive.

Figure 20-1: Captured file on USB drive



20.2 Example of Capturing Data on an Interface

This example captures data from the Ethernet interface on the WAN side and sends it to a TFTP server:

```
MSBR# debug capture data interface gigabitethernet 0/0 proto all
host any port any tftp-server 192.168.0.50

.............................
MSBR#
```

This page is intentionally left blank.

21 PacketSmart

This chapter describes how to setup the BroadSoft's BroadCloud PacketSmart embedded agent that is bundled with AudioCodes Mediant 500, Mediant 500L and Mediant 800 Gateway and E-SBC products.

PacketSmart is a powerful toolkit used for network assessments. Comprised of Assessment, Verification, Diagnostics and Monitoring, PacketSmart is a lifecycle management solution that ensures VoIP services are deployed correctly, accepted by customers and monitored to meet customer satisfaction.

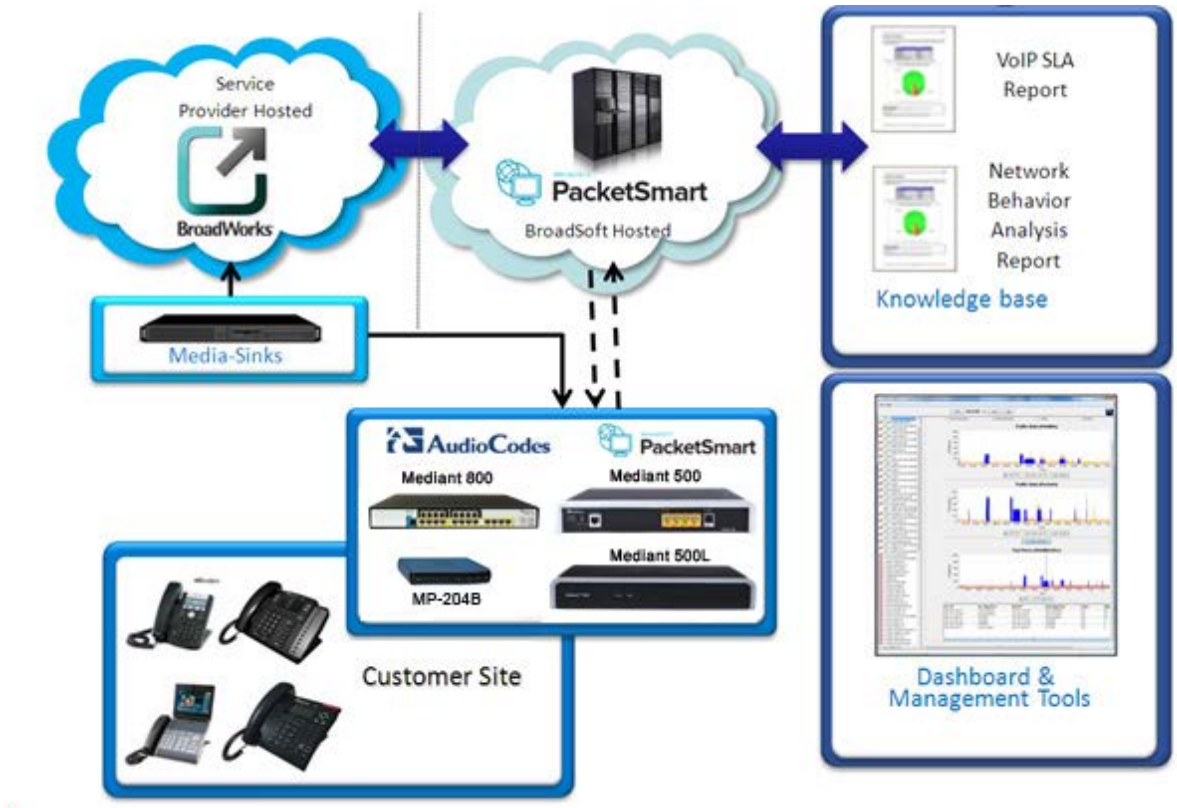
PacketSmart Monitoring observes customer networks and live calls to identify the source of local area network (LAN) and wide area network (WAN) issues that may impact VoIP quality.

PacketSmart uses proactive alerting with automated reporting that enables service providers to address issues prior to customer complaints arising into support groups, thereby reducing overall trouble tickets.

Notes:

- You must configure the Gateway or SBC before enabling PacketSmart. Refer to the *Mediant 800B Gateway and E-SBC User's Manual Ver.7.0.*
- PacketSmart functionality requires a Feature key.

Figure 21-1: PacketSmart Management Solution



The following figures show typical deployment models for the SBC and Gateway.

Figure 21-2: SBC in DMZ Model

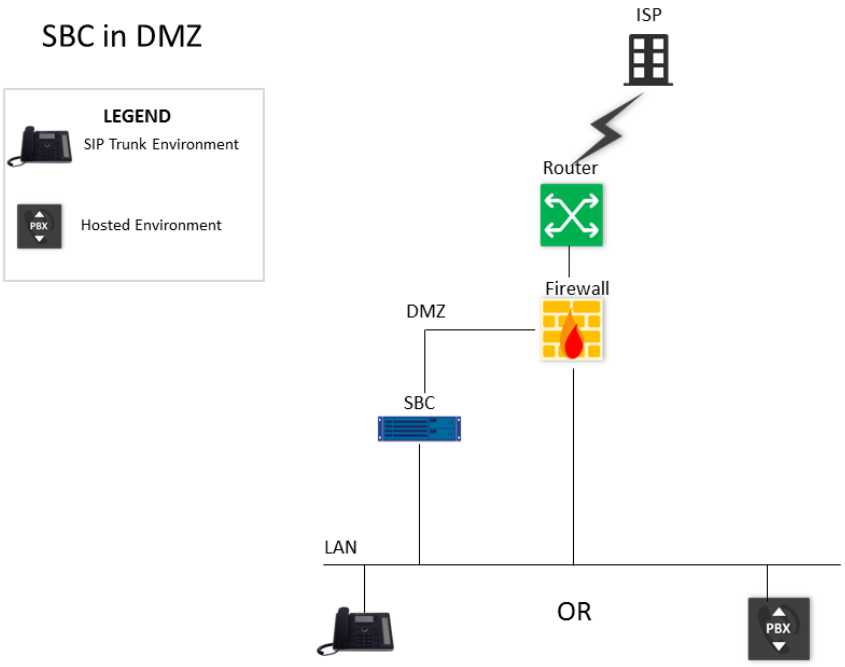


Figure 21-3: SBC on LAN Model

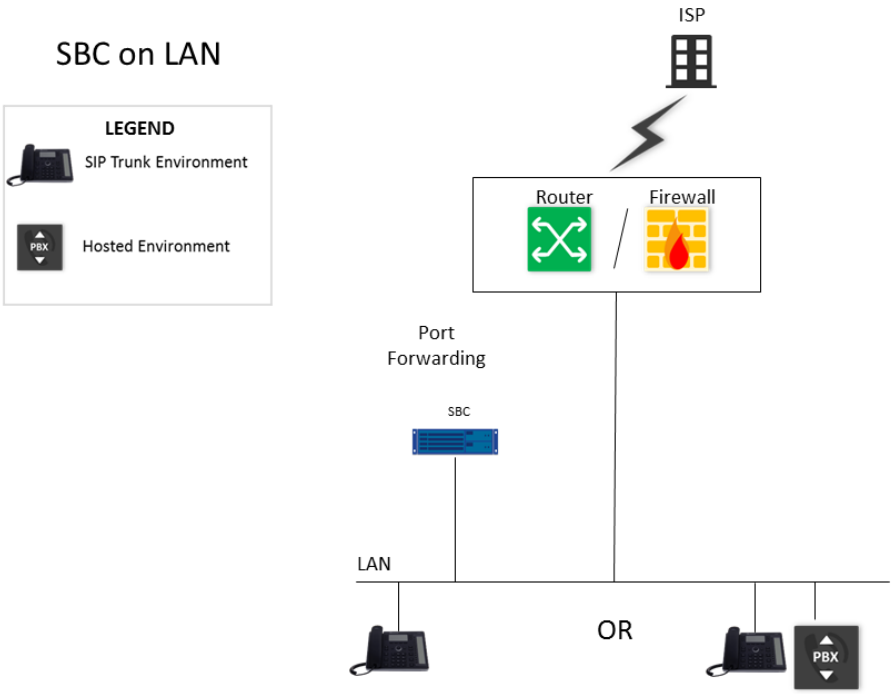
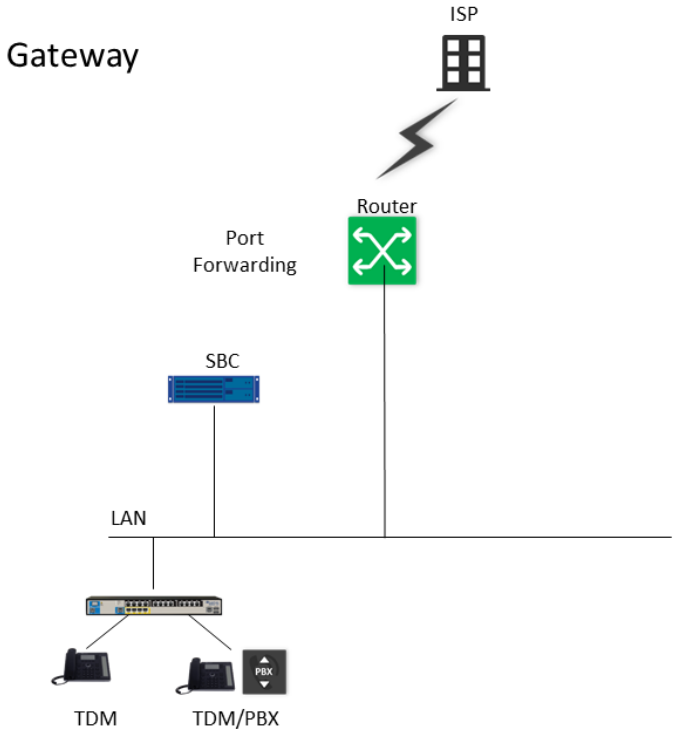


Figure 21-4: Gateway Model



21.1 Configuring the Device for PacketSmart

The device can be configured for PacketSmart through the device's Web interface or CLI.

Notes:

- The parameters become active only after RESET (Off Line Mode).
- The network interface is usually the WAN interface that is configured on the MSBR.

21.1.1 Configuring the PacketSmart Agent through CLI

The following procedure describes how to configure the PacketSmart agent through CLI.

➢ **To configure the PacketSmart agent through CLI:**

Command	Description
`configure system`	Go to system configuration context
`(config-system)# packetsmart enable`	Enables the embedded PacketSmart agent.
`(config-system)# packetsmart server address`	Defines the IP address of the PacketSmart server to which the PacketSmart agent connects. The default is 0.0.0.0.
`(config-system)# packetsmart server port`	Defines the TCP port of the PacketSmart server to which the PacketSmart agent connects. The default is 80.

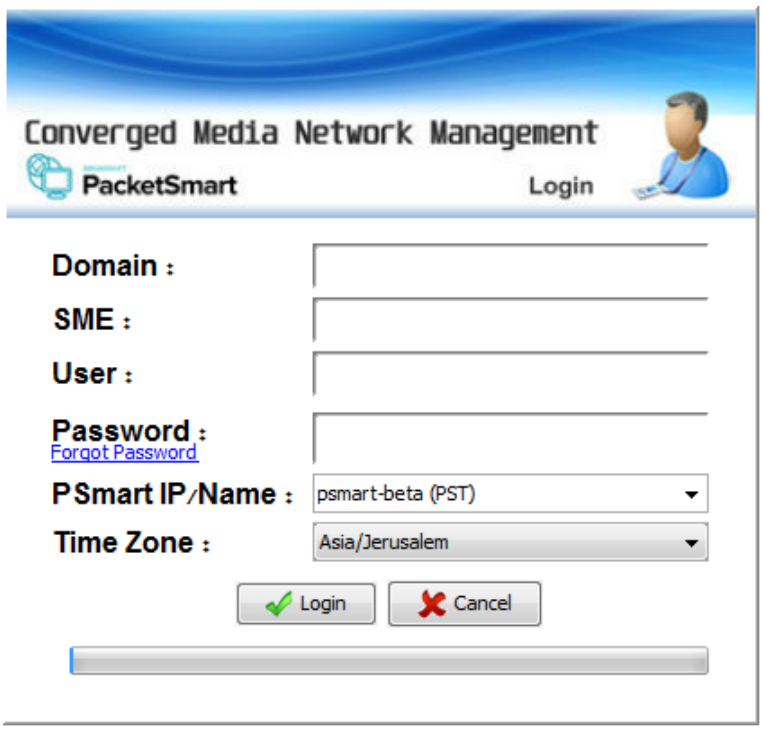
21.1.2 Viewing PacketSmart Statistics

The PacketSmart Web client is the user interface to the PacketSmart cloud-based service platform.

➢ **To view PacketSmart statistics:**

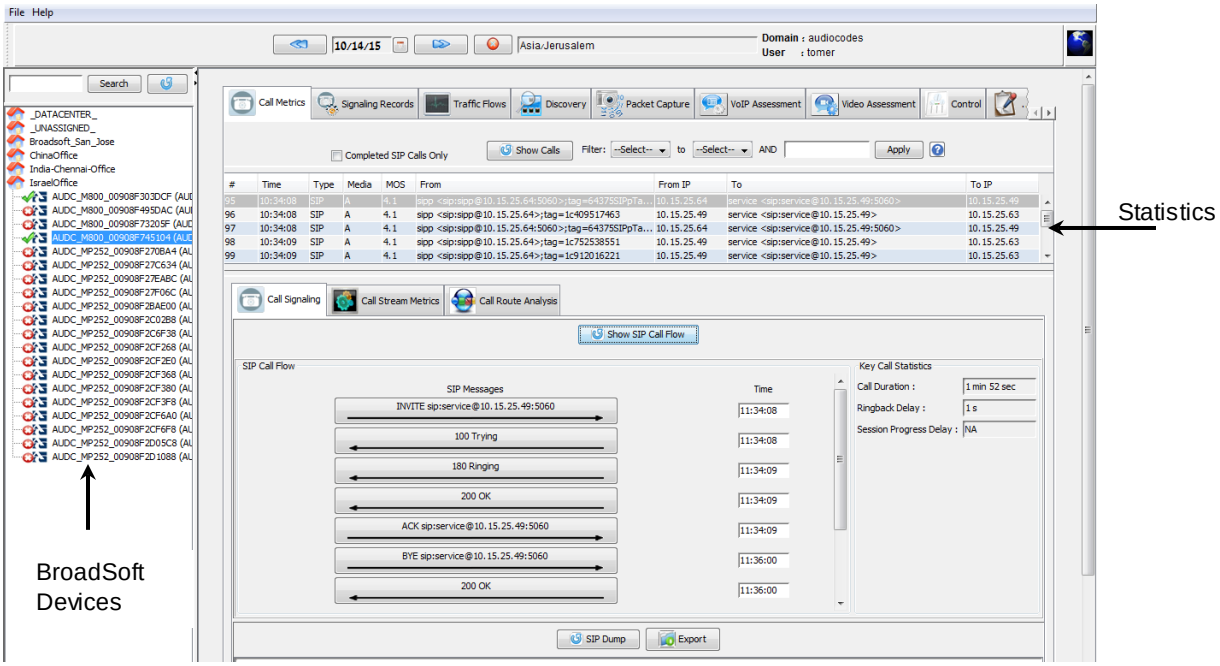
1. Download and launch the PacketSmart Web GUI client.
2. In the PacketSmart Login screen, enter the login credentials you received from BroadSoft.
3. Click **Login**.

Figure 21-5: Converged Media Network Management



4. PacketSmart statistics appear on the screen.

Figure 21-6: BroadSoft Server View



5. Confirm that the devices are connected to the BroadSoft server.

International Headquarters
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.
27 World's Fair Drive,
Somerset, NJ 08873
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: www.audiocodes.com/info
Website: www.audiocodes.com

Document #: LTRT-31614

AudioCodes