

# A Conference Gateway Supporting Interoperability Between SIP and H.323

Jiann-Min Ho  
Information Networking Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
1-412-4227195  
jiannmin+@cs.cmu.edu

Jia-Cheng Hu  
Information Networking Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
1-412-4227195  
jiacheng+@cs.cmu.edu

Peter Steenkiste  
School of Computer Science  
Dept. of Elec. and Comp. Eng.  
Carnegie Mellon University  
Pittsburgh, PA 15213  
1-412-2683261  
prs+@cs.cmu.edu

## ABSTRACT

Increased network bandwidth is making desktop video conferencing an attractive application for an increasing number of computer users. Unfortunately, two competing standards for video conferencing signaling are in use, H.323 and SIP. In this paper we look at the interoperability between these two standards by developing a conferencing gateway that supports conferences involving both SIP and H.323 clients. By appropriately translating between H.323 and SIP operations, our prototype gateway supports basic multi-party video conferencing between NetMeeting (an H.323 client) and VIC (a SIP client) without modifications to the clients. However, our experiments also show that seamless interoperation would require changes to the client implementations and the standards.

## Categories and Subject Descriptors

D.3.3 [Information systems applications]: Communications Applications – *Computer conferencing, teleconferencing, and videoconferencing.*

## General Terms

Experimentation.

## Keywords

H.323, SIP, video conferencing signaling protocols, video conferencing gateway, interoperability.

## 1. INTRODUCTION

Improvements in the wide-area, local-area, and home connectivity network technologies are making the use of desktop video conferencing a practical application for a rapidly increasing number of computer users. However, while the transport of video and audio data is controlled by a single widely-used standard (RTP[10]), two standards are competing for dominance of video conference signaling function: the H.323 protocol suite by ITU-T [13], and the Session Initiation Protocol (SIP) by IETF [3]. Both of these signaling protocols provide mechanisms for call

establishment, call control, capability exchange, and supplementary services.

While the two signaling protocols have similar functions, they perform these functions in radically different ways. H.323 follows a client-server architecture. Endpoints (clients) interact both for data transport and control with one or a small number of servers that coordinate and control the video conferencing session. SIP, in contrast, has a highly decentralized architecture. Data transport is typically performed using IP multicast without central control, i.e. many of the functions supported in a centralized fashion by H.323 servers are performed by SIP endpoints.

The existence of two incompatible standards is a real problem for users, since they have to choose between two solutions that have both advantages and disadvantages. Most commercial products use H.323, but they are only supported on a limited number of platforms, do not use IP multicast, and require the use of expensive servers for multi-point conferencing. SIP is used by the Mbone tools [17], which are freely available on a number of platforms and use IP multicast for multi-point conferencing, but they are not as well supported or as user-friendly as commercial H.323-based systems. They also lack the tight session control that is often needed in a business environment.

This situation motivated us to explore the interoperability between the two standards. Our goal was two-fold: 1. build a system that allows H.323 and SIP clients to participate in a single video conferencing session, and 2. come up with a set of recommendations for developers and standards bodies that would allow them to improve interoperability in future systems.

We designed a Generic Conference Control Gateway (GCCG) that can participate in both H.323 and SIP sessions. It supports mixed video conferencing by translating signaling messages and media streams between H.323 and SIP endpoints. We built a prototype GCCG and demonstrated it using NetMeeting 3.0, an H.323 compliant terminal from Microsoft, and VIC 2.9, a SIP compliant Mbone video conference tool from University College London, as endpoints. While our GCCG allows interoperation without any modifications to the clients, we identified several areas where changes in the standards, their implementation, or user interfaces would simplify combining users using the two different standards in a single session.

The remainder of this paper is organized as follows. We first briefly describe the main differences between H.323 and SIP. In Section 3 we present our design of a video gateway that can

RESERVED AREA FOR COPYRIGHT NOTICE

support multi-party video conferencing involving both H.323 and SIP clients. In Section 4, we describe an implementation of the video gateway and its evaluation using NetMeeting and Mbone clients. We discuss the lessons we learned from building the video gateway design in Section 5. Finally, we present related work in Section 6 and conclude in Section 7.

## 2. Video conferencing signaling protocols

The challenges of integrating the H.323 and SIP call control signaling protocols are rooted in their different design philosophies with respect to protocol syntax and semantics. This section briefly describes properties and significant components of the two protocols and then identifies areas where interoperability might be a problem.

### 2.1 The H.323 Protocol Suite

ITU-T recommendation H.323 defines system aspect requirements for multimedia communication systems over a packet switching network. Its scope includes registration, admission and status (RAS) control, and call setup signaling as defined in H.225.0 [14]; call control defined as defined in H.245 [15]; audio/video codecs (e.g. G.711 for audio and H.261 for video); and real-time media transport protocols (RTP and RTCP).

An H.323 conference system for packet switched networks can include one or more of the following functional components: terminals, gatekeeper (GK), multipoint controller (MC), multipoint processor (MP) and multipoint control unit (MCU). The H.323 control messages and procedures define how these components communicate. The H.323 GK provides services such as address translation, RAS control, call redirection and resource management to H.323 clients. The H.323 MC and each H.323 participant in the conference establish an H.245 control connection to negotiate media communication types. The MP provides media switching and mixing functionality, e.g. the MP decides which of the media streams generated by the clients will be forwarded or mixed as a single stream. H.323 supports two communication modes for multi-party conferences, namely, centralized and decentralized. The centralized mode requires that an MP, operating with the MC, establish (unicast for transfers from participants to the MP, but multicast for transfers from the MP to the participants) media channels with each H.323 participant in the conference. These channels will be used to distribute the media streams selected by the MP. In the decentralized mode, each H.323 participant must have MP functionality and must be able to process multicast media streams. The MP of each client will then decide what streams to replay. The H.323 MC component is responsible for selecting unicast or multicast media transmission and for choosing network/transport addresses.

The H.323 call signaling procedure begins when an originating H.323 endpoint issues an admission request (ARQ) to the gatekeeper in its zone. After the endpoint receives a confirmation message (ACF) from the gatekeeper, the call setup procedure continues with a SETUP and CONNECTION message exchange. Finally, both endpoints follow the H.245 capability exchange procedure: they exchange terminalCapabilitySet messages and open media channels. Clients can reduce signaling overhead by using the Fast Connection procedure, which allows them to start media communication after one round-trip message exchange instead of three. The Fast Connection procedure is initiated by including a fastStart element in the SETUP message. The

FastStart element carries the proposed media channel description, OpenLogicalChannel, which identifies the media capability of the originating endpoint.

### 2.2 The Session Initiation Protocol (SIP)

There are two major architectural elements to SIP: the user agent (UA), and the network server. The UA resides at the SIP end stations, and contains two components: a user agent client (UAC), which is responsible for issuing SIP requests, and a user agent server (UAS), which responds to such requests. There are three different network server types: a redirect server, a proxy server, and a registrar. As a first approximation, the SIP User Agent is equivalent to an H.323 terminal, and the SIP network servers are similar to an H.323 gatekeeper.

While servers are needed to use some of the more powerful SIP features such as transcoding, it is possible to set up simple multi-party conferencing sessions without using servers. SIP has been specifically designed so it can make use of IP multicast both for control and for data transport. IP multicast allows SIP clients to set up a conferencing session by exchanging SIP messages directly.

A generic SIP operation involves a SIP UAC issuing an invitation, a SIP proxy server acting as end-user location discovery agent, and a SIP UAS accepting the call. A successful SIP invitation consists of two messages: INVITE followed by an ACK. The INVITE message contains a session description that informs the called party what type of media the caller can accept and where it wishes the media data to be sent. SIP addresses are referred to as SIP Uniform Resource Locators (SIP-URLs), which are of the form sip:user@host.domain. The SIP message format is based on the Hyper Text Transport Protocol (HTTP) message format, which uses a human-readable, text-based encoding.

The Session Announcement Protocol (SAP) and the Session Description Protocol (SDP) support the establishment of multi-party conferencing sessions. SAP defines the procedures for advertising conferencing sessions by periodically multicasting information about active sessions. SDP supports the description of multimedia sessions, including the specification of preferred media types and scheduling information. The SAP and SDP combine to provide a means of advertising sessions so interested parties can join. In this context, a multimedia session is defined as a set of media streams that exist for a time duration.

### 2.3 Comparison between H.323 and SIP

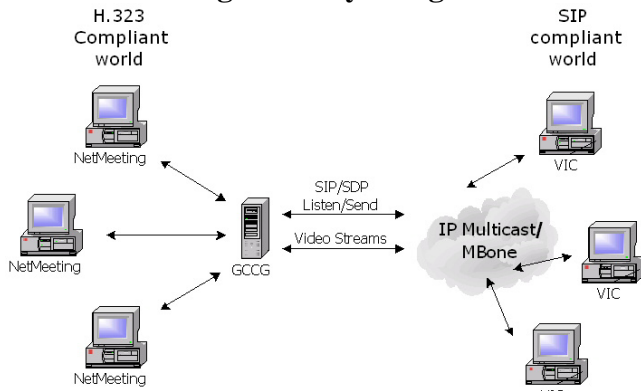
Video conferencing in H.323 is based on centralized server that uses a set of tightly integrated protocols to control sessions. In contrast, SIP is often used without a server, and its control mechanisms are much more loosely coupled: SIP clients can join and leave a conference via UDP signaling without centralized control. The differences in the original design goals and target network environments of H.323 and SIP resulted in a different functional breakdown, incompatible capabilities for conference advertisement and common media mode determination, and different message presentations.

Table 1 summarizes the main differences between H.323 and SIP. In the next section, we describe how these differences affect interoperability, and we present a conferencing gateway architecture that supports interoperation for simple multi-party sessions.

**Table 1. Interoperability Issues**

	<i>H.323</i>	<i>SIP</i>	<i>Interoperability Issues</i>
<i>Modularity</i>	All-in-One module: combined call signaling and media channels	Separate modules for different functions: independent call signaling and media channels	Synchronization of the mapping procedures during operation
<i>Message Presentation</i>	Binary code: ASN.1-PER	Text: ASCII code	Translation of syntax
<i>Mapping of an H.323 Conference and a SIP Session</i>	Unique global conference ID	Unique global session ID	Map H.323 Conference ID and SIP Session ID
	Media capability description: H.245 <i>capabilityDescriptor</i>	Media capability description: SDP message	Map the H.245 <i>capabilityDescriptor</i> structure to SDP syntax
<i>Advertising</i>	Obtain conference information via LDAP (optional)	Announce session information via SAP	Conference information availability
<i>Determination of Conference Media Capability</i>	Centralized determination of conference media communication modes via an MC	Session media capability determined by a SIP creator	Applicable solution: Central determination via an intermediate agent serving as an H.323 MC
	Dynamically changeable via H.245 negotiation procedures	Updated through advertised SDP messages from the session creator.	
<i>Control over Membership</i>	RAS procedures defined in H.225 for conference membership control	No admission control mechanism to manage session membership	Generic admission control mechanisms and strategies handling conference membership

### 3. Conferencing Gateway Design



**Figure 1. Overview of GCCG Architecture**

**Figure 1** illustrates the use of a gateway to achieve transparent interoperation between SIP (e.g. MBone tools) and H.323 (e.g. NetMeeting) endpoints. The key component in this architecture, the Generic Conference Control Gateway (GCCG), participates in both the H.323 and the SIP protocols. It functions as an H.323 gatekeeper and MC, listens to session announcements from the MBone cloud, and sends out SAP messages to advertise newly created H.323 conferences. The GCCG handles the mapping and translation between the signaling protocols, and also forwards media streams between H.323 and SIP clients. Our goal is develop a GCCG that supports interoperation between H.323 and SIP clients without having to modify the H.323 or SIP protocols and without having to modify the client software. Ideally, neither SIP nor H.323 clients would be aware of the fact that some endpoints are using a different protocol.

The biggest challenge in the design of the GCCG is to map the functions and procedures provided by each protocol onto equivalent functions in the other protocol. Based on the significant differences between H.323 and SIP, there are some

situations in which it may be difficult or even impossible to devise a mapping that is transparent to the clients. First, some functions provided in one protocol may not be supported in the other protocol. For example, conference media capability negotiation procedures are provided in H.323 but no corresponding functions are supported in SIP. Second, one procedure in one protocol maps to several separate procedures in the other one, or equivalent procedures may be performed at different times in the two protocols. One example is the determination of conference descriptions. In SIP, this is accomplished before a SIP session creator sends out SDP session announcements. In H.323, it is separated into several steps, as new participants join.

#### 3.1 Key Design Decisions

We describe our design decisions in mapping between the two protocols.

##### Conference Call Messages Translation

H.323 provides a bundled protocol suite of call setup procedures, control functions and media channel control procedures in order to conduct multimedia communication. In SIP, media transmission can be initiated immediately after the call setup procedure since the media channel description can be contained in the request message. In other words, one call setup procedure in SIP may be mapped to several message exchanges in different procedures of H.323. Based on this scenario, the GCCG should synchronize the call signaling/control information exchanges and map procedure(s) in SIP to the corresponding procedure(s) in H.323. The GCCG also translates the call signaling and control message coding formats. H.323 uses ASN.1 PER, while SIP uses ASCII.

**Table 2. Signaling Translation**

<i>Call types</i>	<i>H.323</i>	<i>Call Direction</i>	<i>SIP</i>
<b>Conference CREATE</b>	H.225 <i>SETUP</i> (conference goal: <i>CREATE</i> ) (with <i>Fast Connect</i> procedure supported) Or H.225 <i>SETUP</i> (conference goal: <i>CREATE</i> ), and H.245 <i>CapabilitySet</i> message(s) (without <i>Fast Connect</i> procedure supported)	⇒	SDP message
	N/A (SAP not supported)	⇐	SDP message
<b>Conference INVITE</b>	H.225 <i>SETUP</i> (conference goal: <i>INVITE</i> ) (with <i>Fast Connect</i> procedure supported) <Scenario: create a new conference>	⇒	SIP <i>OPTION</i> message (requesting media capability set) and, SIP <i>INVITE</i> message (containing SDP message)
	H.225 <i>SETUP</i> (conference goal: <i>INVITE</i> ) <Scenario: using an ongoing conference>		
	H.225 <i>SETUP</i> (conference goal: <i>INVITE</i> ) H.245 <i>CapabilitySet</i> message(s) (without <i>Fast Connect</i> procedure supported)	⇒	SIP <i>INVITE</i> message (without session description) SIP <i>OK</i> message (client capability set)
	H.225 <i>SETUP</i> (conference goal: <i>INVITE</i> ) (NOTE: MC determines the common media capability types of the conference via the media capability information in the SDP message)	⇐	SIP <i>INVITE</i> message (containing SDP message)
<b>Conference JOIN</b>	H.225 <i>SETUP</i> (conference goal: <i>JOIN</i> )	⇒	N/A, (IGMP messages sent out to a corresponding multicast router)
	N/A	⇐	N/A

The three main conference call types defined in H.323, CREATE, INVITE and JOIN, must be mapped onto equivalent SIP protocol functions. Clients expect these conference call types to be available. Similarly, SIP and SDP messages must be mapped onto equivalent H.323 procedures. Table 2 summarizes our mapping of call signaling messages between H.323 and SIP. Note that in several cases, one message in one protocol translates into several messages in the other protocol. Also, to support the distribution of conference media type information, the GCCG must translate an H.245 terminalCapabilitySet structure containing multiple media capability sets into multiple SDP messages.

The above H.323 and SIP functions are accomplished through asynchronous and asymmetric message exchanges. The GCCG must not only be capable of handling those signaling messages, but must also keep two sets of protocol state in order to achieve conference call signaling and media information exchange.

**Central Determination of Conference Media Capability**

In H.323, before joining a conference, the incoming endpoint provides its media capability set to the MC, so that the MC can determine what communication modes the new endpoint has in common with the other participants. The MC then selects a media mode that all members have in common. This means that every time there is a member change, the common media determination procedure is invoked to decide which media type is the most suitable. SIP does not normally reevaluate the media type of a conference on a regular basis. In SIP, every member follows the description of the SDP messages, which is either contained in an 'INVITE' message or extracted from the regularly advertised Session Announcement message.

In our design, the GCCG finds out the media communication capabilities of each H.323 or SIP client and determines the applicable common media types in a conference. Therefore, the GCCG serves as an H.323 MC. For H.323 clients, the GCCG simply follows the H.245 procedures to learn about the client's capabilities. Dealing with SIP clients is more difficult. If the conference is initiated by an H.323 client, the GCCG will simply use the conference media mode that it selected in its role as MC in the SDP messages that it multicasts over MBone. If the conference is initiated by a SIP client, the GCCG will use the SDP messages that it receives through SAP to determine the conference media mode that the SIP clients would like to use.

The above approach for determining the conference media mode is simple. However, it is not completely general since it does not consider the full media capabilities of SIP clients. In Section 5, we describe a more general solution that uses the SIP OPTION message to obtain media capability information from SIP clients. In practice, only a small number of media options are used, so this simple solution will often be sufficient.

**Ongoing Conference Information**

SIP endpoints can learn about current sessions by listening to the periodic announcements generated by SAP (Session Announcement Protocol). These announcements contain an SDP message that provides session information for active sessions. H.323 does not have a similar mechanism for disseminating information about ongoing conferences. Some H.323-based products (e.g. NetMeeting) use an LDAP directory to keep track of ongoing conferences and their members. H.323 clients can use LDAP to query the directory, but there is no mechanism to automatically broadcast conference information.

In our system, the GCCG is responsible for keeping track of active conferences. When a conference is created by an H.323 client, the GCCG uses SAP to advertise the session over the MBone cloud to potential SIP clients. In the other direction, when a SIP client creates a new conferencing session, the GCCG maps the new SIP session to a virtual H.323 conference. The GCCG generates a conference ID for the H.323 conference and keeps track of the mapping between the MBone and the H.323 session IDs. We did not implement an LDAP directory for the GCCG, so it is not possible for H.323 clients to automatically learn about sessions created by SIP clients. Therefore, when an H.323 client wants to participate in a session created by a SIP client, it will have to obtain the H.323 conference ID via some out-of-band method (e.g. e-mail, or from a web page).

SIP and H.323 use different representation mechanisms to guarantee global uniqueness of conference IDs. However, both mechanisms refer to node network location and conference creation time. In H.323, a conference ID is a 128-bit value specified in the SETUP message; in SIP, it is a text string attribute in the SDP message. Because the identifiers are unique in their respective domains, the GCCG can simply keep a one-to-one mapping in both the H.323 conference and SIP session information databases. Thus, a session identifier stored in an H.323 conference descriptor (which is built when the conference is created) can be used to find the corresponding SIP session information and vice versa.

### Conference Management: Membership Control and Session Management

An H.323 client must register with an H.323 gatekeeper before it can issue a call request for a conference or join an existing conference. Therefore, one component in the GCCG must have "gatekeeper" functionality to handle RAS messages and to keep a registry of all active H.323 clients. The gatekeeper decides to accept or reject client requests; its decision is based on a variety of factors, including policy rules and server load. SIP, on the other hand, does not have an equivalent admission control protocol. The closest SIP function is an optional name server component that can resolve addresses or user identifications if no other naming schemes are supported, e.g. to support the delivery of INVITE messages for mobile users.

Our prototype GCCG provides basic H.323 gatekeeper functionality. However, we do not extend this function to SIP clients; a SIP client does not have to register to a SIP server before it can participate in a conference. In order to provide address or user location resolution in a heterogeneous conferencing session, a conference control server having the dual role of a gatekeeper in H.323 and a proxy server in SIP would be required to conduct membership and media session management.

## 3.2 Functional Design

Figure 2 shows the primary functional components of the GCCG. The left hand side of the figure implements the SIP and SDP protocols. It has the following properties:

- *IP-multicast capability:* All the SIP/SDP messages and media streams are forwarded to participants by IP multicast.
- *Session Announcement:* GCCG must periodically broadcast a Session Description message on behalf of H.323 terminals. This entails looking for unused multicast addresses and composing SDP Session Description messages. In addition,

the GCCG must send an SDP update message if it determines that a change in media stream type is needed to support the requirements of new members.

- *SIP message parsing:* GCCG parses incoming SIP messages, forwards them to appropriate message handlers, and extracts media channel information.
- *SDP message parsing:* GCCG must parse incoming SDP messages and store the message for future reference. For example, if an H.323 endpoint wants to join a video conference session, GCCG will search its table to see if the conference exists and which media channel it uses.

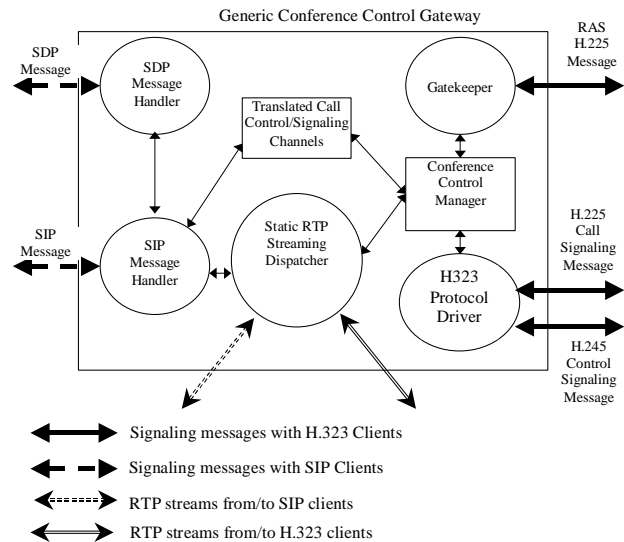


Figure 2. Functional Components of GCCG

The right side of the picture implements the H.323 protocol stack. It has the following properties:

- *Gatekeeper:* This component in the GCCG accomplishes the RAS signaling function by using H.225.0 messages to perform registration, admissions, bandwidth changes, status, and disengage procedures.
- *H.323 protocol driver:* This component implements the H.323 protocol, including ASN.1 encoding and decoding, call set-up procedures and other call control functions (e.g. open media channels).
- *Conference control manager:* This component translates call signaling messages, manages all connections with in-call H.323 clients, and handles control messages for all media channels among H.323 and SIP clients.

Finally, the center components in the block diagram in Figure 2 are responsible for protocol translation and for keeping session state:

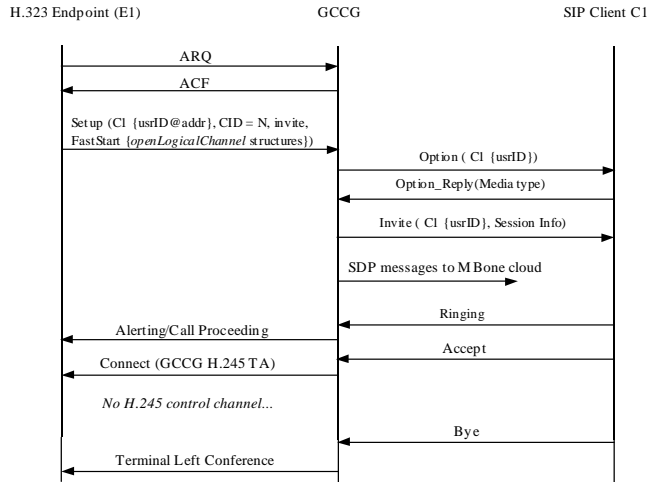
- *RTP/RTCP stream forwarding:* All media streams are carried using RTP, so GCCG must be able to forward the message to appropriate receivers. For example, GCCG should send video streams to SIP clients (VICs) using specific IP multicast channels. However, it can only send a single stream to H.323 endpoints, because NetMeeting is unable to process multiple video streams.

- *Session management*: The GCCG must listen to all the session announcements on MBone to prevent session address clashes, and it must periodically make SDP announcements on behalf of H.323 endpoints. It also must remember the relationship between endpoints and videoconference sessions

### 3.3 Example - Conference Invite and Join

In order to illustrate the mapping between the H.323 and SIP call signaling procedures performed by the GCCG, we describe a simple scenario in which an H.323 client first invites a SIP client to conferencing session via *Fast Connect* mode, and then an H.323 client joins the newly created session.

**Figure 3** shows the call signaling procedure between the GCCG, the H.323 client E1, and the SIP client C1 for the first step of the example. H.323 client E1 contacts the GCCG to set up a new session via *Fast Connect* mode, and it invites SIP client C1. The GCCG also announces the new session using SAP. Note that since we use *Fast Connect* mode, no H.245 control channel is established.

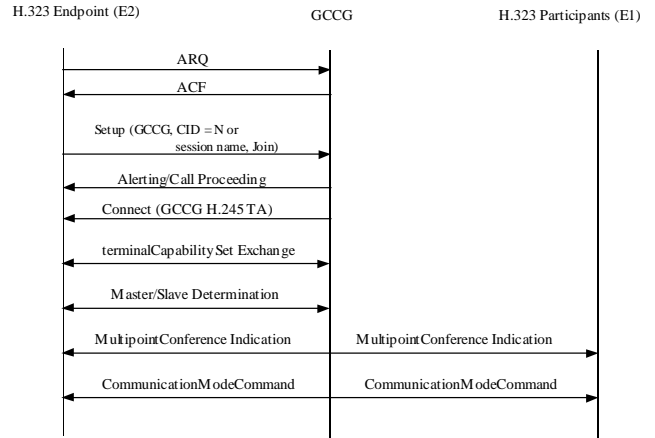


**Figure 3. H.323 Client Invites SIP Client via FastStart Mode**

After the H.323 and SIP client create a conference, another H.323 client E2 may join the ongoing conference/session via standard procedures defined in H.323 (ad hoc multipoint conference procedure in gatekeeper routed mode) as shown in **Figure 4**.

After client E2 has been allowed to join the conferencing session and has given its capability set to the GCCG, the GCCG sends out a MultipointConference indication to all H.323 participants. It then determines what communication mode will satisfy all participants of the session. Next, all H.323 participants will receive a CommunicationModeCommand message from the GCCG, which tells them whether the communication mode has changed. If it has, they need to close existing media logical channels and open new channels with the media communication mode specified in the CommunicationModeCommand message.

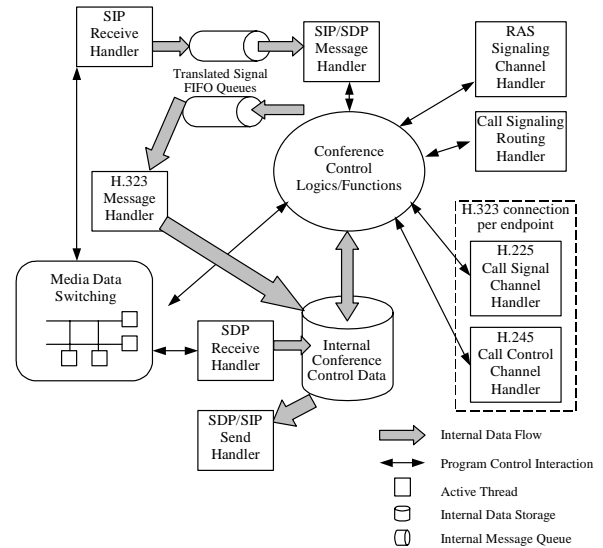
On the MBone side, the GCCG must send an updated SDP message when there is a change in the conference media type.



**Figure 4. H.323 Endpoint joins an ongoing conference/session**

## 4. Implementation

We describe the internals of our GCCG prototype.



**Figure 5. GCCG Internal Architecture**

### 4.1 Component Description

**Figure 5** shows the main GCCG components, implemented as threads in our prototype, and their interactions. Our GCCG prototype supports call signaling translation, protocol semantics adaptation, media stream forwarding/switching and conference management.

The SIP/SDP functions are again shown on the left in **Figure 5**. The SDP\_Receive\_Thread listens for, parses, and records SDP announcements. The SDP\_SIP\_Send\_Thread periodically announces SDP information, and after changes in media type resulting from capability determination, it re-announces the updated session information. Before sending an INVITE to a SIP client, this thread also sends an 'OPTION' message to request the media capability of that client, so the GCCG can decide which media type is suitable for the conference. Finally, the SIP\_Receive\_Thread forwards SIP messages to H.323 endpoints after translation.

The right side of **Figure 5** groups the H.323 related functions. The RAS Signaling Channel Handler receives the RAS PDU from H.323 clients and responds according to the procedures defined in H.225. The Call Signaling Routing Handler listens to sync PDUs from H.323 clients via a TCP connection at service port 1720. This thread is in charge of establishing new connections with H.323 clients that conference service and of handling H.225 call signaling messages and procedures. The H.225 Call Signal Channel Handler and the H.245 Call Control Channel Handler handle H.225.0 and H.245 messages and procedures respectively (e.g. MasterSlaveDetermination, TerminalCapabilityExchange).

Finally, the threads shown in the center of **Figure 5** implement protocol translation and session management functions. The Conference Control Functions component is a set of internal control callback functions handling H.323/SIP message translation, SIP/H.323 semantics adaptation, H.323 multipoint control, conference control data management, RTP stream switching configuration and other bookkeeping procedures. The Internal Conference Control Database stores data such as the H.323 endpoint registration table, conference list information table, Mbone session list information table, active endpoint information table, and SDP session information. Finally, the Media Data Switching block is responsible for forwarding video streams between SIP and H.323 endpoints. As mentioned earlier, our H.323 endpoints cannot handle multiple incoming video streams. Therefore, GCCG will receive RTP packets from every endpoint in the conference and unicast packets to H.323 endpoints and multicast to SIP endpoints respectively.

Our implementation uses the H.323 protocol stack available from the OpenH323 Project [18]. For SIP, we used some of the SIP/SDP functions from the VIC and SDR release that is part of the Mbone tools suite from University College London [17].

## 4.2 Operation and Status

Our prototype GCCG implementation includes most of the components described above, and the three major call types, CREATE, INVITE and JOIN, are completely functional.

The GCCG server does not interfere in pure SIP or H.323 sessions but it will quietly keep track of any conference information necessary to perform translation, if needed. When a NetMeeting user creates a session, GCCG periodically advertises the session information using SAP/SDP to the Mbone community. NetMeeting and VIC users are able to join any active video conference created by either an Mbone or NetMeeting client. Moreover, both can invite NetMeeting and VIC clients. The GCCG not only translates signaling messages appropriately, but it also maps the session IDs used in the two signaling protocols. Media streams are forwarded using RTP/RTCP. GCCG uses multicast to forward packets from NetMeeting to VIC users, and uses unicast to forward packets from VIC to NetMeeting users. Since NetMeeting cannot process media streams from multiple sources, the GCCG will ensure that NetMeeting receives media streams from only a single source.

Our GCCG prototype does not currently use an LDAP server to keep track of sessions (see Section 8). This feature requires an H.323 implementation that can closely collaborate with an LDAP server, including support for registering a session with the server and for requesting session information from the server. Moreover, our capability exchange function is still very simple.

Finally, since we have focused on video, we have not tested Mbone's audio application with our prototype.

## 4.3 Functionality of the Prototype

The GCCG prototype satisfies our goal of supporting hybrid H.323/SIP conferencing sessions without requiring changes to the H.323 or SIP protocols and without requiring any changes to the H.323 and SIP client software. Specifically, it achieves this for Microsoft NetMeeting and for clients using the Mbone tools, as described above.

From the perspective of the end-users, the fact that two different signaling protocols are used is largely invisible. The only exception is that since NetMeeting does not provide an appropriate user-interface to access information about active sessions, NetMeeting users have to type in short instructions in NetMeeting's address bar to join specific sessions. For example, a NetMeeting user has to type 'INVITE:TestConfName|Jason' to invite another registered NetMeeting user, Jason, to join a session named 'TestConfName', which was created by the caller. To invite a SIP user, on the other hand, the NetMeeting user has to type the command 'INVITE:TestConfName|Jason?128.2.34.56'. The IP address is the address of Jason's desktop system. The GCCG needs it so it can contact Jason and invite him to the conference.

While our prototype achieves interoperability between H.323 and SIP, it is important to realize that the clients that we use do not use all features of the H.323 and SIP protocols, and, as we mentioned in Section 3.1, the GCCG prototype also supports only a subset of the protocol features. We discuss the challenges involved in achieving more complete interoperability in Section 5.

## 4.4 Performance Evaluation

We believe the overhead resulting from translation of call signaling messages and interaction of control signaling between clients and servers is not as significant as processing or forwarding media streams. In addition, the latency of call signaling is not as critical as that of real time media streams.

In order to investigate this issue we measured the transmission efficiency between our prototype GCCG server and clients under various load conditions. The specific metric we selected is the round trip time (RTT) between a GCCG server and an H.323 client, which is based on an H.245 control signaling procedure used for control channel maintenance. We measured the roundtrip time for different scenarios involving between one and five SIP or H.323 clients. Our goal is to quantify the effect on the RTT of an increasing number of participants in a conference. Moreover, it shows the performance gain from multicast.

### Evaluation environment

The performance significantly depends on hardware capability of the GCCG server and network environment. Our measurements were made in the following environment:

- **GCCG Server:** 400 MHz PC with 128 MB RAM and a 10 Mbs Ethernet interface, running Linux Redhat 6.1 (kernel v2.2.5).
- **Clients:** 266 MHz PCs with 128 MB RAM, a 10 Mbs Ethernet interface, and a Logitech QuickCam, running MS Windows 98/NT.

- **Video Conferencing Software:** MS NetMeeting version 3.01 and the Mbone tools SDR v2.9 and VIC v2.8. The common communication mode is H.261 encapsulated in RTP.
- **Network Configuration:** 10 Mbs shared Ethernet.

### Evaluation Scenarios

In our first scenario (Figure 6), all five machines are running NetMeeting as H.323 clients, and we measure the roundtrip time between the GCCG server and the first participant for different numbers of participants, in a newly created conference.

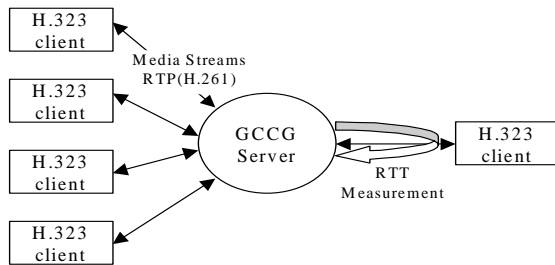


Figure 6. Scenario 1: Five H.323 Clients

In the second scenario (Figure 7), the first participant is an H.323 client using NetMeeting, while the remaining systems are SIP clients running VIC with SDR. The roundtrip time is again measured between the first participant (the H.323 client) and the GCCG server.

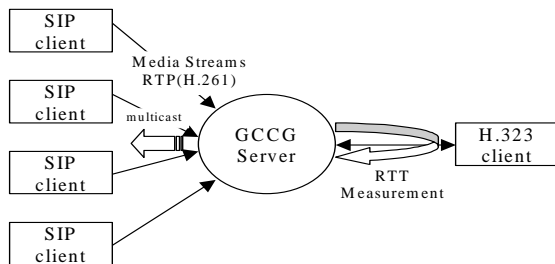


Figure 7. Scenario 2: one H.323 client and four SIP clients

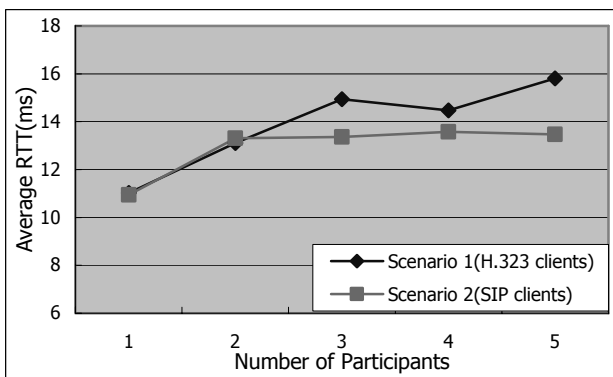


Figure 8. Comparison of Average RTT

We summarize our measurements in Figure 8. Each data point represents the average of 36 measurements, and the standard deviation is in the range of 0.77 to 1.00 millisecond.

In scenario 1, each H.323 client transmits its own video stream to the GCCG and the GCCG then forwards the video stream to each H.323 client. The video RTP streams are transmitted via unicast channels. From the measurement in Figure 8, we can see that the

roundtrip time increases with the number of participants. This is not surprising given the increased load on the network and on the server. While there are too few measurement points to reliably extrapolate, the increase of the RTT appears to be linear with the number of participants.

Scenario 2 differs from scenario 1 in two ways. First, the GCCG server has to do the conferencing signalling protocol translation. Second, the GCCG server can use IP multicast to forward media data to SIP clients, thus reducing the load on both the network and the server. We observe from Figure 8 that the roundtrip times are lower than in scenario 1, especially when there are more clients. Clearly, the use of multicast pays off. The roundtrip times for two clients are basically identical, which suggests that the overhead of the signalling protocol mapping is negligible (the video transfer costs should be similar in that case).

## 5. Discussion

While our GCCG supports interoperation between NetMeeting and the Mbone tools, our tasks was simplified by the fact that the two clients use only a subset of the protocol features. Based on our experience with GCCG, we discuss some broader interoperability issues.

### Negotiation of Media Capabilities

H.323 and SIP have different approaches for establishing conference media types. In H.323, the common media modes for the conference are selected by the H.323 MC, based on information obtained via the H.245 capability negotiation procedures. The media selection procedure is executed every time a new H.323 client joins the session. In SIP, the session creator specifies the preferred media transmission types, and it uses SAP messages to periodically multicast this information to participants and potential participants. The media types are typically selected before the session starts and they do not change during the session. Section 3 describes a simple method for bridging the gap between these very different solutions. While this solution is likely to work in most cases, it is not completely general. The specific problem is that there is no media capability negotiation procedure defined in SIP that would allow the SIP session creator to modify the existing media capability types if new SIP clients with different media capabilities want to join the session.

One possible solution is to have a SIP proxy server that can determine common media capability modes on behalf of the SIP session creator. The SIP proxy server can request media capability information from each SIP client via SIP OPTION messages and store this information for the duration of the session. In our GCCG architecture, the SIP proxy server component can reside in the GCCG to reduce signaling overhead. Every time a new client joins the conference, the proxy server can recompute the common media capabilities and multicast the new preferred media types using SAP.

### Session advertising

A SIP client can learn about a new conference created by H.323 clients via the GCCG and SDR. However, an H.323 client in the zone with the GCCG cannot be informed of a newly created conference in Mbone. A possible solution is that the GCCG uses a DAP (directory access protocol) server to keep track of session information. An H.323 client can then actively request directory service from the DAP server (via LDAP). This also requires that the GCCG is capable of translating SDP messages into a format



(e.g. BER in LDAP) decodable by H.323 clients. However, the transformation or interpretation between SDP and DAP has not been standardized.

In H.323, a conference identifier (conferenceID) is generated by the conference creator. Similarly, a session identifier (sessionID) is generated by the SIP session creator. Therefore, in order to apply the existing protocol syntax and semantics without modification, conference information mapping between these two protocols should be standardized. This would make it possible to have a user interface in SIP and H.323 client applications that gives users information about sessions, independent of how they were created.

Our GCCG implementation does not use an LDAP server. Instead, the GCCG internally maintains a list of conferences and the mapping between the H.323 and SIP IDs. The GCCG can reply to conference list request using the ConferenceListChoice structure of the Facility message defined in H.323.

### Adaptation of Call Signaling Semantics

If H.323 clients can set up a conference with either H.323 or SIP clients via the *Fast Connect* procedure, the call signaling and the synchronization overhead on the GCCG are reduced significantly since the conference setup procedures of SIP and H.323 are symmetric. This simplifies the message procedures. Without *Fast Connect*, it is necessary to create an additional H.245 control channel per client. However, H.323 compliant client applications are not required to support the *Fast Connect* procedure. New H.323 compliant clients applications should support this mode.

### Conference Control

Conference control differs significantly between H.323 and SIP: the former has a tightly coupled conference control standard and the latter is a loosely coupled one. In ITU-based conference systems, conference control mechanisms are specified in the T-series standards (e.g. Generic Conference Control Protocol, T.124). The standards offer conference control services such as floor control, chair control, voting, and some management procedures. ITU-T Recommendation T.124 must coexist with companion Recommendations T.122 and T.125 (Multipoint Communication Service; MCS) and T.123 (transport layer support) in each H.323 terminal or MCU to provide conference control. Currently, no related protocols are applicable for conference control services in SIP. The idea is to offer these functions in higher layer protocols. Ideally, these protocols should mesh well with the syntax and features in the ITU-T T-series standards.

The GCCG bridges this difference by participating in both types of protocols. It serves as a lightweight conference control server (e.g. admission control, RAS, bandwidth management) for H.323 clients and as a session directory server for SIP clients. Since most H.323 compliant endpoints (e.g. Netmeeting and CU-SeeMe) do not provide advanced conference control services, we did not implement them in our GCCG prototype.

### IP Multicast

Our GCCG server exchanges media streams/RTCP messages with H.323 endpoints using unicast, since H.323 clients are not required to have IP-multicast capability. In addition, GCCG will also advertise sessions created by H.323 endpoints and forward media streams to the Mbone community via IP-multicast. This design allows efficient message/media stream exchange,

especially when there are many SIP clients but only a few H.323 endpoints. This raises two design issues.

First, the IP multicast architecture is very open, and any host can subscribe to an IP multicast session. As a result of using IP multicast on the SIP side of the GCCG, any (unauthorized) user can receive data from a specific conferencing session or send data to the conference, so we lose control of the administration of a video conference as it exists in H.323. This violates the intent of the H.323 standard, and even in a loosely-controlled environment, like SIP, this may not be desirable, e.g. it makes it too easy to listen in on private conferences.

This problem can be addressed in two ways. First, SIP allows a conference to use authentication and encryption. However, in the spirit of SIP and Mbone tools, the administration (e.g. key management) is not built into SIP but has to be provided by a separate application. To extend the conference control as it exists in H.323 to SIP clients, we could add a component to the GCCG that uses and the authentication and encryption support in SIP to control who can participate in the session. To rectify this problem in a more fundamental way, IP-multicast would have to be changed to provide control over who joins a session. Many applications could benefit from this, and changing the multicast model to better match the needs of applications has been an active area of research, e.g. [6].

A second design issue centers around the use of multicast for media transfer between H.323 clients. H.323 clients in a multi-party conference can exchange their media streams via multicast assuming they maintain H.245 control channels with an H.323 MC. The multicast addresses of the conference media channels are determined by the MC and delivered to each H.323 participant via the CommonCommunicationMode-Command message defined in H.245. The GCCG could make use of these commands to extend the use of multicast. If the conference is created by a SIP client who specifies the multicast addresses, the MC can retrieve the media multicast addresses from the advertised SAP messages. This can then be used by an MC component to use multicast in an interoperable way for H.323 clients.

In our GCCG prototype, we only use IP-multicast for SIP clients; NetMeeting does not support the use of IP multicast. We do provide admission control for SIP clients.

## 6. Related work

SIP and H.323 are both still evolving protocols and neither of them will dominate the whole market in the foreseeable future. As a result, several research groups have started to work on the problem of interoperability between SIP and H.323. The aHIT! group from IMTC, ITU-T SG 16 and TIPHON are several of the most active groups. So far, most of the work is in its initial stage, and the focus is on problem analysis and architecture design. The IETF has proposed an Internet Draft [12] that defines basic procedures of call establishment and termination. As with our research, they also realized the necessity of translation between H.245 and SDP, and suggested that the *Fast Connect* procedure simplifies the process. Optional extensions of these two protocols and advanced features have not been fully discussed. Related tasks, like billing, security and other supplement services for an inter-networking environment, are still under study. However, as far as signal mapping is concerned, the IETF draft provides a thorough guideline of basic conversion between these two protocols, and it is also a comprehensive analysis that pinpoints

the difficulty of interoperation. Moreover, it presents a solution to alias address resolution and to the calculation of common subset capabilities.

Based on this draft, Columbia University implemented a SIP/H.323 signaling gateway [8]. Similar to our implementation, it is used to bridge between SIP and H.323 client applications. Their gateway integrates a SIP server and an H.323 gatekeeper, and client registration and address resolution procedures are more symmetric and simplified. Our gateway is similar but focuses on different system aspects:

- 1) The Columbia gateway focuses on Voice of IP (VoIP) services, while our focus is on multi-party multimedia conferences, which requires support for multiple media stream processing and related signaling, e.g. session create and join. Therefore, advertising conference session information to SIP and H.323 clients is a key feature in our design.
- 2) We support IP-multicast in our design so that media streams can be transmitted efficiently. This is especially important in high quality multimedia conferences involving video.

## 7. Conclusions

In this paper we looked at the issue of interoperability between video conferencing clients that use one of two competing standards, H.323 and SIP. We identified some key differences between the two protocols and proposed a generic conference control gateway (GCCG) as a way of bridging the two worlds. We presented the design of the GCCG and described our prototype implementation. The GCCG allowed us to set up basic multi-party video conferences between unmodified H.323 clients (Microsoft NetMeeting 3.1) and SIP clients (VIC 2.9 from UCL).

While our GCCG prototype supports interoperation between H.323 and SIP clients, it also helped us identify some problem areas. First, differences in the establishment of sessions create problems with session identification and negotiation of session parameters. While we were able to map between H.323 and SIP sessions, doing so consistently, requires an explicit session directory and support in the video conferencing GUI. Bridging the differences in the negotiation of session parameters will require changes in the standards. A second issue is conference management, which is handled in a tightly integrated way in the H.323 standards family and in a very decentralized way in SIP. While we were not able to investigate this issue in depth because of the limited capabilities of the clients, it did not appear that interoperability would be a problem, assuming matching protocols exist in the SIP world. Finally, while multicast plays a key role in SIP-based video conferencing, it is less well supported in H.323 conferencing. This has an impact on efficiency and how tightly the conferencing session can be controlled. Improving interoperability has the potential of also improving both standards. H.323 could benefit from better support for multicast-based media transport, while SIP could benefit from tighter session control.

## Acknowledgements

This research was sponsored in part by the Defense Advanced Research Project Agency and monitored by AFRL/IFGA, Rome NY 13441-4505, under contract F30602-99-1-0518.

## 8. REFERENCES

- [1] E. Amir, S. McCann, and H. Zhang, "An Application Level Video Gateway," Proceedings of ACM Multimedia '95, San Francisco, CA, Nov. 1995.
- [2] M. Handley, C. Perkins, E. Whelan, "Session announcement protocol", RFC 2974, IETF, Oct. 2000.
- [3] M. Handley, H. Schulzrinne, E. Schooler and J. Rosenberg, "SIP: session initiation protocol," RFC 2543, March 1999.
- [4] M. Handley, I. Wakefield, and J. Crowcroft, "CCCP: Conference Control Channel Protocol-A Scalable Base for Building Conference Control Applications", ACM Computer Communication Review, vol 25, pp. 275-287, Oct. 1995.
- [5] V. Jacobson and M. Handley, "SDP: Session Description Protocol", RFC 2327, IETF, April 1998.
- [6] C. Shields, J.J. Garcia-Luna-Aceves, "KHIP - A Scalable Protocol for Secure Multicast Routing," Proceedings of ACM Multimedia '99, Cambridge, MA, Aug. 1999.
- [7] N. Kausar, J. Crowcroft, "General Conference Control Protocol", IEEE Telecommunication 1998.
- [8] K. Singh, H. Schulzrinne, "Interworking Between SIP/SDP and H.323". Proceedings of the 1st IP-Telephony Workshop (IP Tel'2000), Apr. 2000.
- [9] S. McCanne, and V. Jacobson, "vic: A Flexible Framework for Packet Video". Proceedings of ACM Multimedia '95, San Francisco, CA, Nov. 1995
- [10] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications". RFC 1889, IETF, Jan. 1996.
- [11] H. Schulzrinne and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony", Network and Operating System Support for Digit Audio and Video (NOSSDAV), Cambridge, England, Jul. 1998.
- [12] Internet Draft, "SIP-H.323 Interworking", work in progress, Feb. 2001.
- [13] ITU-T Recommendation H.323, "Packet-based Multimedia Communication Systems", Sep. 1999.
- [14] ITU-T Recommendation H.225.0, "Call Signaling Protocols and Media Stream Packetization for Packet-based Multimedia Communication Systems", Feb. 2000.
- [15] ITU-T Recommendation H.245, "Control Protocols for Multimedia Communication", Feb. 2000.
- [16] ITU-T Recommendation Q.931, "ISDN User-Network Interface Layer 3 Specification for Basic Call Control", 1993
- [17] Mbone Conferencing Applications.  
<http://www-mice.cs.ucl.ac.uk/multimedia/software/>
- [18] Open H323 Project  
<http://www.openh323.org>