AN ANALYSIS OF IP TELEPHONY SIGNALING USING THE SESSION INITIATION PROTOCOL (SIP)

by
James R. Alumbaugh

A Master's paper submitted to the faculty of the School of Information and Library Science of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in Information Science.

Chapel Hill, North Carolina

April, 2000

Approved by:

_____

Advisor

1 INTRODUCTION

IP (Internet Protocol) telephony is emerging as a popular and bleeding-edge technology for different groups in the data and telecommunications sectors. For those that engineer, operate, and are responsible for the day-to-day maintenance of both public and private data and telecommunications networks, IP telephony (also referred to as Voice over IP, or VoIP) networks represent a cheaper and more efficient solution on which to transport voice telephony traffic. Likewise, data and telecommunications hardware and software vendors are rushing to capture portions of this potentially lucrative market by investing large amounts of capital into the production of IP telephony products. A new signaling protocol, called Session Initiation Protocol (SIP), is beginning to gain momentum in this market space as an alternative for providing signaling services for real-time media such as voice and video over IP packet-switched networks.

1.1 Technical Advantages of IP Telephony

For the providers of telecommunications network services, IP telephony simply provides a more resource efficient and cheaper alternative to traditional time-division multiplexed (TDM) circuit-switched networks. Circuit-switched networks allocate a entire 56kb or 64kb circuit for each and every telephony session (phone call); packet-switched IP networks make use of more efficient multiplexing and

compression mechanisms so that the bandwidth allocated per call can be reduced to as low as 5kb to 8kb per call. VoIP technologies employ silence suppression techniques that prohibit the transmission of empty packets onto the network resulting from natural gaps or silence in conversation, which frees up bandwidth and resources for other services. In addition, these packet-switched networks can provide dual functionality by offering data as well telephony services over the same IP infrastructure. Typically, public switched telephone network (PSTN) switches are large in physical size, proprietary in nature, and are not necessarily interoperable with other vendors' switches. These switches are also far more expensive than the equivalent smaller IP telephony devices; the latter take up less physical space and mostly use standard interoperable protocols. These advantages effectively enable telecommunications service providers to do more for less.

1.2 Cost Advantages and the Market Potential of IP Telephony

The cost benefits of implementing and offering VoIP services can be significant. For those who maintain private networks, e.g. corporate or enterprise networks, the implementation of telephony services over their existing IP infrastructures can substantially reduce the costs of providing intra-company telephone services. For example, consider a company with five U.S. locations, each with 2,500 end-users who on average make fifteen minutes of intra-company calls per day at six cents per minute; the cost per month for just these intra-company calls would be $45,000 per location, or $225,000 per month. The cost to implement VoIP services at each site would be approximately $60,000, for a total of $300,000 for all

six sites. The cost per intra-company VoIP call would be approximately one to two cents per minute, for a savings of at least four cents per minute. In order to recoup the $60,000 invested per site, users at each site must make 1.5 million minutes of calls (1.5 million minutes multiplied by 4 cents). Users on average are making 750,000 minutes of calls per month (2,500 users multiplied by 15 minutes), therefore this VoIP implementation would pay for itself in two months. (Friedrichs, 1998)

For providers of public telephony services, the cost savings can be even more dramatic. A call from Brazil to the United States might normally cost a provider 20 cents per minute to transport, while the same call placed over an IP telephony network would cost only two and a half cents per minute (Henderson, 1999b). IP telephony services have also become popular on an international scale in certain emerging and less developed regions where international calls are very expensive; a significant percentage of this expense results from regulatory taxes imposed on long-distance voice traffic (Dalgic & Fang, 1999). The IP networks in regions such as the Middle East, Eastern Europe, and Africa can be unregulated which enables providers to evade accounting rate settlements and provide cheap and sometimes higher quality services than existing circuit-switched networks. Finally, International Data Corporation expects the use of IP telephony to grow from 2.7 billion minutes by the end of 1999 to 135 billion minutes by 2004. The market for IP telephony services is predicted to grow from $480 million in 1999 to $19 billion by 2004. (Henderson, 1999a) (Henderson, 1999b)

IP telephony signaling provides the means for call setup and teardown, call control and services, and call capability exchange. Signaling will play a crucial role

in any success that IP telephony achieves within the next few years. Currently, the two most popular signaling protocols available for use in VoIP networks are the Session Initiation Protocol, a Internet Engineering Task Force (IETF) standard, and H.323 (International Telecommunications Union [ITU], 1998a), which is an International Telecommunications Union (ITU) standard. H.323 currently enjoys leadership over SIP in terms of current live deployment. However, SIP is threatening to challenge H.323 in popularity due to its "simplicity, scalability, extensibility, and modularity" (Schulzrinne & Rosenberg, 1998c, p. 144).

2 THE SESSION INITIATION PROTOCOL

Before we begin a more detailed description of SIP functionality, it is important to define IP telephony signaling and differentiate it from the transport or media connection functionality of IP telephony sessions. IP telephony can be defined as "synchronous voice or multimedia communication between two or more parties", such as two- or multi-party phone calls and multimedia conferences, and "requires a means for prospective communications partners to find each other and to signal to the other party their desire to communicate". Functions that signaling protocols are responsible for include *name translation and user location, user availability, feature negotiation* or *capabilities exchange, call participation management* (including call setup and teardown), and *feature changes*. (Schulzrinne & Rosenberg, 1998a, p. 2)

Name translation and user location entails determination of the end system to be used for communication and associating between names of different levels of

abstraction, of which Domain Name System (DNS) (Mockapetris, 1987) name server queries are an example. The support of name translation and redirection services also enables the implementation of personal mobility features via Integrated Services Digital Network (ISDN) and Intelligent Network (IN) telephony subscriber services. Personal mobility is defined as " the ability of end users to originate and receive calls and access subscribed telecommunication services on any terminal in any location, and the ability of the network to identify end users as they move". *User availability* determines the willingness of the called party to engage in communication. *Feature negotiation* or *capabilities exchange* involves end systems agreeing on which types of media to exchange and what parameters are to be used, such as compression, etc. *Call participation management* can include not only call setup and teardown, but call transfer, call hold, whether call invitations will be multicast or unicast, etc. *Feature changes* allow telephony session participants to "adjust the composition of media sessions during the course of a call", such as adding video or whiteboard capabilities during the active session. (Schulzrinne & Rosenberg, 1998a, p. 2) (Handley, Schulzrinne, Scholler, & Rosenberg, 1999, p. 6)

As stated previously, IP telephony signaling is engineered as a separate and distinct function within the network. This allows for far greater flexibility when engineering other separate functions of the network. For example, in circuit-switched networks "the SS7 [Signaling System 7] telephony signaling protocol encompasses routing, resource reservation, call admission, address translation, call establishment, call management, and billing". In an IP network, many of these other functions are controlled by separate protocols. This allows the network architect to

design the network by layering different protocols onto other protocols depending on the needs or requirements of the network and users. An example of possible protocols and their layering is included in Figure 1. (Schulzrinne & Rosenberg, 1998a, p. 2-3)

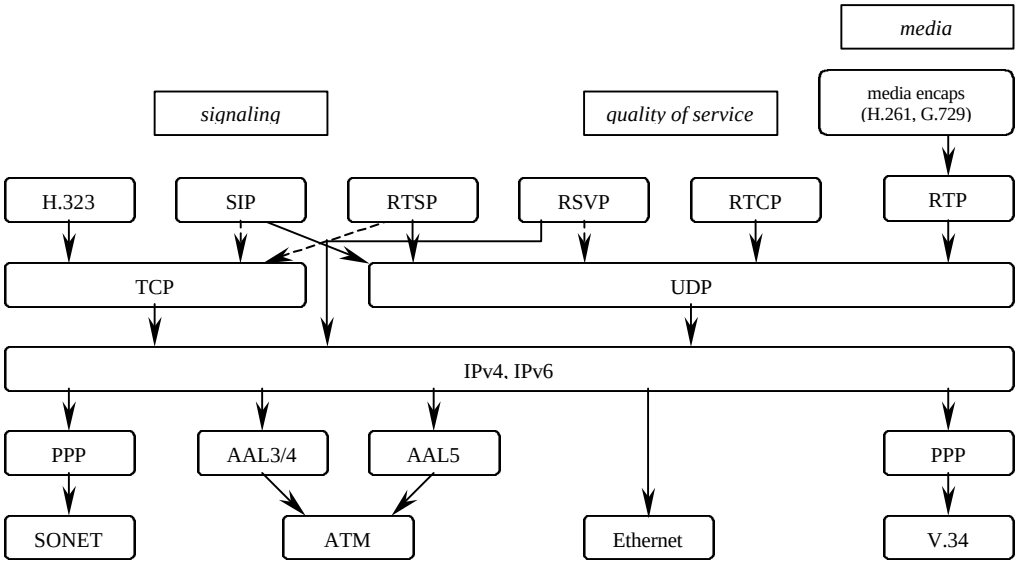

Figure 1: Possible protocol architecture and layering for IP telephony services (Schulzrinne & Rosenberg, 1998a, p. 3)

SIP is an application layer, text-based, and client-server protocol where requests are sent by the client and responses to these requests are returned by the server, and it is modeled after the simple mail transfer protocol (SMTP) (Postel, 1982) and the hypertext transfer protocol (HTTP) (Fielding, Gettys, Mogul, Nielsen, Berners-Lee, 1997). SIP actually "reuses much of the syntax and semantics of HTTP, including its response code architecture, many message headers, and its overall operation" (Schulzrinne & Rosenberg, 1998c, p. 146). In addition, as Figure

1 demonstrates, SIP can use either TCP (transmission control protocol) or UDP (user datagram protocol) as its lower-level transport protocol. By utilizing UDP, SIP is able to use and take advantage of multicast functionality for tasks such as group invitations. The use of UDP also brings with it certain performance advantages; TCP requires that each server or client must keep state for the duration of a particular communication session. Also, with TCP multiple messages are required to synchronize the two endpoints, whereas UDP, though inherently unreliable, does not require this kind of synchronization and is faster in the call setup phase. These latter two performance advantages allow SIP servers to scale to accommodate larger numbers of users and sessions.

2.1 SIP Components

A SIP implementation has essentially two components, a user agent (UA) and a network server of some type. A user agent resides at SIP end stations and typically contains two components, a user agent client (UAC) and a user agent server (UAS). The UAC initiates SIP requests while the UAS responds to said requests; or to state more plainly, the UAC makes the phone call and the UAS answers the call. There are three varieties of network servers: redirect, registrar and proxy. Different SIP implementations can utilize different combinations of these servers; it is not necessary to use all three servers within a SIP network. In addition, multiple SIP server types can reside on a single physical hardware platform. Finally, simple SIP call functionality can be attained without the use of any network servers at all.

However, the more powerful features of SIP are reliant upon the utilization of these different servers (Dalgic & Fang, 1999).

Redirect servers respond to requests or call setups but do not forward them to the client. It responds back to the calling client with the called SIP end stations address or the next hop servers address so that the calling client can contact the called end station or next hop server directly. Registrar servers store addresses and the associated IP addresses for UACs, which forward this information to the registrar servers when they first boot or initialize. The registrar server then stores this information, which can be accessed and used by a proxy or redirect server co-located on the same physical platform to forward call setup requests to the appropriate location. Proxy servers behave similar to HTTP proxy servers; they perform application routing of SIP requests and responses. Proxy servers receive requests and then forward these requests toward the current location of the called SIP end station. The next hop that it forwards the request to may be a UAS, a redirect server, or another proxy server. Proxy servers can also *fork* incoming requests or call setups if it believes that there may either be multiple possible next hops to the destination or the called party may be currently located at one of multiple locations. Therefore as an example, if a proxy server knew of three possible next hops that could be used to route a setup toward a called party, it could fork a single incoming request for this party into three individual requests which would be forwarded down to each different next hop. Likewise, a forking proxy could ring two different phones in search of a called party by utilizing the forking capability inherent within SIP proxy design. Rules exist for how subsequent responses emanating from these requests are

merged and returned to back to the UAC. (Dalgic & Fang, 1999) (Schulzrinne & Rosenberg, 1998c, p. 146-147)

2.2 SIP Proxy Server Stateful and Stateless Operation

Different messages from individual SIP sessions can take different routes through the network. Individual proxy or redirect servers need not, in most cases, process all requests and responses for a particular SIP session. This is due to the ability of SIP network servers to operate in a *stateless* fashion, i.e., they need not maintain the call state once a particular transaction is complete or message has been forwarded. The notion of stateless servers contributes to SIPs reliability because if a stateless server failed, it would not have any effect on any currently active calls whose setup request messages were processed by the failed server; any subsequent messages relevant to those particular sessions would simply be routed through some other currently functioning proxy or redirect server in the network (Schulzrinne & Rosenberg, 1998c, p. 147). This is similar to how next-hop routing is conducted in IP networks. A stateless proxy or redirect server is also capable of scaling far greater than a stateful server; maintaining call states for all sessions whose messages have been processed by a particular server requires significant resources and can potentially inhibit the performance of the server.

However, depending on the functionality required, servers may be stateful if necessary. RFC 2543 recommends that forking proxies be stateful so that responses from multiple call setup requests can be merged and the ensuing multiple active sessions can be maintained or torn down if necessary. RFC 2543 also states that

proxies that accept TCP as means for SIP signaling transport must be stateful because if a stateless “proxy were to lose a request, the TCP client would never retransmit it”. (Handley et al, p. 97)

2.3 SIP Addressing

SIP uses the most common method of addressing requests in the Internet by addressing all requests to users at hosts, i.e. “user@host”. This is often referred to as a SIP URL (uniform resource locator) (Berners-Lee, Masinter, & McCahill, 1994) and takes the form of “sip:user@host”, “sip:user@domain”, or “sip:phone-number@gateway”. The user portion of the address can either be a user name or a telephone number while the host portion can be either a domain name or a numeric network address, such as an IP address. The domain name can be “either the name of the host that a user is logged in at the time, an email address or the name of a domain-specific name translation service” (Schulzrinne & Rosenberg, 1998a, p. 6). According to RFC 2543, “a user’s SIP address can be obtained out-of-band, can be learned via existing media agents, can be included in some mailers’ message headers, or can be recorded during previous invitation interactions” (Handley et al, p. 11). Similar to the use of the “mailto:” tag (Hoffman, Masinter, & Zawinski, 1998), SIP addresses can also be embedded within web pages in the form of “sip://user@domain.com”, which can then be clicked by a user to place a SIP call to the specified address (using a browser that supports SIP URLs).

2.4 SIP Messages

There are two basic types of SIP messages, requests issued by a client to a server and responses sent by server to a client, and these messages contain different headers depending on the message type or information that is to be transported. As stated earlier, SIP is a text-based protocol and uses the ISO 10646 character set in Universal Character Set Transformation format 8 (UTF-8) (Yergeau, 1996) encoding. A significant portion of the message syntax and header fields are identical to those used in HTTP version 1.1 (Handley et al, p. 24). Because text-based protocols can be difficult to parse due to irregular structure, SIP has been designed with a common structure for all messages and header fields; this allows use of a more generic parser (Fingal & Gustavsson, 1999, p. 16). Request and response messages consist of start-line, one or more headers, an empty line which indicates the end of header fields, and an optional message body.

2.4.1 Header Fields

Header fields contain important specifics and parameters about the telephony session, such as subject, calling party, called party, length of message body, etc. SIP defines four different groups of headers. *General header fields* apply to both request and response messages. *Entity header fields* define meta-information about the message body, and if a message body is absent, then this header contains information about the resources identified by the request. *Request header fields* are a mechanism which allows the client to send additional information about itself and

the request to the server. *Response header fields* allow a server to pass additional information within the response message which cannot be placed in the response Status-Line. Header fields are listed according to relevant group in Figure 2.

Three more important header fields are the Via header, the Route header, and the Record-Route header. The Via header field indicates the path that the request or response message has taken so far and is used to prevent looping. Each SIP server, or hop, inserts a Via header with its own address into the message and if a server processes a message which already contains a Via header field with its own address, an error message is generated back to the previous sender. The Route and Record-Route header fields can be used by proxy servers to ensure that they are included on the signaling path for any subsequent transactions of a particular telephony session. A proxy server will insert its Request URI (uniform resource identifier) (Berners-Lee, Fielding, & Masinter, 1998) into the Record-Route header field when it wishes to be involved in the path for any future messages of a particular call. (The Request-URI is simply a SIP URL. It typically indicates the user to which the request is addressed, but it differs from the To: field in that it may be overwritten by proxy servers.) Conversely, the Route header field determines the route taken by a particular request. Each host removes the first Route header field entry and then proxies the request to the host listed in that entry.

General-header	entity-header	request-header	Response-header
Accept	Content-Encoding	Authorization	Allow
Accept-Encoding	Content-Length	Contact	Proxy-Authenticate
Accept-Language	Content-Type	Hide	Retry-After
Call-ID		Max-Forwards	Server
Contact		Organization	Unsupported
CSeq		Priority	Warning
Date		Proxy-Authorization	WWW-Authenticate
Encryption		Proxy-Require	
Expires		Route	
From		Require	
Record-Route		Response-Key	
Timestamp		Subject	
To		User-Agent	
Via			

Figure 2: SIP header fields (Handley et al, p. 26).

2.4.2 Request Messages

Request messages begin with what is referred to as a method token which is followed by a Request-URI and the SIP version. There are six different request types, or methods: INVITE, ACK, OPTIONS, BYE, CANCEL, and REGISTER. Methods that are not supported by a proxy or redirect server are regarded as an OPTIONS method and forwarded as such. The INVITE method specifies the called SIP party is being invited to participate in a session. The INVITE message body contains a description of the session to which the called party is invited. "For two-party calls, the caller indicates the type of media it is able to receive and possibly the media it is willing to send as well as their parameters such as network destination. A success response indicates in its message body which media the callee wishes to receive and MAY indicate the media the callee is going to send" (Handley et al, p. 27).

Figure 3 provides an example of an INVITE message. The first line indicates that the message is an INVITE and includes the Request-URI of the called party and the SIP version used. The next two fields are Via fields and they list the hosts that processed the request along the path from the calling endpoint to the called endpoint. The first Via field indicates that the request was last multicast by the host csvax.cs.caltech.edu to the 239.128.16.254 group with a time-to-live (ttl) of 16. The From: and To: fields are fairly self-explanatory, although it is important to note that the Request-URI in the To: field is more generic than the Request-URI found on the first line; this indicates that the last proxy which processed the request did a lookup on the address and found a more specific hostname for which to send the request. The Call-ID is a unique number generated by the calling party and must remain unique to that particular call. The CSeq contains the request method type and a sequence number for that method within the context of the session. The Content-Type header states that the Session Description Protocol (SDP) is indicating the content or session description. The header is terminated with an empty line and a new message body indicates the start of the session description, which will be explained in section 2.7.

```
INVITE sip:schooler@cs.caltech.edu SIP/2.0
Via: SIP/2.0/UDP csvax.cs.Caltech.edu;branch=8348
   ;maddr=239.128.16.254;ttl=16
Via: SIP/2.0/UDP north.east.isi.edu
From: Mark Handley <sip:mjh@isi.edu>
To: Eve Schooler <sip:schooler@caltech.edu>
Call-ID: 2963313058@north.east.isi.edu
CSeq: 1 INVITE
Subject: SIP will be discussed, too
Content-Type: application/sdp
Content-Length: 187

v=0
o=user1 53655765 2353687637 IN IP4 128.3.4.5
s=Mbone Audio
i=Discussion of Mbone Engineering Issues
e=mbone@somewhere.com
c=IN IP4 224.2.0.1/127
t=0 0
m=audio 3456 RTP/AVP 0
```

Figure 3: Example of SIP INVITE request.
Note method type on first line. (Handley et al, p. 120)

The ACK method is confirmation that the client has received a final response to an INVITE request, such as a 200 OK (ACK methods are only used with the INVITE request process). The ACK request may include the final session description to be used by the called party within the message body; if this is empty then the called party uses the session description in the previously sent INVITE request. The OPTIONS method indicates a server is being queried for its capabilities; however, it does not set up any connection. UAS are the only servers that respond to such methods – proxy and redirect servers just forward these requests without indicating capabilities. BYE methods signify that the client agent wishes to inform the server that it wants to release the call. This method may be sent by calling or called party. RFC 2543 states that a session participant should issue a BYE request before releasing a call. Likewise, all parties that receive a BYE request

must subsequently cease transmitting media streams to the originator of the BYE request.

The CANCEL request method cancels a pending request message that contains the same Call-ID, To, From, and CSeq headers; however, this does not affect a completed request. "A request is considered completed if the server has returned a final status response" (Handley et al, p. 29). User agents, clients or proxies may issue a CANCEL request at any time. The CANCEL is typically used when call setups have been forked to different destinations. If one destination answers the call before the other called destinations, the proxy server may send a CANCEL to the remaining parties that have not yet responded to the setup. Finally, the REGISTER method is used by a client to register its current location with a SIP registrar server. Typically, the UA might register on startup with a local server by sending a REGISTER request to a well-known multicast address. Otherwise, the UA may be hard-coded with an IP address of a registrar server to which it sends a REGISTER message upon startup.

2.4.3 Response Messages

The called party responds with a SIP response message after it receives and processes a request message. There are six main classes of responses with multiple possible responses within each class. Each class is represented by a Status-Code, in which the first digit defines the category of response. (Response Status-Codes are listed in Appendix B.) Informational class status codes are defined as provisional, meaning that the code indicates progress of some kind but does not indicate the

termination of the request. Success, Redirection, Client-Error, Server-Error, and Global-Failure response classes are defined as final, meaning that a SIP request is terminated by the response (Fingal & Gustavsson, p. 19). RFC 2543 states that "SIP applications are not required to understand the meaning of all registered response codes, though such understanding is obviously desirable. However, applications MUST understand the class of any response code, as indicated by the first digit …" (Handley et al, p. 37). One of the more common response codes is 200 OK which indicates the success of a previous request, such as an INVITE.

An example of a SIP 200 OK response is provided in Figure 4. The first line indicates that the SIP version is 2.0 and the response is a 200 OK. The Via headers are taken from the original INVITE message and then removed hop by hop as the response works its way back to the calling party. The From:, To:, Call-ID:, and CSeq: fields remain as they were in the original request message. The Contact: field provides details of where the called user was actually located. However, this field may instead include a proxy contact point that must be reachable by the calling party (Handley et al, p 121).

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP csvax.cs.Caltech.edu;branch=8348
   ;maddr=239.128.16.254;ttl=16
Via: SIP/2.0/UDP north.east.isi.edu
From: Mark Handley <sip:mjh@isi.edu>
To: Eve Schooler <sip:schooler@caltech.edu> ; tag=9883472
Call-ID: 2963313058@north.east.isi.edu
CSeq: 1 INVITE
Contact: sip:es@jove.cs.caltech.edu
```

Figure 4: Example of SIP 200 OK response (Handley et al, p. 122).

2.5 Basic SIP Call Setup and Tear Down

Basic SIP call setup and tear down are illustrated in the simple call flow given in Figure 5. User A and User B both register with the registrar server (coexisting with the proxy server on the same platform) and their registration requests are acknowledged. User A tries to initiate a session with User B by sending an INVITE to the proxy server that has a final destination of User B. The INVITE is received by the proxy and then forwarded along to User B. Immediately after forwarding the INVITE, the proxy sends an informational response message indicating it is trying User B. User B then sends an alerting back to the proxy by issuing a "180 ringing" informational response message, which the proxy then forwards back to User A. User B answers the call, which initiates a 200 OK being sent back to User A via the proxy. This message is acknowledged by User A, again via the proxy. At this point, the two-way media stream is established, via the real-time transmission protocol (RTP) (Schulzrinne, Casner, Frederick, & Jacobson, 1996) or some other protocol. At some point, User A decides to hang up the call by

sending a BYE (which is propagated back to User B by the proxy) who then sends back a 200 OK success response message to terminate the session.

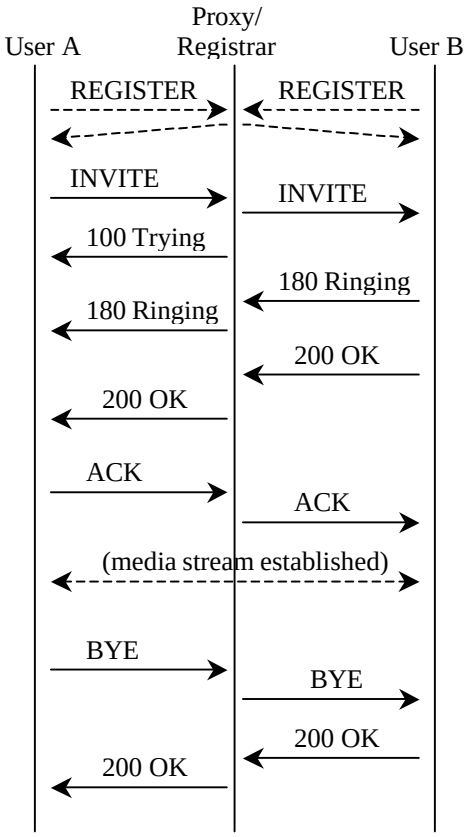

Figure 5: Basic SIP Call Setup and Tear Down (Sparks et al, 1999).

2.6 Advanced Operation

Schulzrinne and Rosenberg (1998c) provide an example which illustrates some of the more advanced personal mobility functionality inherent within SIP. This is depicted in Figure 6. User B works at Lucent and also inhabits an office and

lab space at Columbia University. Despite his multiple work locations, he publishes and gives only one IP telephony address out to others: “userB@lucent.com”. When working from Columbia, User B sends a REGISTER message upon startup to the Lucent SIP registrar server (1) using the address “userB@columbia.edu” as a forwarding addresses. He also registers his lab portable computer, “userB@lab.columbia.edu” (2), and his office machine, “userB@office.columbia.edu”, with the Columbia SIP registrar server (3). In addition, when previously at Lucent, User B had configured his lab portable computer to automatically forward calls to his Lucent address. Not remembering this configuration, User B retains this older configuration when he starts the SIP user agent on the portable in the Columbia lab.

At some point, User A (userA@att.com) makes a call to userB@lucent.com; the address “lucent.com” is resolved using DNS to the address of the Lucent SIP server. The Lucent SIP server receives the INVITE setup (4) and references its registration database (5), and based upon this information chooses to forward the INVITE to userB@columbia.com (after resolving columbia.com in DNS to the Columbia SIP server [6] address). When the INVITE arrives at the Columbia SIP server, the server looks up userB@columbia.com in its registration database (7). Since it can contact User B at one of two addresses, the server forks and forwards the call setup to both lab and office addresses (8, 9); at this point, the office machine rings but, due to the outdated configuration on the portable machine, the portable machine forwards the INVITE back to the Lucent SIP server (10). However, because of SIPs previously mentioned ability to detect loops using the Via header

field, the Lucent server identifies there is an error and returns an error response to the portable (11), which consequently sends an error to the Columbia SIP server (12).

User B answers the office machine which sends a 200 OK message back to the Columbia SIP server (13). The server propagates the message back to the Lucent server (14), who sends it back to User A (15). If so desired, all call states can be destroyed at this point since SIP servers can operate in stateful or stateless mode; any future transactions related to this particular telephony session may bypass the SIP servers and be processed directly between User B and User A (16). This example demonstrates four powerful capabilities of SIP: 1) how a INVITE request can be used to effectively track down a user by traversing multiple SIP servers, 2) the ability to detect and prevent loops, 3) the ability to fork requests so that the called party can be contacted more rapidly, and 4) how a SIP server can shift from stateful to stateless operation within the same telephony session. (Schulzrinne & Rosenberg, 1998c, p. 151)

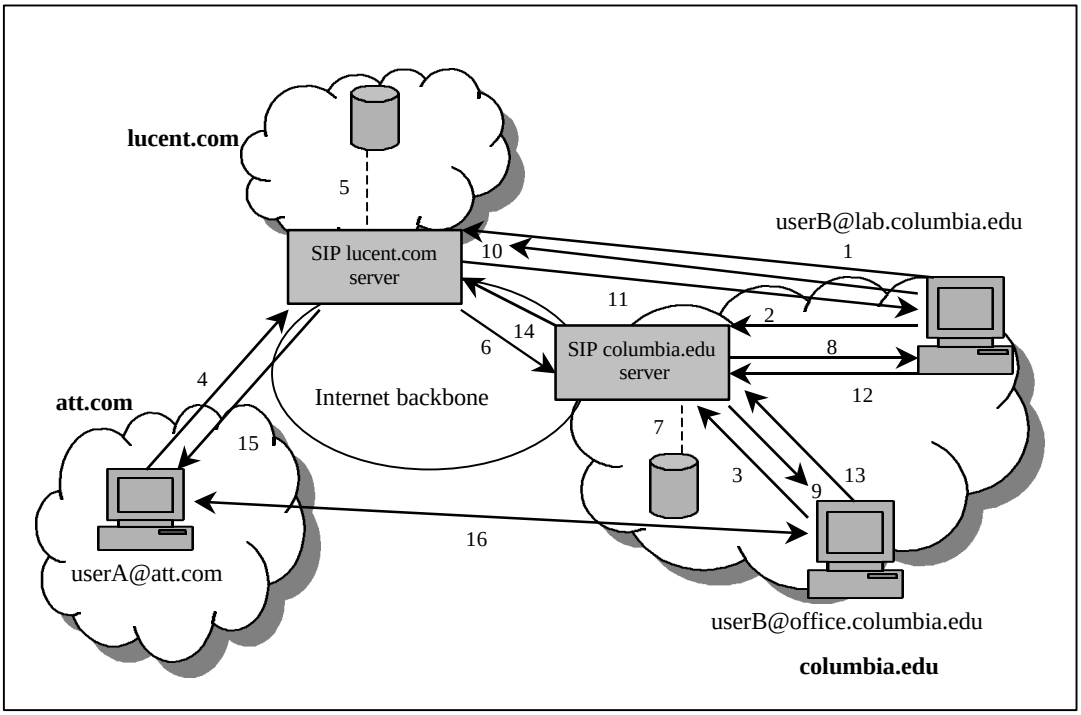

Figure 6: Example of advanced SIP personal mobility services (Schulzrinne & Rosenberg, 1998c, p. 152)

2.7 The Session Description Protocol

The Session Description Protocol is used to describe the multimedia session within the SIP request. SDP, as defined within RFC 2327, is intended "for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia initiation"; the latter three tasks would be performed by another protocol such as SIP (Handley & Jacobson, 1998, p. 1). SDP is specifically used to convey information about media streams in multimedia sessions so to allow the recipients of a session description to participate in a session (Handley & Jacobson, 1998, p. 3). However, it is important to note that SDP is not

intended for negotiating media encodings, rather it simply describes them for a particular session.

SDP includes 1) session name and purpose, 2) time the session is active, 3) the media comprising the session and, 4) information about entities that will be receiving the media in question (such as addresses, ports, etc.). In addition, SDP may also include information regarding bandwidth for the session and contact information for the person responsible for the session. In terms of information related to session media, SDP conveys information about the type of media (video, audio, etc.), the transport protocol being used (RTP/UDP/IP, H.320), and the format of the media (H.261 video, MPEG). (Handley & Jacobson, p. 3-4)

Like SIP, SDP session descriptions are textual based and use the ISO 10646 character set with UTF-8 encoding. A session description consists of a number of lines in the form of <type>=<value> <type>. The description first consists of a session-level section that is optionally followed by one or more media-level sections. The session-level section begins with "v=" line; the media description section begins with a "m=" line and continues to the next media description section or until the end of the session description itself.

Figure 7 gives an example of an SDP session description. The first line, denoted by a "v=" represents the start of the session description and the session-level portion of the description. In this case, this line also specifies the protocol version. The second line ("o=") specifies the originator of the session, including the username and IP address of the host, and the session identifier and version number. The field starting with the "s=" indicates the session name, of which there should

only be one for every session description. The “i=” field is the session description field and simply contains information about the session. The session description can contain a line beginning with “e=” or ”p=”, which contain either the email address or phone number, respectively, of the person responsible for the session. Connection data is specified in the “c=” field, and in the example in Figure 7, this happens to be a multicast IP address along with subnet information (the “IN” refers to Internet). The “t=” field indicates start and stop time of the session; in this case the 0 fields specify that the start and stop times are not bounded, and therefore the session is permanent. The “m=” line indicates the start of the media description portion of the session description and contains several sub-fields. The first sub-field specifies media type, the second is the transport port to which the media will be sent, the third specifies the transport protocol being used, and the fourth sub-field identifies the media format of the session. Therefore, the field in the example indicates an audio-only session using port 3456 and RTP over Audio/Video profile (AVP) which is carried over UDP. The final sub-field value of “0” indicates a media format type of u-law PCM coded single channel audio which is sampled at 8KHz.

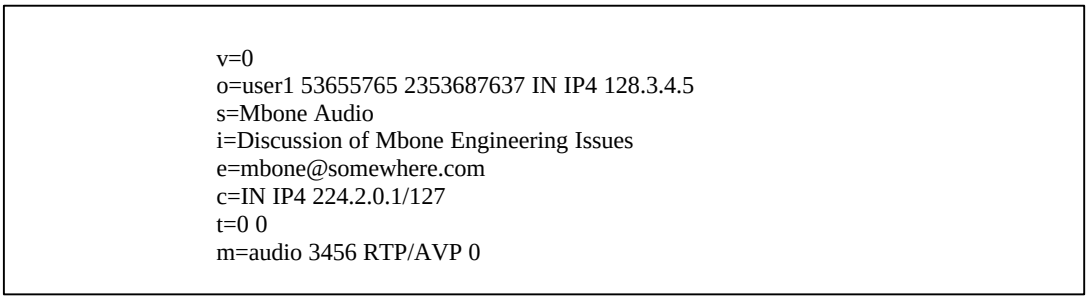

Figure 7: Example of SDP session description (Handley et al, p. 122).

3 BENEFITS OF SIP

According to the Schulzrinne and Rosenberg (1998c), SIP presents many significant benefits. These include extensibility, scalability, simplicity, the ease of integration and modularity.

3.1 Extensibility

SIP has a number of built-in extensibility and compatibility functions. Firstly, unknown headers and values are ignored by the protocol; any headers or features that are required to be understood can be indicated within the Require header field. If certain features are not understood, a server can return an error code to the client indicating which features are not supported. The client can then back-off and resort to a simpler operation if needed. In addition, developers can create new features for SIP and then register a name for them with the Internet Assigned Numbers Authority (IANA). The compatibility of these features is maintained across different SIP versions. Similar to HTTP, numerical error codes are hierarchically organized according to class; SIP terminals need only to understand the class of the response, not necessarily the specific error code itself.

3.2 Scalability

Schulzrinne and Rosenberg define scalability in terms of domains, server processing, and conference sizes. Firstly, because of existing scalable Internet services and routing protocols, such as DNS and the border gateway protocol (BGP) (Rekhter & Li, 1995), SIP can leverage off of these powerful technologies to scale to

large areas of operation or domains. As discussed previously, the ability of SIP servers to run in stateful or stateless mode or to use UDP as means of transport allows SIP servers to make more efficient use of their processing resources. It also allows network engineers to architect the network such that SIP servers operating at the edge of the network can offer more complex services by operating in stateful mode, while those servers in the core can be run in stateless mode, where processing and transaction speed is crucial. Finally, SIP can scale to different conference sizes and does not require the use of a centralized multipoint control unit (MCU) to coordinate the conference, as does H.323.

3.3 Simplicity

Because SIP is a text-based protocol, the parsing, generation and debugging of SIP messages is relatively easy and can be done with simpler scripting languages such as Perl or Tcl. This represents a "low cost of entry" for potential developers because client and server implementations can be rapidly built using these scripting tools whose natural data type is text (Schulzrinne & Rosenberg, 1998a, p. 11). In addition, because of the inherent simplicity of the protocol, a basic but "legal" SIP telephony implementation need only use four headers (To, From, Call-ID, and CSeq) and three request methods (INVITE, ACK, and BYE).

3.4 Ease of Integration

Because SIPs design is similar to HTTP, SMTP, and other Internet protocols and applications, it is currently capable of easily integrating with the World Wide

Web, e-mail, and other streaming media applications. For example, the ability to launch a telephony session by clicking a SIP URL within a SIP-capable browser could represent a powerful and popular method for initiating telephone calls in the future. Previous applications and protocols that have been designed with similar ease of integration in mind have grown to be wildly popular in the Internet.

3.5 Modularity

As previously shown in Figure 1, SIP fits well within a modular infrastructure like the Internet or other IP network. SIP is solely responsible for telephony signaling; session descriptions are handled by SDP, Quality of Service (QoS) is handled by protocols such as RSVP (Braden, Zhang, Berson, Herzog, & Jamin, 1997), IP routing is determined by OSPF (Moy, 1998) and BGP, etc. This modularity has allowed other IP-based applications and protocols to flourish. Changes to a particular protocol won't necessarily impact the integrity of other protocols; any changes to one protocol will most likely not affect another protocols use of its services.

4 A BRIEF COMPARISON OF SIP AND H.323

As of today, the ITU standard H.323 enjoys more popularity and is generally more accepted than SIP for the purposes of IP telephony signaling; it is more widely deployed than SIP and most IP telephony vendors include support for H.323 in their products. Multiple authors have compared SIP and H.323, including Schulzrinne (2000), Schulzrinne and Rosenberg (1998b), Dalgic and Fang (1998), Kraskey and

McEachern (1999), and Woods (1999). The following section summarizes these authors' findings on some of the more salient differences between the two protocols.

4.1 A Brief Overview of H.323

H.323 is actually a series of recommendations for providing multimedia communication systems over packet-based networks, including IP networks (ITU, 1998a). H.323 consists of a set of protocols and is an "umbrella specification" where various aspects of the protocol are specified in several different ITU-T recommendations.

There are four major components within a H.323 system: terminals, gateways, gatekeepers, and multipoint control units (MCUs). Terminals are simply client endpoints that provide and participate in two-way real-time communications (similar to UACs in SIP) with other H.323 objects. Terminals must support signaling and control, real-time communication, and codec functionality. Signaling and control capabilities are implemented by using three different protocols: H.245 (ITU, 1996) for channel usage and capabilities, H.225 (ITU, 1998b) for call signaling and establishment, and the Registration Admission and Status (RAS) protocol which is used for communication with gatekeepers. All three of these protocols use the Abstract Syntax Notation One (ASN.1) and the packed encoding rules (PER), binary representations, for encoding messages. Real-time communication is accomplished by requiring that terminals support RTP and the RTP control protocol (RTCP), which controls the sequencing of audio and video packets. Finally, codecs, which compress/uncompress audio and video before and

after transmission, are supported through different ITU G-series recommendations. Each H.323 terminal is required to support G.711, a 64kb codec.

Gateways are simply portals between packet-switched networks and circuit-switched networks. They provide call setup and control functionality and they translate between transmission formats and communication procedures of these two different types of networks. Gateways can also provide translation between different codecs if necessary. Gatekeepers are optional components within a H.323 system, but essentially allow the protocol to scale to larger numbers of users and terminals. When a gatekeeper is used on a H.323 network, all other endpoints are required to register with it and request permission from it previous to making a call. Gatekeepers are required to perform four different responsibilities: 1) address translation (for example, between E.164 phone numbers and IP addresses), 2) admission control, 3) bandwidth control, and 4) zone management. H.323 utilizes the concept of zones, where typically a gatekeeper or group of gatekeepers will be responsible for providing the above functionality within a zone and directly communicate with other gatekeepers in other zones. Gatekeepers may also provide four optional services: 1) call control signaling, 2) call authorization, 3) bandwidth management, and 4) call management.

MCUs support conferencing between three or more endpoints. Within the MCU reside two different components, the multipoint controller (MC) and possibly one or more multipoint processors (MP). The MC provides the control functionality between terminals while the MP performs any necessary processing on conference media streams, such as audio mixing.

Finally, H.323 employs four different channels to architect the communication exchange between different components. The *RAS channel* provides for communication between endpoint and a gatekeeper. This channel is used to register with the gatekeeper and request permission to place calls with other H.323 terminals. The *call signaling channel* uses H.225 and H.450 (ITU, 1998c) for call control and supplementary service control features, and is similar to Q.931 (ITU, 1998f). The *H.245 control channel* carries messages for media control, which includes support for capabilities exchange (similar to the use of SDP within SIP). Lastly, the *logical channel for media* transports the audio, video or other media in the network. Each media type is transported in a separate pair of unidirectional channels using RTP and RTCP. (Dalgic & Fang, 1998) (ITU, 1998a)

4.2 Complexity

One of the biggest drawbacks of H.323 is its complexity. Since it is an umbrella specification, it contains several complex protocols such as H.225, H.245, H.332 (ITU, 1998g) for large conferences, H.450 for supplementary services, H.235 (ITU, 1998e) for security, and H.246 (ITU, 1998d) for circuit-switched interoperation. Many H.323 services require that several of these different protocols interact to some extent; in addition, these are all ASN.1 PER binary encoded protocols, all of which make the debugging and the development of H.323 protocols and applications more of a complex exercise. This is in contrast to SIP, which is text-based and can be developed or customized using simpler high-level

programming tools such as Perl, Tcl, or Visual Basic, and which a simple implementation need only contain four headers and three request methods.

4.3 Scalability

With the number of worldwide Internet and IP users increasing at an exponential rate, the ability for an IP telephony protocol to scale to support large numbers of users over large geographic areas will become more essential as time goes on. We can highlight the ability of H.323 and SIP to scale in terms of the following two areas.

4.3.1 Stateful vs. Stateless Server Processing

In H.323 versions 1 (v1) and 2 (v2), gatekeepers must be stateful so they must keep track of all call states, as well as TCP connections since TCP is used for transport within these versions. This increases the processing load on the gatekeeper and limits its ability to scale to larger numbers of users. H.323 version 3 (v3) is similar to SIP in that the gatekeeper can function in stateless or stateful mode and either TCP or UDP can be used as the transport protocol.

4.3.2 Loop Detection

Forwarding loops can occur in IP telephony networks when there are several H.323 gatekeepers or SIP proxy servers involved in the setup of individual calls. As already discussed, SIP provides a loop detection mechanism using the Via header field, which is similar to the loop detection algorithm employed in BGP. H.323 v1 and v2 provide no means for loop detection and prevention; H.323 v3 makes use of a

PathValue field, which is similar to a ttl field, and specifies the maximum number of gatekeepers that a signaling message can traverse before being dropped. However, Dalgic and Fang contend that this mechanism is not as efficient as the mechanism employed within SIP; firstly, the PathValue field simply contains an integer value and does not use the names of gatekeepers, so a signaling message involved in a loop will not be discarded until reaching the value specified within the PathValue field. Secondly, if the architecture of the network changes, the PathValue may need to be changed to adequately support this change, which therefore increases the complexity of changing and maintaining the network.

4.4 Call Setup and Teardown

The call setup delay in H.323 v1 can be very large; call setup can utilize approximately one dozen packets and about six to seven round-trips. If a network is experiencing moderate packet loss, this can cause TCP retransmits, which in turn can result in even longer setup delays. H.323 v2 has rectified this problem somewhat with a fast setup procedure; this lowers the roundtrips down to about three for a H.323 v2 setup. H.323 v3 and SIP call setup times are very comparable, primarily due to the fact that both can use UDP as transport and thus, roundtrip delay due to TCP retransmits is not an issue. H.323 v3 does have some advantages over SIP in this area however. Version 3 sets up a TCP and a UDP connection almost simultaneously, so that if UDP fails TCP can take over the setup process. In SIP, this process occurs sequentially; TCP waits until UDP fails to begin the call setup process, which can introduce additional round-trip delay.

An example of a basic H.323 call setup and teardown is shown in Figure 8 in Appendix C. As shown, when using a gatekeeper that is operating in gatekeeper routed signaling mode two different protocols are required for simple call setup and release. The RAS protocol begins the process when both User A and B register with the gatekeeper using the Registration Request (RRQ) and Registration Confirm (RCF) messages. When User A tries to setup a call with User B, he must first send a RAS Address Request (ARQ) for User B to the gatekeeper and the gatekeeper responds with a Address Confirm (ACF) which tells User A to route all signaling messages through the gatekeeper. User A sends a H.225 setup message to the gatekeeper, who then forwards the setup to User B. User B sends a H.225 call proceeding message right away back to the gatekeeper, which is propagated back to User A. In the meantime, User B also sends an ARQ to the gatekeeper, and as in the previous ACF response to User A, the gatekeeper tells User B to route all call signaling through him. A H.225 alerting message (which indicates the phone ringing) is sent from User B through the gatekeeper to User A; this is followed by a H.225 connect (that indicates the phone going off-hook), which follows the same path as the previous message. At this point a media stream is established directly between the two endpoints. Finally, when User A wants to release the call, he sends a release complete back through the gatekeeper to User B. This must also be followed up by a Disengage Request (DRQ) to the gatekeeper by both users, which the gatekeeper responds to with a Disengage Confirm (DCF); this enables the gatekeeper to free up the bandwidth that was associated with this particular call.

4.5 Packet Loss and Reliability

H.323 v1 and v2 use TCP as means for overcoming packet loss in the network and achieving message reliability. But, since H.323 v3 supports both UDP and TCP, another mechanism is needed for providing message reliability when the former unreliable protocol is used. H.323 v3 introduces five new timers on both the sending and receiving sides to provide this reliability. On the sending side, the calling endpoint starts two timers after sending a setup message, T1 and T4. If T1 expires before it receives a response from the called endpoint or gatekeeper, it resends the setup and starts a new timer called T3. If T3 expires, another call setup is sent and the T3 timer is restarted; if this timer expires again, the calling endpoint stops retransmitting the setup and begins the call setup process with TCP instead. On the receiving side, the called endpoint starts T1 after the first response transmission. If T1 expires, it resends the response and starts timer T3 which, if timeout for T3 occurs, is restarted and the response is sent again. If T3 times out again, the called endpoint stops the retransmissions and starts timer T5. If T5 expires, the called endpoint dispenses with all associated call and state information and considers the setup of this particular call as failed.

SIP maintains reliability by retransmitting requests every .5 seconds or until either a 1xx progress report or final status (greater than 2xx) response is received. Servers provide reliability by retransmitting an original final response until an ACK is received, while SIP clients retransmit an ACK after every final message.

4.6 Extensibility

SIPs approach to extensibility has already been discussed here; error codes are divided among classes and SIP platforms are only required to understand the class definition, not the individual error code. Any new features can be developed by a third-party for SIP, and the feature names can easily be registered with the IANA.

On the other hand, H.323 can only be extended using the vendor-defined nonstandardParam fields which are placed in various locations in ASN.1. These parameters contain a vendor code and a value which is typically only understood by that particular vendor. This limits vendors to writing extensions where only nonstandardParam fields are located; if a vendor wishes to add a value or component to an existing parameter and no nonstandardParam field exists, there is no recourse.

In terms of codec support, SIP can support any codec, standard or non-standard, third-party or developed in-house. H.323 requires that each codec be registered and standardized as a G-series ITU recommendation. Schulzrinne and Rosenberg (1998b) argue that since many codecs contain significant intellectual property, there is no freely available sub-28.8kbs codec which can be used in H.323 systems by less wealthy institutions, such as universities and small companies.

4.7 Interoperability

Dalgic and Fang discuss interoperability in terms of a signaling protocol being interoperable with itself but across different versions, interoperable with other

vendors implementations, and interoperable with other signaling protocols. Due to the fact that these signaling protocols will most likely be widely deployed across the globe in different versions and will be required to interoperate with other protocols, interoperability is a significant issue.

H.323 is required to be backward compatible from one version to the next so that different versions can be integrated without compatibility problems. This can have positive and negative implications. It is good that different versions can interoperate because this typically means that an existing provider can deploy a new version of H.323 within their network and expect features from previous versions to still work. However, the requirement that all versions must interoperate means the code base for later versions will grow to be quite large because of the legacy features these versions are required to support. This could make any future customization or debugging to be quite complex and difficult. SIP suffers from the same problem, albeit in the reverse direction. A newer version of SIP may discard old features that are not expected to be used any longer; this can reduce the overall size of the code base, but it may lead to certain features not being supported in later releases.

In terms of interoperability among implementations, proponents of H.323 have provided numerous tools and pieces of documentation to help clarify the protocol and different implementations of the protocol for vendors. Since SIP is still immature and is in the early development stages, interoperability tests have only recently begun. SIPs adherence to interoperability amongst implementations will only be clarified as time goes on.

H.323 is positioned well to interoperate with User-to-Network Interfaces (UNI) in the PSTN, such as Q.931. Some procedures within H.323 are very similar to Q.931. However, there is currently no established standard for the translation of the Network-to-Network Interface (NNI) of SS7 ISDN User Part (ISUP) messages across H.323. Currently, there is not a standard by which to translate SIP messages to SS7 signaling messages. However, there is an Internet Draft available which gives a high-level description of a SIP-to-PSTN gateway, suggesting the Media Gateway Control Protocol (MGCP) as a possible interface between SIP and SS7 (Donovan & Cannon, 1998). But further work needs to be done in this area.

4.8 Fault Tolerance

SIP at this time provides no means itself for bypassing network faults such as failed proxy servers, etc., other than its ability to operate in stateless mode, and therefore, it is not required to route all session-related messages through the same server (in case failure occurs). H.323 v3 introduces redundant gatekeepers and endpoints, and gatekeeper clustering so that there is a notion of redundancy within the network. During the RAS registration process, the gatekeeper can also designate alternate gatekeepers to the H.323 endpoints in case the primary gatekeeper fails for some reason.

5 CONCLUSION

IP telephony represents the next big thing in the telecommunications industry. Service providers and telephony carriers will find that they need to take this technology shift seriously and at least consider the possibility of implementing IP telephony services of some sort; the competition in the industry is currently far too fierce which is, in turn, causing both prices that consumers are paying for voice calls and service providers margins to drop. Service providers will need newer and cheaper methods for offering new and existing services in this highly turbulent market (Kraskey & McEachern, 1999). Likewise, corporations are targeting IP telephony as a way to shunt the bulk of their long-distance voice traffic over their existing IP infrastructure, and thus save significant amounts of capital for a relatively modest investment.

However, the technical challenges in providing time-sensitive services such as telephony traffic over IP networks will be a formidable task. Bandwidth can suddenly become very scarce within an IP network, and this factor alone makes many a skeptic that IP telephony will ever become reality. Anyone who has used the World Wide Web can relate to the experience of having to wait five, ten, and even more seconds for their favorite web page to download to their computer. When this kind of delay is experienced browsing the web it is a nuisance, but when it is experienced in the middle of a telephone conversation it can make the conversation downright unintelligible and bring it to a grinding halt. If IP telephony services are going to be sharing resources with bandwidth-intensive applications such as HTTP and FTP, then Quality of Service tools will need to be reliable and dependable so

that bandwidth can be virtually guaranteed for these real-time applications. Along the same lines, the TDM switches that are currently deployed in carrier networks may be excessively expensive to purchase and maintain and they may take up more physical space than their IP telephony counterparts, but they work and they are dependable. If IP telephony services are going to achieve the large-scale deployment that proponents are predicting, the equipment and software will need to be similarly well-engineered and designed.

However, despite the risks and challenges mentioned above, providers and carriers are continuing to deploy IP telephony networks. A vast majority of these networks are running H.323 of some flavor or another, and therefore, only time will tell if SIP will be capable of challenging H.323s popularity in terms of live deployment. Given that H.323 is an ITU-standard protocol, another advantage it enjoys is its ability to interconnect with PSTN services relatively seamlessly; H.225 messages map effectively with Q.931 messages and, although there is no current standard for doing so, the basic framework exists to map H.323 to SS7 messages. And although SIPs architecture enjoys some advantages over previous versions of H.323, Dalgic and Fang (1999) contend that H.323 v3 and SIP are fairly comparable in terms of scalability, support of both UDP and TCP, call setup times, and fault tolerance. However, if SIP can gain momentum in terms of live deployment, it has the potential to achieve popularity similar to that of H.323. The remaining and singular advantage SIP maintains is its ease of implementation; consider how many new and popular third-party applications emerged after the World Wide Web and HTTP became popular. If developers can use higher-level programming tools such

as Tcl and Perl to develop SIP applications and enhancements, then SIP could overtake H.323 merely due to the popularity of the third-party applications that support SIP.

APPENDIX A: GLOSSARY

ASN.1	Abstract Syntax Notation One. Standard way to describe message that can be sent or received in a network system.
BGP	Border Gateway Protocol. Protocol used for exchanging routing information between gateway hosts in a network of autonomous systems.
DNS	Domain Name System. Translates Internet domain names into IP addresses and vice versa.
H.323	ITU umbrella specification for providing multimedia communication systems over packet-based networks.
HTTP	Hypertext Transfer Protocol. Application protocol and set of rules used for exchanging files in the World Wide Web.
IANA	Internet Assigned Numbers Authority. Responsible for registering any “unique parameters and protocol values” used for operation within the Internet.
IETF	Internet Engineering Task Force. Organization that defines standard Internet operating protocols.
IN	Intelligent Network. Telephone network architecture designed by Bellcore, where service logic for a phone call is separately located from switching facilities which allows services to be added without the need for redesigning the switching equipment.
ISDN	Integrated Services Digital Network. Set of CCITT and ITU standards for digital transmission over telephone copper wire and other media.
ISUP	ISDN User Part. Transport, or layer 4 protocol used within SS7 telephony signaling networks.
ITU	International Telecommunications Union. International body which fosters cooperative standards for telecommunications equipment.
MCU	Multipoint control unit. Used in H.323 systems for supporting and controlling conferences between two or more endpoints.

MGCP	Media Gateway Control Protocol. Signaling control protocol which controls media gateways or servers and can be used as a network-to-network interface to the PSTN (SS7, for example).
PSTN	Public Switched Telephone Network.
Q.931	ITU-T specification for signaling to establish, maintain, and clear ISDN network connections.
QoS	Quality of Service. Notion of providing guaranteed transmission and level of service over IP networks.
RAS	Registration, Admission and Status Protocol. Protocol used within H.323 for discovering and communicating with gatekeeper.
RFC	Request for Comments. Internet formal document or standard which is reviewed by interested parties.
RSVP	Resource ReSerVation Protocol. Allows for in-band reservation of resources for audio and video multicast transmissions.
RTCP	Real-Time Control Protocol. Signaling protocol which controls RTP transmissions.
RTP	Real-Time Protocol. Provides end-to-end transport functions for real-time applications such as voice and video.
SDP	Session Description Protocol. Describes multimedia sessions and media streams within session initiation signaling messages.
SIP	Session Initiation Protocol. Protocol for transmitting and receiving packet telephony signaling information.
SMTP	Simple Mail Transfer Protocol. Protocol used in the sending and receiving e-mail over IP networks.
SS7	Signaling System #7. Out-of-band overlay packet network used for telephony signaling within the PSTN.
TCP	Transmission Control Protocol. Transport, or layer four protocol used with IP to reliably transport data in IP networks.
UAC	User Agent Client. Initiates SIP requests.

UAS	User Agent Server. Responds to SIP requests.
UDP	User Datagram Protocol. Transport, or layer four protocol used with IP to transport data unreliably in IP networks. An alternative to TCP.
URI	Uniform Resource Identifier. Addressing format used to identify resources on the Internet. The most common form of the URI is the URL.
URL	Uniform Resource Locator. The address of a resource on the Internet.
UTF-8	Universal Character Set Transformation format 8. Eight-bit encoding system used for 16-bit that preserves the full US-ASCII range.
VoIP	Voice over IP. Term used to describe the transport of voice calls over IP networks.

APPENDIX B: RESPONSE STATUS CODES

Informational

100	Trying
180	Ringing
181	Call Is Being Forwarded
182	Queued

Table B 1: Information Response Status Codes 1xx (Handley et al, p. 74).

Successful

200	OK

Table B 2: Success Response Status Codes 2xx (Handley et al, p. 75).

Redirection

300	Multiple Choices
301	Moved Permanently
302	Moved Temporarily
303	See Other
305	Use Proxy
380	Alternative Service

Table B 3: Redirection Response Status Codes (Handley et al, p. 75)

Client-Error

400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
413	Request Entity Too Large
414	Request-URI Too Large
415	Unsupported Media Type
420	Bad Extension
480	Temporarily not available
481	Call Leg/Transaction Does Not Exist
482	Loop Detected
483	Too Many Hops
484	Address Incomplete
485	Ambiguous
486	Busy Here

Table B 4: Client-Error Response Status Codes 4xx (Handley et al, p. 77)

Server-Error

500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-out
505	SIP Version not supported

Table B 5: Server-Error Response Status Codes 5xx (Handley et al, p. 81)

Global-Failure

600	Busy Everywhere
603	Decline
604	Does not exist anywhere
606	Not acceptable

Table B 6: Global-Failure Response Status Codes (Handley et al, p. 82)

APPENDIX C: BASIC H.323 CALL SETUP AND TEARDOWN

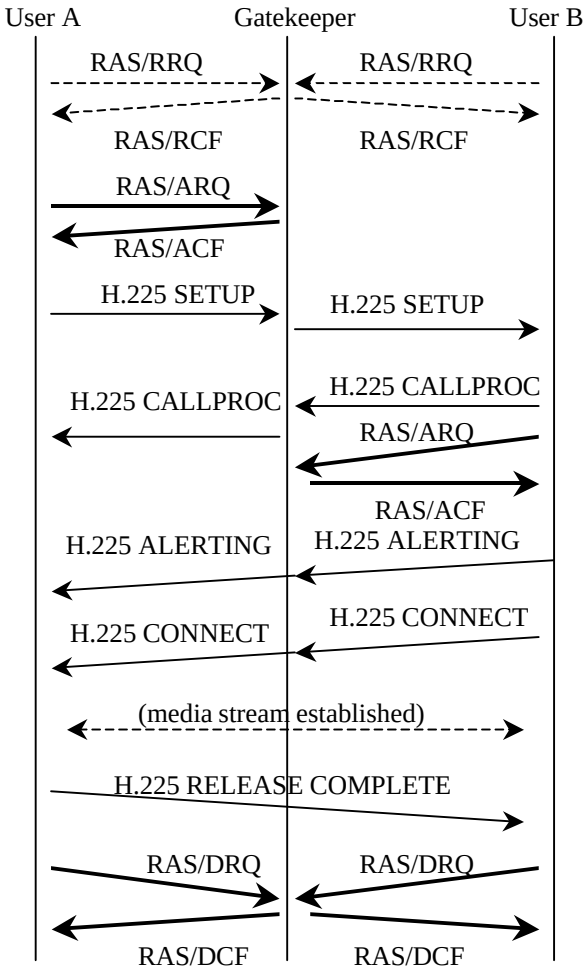

Figure 8: Basic H.323 Call Setup and Teardown using a single gatekeeper (ITU, 1998a).

REFERENCES

Berners-Lee, T., Fielding, R., & Masinter, L. (1998, August). Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc2396.html

Berners-Lee, T., Masinter, L., & McCahill, M. (1994, December). Uniform Resource Locators (URL). RFC 1738. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc1738.html

Braden, B., Zhang, L., Berson, S. Herzog, S., & Jamin, S. (1997, October). Resource ReSerVation Protocol (RSVP): Version 1 Function Specification. RFC 2205. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc2205.html

Dalgic, I., & Fang, H. (1999, September). Comparison of H.323 and SIP for IP Telephony Signaling. Proceedings of Photonics East. Retrieved March 5, 2000. Available: http://www.cs.columbia.edu/~hgs/papers/others/Dalg9909_Comparison.pdf

Donovan, S., & Cannon, M. (1998, November). A Functional Description of SIP-PSTN Gatway. Internet Draft. Internet Engineering Task Force. Work in progress (expired). Retrieved March 4, 2000. Available: http://www.cs.columbia.edu/sip/drafts/draft-donovan-sip-gw-client-00.txt

Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Berners-Lee, T. (1997, January). Hypertext Transfer Protocol: HTTP/1.1. RFC 2068. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc2068.html

Fingal, F., & Gustavsson, P. (1999, February). A SIP of IP-telephony. Master's Thesis. Department of Communication Systems, Lund University. Retrieved March 11, 2000. Available: http://www.cs.columbia.edu/sip/drafts/Fing9902_SIP.pdf

Friedrichs, T. (1998, November). Talk is Cheaper With Voice Over IP. Network Magazine. Retrieved March 9, 2000. Available: http://www.data.com/issue/981121/talk.html

Handley, J, & Jacobson, V. (1998, April). SDP: Session Description Protocol. RFC2327. Internet Engineering Task Force. Retrieved March 4, 2000. Available: http://www.normos.org/rfc/rfc2327.txt

Handley, M., Schulzrinne, H., Scholler, E., & Rosenberg, J. (1999, March). SIP: Session Initiation Protocol. RFC2543. Internet Engineering Task Force. Retrieved March 3, 2000. Available: ftp://ftp.isi.edu/in-notes/rfc2543.txt

Henderson, K. (1999a, November). Teleglobe Offers Wholesale, Retail Global IP Telephony Services. Sounding Board Magazine. Retrieved March 9, 2000. Available: http://www.soundingboardmag.com/articles/9b1new13.html

Henderson, K. (1999b, November). Unwrapping the Opportunity of VoIP Resale: As Arbitrage Abates, More Colorful Services will Surface. Sounding Board Magazine. Retrieved March 9, 2000. Available: http://www.soundingboardmag.com/articles/9b1feat1.html

Hoffman, P., Masinter, L., & Zawinski, J.. (1998, July). The mailto URL scheme. RFC 2368. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc2368.html

International Telecommunication Union. (1996, March). "Control Protocol for Multimedia Communication". Recommendation H.245. Telecommunication Standardization Sector of ITU. Geneva, Switzerland.

International Telecommunication Union. (1998a, February). "Packet-Based Multimedia Commmunications Systems". Recommendation H.323. Telecommunication Standardization Sector of ITU. Geneva, Switzerland.

International Telecommunication Union. (1998b, February). "Call Signalling Protocols and Media Stream Packetization for Packet-Based Multimedia Communication Systems". Recommendation H.225.0. Telecommunication Standardization Sector of ITU. Geneva, Switzerland.

International Telecommunication Union. (1998c, February). "Generic Functional Protocol for the Support of Supplementary Services in H.323". Recommendation H.450.1. Telecommunication Standardization Sector of ITU. Geneva, Switzerland.

International Telecommunication Union. (1998d, February). "Interworking of H-Series Multimedia Terminals with H-Series Multimedia Terminals and Voice/Voiceband Terminals on GSTN and ISDN". Recommendation H.246. Telecommunication Standardization Sector of ITU. Geneva, Switzerland.

International Telecommunication Union. (1998e, February). “Security and Encryption for H-Series (H.323 and other H.245-based) Multimedia Terminals”. Recommendation H.235. Telecommunication Standardization Sector of ITU. Geneva, Switzerland.

International Telecommunication Union. (1998f, May). “ISDN User-Network Interface Layer 3 Specification for Basic Call Control”. Recommendation Q.931. Telecommunication Standardization Sector of ITU. Geneva, Switzerland.

International Telecommunication Union. (1998g, September). “H.323 Extended for Loosely Coupled Conferences”. Recommendation H.332. Telecommunication Standardization Sector of ITU. Geneva, Switzerland.

Kraskey, T., & McEachern, J. (1999, December). Next-Generation Network Voice Services. Network Magazine. vol. 14 no. 12, pgs. 96-102.

Mockapetris, P. (1987, November). Domain Names: Implementation and Specification. RFC 1035. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc1035.html

Moy, J. (1998, April). OSPF Version 2. RFC 2328. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc2328.html

Postel, J. (1982, August). Simple Mail Transfer Protocol. RFC 821. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc821.html

Rekhter, Y., & Li, T. (1995, March). A Border Gateway Protocol 4 (BGP-4). RFC 1771. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc1771.html

Schulzrinne, H. (2000, March). Comparison of H.323 and SIP. SIP Homepage. Retrieved March 4, 2000. Available: http://www.cs.columbia.edu/sip/h323-comparison.html

Schulzrinne, H., & Rosenberg, J. (1998a, February). Signaling for Internet Telephony. Columbia University Dept. of Computer Science Technical Report CUCS-005-98. Retrieved March 11, 2000. Available: http://www.cs.columbia.edu/~hgs/papers/Schu9802_Signaling.ps.gz

Schulzrinne, H., & Rosenberg, J. (1998b, July) A Comparison of SIP and H.323 for Internet Telephony. Network and Operating System Support for Digital Audio and Video (NOSSDAV). Cambridge, England. Retrieved March 5, 2000. Available: http://www.cs.columbia.edu/~hgs/papers/Schu9807_Comparison.ps.gz

Schulzrinne, H., & Rosenberg, J. (1998c, October). The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet. Bell Labs Technical Journal. pgs. 144-160. Retrieved March 4, 2000. Available: http://www.lucent.com/ideas/perspectives/bltj/oct-dec1998/pdf/paper09.pdf

Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (1996, January). RTP: A Transport Protocol for Real-Time Applications. RFC 1889. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc1889.html

Sparks, R., Cunningham, C., Johnston, A., Donovan, S., & Summers, K. (1999, October). SIP Telephony Service Examples With Call Flows. Internet Draft. Internet Engineering Task Force. Work in progress. Retrieved March 5, 2000. Available: http://www.cs.columbia.edu/sip/drafts/draft-ietf-sip-call-flows-00.txt

Woods, D. (1999, December). Translating Menus at the VoIP Café. Network Computing. Retrieved March 9, 2000. Available: http://www.nwc.com/shared/printArticle?article=nc/1026/1026ws1.html&pub=nwc

Yergeau, F. (1996, October). UTF-8: A Transformation Format of Unicode and ISO 10646. RFC 2044. Internet Engineering Task Force. Available: http://www.faqs.org/rfcs/rfc2044.html