AudioCodes

WHITE PAPER

SIP Conferencing/ Collaboration

Leveraging the SIP Architecture to build IP conferencing and collaboration solutions

Version 1
December 2007

WHITE PAPER

Introduction

Global organizations utilize conference calls as a very important business tool for collaboration. Multi-branch organizations were the first to recognize the value in voice and video conferencing services to economize on travel costs and to coordinate business activities. Other smaller organizations have also begun to recognize that having access to easy-to-use conferencing resources speed up collaboration efforts with clients and suppliers. Whether using a traditional TDM PBX, an IP-PBX or a hosted service provider, SIP is seen as a key technology going forward to help tie organizations together and dramatically reduce the costs of conferencing.

Conferencing and collaboration applications can include a broad range of different functions and media types, including:

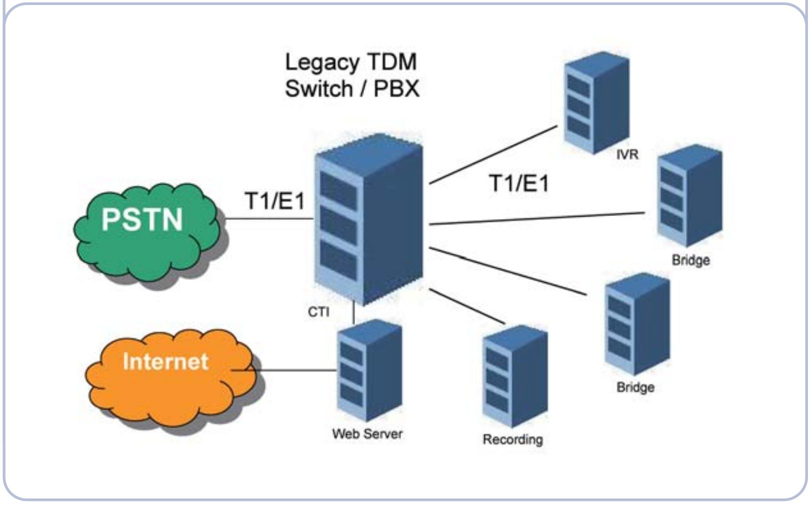
- Voice
- Video
- Slide push
- Desktop sharing
- Chat
- Q/A
- and surely many others...

As voice is virtually ubiquitous in all conferencing applications, this whitepaper will focus on the voice component, whether from PSTN callers or from web-based conference clients. We'll save the other media types for future whitepapers.

The Legacy TDM Conference Bridge

The legacy conference bridge was built around a TDM switch or PBX equipment, adding a number of conferencing application functional blocks as separate point solutions, generally each of these are integrated into the switch via T1/E1 links.

Figure 1: Legacy TDM Conferencing Architecture



The vast majority of in-bound conferencing applications start with callers reaching an **Interactive Voice Response** (IVR) session that collects the caller's conference ID and optional passcode digits. From there, routing logic within the application transfers the call to one of the conference bridge systems to participate in the live conference. You may not know it, but the "Please hold while your conference ID is verified" announcement is really a cover for the transfer process.

The **conference bridge** systems have responsibility for performing the actual mixing of the multi-party conference. This includes speaker detection, tone detection and clamping, automatic gain control, noise cancellation and other media manipulation to maintain a quality conference session.

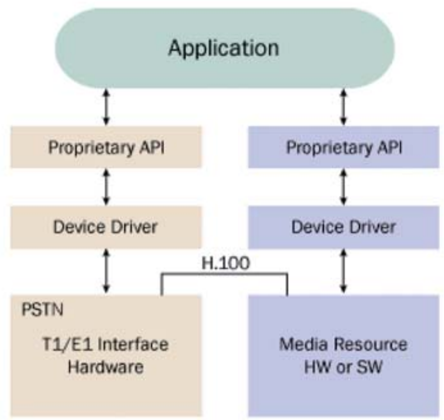
In many applications, a separate server performs optional **conference recording** operations. This is accomplished by allocating a port on the recording server and routing it via the switch to one of the active conference servers.

WHITE PAPER

Many conference services also include web servers, used to push slide content to conference participants (a.k.a. “conferees”) and their desktops. Surprisingly, most existing conference solutions have not integrated the functions of the voice conferencing and the web content. In some cases, they are totally independent requiring both the moderator and conferees to log into two separate systems. Only recently have the applications begun to merge, bringing information from one to another about who is connected and who is speaking.

Each of the IVR, conference bridge and recording functional blocks noted above have almost universally been built on the Legacy CTI Architecture. This architecture used proprietary APIs, device drivers and board technologies that were ground-breaking in their time, but now are widely recognized as obsolete and suffering from a number of serious limitations.

Figure 2: Legacy CTI Architecture



Proprietary APIs - As the hardware and software components in this architecture originated from a number of different vendors, the APIs that control them are very different from one another and very complex. To work in this environment, developers require very specialized knowledge.

Proprietary Device Drivers - The device drivers provided by the vendors are tightly tied to the operating system. The net of this is that the enabling technology vendors dictated which operating systems the developer could use.

Limited Packaging Options – Because the CTI architecture depends on the API to communicate with boards or software via device drivers, all the resources must reside in the same server. This limits features and drives up costs.

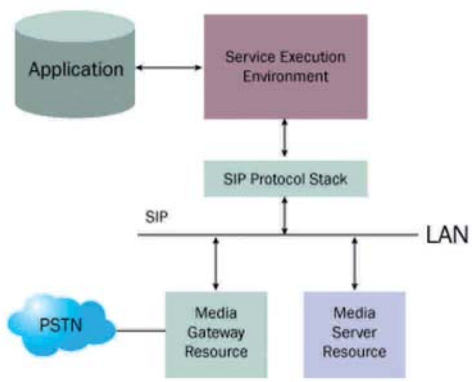
Scaling – Once an installation grows to the point where the resources required outgrow the hardware platform, the addition of more boards and/or servers is usually complex and expensive.

Pace of Change – Due to the complexity, modifications to applications based on APIs is extremely difficult and requires significant effort to completely test and debug development issues. This can delay new features and capabilities, leaving room for competitors to capture market share.

THE SIP Architecture

The SIP Architecture approach resolves these issues by using SIP to standardize the interfaces between functional blocks, whether they are DSP boards, separate servers or software modules. The architecture separates the application execution, the connectivity and the media processing requirements into a number of separate functional blocks as shown below:

Figure 3: SIP Architecture for PSTN



The SIP architecture has a number of advantages that are important to both the developers and customer solution designers:

Industry Standard Interfaces – by leveraging open protocol standards of SIP (RFC 3261), NetAnn (RFC 4240) and MSCML (RFC 4722), developers have access to a wide range of products, each offering different capabilities from various vendors – all sharing common interfaces.

Operating System and Platform Independence – by using SIP as a control protocol and not using proprietary device drivers, virtually any operating system and any hardware platform can be used to create applications.

Broad Packaging Options – applications based on the SIP architecture can leverage a wide range of physical packaging options. From PCI-based commercial servers, to AdvancedTCA and other blade server form factors, to pre-packaged appliances.

Scaling - the SIP architecture solves a number of scaling issues. Application servers can manage multiple media gateways and/or multiple media servers – allowing the integrator to scale up or down over a very broad range.

Agility and Time to Market – as the SIP architecture is more modular, application modifications can be implemented and deployed more efficiently, helping developers add features and remain competitive in a fast-paced marketplace.

All of these advantages add up to a very compelling argument for adopting the SIP architecture in conferencing / collaboration solutions.

Adopting SIP in Conferencing - The Driving Forces

A number of driving factors in both business and technology have made the SIP architecture viable for use in a range of conferencing and collaboration solutions:

Multimedia Web Penetration – virtually all portable and office-based computers now are equipped with multi-media features and have access to broadband internet connectivity. This allows for new ways to access conferencing / collaboration sessions via the internet and avoiding the costly and long conference telephone calls.

Maintenance costs – as the legacy conferencing systems age, the maintenance costs required to keep them operating grow relative to the services they offer. Once the systems are fully depreciated, a wise organization will look to migrate to newer equipment with lower maintenance and operational costs.

Mobile and Distributed Workforce – greater mobility and work-at-home employees have created strong demand for easier access and low-cost conferencing / collaboration environments. This allows employees and customers to meet in virtual locations and share ideas without fighting the long lines at the airport

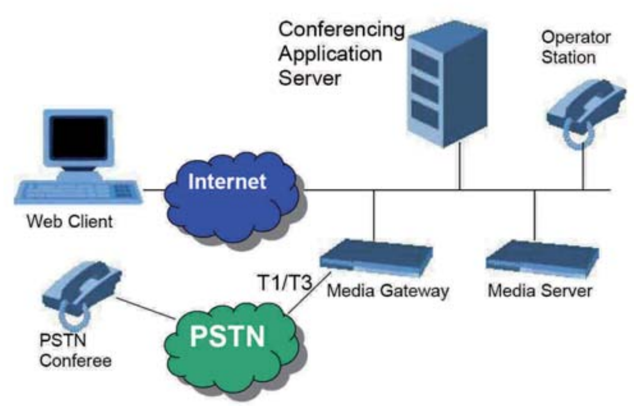
Protocol Evolution - SIP and the companion media control protocols **NetAnn** and **MSCML** have evolved over the last few years to finally encompass capabilities needed to implement complex conferencing applications. Prior to these standard protocols, developers had no choice but to use complex proprietary APIs to create the application elements. The new capabilities of these protocols☐ have made complete SIP solutions possible.

Best-of-Breed Standardized Components - The business of building telecommunications equipment has forever changed. Instead of building the application and all of the underlying components themselves, the equipment manufacturers have recognized the value of staying focused on their application feature set, leveraging standard off-the-shelf enabling technology building blocks. SIP has been a key technology for the equipment manufactures to accomplish the move to standardized components and leveraging best-of-breed enabling technologies from a range of suppliers.

The SIP Conference Application

When applied to conferencing, the SIP architecture takes the form as shown below, creating the SIP Conferencing Application:

Figure 4: The SIP Conferencing Applicationn



Application Overview

Following the SIP architecture as a guideline, the SIP Conferencing Application includes a number of elements:

- Conferencing Application Server
- Media Gateway
- Media Server
- Operator Station(s)
- PSTN Conferees and Web client(s)

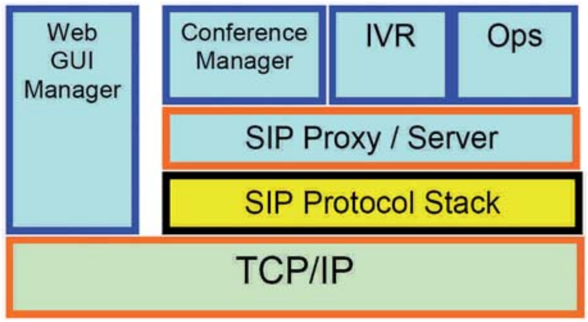
WHITE PAPER

Conferencing Application Server Element

The Application Server provides the core logic and management of the conferencing solution. Instead of completely separate systems, the functions of the Application Server are usually divided into a number of separate software modules, all working together. There are an infinite number of ways to build the conferencing application server software, but the following core functions seem common amongst many implementations:

- Conference Manager – an application-level module that manages the active conference sessions and mid-call functions like waiting room features, muting, recording, summoning the operator, entrance and exiting the conferences
- IVR – also an application-level module responsible for the pre-conference and post-conference IVR interactions used to collect conference IDs, passwords and account management.
- Ops (Operator services) - an application-level module responsible for managing the interacting of operators and the different conferences. Functions like conference GUI management, Q/A queuing, pre-recorded announcements, active talker identification and muting
- SIP Proxy Server – responsible for call switching, transfers and routing. Based on the implementation, the SIP Server function may be either completely integrated into the application or an adjunct software module from other commercial or open-source vendors
- SIP Protocol Stack – provides connectivity between the Application Server and other elements within the application
- TCP/IP Stack – generally provided by the operating system, allowing real-time communications.

Figure 5: Software Modules within Application Server



Based on the size of the conferencing application, one or more physical servers may be used to build the solution. The servers may execute just one of the software modules or combine dtogether in a clustered server model - spreading the workload over two or more servers.

Media Gateway Element

Positioned between the Public Switched Telephone Network (PSTN) trunk lines and the conferencing application, the media gateways provide connectivity to the outside world.

Trunking media gateways have a number of duties, including:

- PSTN Signaling
- Detecting ring voltage
- Converting and compressing the voice into packets
- Echo cancellation
- Tone detection and generation
- Call Progress and Answering Machine detection
- IP control protocols
- Network performance measurement
- And more...

Media Server Element

The media server plays a key role in handling all of the media processing required for the conferencing application, including:

- Playing or streaming of announcements and prompts for conferee authentication
- Music on Hold streaming for “waiting room” features
- Conferencing and mixing
- Active Speaker detection
- DTMF detection and generation
- Call progress tone detection
- Answering Machine Detection
- Training
- Conference Recording
- Transcoding

PSTN Conferees and Web Clients

These are the participants in the conferencing session, either calling in via the PSTN or connecting via a web-based software client that includes an integrated SIP User Agent.

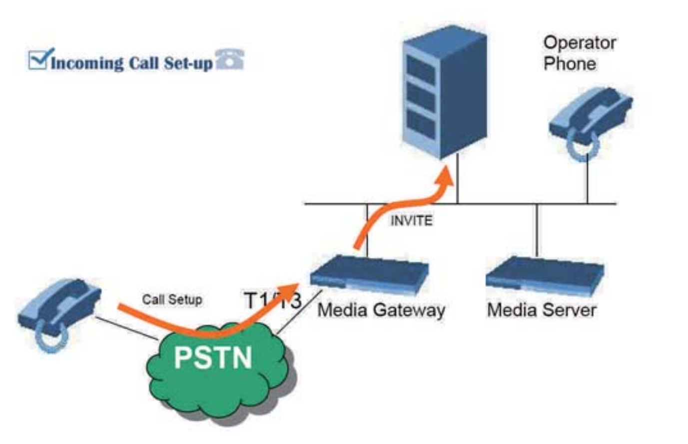
Sample Inbound Call Flow

“A picture is worth a thousand words”

As per the saying, the interaction between the different elements in typical applications in a pictorial form should be very useful in building an understanding of how SIP impacts the conferencing application. In the series of high-level call flow diagrams below, a real-world series of typical conferencing activities is played out as follows:

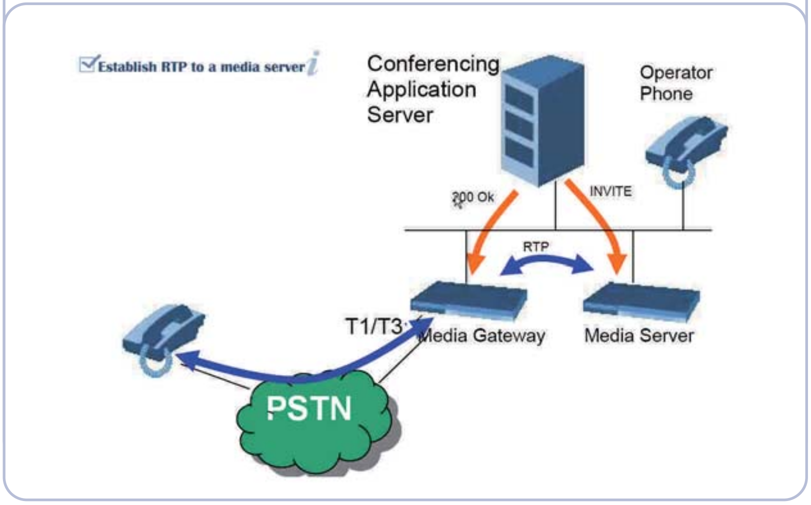
> *The moderator for an ad-hoc conference calls into the conferencing center to activate a live conference for a number of participants. Upon calling the conference center, the moderator is initially prompted for a conference ID and then prompted for a pass-code. After making a valid entry, the conference bridge is established and a number of other callers join in the conference.*

Figure 6: Call Flow- Step #1



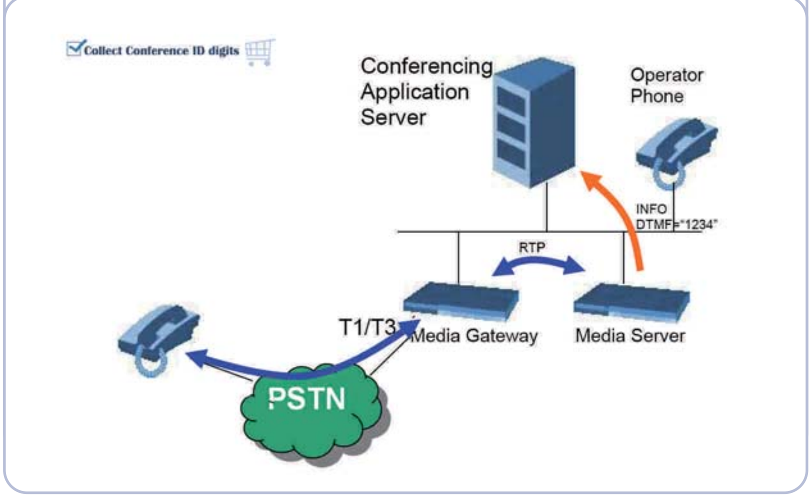
The moderator calls into the conference center – the media gateway converts the ISDN Call Setup message from the PSTN to a SIP INVITE and forwards the message to the SIP Server within the application.

Figure 7: Call Flow- Step #2



Based on available resources and other factors, the application selects one of the media servers to perform the initial IVR session to authenticate the caller. To do this, the application directs the call to one of the media server resources within the installation using an INVITE to the media server and 200 Ok to the media gateway. SDP negotiation is carried out using INVITEs, ACKs, and OKs. At the end of this negotiation, RTP is connected in both directions between the gateway and the media server resource.

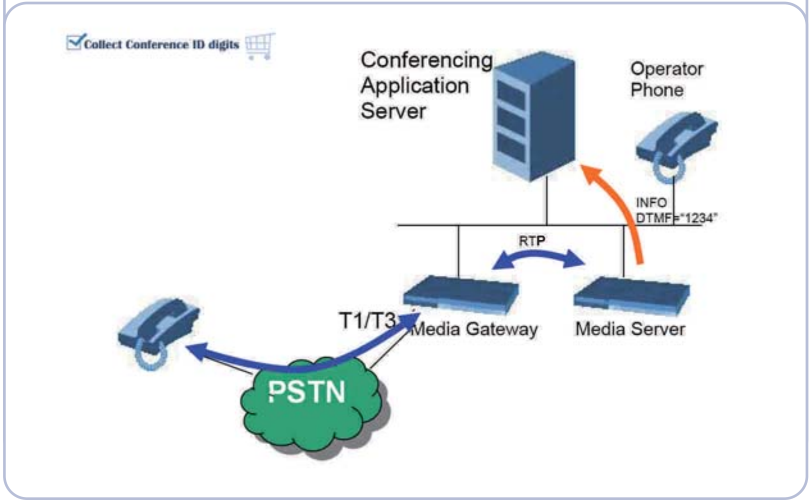
Figure 8: Call Flow- Step #3



The application then uses the media server and an MSCML Play/Collect command to play an announcement to the caller, asking for their account number.

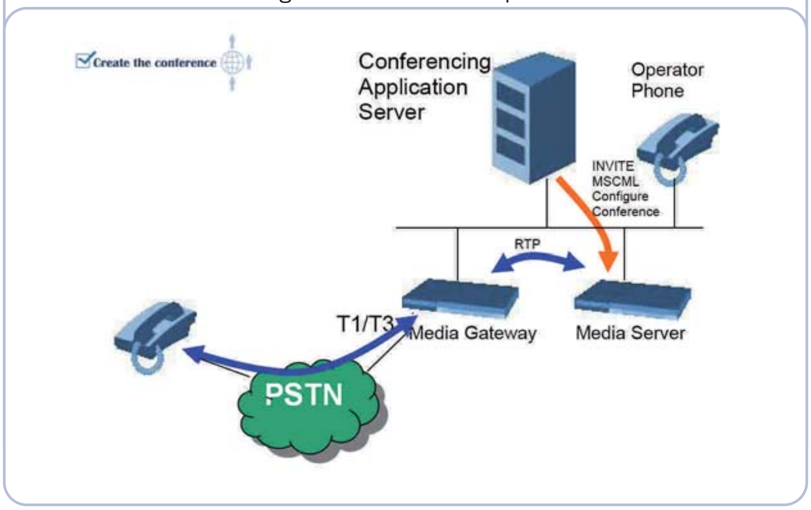
Whitepaper

Figure 9: Call Flow- Step #4



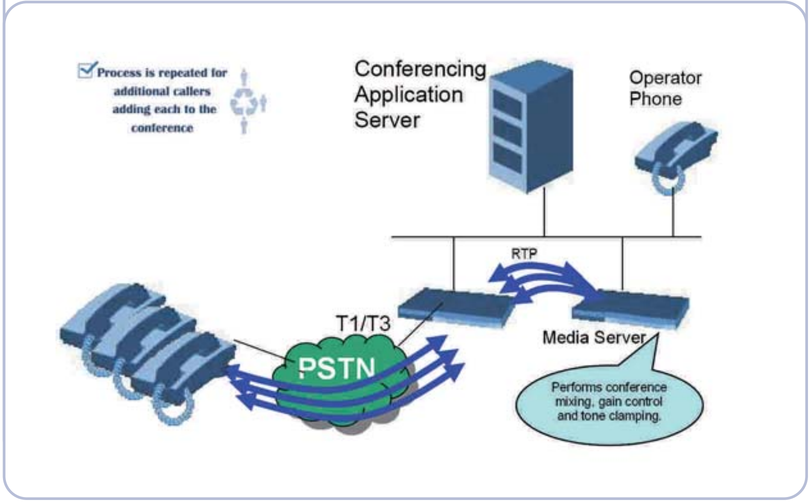
In this example, the moderator enters “1234” in response to the announcement, specifying his/her own conference ID value. The process is repeated for the security pass code. At this point the application now has both the conference ID and pass code for the moderator and with the valid combination, the decision is made to create a conference.

Figure 10: Call Flow- Step #5



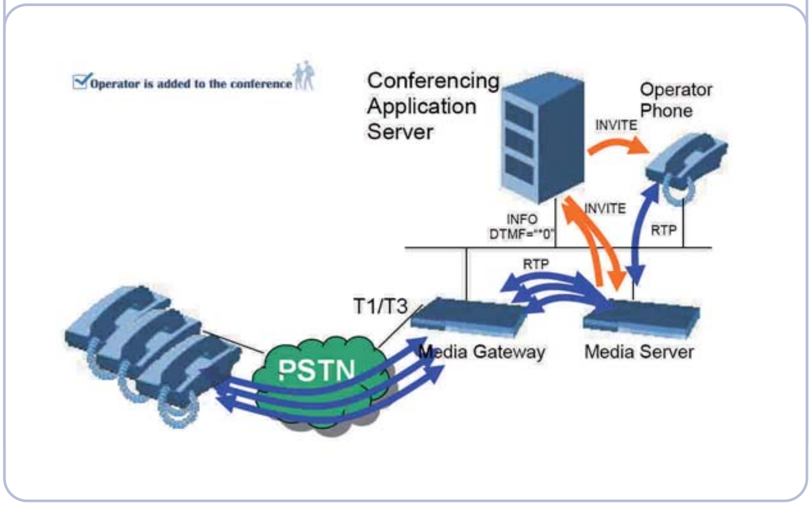
To create a conference, the application establishes another SIP dialog with either the same or a different media server, using a SIP INVITE with a MSCML Configure Conference command in the payload section. The connection with the moderator is then redirected to the newly created conference by the application server by sending a reINVITE (not shown) to the gateway. At the end of this process, the moderator is connected to the conference session within the media server.

Figure 11: Call Flow- Step #6



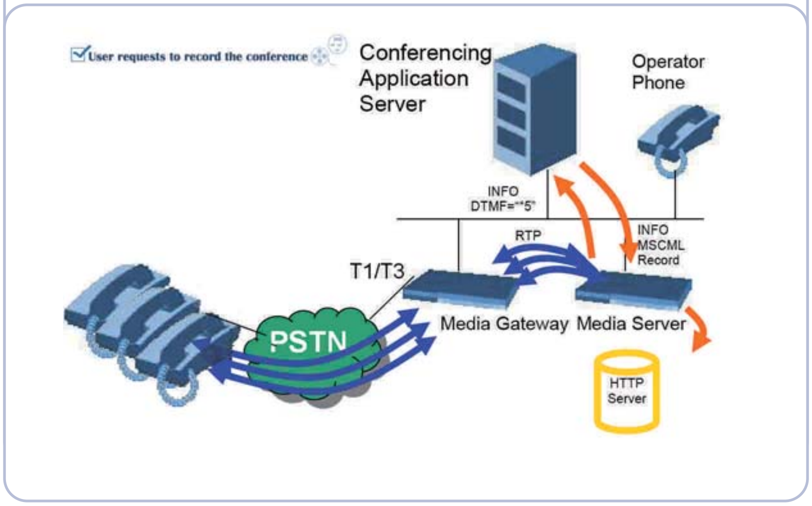
The process is repeated for each additional conferee, prompting for the conference ID and passcode, eventually directing all the RTP streams to the allocated conference resource instance. At this point, the conferees can have their discussion without further interaction by the conference application server.

Figure 12: Call Flow- Step #7



If however, the moderator needs to pull an operator into the conference, a DTMF command is a common way to indicate the request. In Step #7 above, a "*0" is issued by the moderator, which is relayed to the application.The application would then interpret the command and allocate an operator and use a set of INVITEs to add the operator to the conference bridge. (Most likely, the application would also issue a GUI screen pop to the operator's console, providing information about the status of the active conference session).

Figure 13: Call Flow- Step #8



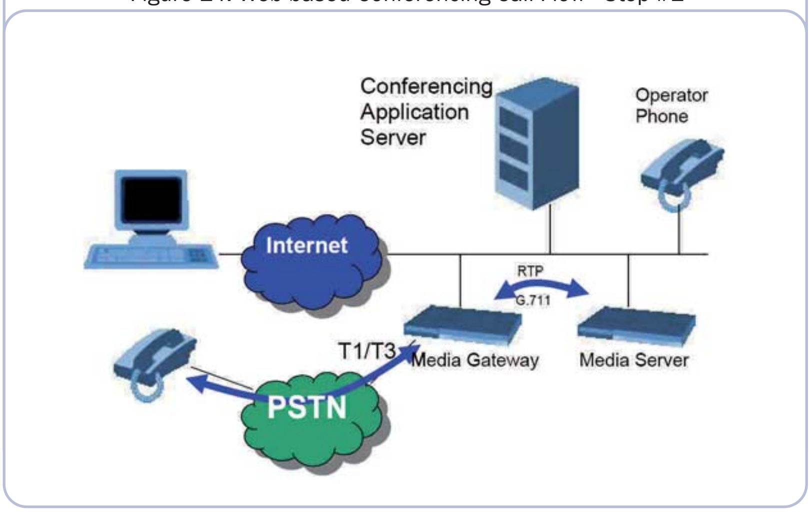
Another common feature of conferencing is recording conference calls for later review by those that missed the live event. As shown the example in Step #8 above, the moderator could press "*5" which would be sent to the application via a SIP INFO. The application would interpret the command and use a SIP message to pass an MSCML command to the media server. The media server would then use HTTP streaming to record the conference to a network storage device. MSCML allows for any one of the conference legs, or the entire conference to be recorded.

Web-based Conferencing

Enabling both web-based and PSTN conferees in the same conference bridge is an increasingly common requirement for service providers and enterprise users. Many applications to date have used clumsy hybrid solutions that depend on traditional TDM conference bridges and isolate web-only conferences to separate facilities. All of these attempts have proven to be far from effective at bridging the different media types required by both types of participants. To truly accommodate both web-based and PSTN conferees, they must both be integrated into the same conference bridge, and offered identical services.

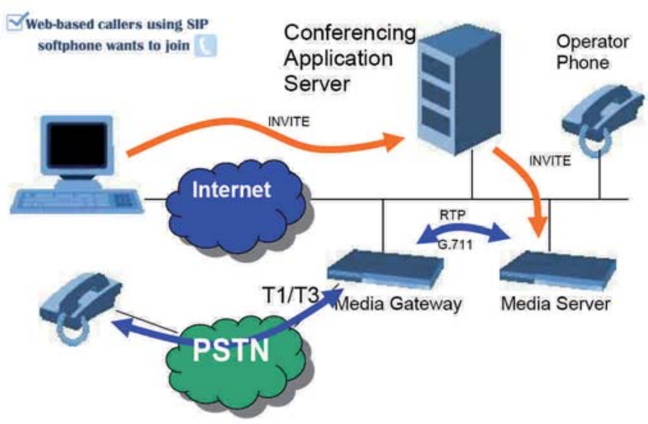
Continuing with the example of call-flow diagrams, from where we left off above, a series of diagrams that follow show the interaction between a web-based client and the conferencing application.

Figure 14: Web-based Conferencing Call Flow - Step #1



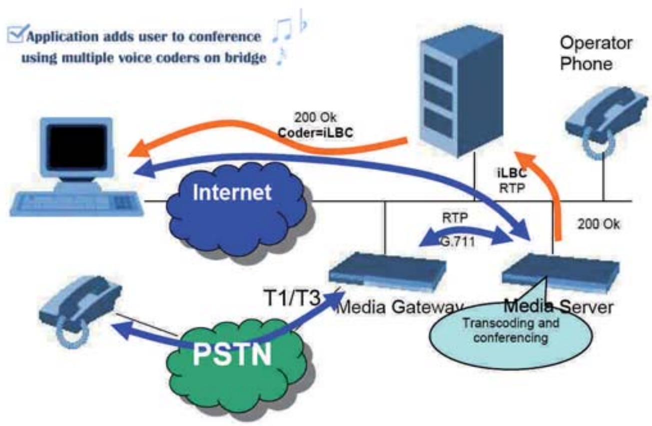
The process begins where we left off with one or more PSTN conferees connected to the media server and participating in an active conference session. Note that in this case, the PSTN caller(s) are using G.711 as the voice coder between the media gateway and the media server – since bandwidth within the datacenter is rarely an issue, G.711 is a good choice.

Figure 15: Web-based Conferencing Call Flow- Step #2



For a web-based client to join the conference, our application will assume the client uses an embedded SIP User Agent to provide the multi-media connectivity to the conferencing center. In our example, the web client initiates a call over the internet and that call is delivered to the Conferencing Application Server. Conference ID information can be easily carried in the SIP INVITE payload, indicating to the conferencing application which conference the web-based conferee would like to join. Based on that information, the application would then initiate an INVITE to the appropriate media server and conference instance, requesting the addition of the web client.

Figure 16: Web-based Conferencing Call Flow- Step #3



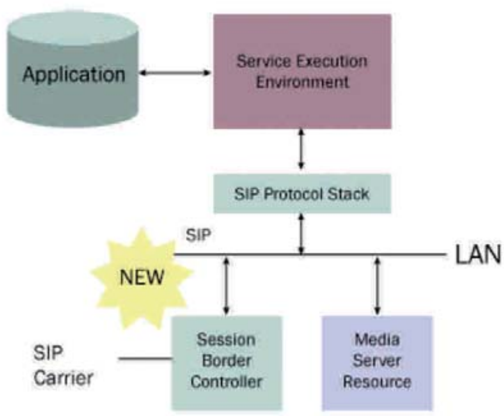
However, since the web clients are generally using bandwidth-limited broadband connections, it is common to use a low-bit-rate coder between the client and the conference bridge. Using a low-bit-rate coder dramatically reduces the traffic on the broadband connection, reducing congestion and dropped packets, frequently improving voice quality. In our example above, the web-client chooses iLBC as the coder of choice, and establishing an RTP session to the media server with that coder selection.

This requires that the media server must transcode the iLBC coded RTP before being mixed into the conference. Fortunately, the AudioCodes media server shown can perform both the required transcoding and conferencing in one operation, keeping latency down and improving voice quality.

Migrating to SIP Trunking

A question you may be asking yourself now is “What about when the
SIP architecture addresses this change and provides a smooth transi

Figure 17: SIP Architecture for SIP Trunking



In deployments where a direct interface to a SIP carrier is available and there is no PSTN, the media gateway function is replaced with a Session Border Controller (SBC). The SBC provides important security and interoperability functions, including:

- Authentication
- Public/Private Network Address Normalization
- NAT Traversal
- Topology Hiding
- Session Admission Control
- Session-based Firewalling
- Encryption/Decryption
- Service Level Agreement (SLA) assurance
- Rouge RTP Detection and Deep Packet Inspection
- Protocol conversion (H.323 to SIP, or SIP to SIP)
- Protocol variant interoperability

Security

A common area of concern is confidentiality and security within conferencing applications, especially within the Government, Financial Services and Health Care markets. Beyond the federally mandated regulations, a wide range of companies are only now beginning to realize their potential vulnerabilities and as a result, CIOs are requiring that their voice and data infrastructure be secure from end-to-end. This is of particular concern as conferences are commonly used for confidential communications between companies and other negotiating parties.

The SIP architecture addresses these needs with two key features:

SIP over TLS (SIPS) – using the well understood security features of Transport Layer Security (RFC 4346) are leveraged to secure the SIP messages, keeping the call control information private.

SRTP – in addition to securing the call control, encryption of the voice path is accomplished via Secure Real-time Transport Protocol (SRTP). SRTP (RFC 3711) provides confidentiality, message authentication, and replay protection to the RTP traffic and to the associated control traffic.

Embedding these protocols within the media gateways, media servers and session border controllers enables security from end-to-end and greatly improves application security.

AudioCodes Solutions

When designing or deploying SIP-based Conferencing/Collaboration solutions, AudioCodes offers a number of products that can help with your efforts.

Media Gateways

MediaPack™ Gateways– from as few as two FXO or FXS analog ports up to 24 FXS interfaces or BRI ISRN interfaces, AudioCodes MediaPack media gateway products provides complete turnkey analog gateway functionality. All MediaPack media gateway products include an onboard SIP protocol, making them ideal for us in the SIP Architecture.

Mediant Gateway Systems – from as few as one T1 span (24 ports) up to three DS3 interfaces, AudioCodes Mediant media gateway products provides complete turnkey digital gateway functionality. All Mediant media gateway products include an onboard SIP protocol, also making them ideal for us in the SIP Architecture.

TrunkPack – a family of blades that deliver media gateway functionality in a PCI or cPCI form-factor without device drivers or complex APIs. Starting at one T1 span up to three DS3 circuits, TrunkPack media gateways include on-board SIP protocols for use in the SIP Architecture.

Media Server Resources

AudioCodes IPmedia™ Server Platforms – deliver state of the art media processing and protocol technology in a single 1U form factor that is ready to rack and stack. Supporting both SIP and MSCML, the IPmedia line enables complex media processing needed for sophisticated applications.

IPmedia - a family of blades that deliver media resource functionality in a PCI or cPCI form-factor without device drivers or complex APIs. Starting at 30 sessions up to 2,016 sessions per board, the IPmedia resources include on-board SIP, NetAnn and MSCML protocols for use in the SIP Architecture.

Session Border Controllers:

nCite™ Session Border Controllers - provide secure traversal of firewall and network address translation (FW/NAT) systems, as well as denial of service (DOS) attack prevention through deep packet inspection at both the SIP signaling and VoIP media layers. nCite SBCs also provide end-to-end QoS, protocol Interworking, and support of high call processing volumes and over-subscription ratios required for a scalable residential VoIP service offering. AudioCodes nCite solutions offer proven interoperability with all major Softswitches, SIP servers, H.323 gatekeepers, call agents, application servers, media servers, media gateways, IP PBXs and numerous IP-based voice and video endpoints.

Summary

After reading this whitepaper, hopefully you can recognize how the power and flexibility of SIP and MSCML can be leveraged along with AudioCodes SIP products to build advanced conferencing / collaboration solutions.

SIP Conferencing / Collaboration Application Advantages:

- Greater Flexibility
- Greater Security
- Easier to Integrate
- Application components all sharing hardware platform
- Common COTS Application Servers
- Integrates both PSTN and Web-based callers on the same bridge
- Multi-media integration
- Independent of expensive PBX equipment
- No Proprietary API Development
- More Cost Effective
- Faster time to market

About AudioCodes

AudioCodes Ltd. (NASDAQ: AUDC) provides innovative, reliable and cost-effective Voice over IP (VoIP) technology, Voice Network Products, and Value Added Applications to Service Providers, Enterprises, OEMs, Network Equipment Providers and System Integrators worldwide. AudioCodes provides a diverse range of flexible, comprehensive media gateway, and media processing enabling technologies based on VoIPerfect™ – AudioCodes' underlying, best-of-breed, core media architecture. The company is a market leader in VoIP equipment, focused on VoIP Media Gateway, Media Server, Session Border Controllers (SBC), Security Gateways and Value Added Application network products. AudioCodes has deployed tens of millions of media gateway and media server channels globally over the past ten years and is a key player in the emerging best-of-breed, IMS based, VoIP market. The Company is a VoIP technology leader focused on quality and interoperability, with a proven track record in product and network interoperability with industry leaders in the Service Provider and Enterprise space. AudioCodes Voice Network Products feature media gateway and media server platforms for packet-based applications in the converged, wireline, wireless, broadband access, cable, enhanced voice services, video, and Enterprise IP Telephony markets. AudioCodes' headquarters are located in Israel with R&D in the U.S. Other AudioCodes' offices are located in Europe, India, the Far East, and Latin America.

International Headquarters

1 Hayarden Street, Airport City
Lod, Israel 70151
Tel: +972-3-976-4000
Fax: +972-3-976-4040

US Headquarters

2099 Gateway Place, Suite 500
San Jose, CA 95110
Tel: +1-408-441-1175
Fax: +1-408-451-9520

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

Ref. # LTRM-80025 V.1 12/07

Notice

The information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and changes in the business environment, AudioCodes does not assume any obligation to update or correct any information and does not guarantee or assume liability for the accuracy of the printed material nor can it accept responsibility for errors or omissions. Original Date Published: December 2007