A SIP/H.323 Signaling Gateway Implementation for IP Telephony

Bruno F. M. Ribeiro — bruno@land.ufrj.br
Paulo H. de Aguiar Rodrigues[*] — aguiar@ufrj.br
Cesar A. C. Marcondes[†] — cesar@posgrad.nce.ufrj.br

Núcleo de Computação Eletrônica/Instituto de Matemática - UFRJ
Caixa Postal 2324, CEP 20001-970
Rio de Janeiro – RJ – Brasil

Abstract

The Internet has truly become the ubiquitous communication infrastructure, bringing many benefits to the users and making the communication over IP the predominant protocol architecture for data and multimedia transmission (Web). The regular telephone service shall, in a near future, converge to the Internet, giving place to a single integrated voice and data network. IP telephony is growing, following this expected trend, and different protocol standards, as SIP and H.323, are available for this new service. In order to guarantee interoperability and give users the freedom of choosing from equipment and applications based on these protocols, a SIP/H.323 gateway has to exist. We present the implementation of a SIP/H.323 gateway, which allows a transparent interoperation to the users. The implementation unites the two protocol stacks over a simple signaling message channel, allowing partial utilization of two existing open source codes: OpenH323 Opengatekeeper and SIP client IPTele, developed at the Helsinki University of Technology (HUT). The modular approach allows using developed modules for implementing other gateways.

Keywords: IP Telephony; Signaling Gateway; SIP; H.323

1. INTRODUCTION

One the most important changes in telephony in recent years is the migration from a circuit switched architecture to a new packet switched paradigm. Packetized voice is already a reality in the internal network of some service providers and many companies are planning an evolution towards this new paradigm [21,22]. On the other hand, local area networks and Internet access is presently a basic requirement for any enterprise. And even residences are deploying local area networks, wired or wireless, to promote interconnection for multiple users and equipments [1]. With ubiquitous Internet, IP telephony attracts special interest, despite factor as quality, security, availability and reliability have to improve to match existing standards set by common telephone services [15].

In regular telephone we face an old problem: we need to previously know the number where the person we want to talk to is. Sometimes setting up calls to different places is the only way to get a hold of someone. In IP telephony, we have the concept of personal mobility. Mobility is based on an identification or personal user alias [3]. IP telephony signaling locates the user independently of the hardware he is physically accessing. Furthermore, IP telephony allows a more efficient bandwidth usage, due to the statistical nature of packets and use of compression and silence suppression in voice communication, reducing from 64 kbps to only a few kbps the voice bandwidth requirement. Voice and data convergence favors cost reduction, from unifying technical services (maintenance, management and planning) and sharing resources. There is also the possibility of application integration, as deploying unified message systems, using the Internet to create value added services integrated with IP telephony [16].

---

[*] Supported by CNPq Fellowship PDE 200.123/79-3, Visiting Faculty at the Computer Science Department/UCLA.

[†] Supported by IM/NCE Master Graduate Program Fellowship.

Following this hype, many international organizations established standards to support packetized telephony. H.323 [2] was the first standard, proposed by ITU-T (International Telecommunications Union-Telecommunications Section). H.323 is a framework for implementing multimedia communications over packet networks without quality of service guarantees. There are many H.323 implementations in the market today, as Netmeeting from Microsoft, IP Phone from Intel, solutions developed by Picture Tel, among others [19].

To support IP Telephony over the Internet, IETF (Internet Engineering Task Force) proposed a simpler telephony signaling mechanism called SIP (Session Initiation Protocol) [3], based on the wide used HTTP protocol. Very soon, SIP was able to show its versatility and adequateness to merge with other WEB technologies, inducing many application implementations based in itself [17]. SIP has become a major contender in IP telephony.

SIP and H.323 are disputing the IP telephony scenario for long time, and many articles cover their differences [8,9]. Some updated versions of H.323 [2] clearly show a tendency to promote same features in both protocols. Thus, the most recent version of H.323 proposes mechanisms as Faststart over UDP, which is sending media description in a single message in the beginning of a session, similar to the INVITE method in SIP (see section 2.1), while in H.323 v1 the basic signaling works over TCP and media descriptions works in separate H.245 channels (see section 2.2).

Another protocol to join this scenario is MGCP (Media Gateway Control Protocol), an IETF.ITU-T partnership. H.248 is ITU-T equivalent to MGCP IETF. MGCP deals specifically with centralized control of sessions between equipments called Media Gateways, which are soft PBXs in IP networks [10].

Many companies are adopting IP telephony solutions or buying equipment based on those conflicting standards. However, the existence of an integrated communications system where the user have the option to choose its own telephony solution requires a protocol translator, a Signaling Gateway, to glue both H.323 and SIP architectures. Though Columbia University had implemented a gateway [13] as such, the inexistence of an H.323/SIP Signaling Gateway with open code motivated this work. Very recently, an open source gateway implementation was presented [23].

This paper is organized as follows. In Section 2, we present the basics of SIP and H.323, focusing on problems as translation, user location discovery, registration, among others, and ending up with a proposal for the gateway functional architecture. In Section 3, we focus on implementation issues and describe the internal architecture. Interoperability tests and data flow are discussed in Section 4. Finally, conclusions and future work are covered in Section 5.

2. GATEWAY FUNCTIONAL ARCHITECTURE

Initially we present an overview of SIP and H.323 behavior, followed by the proposed gateway functional architecture.

2.1 SIP Environment

SIP (Session Initiation Protocol) [3] is an IETF application level protocol which establishes, modifies and terminates multimedia sessions and/or connections. Sessions can be multimedia conferences, classes over the Internet, telephony over the Internet, and others.

Figure 1 shows a generic SIP environment. SIP main components are: SIP User Agent, SIP Proxy Server and SIP Redirect Server. This set of components working over an IP network is called a SIP network. These components are described in the following table.

SIP Component	Function
SIP User Agent	Client, or multimedia communication end point.
SIP Proxy Server	Server for redirecting requests and responses. It performs signaling as it was the call originator, and redirects response to real originator when it gets it.
SIP Redirect Server	Redirects requests and responses, sending msg to clients with the newly wanted SIP address, and not taking the role of proceeding with the call.
SIP Registrar Server	SIP Server supporting REGISTER requests, which are used to register user information in Location Server.
Location Server	SIP RFC [3] describes only user registration and inquiring features for this server, leaving to the implementor the choice of technology for its realization.

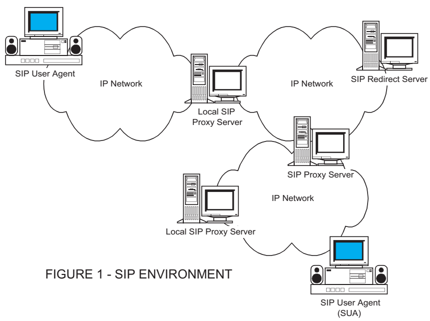


FIGURE 1 - SIP ENVIRONMENT

A SIP network can be accessed using an URI (Uniform Resource Identifier). URI is a compact string to address physical or abstract resources inside a network Some examples of SIP addressees are *alias* (or nickname) as <sip://user@server> or it can be a telephone number as <tel://5556666@gw.ufrj.br>. The URI host part can be a valid alphabetical Internet domain or a numeric IP address. SIP protocol is based on HTTP and uses MIME types (Multipurpose Internet Mail Extensions) to support different payloads. SIP works in a client/server model and its operations involve only request and response methods, as observed in HTTP and RTSP. SIP request methods are: INVITE, ACK, OPTIONS, BYE, CANCEL and REGISTER. LDAP (Lightweight Directory Access Protocol) [18] servers, which hold user profiles and directories, or local databases are used for user location.

For each request or response, there are a group of headers, organized as: general headers, containing important information about the call; entity headers, containing meta-information about message body; and the specific headers, which convey additional information besides those in request or response status line.

When answering requests, responses carry a response class identification number. Many temporary messages can be sent before a final response is finally sent. There are six response classes: Class 1XX, for temporary or informational responses; Class 2XX, for successful final response; Class 3XX, request redirection; Class 4XX, client errors; Class 5XX, server errors; and Class 6XX, network global errors.

Follow in Figure 2 the steps of an invitation flow to a SIP user:

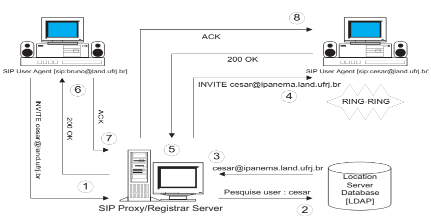


FIGURE 2 - SIP CALL PROCEEDING

(1) User <u>bruno</u> asks to create a session with alias <u>cesar@land.ufrj.br</u>. [SIP INVITE Request]
(2) Proxy Server asks Location Server Database where the user with that alias is (using LDAP).
(3) Server response is the user real location, a SIP mobility feature (it shows that last REGISTER came from ipanema.land.ufrj.br).
(4) Session open request is redirected by Proxy to correct address [SIP INVITE request]. User <u>cesar</u> in machine <u>ipanema.land.ufrj.br</u> will be

alerted by a RING tone [RING-RING].

(5) cesar decides to join the session and its SIP client answer back to Proxy Server confirming that session can be opened [200 OK successful response].

(6) Proxy server redirects this response to the calling client [200 OK to bruno].

(7) Calling client bruno indicates to Proxy that negotiation is over and session is open [ACK request with final media negotiation description].

(8) Finally, Proxy server tells called user that negotiation is over and session is open [ACK request with final media negotiation description].

2.2 H.323 Environment

H.323 conceptually describes terminals, equipments and services for multimedia communication over local area networks without QoS guarantees. H.323 terminals and equipments can handle real-time voice, data and video or any combination of these, as video telephony. LAN where H.323 is operating over can be just a single segment, multiple segments or a very complex network topology. However, H.323 operation over multiples LAN segments or over the Internet can face scalability issues [9].

In H.323, user registers in a network element called Gatekeeper (GK). GK is a server in charge of connection admission, keeping tracking of available bandwidth, and search for registered users. H.323 is based on the notion of administrative zones. Administrative zone is a set of neighboring GKs, i.e. GKs that are in the same administrative region, but have different registered users.

H.323 terminals can be implemented in software in PCs or integrated in independent hardware as videophones or IP phones. Voice support is mandatory, while data and video supports are optional. H.323 treats media transport as a channel abstraction, allowing more than one channel allocation for each media. Other recommendations are part of the H.323 stack: H.225.0 [7], describing RAS (Request, Admission and Status) and synchronizing messages; H.245 [6], describing media control; H.261 and H.263, for video coding; G.711, G.722, G.728, G.729 and G.723, for audio coding; and T.120, for data protocol.

H.323 uses H.245 procedures to open logical channels for each kind of media. Before opening the channel, terminals have exchanged capability messages under H.245 and know which media can be sent or received and what transport is supported by the other terminal. H.323 signaling and call opening procedures are based on ISDN Q.931, using extensions defined in H.225 for the user-to-user optional field (SETUP UUIE). Hence, all basic call control is conducted under Q.931/H.225 and media negotiations under H.245. Fig. 3 shows H.323 architectural elements. MCUs are elements supporting conference control functions for multi point operation.. The gateway in the figure could be a PSTN gateway that allows for transmission format and communication procedures translation, besides detecting and generating DTMF (Dual-Tone Multi Frequency) signals which correspond to H.245 signaling (necessary for PSTN interaction).

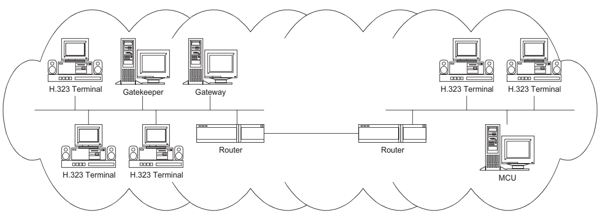
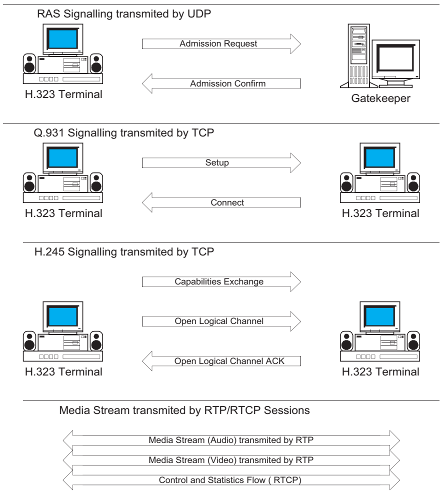


FIGURE 3 - H.323 ZONE ELEMENTS

H.323 signaling is extremely complex, mainly because of its extensive protocol stack and conformance to old ITU-T standards. In Figure 4 we have an idea of this complexity. ARQ

FIGURE 4 - RECOMMENDATION H.323 V.1 SIGNALLING

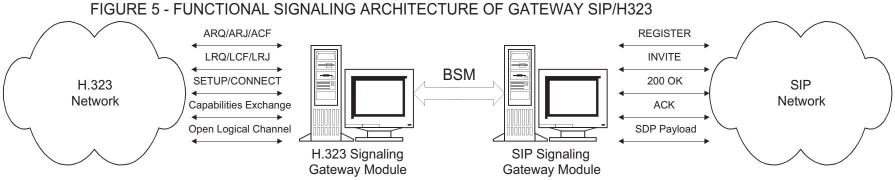


(Admission Request), ARJ (Admission Reject), and ACF (Admission Confirm) messages are exclusive to H.323 terminals. These messages together with LRQ (Location Request), LCF (Location Confirm), and LRJ (Location Reject) messages, which are used by gatekeepers, form the set of messages called RAS (Request, Admission and Status). A terminal registered with a GK will always ask GK for authorization to initiate an IP telephony call. Q.931 messages are SETUP (ISDN call establishment), Call Proceeding (equivalent to SIP RINGING) and CONNECT (call establishment confirmation).During H.245 media initialization phase a TCP port is allocated for negotiating media capabilities and order of preference. H.245 channel is maintained open in case a new media session is opened or an existing one is modified. H.245 basic messages are: Capability Exchange (exchange set of media capabilities between terminals), Open Logical Channel (media control channel opening) and Open Logical Channel Acknowledge. Media transport after negotiation phase occurs at the network level using RTP (Real time Transport Protocol) [5], which is also the same procedure SIP uses to effect media transport.

2.3 SIP/H.323 Funcional Environment

SIP and H.323 interoperability involves more than a simple translation because many architectural differences have to be overcome due to protocol design incompatibility. Addressing, user registration maintenance and media negotiation are among the main problems to be solved. In [13,14], a general SIP/H.323 signaling gateway architecture is proposed. Figure 5 shows our SIP/H.323 SGW (Signaling Gateway) implementation.

FIGURE 5 - FUNCTIONAL SIGNALING ARCHITECTURE OF GATEWAY SIP/H323



Our implemented SGW has a modular software architecture comprising a SIP-BSM module and an H.323-BSM module. Module is a process able of interpreting and mapping a certain telephony protocol signaling into a defined set of functions called BSM (Basic Signaling Messages), which builds an intercommunication channel between the modules. BSM are simple messages whilst flexible enough to allow interconnection with any telephony protocol. BSM is a unique feature of our SGW and will be thoroughly discussed in Section 3.1.

As media transport uses same RTP/UDP/IP stack in both protocols, audio and video transfers are transparent to SGW. Interoperation task involves only issues of signaling and media description, permitting a single SGW to handle a large load of requests/responses without compromising performance. The most important interoperability issues are discussed next.

2.3.1 Call Establishment Procedures

Three pieces of information have to be exchanged to establish a call between the protocols: signaling destination address, remote or local media capabilities, and remote or local transport ports terminals will use for receiving media packets. Difficulty in translating these procedures between SIP and H,323 arises because SIP keeps the whole information in SIP INVITE and following responses, whereas H.323 spreads information over different signaling messages.

There are two actions to solve this problem. From SIP to H.323, information contained in SIP INVITE has to be mapped in parameters of H.225/H.245 signaling messages. In the opposite direction, from H.323 to SIP, the process is more complicate and H.225 and H..245 messages have to be previously collected in the H.323 BSM module. However, H.245 is only activated when the call has already been established through Q.931/H.225 procedures. Therefore, we have two alternatives: send an empty INVITE (without SDP and no media description); or use an OPTIONS SIP message to discover media capabilities of SIP client, and later pass this information to H.323 side and negotiate via H.245. See Section 3 about implementation.

2.3.2 Address Translation

Address translation has to be performed to identify the user. SGW has to create a valid SIP address from an H.323 address, and vice versa. In SIP, addresses are formed as URIs (as an HTTP address). For example, sip: user@host. SIP clients can support URL as "tel:xxx@" for telephone numbers or even H.323 URLs may exist‡. H.323 addresses are coded using ASN.1[2] (in SIP, both message coding and URI are based on text [3]) and may be non structured identifiers (H.323 ID), E.164 for telephone numbers, URLs of different types, host name or IP address, and e mail addresses (Email ID). Nevertheless, local user and host names seem to be the most used in H.323 address formation.

Mapping a SIP address to an H.323 address is easy, sufficing copying SIP URI to H.323 ID. The parts to mount an email ID are taken from SIP URI user@host format: transport ID is taken from host part of SIP URI, and E.164 field is extracted from SIP address user part, if it contains a telephone number, or from "tel: " scheme.

Mapping form H.323 to SIP is more difficult due to many different H.323 representations and translation to a singe SIP address. [13] recommends to first check if H.323 ID is a valid SIP address. If it is, it suffices to copy this address to SIP URI, otherwise it is necessary to form SIP URI host part from H.323 transport ID, when host part does not point to SGW itself. And email ID will be used to format the rest of SIP URI.

2.3.3 User Registration Architecture

To simplify user access to location services, SGW has to directly access user registration servers, independently of the signaling protocol. Figure 6 illustrates three ways to allow user registration mapping in a heterogeneous SIP/H.323 network [13].

First, SGW has an internal SIP Proxy/Registrar, and every time SIP Registrar receives a SIP REGISTER request from a SIP user, SGW generates a RRQ (Register Request) to H.323 Gatekeeper (GK), registering SIP users also into the H.323 network. As SIP information is made available to H.323 gatekeepers, then any H..323 entity can resolve the address of SIP users reachable through SGW. In SIP to H.323 direction, when a SIP user wants to talk to an H.323 users, he sends a SIP INVITE to SGW, and SGW H.323 BSM module sends LRQ (Location Request) to H.323 GKs. GK containing desired user registration responds with user

‡ URL H.323 was proposed by P. Cordell, "Conversational multimedia URLs", Internet Draft, 1997, expired.

IP. Main disadvantage with this approach is the fact that H.323 gatekeepers will have to maintain not only all H.323 addresses for H.323 network, but also all SIP addresses corresponding to SIP network. Larger storage affects scalability, but it speeds up H.323 to SIP communication.

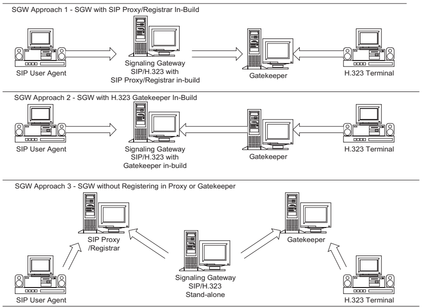


FIGURE 6 - USER REGISTRATION - SGW ARCHITECTURE APPROACHES

In a second approach, SGW contains an internal gatekeeper GSW-GK, which implies in a solution similar to the above, except that SIP Proxy/Registrar is the one who maintains, redundantly, information on users of both networks. When a register request is received by GSW-GK, request will be redirected to SIP Registrar. SIP users get a fast H.323 address resolution as H.323 registrations are in the same SIP domain. On the other hand, when an H.323 terminal wants to talk to a SIP user, it has to send ARQ to GK, asking for permission. GK will try to locate the registration in other GKs sending multicast LRQ, for example. When SGW-GK receives LRQ, SGW-GK will look for the user inside SIP network using the OPTIONS method to check user existence and status. User IP is later sent to H..323 network. SIP to H.323 communication will be speeded up. Scalability problems still persist.

Last approach does not keep redundancy information as in previous ones, but locating users takes more time. SGW does internally have neither H.323 gatekeeper nor Sip Proxy/Registrar. User registrations are kept independently by each network. When a user needs to be located, SGW will simply try to search in the other network. When an H.323 user tries to contact a SIP user, corresponding GK will not find SIP user address in its database and will send an LRQ to neighboring GKs including SGW. SGW then uses SIP OPTIONS method to locate user. If search starts from SIP side, SIP Proxy/Registrar will not find user register and an LRQ will be sent to all known GKs on H.323 side. We have opted for this last approach.

3. SIP/H.323 SIGNALING GATEWAY IMPLEMENTATION

In order to get a fast and distribution free implementation, we decide to use freeware and open code software. We used OpenH323 [19] and IPTele [11], for SIP stack. However, partially reusing existing code has its own problems. In the case of Open H323, which is maintained by an Australian company called Equivalence Pty, H.323 libraries are quite extensive , developed in C++, and with very little documentation. Nevertheless, OpenH323 implementations were quite stable and reliable. For SGW development, we decide to use the code of OpenGatekeeper [19], a gatekeeper implementation that uses OpenH323 libraries. Our goal was to take advantage of a server implementation capable of supporting multiple simultaneous connections and threads for efficient and scalable protocol processing.

Access to SIP User Agent IPTele was granted through a cooperation agreement between IM/NCE/UFRJ and Helsinki University of Technology (HUT), Finland. This SIP client was written in Java and has SIP User Agent Client and User Agent Server functionalities, besides a simple SIP message parser. This software has many limitations we discuss later.

To add minimum changes in the original codes and preserve programming environment, a two independent module implementation was decided, having BSM (Basic Signaling Messages) to perform the communication function between H.323-BSM and SIP-BSM modules. BSM operates over TCP, allowing separate physical deployment of modules if necessary. This distributed modular concept has other advantages. If a third implementation is available, say

MGCP BSM module, then combining the modules automatically gives us H.323 MGCP and SIP MGCP gateways.

Address translation problems (introduced in Section 2.3) were solved in the following way. First, form SIP to H.323 we used H.323 ID instead of email ID or URI ID. Reason is some gatekeepers, as OpenGatekeeper, make a byte to byte address comparison, and H.323 ID uses two bytes per character (multilingual support), differently form email ID and URI ID that use only one byte per character. As all tested H.323 clients (OpenPhone and Netmeeting) send their alias as H.323 ID, we decided to stick to their implementation for compatibility.

In registering users, we decided for approach three (Section 2.3.2) which causes minimum interference in existing SIP and H.323 networks and makes SGW completely transparent. Furthermore, differently from what is suggested in [14], where SGW would have to implement a full set of RAS messages, we use only LRQ/LRJ/LCF messages to search for neighbors gatekeepers. This approach reduces processing load in GKs, besides reducing search delay, as simultaneous communication occurs with neighboring GKs. We have also a reduction on exchanged number of H.323 location messages. This decision simplifies SGW implementation and contributes to performance.

For media capability exchange, when there is a call from H.323 to SIP, [14] proposes using OPTIONS to search for capabilities on SIP side and later sending an INVITE message, or sending INVITE with media capability supported by SIP, allowing the need for an eventual reINVITE. We decided for a different approach, sending an empty INVITE (without SDP, see Section 2.3.1). Therefore, connection can be initiated and H.245 can start capability exchange on H.323 side. SDP with negotiated media capabilities is only sent in the SIP ACK final message. We have identified two advantages on this implementation. First, there is no need for a reINVITE, in case initial sent media is wrong. Second, reduction in message number and time delay for opening a session.

3.1 BSM

BSM is a design option devised to unite, via TCP, public domain modules without the need for code rewriting. Furthermore, BSM solves media capability exchange problems. SIP and H.323 send media capabilities in different session stages: SIP during session establishment and H.323 after session establishment. As soon as BSM makes media information available to a module, media negotiation with corresponding network side is initiated. If media description is tardy, a BSM message can be sent asking the other GSW module to inform what media should be used. Another foreseen advantage is independent module operation. GSW modules can run in separate machines, allowing signaling processing load sharing.

3.1.1 BSM Messages

BSM has the following basic messages: SETUP (open connection), CONNECT (connection established), RINGING (connection alert), RELEASE (connection release), CAPABILITY ARRIVAL (media parameters arrival), and MESSAGE CAUSE (optional extensions).

SETUP has two parameters: origin and connection destination. CAPABILITY ARRIVAL has the following field parameters: "c" (transport address and type), "m" (media type and corresponding parameters), and "a" (auxiliary field, describing SDP header field "m" [2]. CONNECT informs about connection establishment, and RELEASE about its termination. When an IP telephony protocol has some extension, MESSAGE CAUSE is used to indicate the desired extension. The other messages have no parameters.

BSM connections are identified by a pair of numbers, generated by each module from a base number and exchanged when initial TCP session is established. The identifier generated by a module is called *topID*. The identifier sent by the corresponding remote module is called *topRemoteID*. All BSM messages carry an identifier called *messageID*. A BSM message received with *messageID > max (topRemoteID, topID)* does not refer to any open connection, but to some connection still to be opened. BSM connections allow a module to interact simultaneously with multiple remote modules. When sending a message referring to a new connection, *topID* has to be incremented by 100 and message sent with *messageID = topID*. This procedure keeps track of protocol status over its different threads.

3.1.2 BSM Message Detailed Actions

BSM fields are string type, with "new line" as separator. First byte indicates message type.

SETUP = 0, RELEASE = 1, RINGING = 2, CAUSE = 3, CAPABILITY = 4, CONNECT = 5

Setup	Message Description: <ID><to (url)><From (url)>(<field1><field2><field3>	
When to send:	• When SGW gets INVITE (SIP) or Setup (H323 v1,2,3). • When leaving call waiting state	
What to do when receiving:	For H.323 side:	Send H.323 Setup. Remember Faststart (sending all supported media types with SETUP) needs a CapabilityArrival message to arrive before. If capability message occurs before Setup, store data and wait for coming SETUP.
	For SIP side:	Send INVITE. Sending SDP depends on having received a CapabilityArrival message previously.

Release	Message Description: <ID><HOLD bit [0|1]> HOLD indicates call waiting.	
When to send::	• Receiving SIP BYE, or any message between 400 and 500, 600, 602 or 603. • Receiving H.245 Endsession in H.323 (v1) or Q.931 ReleaseComplete.	
What to do when receiving:	For H.323 side:	Send H.245 Endsession (v1), and if timeout occurs, send Q.931 ReleaseComplete.
	For SIP side:	Send BYE.

Ringing	Message Description: <ID>
When to send:	• Receiving RINGIG (SIP) or Alerting (do H.323 v1,2,3).

Cause	Message Description: <ID><cause> Follows same message numbering as SIP response messages.	
When to send:	• Receiving an indication of disconnection or setup resending.	
What to do when receiving:	For H.323 side:	If Release is later received, send motive.
	For SIP side:	If Release is later received, send motive.

Capability Arrival	Message Description: <ID><"c=IN IP4 %s\nm=audio %s RTP/AVP %*d\na=rtpmap:%*d %s\n\n"> First parameter is IPv4 address for RTP session. Second parameter refers to name of media code to be used. Third parameter refers to port where RTP packets are expected	
When to send:	• Receiving media capability from client (in INVITE with SDP, or in H.245).	
What to do when receiving:	For H.323 side:	Send capability using H.245 v1 (H.245 vs2 e 3 to be defined)
	For SIP side::	Insert SDP in corresponding INVITE, 200 or ACK.

Connect	Message Description: <ID>
When to send:	• Every time OK (200 SIP) or Connect (H.323) is received.

Same three classes have been implemented in both modules, allowing a simple inheritance for the core which handles BSM messages in each module. Classes are:

- Translator: responsible for sending BSM msgs to the other SGW side.
- RemoteTranslator: exchanges msgs between Translator instantiations (in each module)
- Msg: responsible for creating/interpreting sent/received Remote/Translator msgs.

More detailed information in SGW programming can be found in [23].

3.2 SIP-BSM Module Implementation

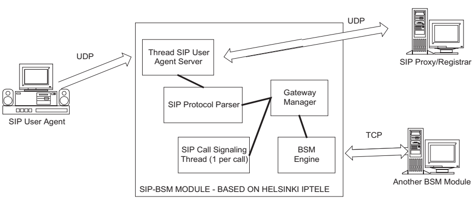


FIGURE 7 - SIP-BSM MODULE

SIP-BSM module was based on SIP User Agent, developed as a master thesis in Helsinki University of Technology [11]. However, because the client program was able to answer only one call at a time, major recoding was done to implement a server capable of processing simultaneous calls. New code uses Java threads and also implements new primitives to help interaction with other SIP Proxies. Client parser was modified to increase the number of messages that can be interpreted and sent. Part of original code dealing with media transport and using Java Media Framework (for media coding/decoding and transport via RTP) was eliminated, as SGW does not need to process media transport. As a result of recoding, a new SIP client is almost ready. Figure 7 shows SIP-BSM module internal structure. This module is written in Java, has 2437 lines of code, and only uses SUN libraries distributed with JRE (Java Runtime Environment).

3.3 H.323-BSM Module Implementation

This module was based on OpenGatekeeper [20], a GK server implementation from open code and complete H.323 protocol stack called OpenH323. Starting from an H.323 client would also have been a possible solution, but a GK server implementation has imbedded LRQ/LCF/LRJ messages support and signaling for a full H.323 terminal, features needed in GSW. Furthermore, OpenGatekeeper is capable of supporting simultaneous multiple signaling calls and threads. Some of the code changes introduced in OpenGatekeeper are:

- Q.931, H.225.0, and H.245 messages interpretation, creation, and parameters insertion. These are features required for an H.323 client and not present in a GK.
- Treating parameters that should be optional according to H.323 recommendation, but required for Microsoft Netmeeting interoperability.
- An Abstraction Layer class , which implements an interface between BSM and H.323 signaling, was created to allow different H.323 versions (e.g., v1 with H.245 and v2 with Faststart) to interoperate transparently with GSW.

Figure 8 shows the internals of SGW H.323-BSM module. Module is object oriented and written in C++. It has 51 classes scattered over 13 files, for a total of 9825 lines of code. Two open libraries were used: OpenH323 (873 classes) and PWLib (359 classes), both object oriented and written in C++.

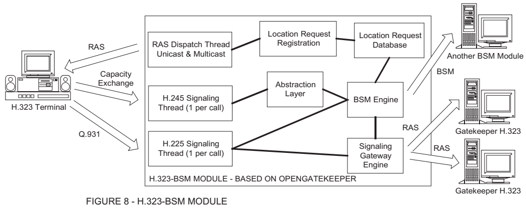


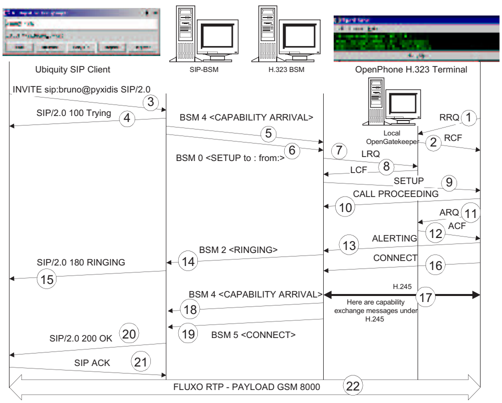
FIGURE 8 - H.323-BSM MODULE

4. INTEROPERABILITY TESTS

SIP-BSM was the first module to be developed. Initially, we tested signaling between two SIP-BSM modules, following scenarios proposed in [12]. After H.323-BSM module was ready, we tested interoperation between SIP clients and H.323 terminals, using Ubiquity SIP User Agent (UA) and OpenH323 OpenPhone terminal. Microsoft Netmeeting was also tested against Ubiquity SIP UA. In this latter test we were unable to close RTP session because Ubiquity SIP UA uses an exclusive media coding not supported by Netmeeting. Figure 9 shows a thorough test over an architecture formed by 2 clients (one H.323 and one SIP), a SIP/H.323 SGW spread over two computers, and a GK. Follow explanations below.

(1) H.323 terminal (OpenPhone) registers in its administrative domain Opengatekeeper via RRQ, and **(2)** receives RCF confirmation. **(3)** Later, SIP side starts a call, with Ubiquity SIP client sending INVITE to some H.323 user via SIP-BSM module (in this test we are not redirecting via Proxy/Registrar). **(4)** SIP-BSM returns TRYING, a temporary response, and sends BSM msgs to H.323-BSM. **(5)** CAPABILITY ARRIVAL msg is sent and INVITE SDP is stored in SGW for future comparison with H.245. **(6)** BSM SETUP (with origin and destination addresses) is sent to allow mounting H.323 connection messages). **(7)** SGW sends LRQ to GK to search for user. **(8)** GK finds user (LCF) and delivers its IP to SGW. **(9)** SGW then sends Q.931 SETUP request to OpenPhone. **(10)** OpenPhone automatically returns Call Proceeding**.** **(11)** OpenPhone sends ARQ to GK, asking for authorization. **(12)** GK authorizes call (ACF)**.** **(13)** OpenPhone alerts ringing. **(14)** H.323-BSM module forwards call ringing sending BSM RINGING msg. **(15)** In SIP side, call ringing becomes 180 RINGING, meaning telephone is ringing but user has not answered yet. **(16)** When user answers the call, a Q.931 CONNECT is sent to H.323-BSM module. **(17)** H.323-BSM module starts H.245 procedures, negotiating media set with H.323 terminal (OpenPhone), using information received in first SDP. **(18)** H.323-BSM module sends CAPABILITY ARRIVAL msg (containing negotiated capabilities) to SIP-BSM module, and **(19)** BSM CONNECT to confirm call. **(20)** SIP-BSM module sends 200 OK msg with negotiated medias (subset from first INVITE). **(21)** SIP client signals with ACK. **(22)** GSM with 8000 bits was the negotiated media.

FIGURE 9 - CALL PROCEEDING FLOW USING SIP/H.323 SIGNALING GATEWAY



5. CONCLUSIONS

In this paper we present a modular implementation of a SIP/H.323 signaling gateway (SGW), with partial reuse of code from OpenGatekeeper (OpenH323) and SIP IPTele client (Helsinki University of Technology). SGW uses a set of basic signaling messages (BSM) to implement communication between modules over TCP, allowing modules to run in separate machines and perform load sharing. Furthermore, each module implementation is completely independent form other modules, enabling module combination to realize new SGWs. Design decisions were focused in supporting many simultaneous calls and high performance operation. Interoperability tests involving Microsoft Netmeeting, Ubiquity SIP client and H.323 OpenPhone terminal were conducted, demonstrating SGW effectiveness, righteous operation and complete adherence to standards. Building a SGW module is not only a matter of an efficient message mapping, but also a way to improve flexibility allowing compatibility with a larger possible number of clients. We were forced to handle optional parameters and minimize the chance of signaling timeouts.

There is space for SGW improvement. We are studying a new BSM formulation to allow increased flexibility with greater implementation simplicity. Scalability tests have still to be performed for high signaling load scenarios. Restricting user search to a single administrative domain is a great limitation in H.323, and it does not help Internet scalability. A location server based on DNS could bring greater coverage for H.323. Long response time to locate users is a negative factor for IP telephony. Cache solutions implemented as extensions to clients or location servers could improve response time. In mobile IP environments, user registration deserves a more some automated design and the effectiveness of using soft states instead of hard states should be investigated. At last, BSM simplicity and SGW modular architecture invites the realization of modules as BSM MGCP or BSM SS7 to combine with existing ones and deploy new gateways.

6. REFERENCES

[1] Christian Huitema, et al. **An Architecture for Residential Internet Telephony Service**. IEEE Network Magazine. May/June, 1999. Vol.13. No. 3 .

[2] ITU T Recommendation H.323. **Packet Based Multimedia Communications Systems**, September, 1999.

[3] M. Handley, et al. **IETF RFC 2543 – SIP : Session Initiation Protocol**, July, 1998.

[4] M. Handley and V. Jacobson, "**SDP: Session Description Protocol,** "Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force, April, 1998.

[5] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "**RTP: A Transport Protocol for Real Time Applications**," RFC (Proposed Standard) 1889, Internet Engineering Task Force, January., 1996.

[6] International Telecommunication Union, "**Control Protocol For Multimedia Communication**," Recommendation H.245, ITU T, Geneva, Switzerland, February. 1998.

[7] International Telecommunication Union, "**Media Stream Packetization And Synchronization On Non Guaranteed Quality Of Service Lans**," Recomm. H.225.0, ITU T, Geneva, Switzerland, Nov., 1996.

[8] H. Schulzrinne, J. Rosenberg. **Internet Telephony: Architecture and Protocols**. Computer Networks, February, 1999. Vol. 31, no. 3.

[9] H. Schulzrinne, J. Rosenberg. **A Comparison of SIP and H.323 for Internet Telephony**. Network and Operating System Support for Digital Audio and Video (NOSSDAV), Cambridge, England, July, 1998.

[10] M. Arango, et al. **IETF RFC 2705 – MGCP : Media Gateway Control Protocol**. October, 1999.

[11] Inmaculada Espigares del Pozo. **An Implementation Of The Internet Call Waiting Service Using SIP**. Master Thesis. Helsinki University of Technology and Politecnica de Valencia. December, 1999.

[12] J. Rosenberg, H. Schulzrinne. **Interoperability Testing for SIP**. April, 1999. Columbia University, N.Y.

[13] K. Singh, H.Schulzrinne. **Interworking Between SIP/SDP and H.323**. Proceedings of 1st IP Telephony Workshop (IPTel'2000), Berlin, Germany, April, 2000.

[14] K. Singh, H. Schulzrinne. **Interworking Between SIP/SDP and H.323**. IETF Draft., on going.

[15] Bill Douskalis. **IP Telephony The Integration of Robust VOIP Services.** Hewlett Packard Professional Books, Prentice Hall, Inc., 2000.

[16] Kundan Singh, Henning Schulzrinne. **Unified Messaging using SIP and RTSP**. IP Telecom Services Workshop, Atlanta, Georgia, USA, September, 2000.

[17] **Succession Solutions Integration of Session Initiation Protocol.** Nortel Networks. Technical Brief. September, 2000.

[18] M. Wahl, S. Kille e T. Howes. **IETF RFC 2251 – Lightweight Directory Access Protocol (v3)**.Dec.,1997.

[19] Site do OpenGatekeeper **http://www.opengatekeeper.org**

[20] A. Vemuri. **Carrier Use of SIP**. Level 3 Communications. Presentation in Voice Over Networks Conference (VON). September, 1999.

[21] T. Hastings. **IP Communications Services Trial**. MCIWORLDCOM. Presentation in Voice Over Networks Conference (VON). September, 1999.

[22] Bruno F. M. Ribeiro. **Implementação de Gateway de Sinalização entre Protocolos de Telefonia IP SIP/H.323.** Final Graduate Report, Comp. Science Dept., Federal University of Rio de Janeiro, Feb., 2001.

[23] Ackermann R., et al. **An Open Source H.323 SIP Gateway as Basis for Supplementary Service Interworking**. 2nd IP Telephony Workshop (IPTel'2001), New York, May, 2001.