Technology Review#2001-03

Voice Over IP – The SIP Way

Vinod K. Bhat

April 2001

Contents

Introduction

Today most of the telephony is still made on the old Public Switched Telephone Network (PSTN). This means that a call reserves the connection between the two users and no one else can use this connection. The difference with Internet Telephony, also called Voice-over-IP (VoIP), is that the transport is made on an IP-network. It is possible to send packets between two or more parties without reserving the connection.

IETF (Internet Engineering Task Force) and other working groups have put lot of efforts to come up with a protocol, which could lay standards for Internet Telephony. These efforts gave birth to Session Initiation Protocol (SIP). The imminent acceptance of the SIP as an official IETF standard marks an important milestone to the IP telephony industry. That milestone is the merging of Internet based distributed technologies with traditional telephony.

SIP standardization has moved from MMUSIC (Multiparty Multimedia Session Control) to the SIP Working Group (WG). SIP WG has primary responsibility for the future development of SIP, but SIP-related work occurs in a number of IETF working groups.

About SIP

SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions (conferences) or Internet telephony calls. SIP can invite participants to unicast and multicast sessions; the initiator does not necessarily have to be a member of the session to which it is inviting. Media and participants can be added to an existing session. SIP transparently supports name mapping and redirection services, allowing the implementation of ISDN and Intelligent Network telephony subscriber services. These facilities also enable personal mobility, which provides capability to reach a called party at a single, location-independent address. As a traditional text-based Internet protocol, it resembles the hypertext transfer protocol (HTTP) and simple mail transfer protocol (SMTP). Like these protocols, SIP is a textual protocol based on the client-server model, with requests generated by one entity (the client), and sent to a receiving entity (the server) which responds them. A request invokes a method on the server and can be sent either over TCP or UDP. The most important SIP method, of the currently six, is the INVITE method, used to initiate a call between a client and a server. The other SIP methods are ACK, OPTIONS, BYE, CANCEL and REGISTER. A new method INFO has also been proposed as part of SIP-extensions and is detailed in RFC 2976. SIP uses Session Description Protocol (SDP) for media description. SIP supports five aspects of establishing and terminating multimedia communications; which are user location, user capabilities, user availability, call setup and call handling. SIP 2.0 is detailed in RFC 2543.

SIP Components

There are three components in SIP architecture, namely, user agents, network servers and SIP messages.

User Agents

A user agent is an application that acts on behalf of a user. It can act both as a User Agent Client (*UAC*) and User Agent Server (*UAS*); as the user probably is wishing to both be able to call and to be called. *UAC* is used to initiate a SIP request. *UAS* receives requests and returns responses on behalf of the user. The response *accepts*, *rejects* or *redirects* the request. These user agents contain the full SIP state machine and can be used without intermediate servers.

Network Servers

There are three kinds of network servers, namely, *proxy servers, redirect servers* and *registrar servers*. SIP servers, on occasion, will need to contact an external *location server* to determine callee's possible location(s).

Proxy Server

A SIP proxy server forwards requests to the next server after deciding which it should be. A proxy server interprets, and, if necessary, rewrites a request message before forwarding it. This next server could be any kind of SIP server; the proxy does not know and does not have to know. Before the request has reached the UAS it may have traversed several servers. As a proxy server issues both requests and responses it contains both a client and a server. A proxy server can either be stateful or stateless. When stateful, a proxy remembers the incoming request, which generated outgoing requests, and the out-going requests. A stateless proxy forgets all information once an outgoing request is generated. A proxy server can fork the incoming request to multiple locations if the callee has multiple-location registrations with the server. A forking proxy is always stateful because it needs to remember the states of all the branches to which the incoming SIP request was forked.

Redirect Server

Redirect server, does not forward requests to the next server. It accepts a SIP request and maps the address to zero or more new addresses and returns these addresses to the client and then client can contact the server directly. Unlike a *proxy server*, it does not initiate its own SIP request. Unlike a *user agent server*, it does not accept calls.

Registrar Server

A registrar is a server that accepts REGISTER requests and maintains the availability details of various servers and clients. A registrar is typically co-located with a proxy or redirect server and may sometimes offer location services also.

SIP Messages

SIP messages are typically of type *requests* and *responses*. Requests flow from client to a server and a response from a server to a client. These, requests and responses, include different headers to describe the details of the communication.

SIP being a text-based protocol makes its header largely self-describing and minimizes the cost of entry. SIP maintains a common structure of all messages and their *header* *fields*, allowing a generic parser to be written. Request and response use a generic-message format, which consists of a start-line, one or more header-fields ("headers"), an empty line indicating the end of the header fields, and an optional *message-body*. SIP was designed for character-set independence, so that any field can contain any ISO 10646 character. Together with the ability to indicate languages of enclosed content and language preferences of the requester, SIP is well suited for international use.

To make SIP signaling more secure, encryption and authorization can be used. Encryption can for example be used to prevent packet sniffers and other eavesdroppers from seeing who is calling whom. Authorization is used to prevent an active attacker from modifying and replaying SIP requests and responses.

SIP Header-fields

SIP header fields are similar to HTTP header fields in both syntax and semantics. Messages use header-fields to specify such things as caller, callee, the path of the message, type and length of message body and so on. Some of the header fields are used in all messages, the rest is used when appropriate. A SIP application does not need to understand all these headers, though it is desirable. The entity receiving simply silently ignores headers that it does not understand. The order in which the headers appear is generally of no importance, except for the *Via* field and that hop-by-hop headers appear before end-to-end headers.

There are 44 SIP headers listed in the Internet draft of RFC 2543, dated November 2000. In Figure 1 below, 37 headers have been mentioned. These headers can be divided into four different groups of headers:

General header fields apply to both request and response messages.

Entity header fields define information about the message body or, if no body is present, about the resources identified by the request.

Request header fields act as request modifiers and allow the client to pass additional information about the request, and about the client itself, to the server.

Response header fields allow the server to pass additional information about the response, which cannot be placed in the Start-Line (in responses it is called Status-Line).

These header fields give information about the server and about further access to the resource identified by the Request-URI

General-headers	**Request-headers**	**Response-headers**	**Entity-headers**
Call-ID	Accept	Allow	Content-Encoding
Contact	Accept-Encoding	Proxy-Authenticate	Content-Length

General-headers	**Request-headers**	**Response-headers**	**Entity-headers**
CSeq	Accept-Language	Retry-After	Content-Type
Date	Authorization	Server	
Encryption	Contact	Unsupported	
Expires	Hide	Warning	
From	Max-Forwards	WWW-Authenticate	
Record-Route	Organization		
Timestamp	Priority		
To	Proxy-Authorization		
Via	Proxy-Require		
	Route		
	Require		
	Response-Key		
	Subject		
	User-Agent		

Figure 1: SIP Header fields

Some of the above headers are shortly explained below, the rest are explained in RFC2543.

Header	**Explanation**
Allow	The Allow header field lists the set of methods supported by the resource identified by the Request-URI.
Call-ID	Uniquely identifies a particular invitation or all registrations of a particular client.
Call-Info	The Call-Info general header field provides additional information about the caller or callee, depending on whether it is found in

Header	Explanation
	a request or response.
Contact	Provides URL(s), where the user can be reached for further communications. It is used in INVITE, OPTIONS, ACK and REGISTER requests.
Content-Length	Indicates the size of the message body sent to the recipient.
Content-Type	Indicates the media type of the message body sent to the recipient.
CSeq	(Command Sequence) Uniquely identifies a request within a Call-ID
Encryption	Specifies that the content has been encrypted.
From	Indicates the initiator of the request
Route	The Route request-header field determines the route taken by a request.
Subject	Indicates the nature of the call.
To	Specifies the recipient of the request.
Via	Indicates the path taken by the request so far.
WWW-Authenticate	It is response-header field and MUST be included in 401 (Unauthorized) response messages; specifying client to sent the authorization information.

SIP Requests

The request is characterized by the *Start-Line*, called *Request-Line* and starts with a method token followed by a Request-URI and the protocol version. There are six different kinds of requests in the current version of SIP (version 2.0). They are referred to as methods and are here listed with their functionality. New SIP method INFO is also proposed as part of the SIP-extensions.

REGISTER: Conveys information about a user's location to a SIP server.

INVITE: The INVITE method indicates that the user or service is being invited to participate in a session. The message body MAY contain a description of the session to which the callee is being invited. For a two-party call, the caller indicates the type of media it is able to receive as well as their parameters such as network destination. A success response indicates in its message body which media the callee wishes to receive.

ACK: The ACK request confirms that the client has received a final response to an INVITE. ACK is used only with INVITE requests. It may contain a message body with the final session description to be used by the callee. If the message body is empty, the callee uses the session description in the INVITE request.

OPTIONS: The OPTIONS method queries the capabilities of the server/end system, but does not set up a connection.

BYE: The user agent client uses BYE to indicate to the server that it wishes to release the call.

CANCEL: The CANCEL request cancels a pending request with the same Call-ID, To, From and CSeq (sequence number only) header field values, but does not affect a completed request or existing calls. (A request is considered completed if the server has returned a final response).

INFO: An additional SIP method proposed, as part of the SIP-extensions is *INFO* method. The intent of the INFO method is to allow for the carrying of session related control information that is generated during a session. INFO method is detailed in RFC 2976. Other SIP extension methods are also being proposed.

Following the Request-Line, after the SIP headers, the request may contain a message body, which is separated from the headers with an empty line. The message body is always a session description and if present the type of Internet media in it is indicated by the Content-Type header field.

SIP request example:

SIP Request	**Explanation**
Header	
INVITE sip:vinod@mumbai.tcs.co.in SIP/2.0	Method type, request URI and SIP Version
Via:SIP/2.0/UDP anilworkstation.com	IP address and port of previous hop
From: Anil <sip:anil@delhi.tcs.co.in>	Caller
To: Vinod Bhat <sip:vinod@mumbai.tcs.co.in>	Callee
Call-ID:123456789@anilworkstation.com	Globally unique ID for this call
Content-Type:application/sdp	The body type, an SDP message

Cseq:1 INVITE Content-Length: …	Command Sequence number and type Length of the body of the SIP method
Blank line separates header from body	
Body	
v=0 o=anil 28960783 0 IN IP4 157.227.12.184 s=Urgent phone call from Anil c=IN IP4 anilworkstation.com t=3126288799 3126289399 m=audio 5002 RTP/AVP 0 3.5	SDP version Owner/creator and session identifier The name of the session Connection information Time the session is active Media name and transport address

SIP Responses

The recipient, after receiving and interpreting a request message, responds with a SIP response message, indicating the status of the server, success or failure. The responses can be of different kinds and the type of response is identified by a status code, a 3-digit integer. The first digit defines the class of the response. The other two have no categorization role. The six different classes that are allowed in SIP are here listed with their meaning. These classes can be categorized by provisional and final responses. A provisional response is used by the server to indicate progress, but does not terminate a SIP request. A final response terminates a SIP request. 1xx response codes are provisional responses and 2xx onwards responses are final responses.

Response Code Series	Explanation	Example Response Codes
1xx	Informational	100 – Trying 180 – Ringing 181 – Call is being forwarded
2xx	Successful	200 – Ok
3xx	Redirection	301 – Moved Permanently 302 – Moved Temporarily 305 – Use Proxy 380 – Alternative Service
4xx	Request Failure	400 – Bad Request 401 – Unauthorized 403 – Forbidden 404 – Not Found 407 – Proxy Authentication Required
5xx	Server Failure	500 – Server Internal Error 501 – Not Implemented 502 – Bad Gateway 504 – Server Time-out

6xx	Global Failures	600 – Busy Everywhere 603 – Decline 604 – Does Not Exist Anywhere

SIP applications are not required to understand the meaning of all registered response codes, though it is desirable. However applications must be able to recognize the class of the response and treat any unrecognized response as being the x00 response code of the class.

SIP response example:

SIP Response	**Explanation**
Header	
SIP/2.0 200 OK	SIP Version and response code
Via:SIP/2.0/UDP hop.mumbai.tcs.co.in Via:SIP/2.0/UDP anilworkstation.com	IP addresses and port of previous hops
From: Anil <sip:anil@delhi.tcs.co.in>	Caller
To: Vinod Bhat <sip:vinod@mumbai.tcs.co.in> ;tag=35453231	Callee
Call-ID:123456789@anilworkstation.com	Globally unique ID for this call
Content-Type: application/sdp	The body type, an SDP message
CSeq: 1 INVITE	Command Sequence number and type
Content-Length: …	Length of the body of the SIP method
Blank line separates header from body	
Body	
v=0	SDP version
o=vinod 34234567 12354354 IN IP4 197.27.12.14	Owner/creator and session identifier
s=Ok	The name of the session
c=IN IP4 vinodworkstation.mumbai.tcs.co.in	Connection information
t=3126288799 3126289399	Time the session is active
m=audio 5002 RTP/AVP 0	Media name and transport address

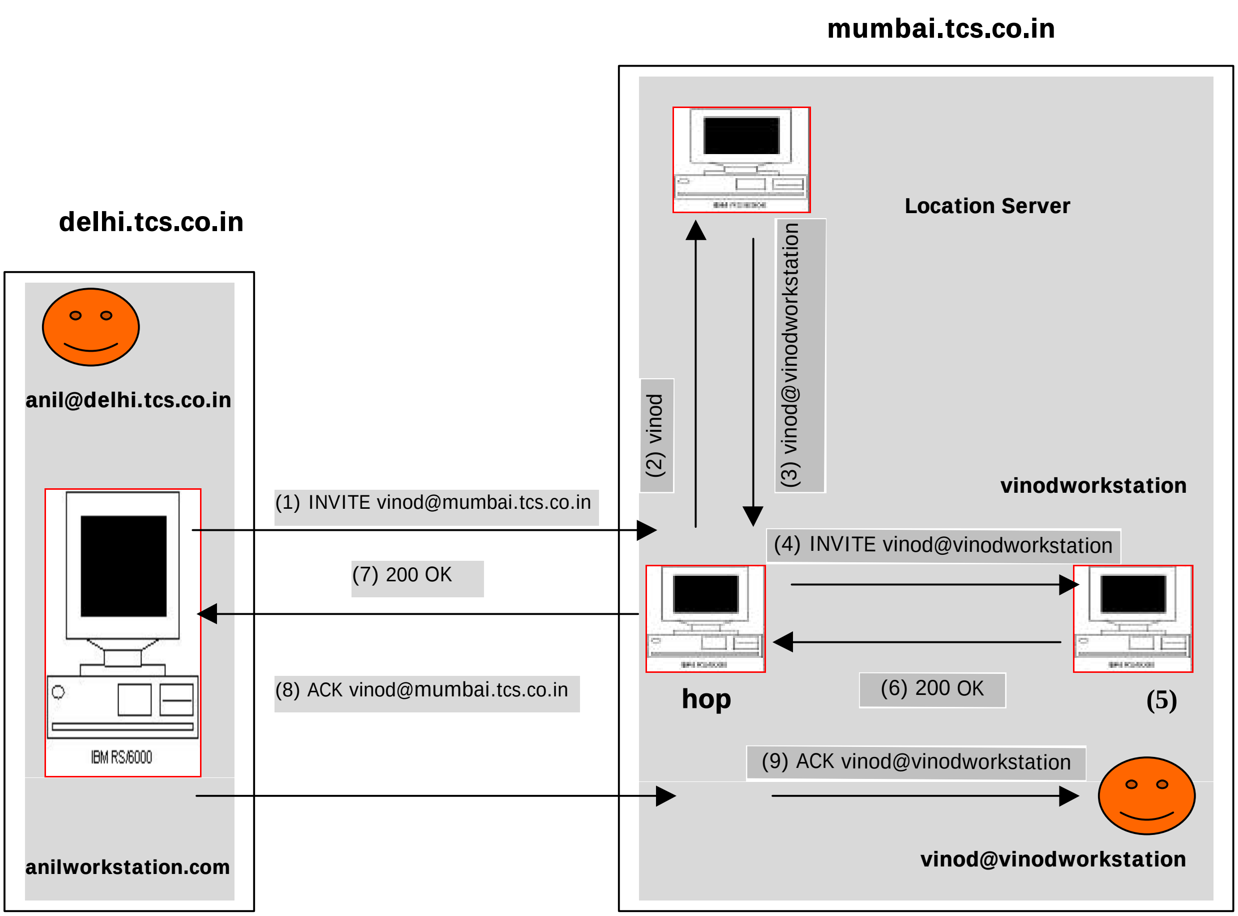
SIP Operation

As mentioned in the previous section, there are two different ways of handling an incoming SIP request at a SIP server.

These are here illustrated using the most important SIP operation, inviting a participant to a call, to show the basic operation of SIP. Figure 2 and Figure 3 each shows a sample example with simplified messages and the provisional responses left out. The user *anil* at host *anilworkstation.com* wants to invite user *vinod.* He obtains *vinod's* address from his email address, of form name@domain, which is *vinod@mumbai.tcs.co.in*.

The client then translates the domain part to a numeric IP address, by a DNS lookup, where a server may be found. This results in the server *hop* of domain *mumbai.tcs.co.in*.

As shown in figures 2 and 3, an INVITE request is then generated and sent to this server (1). The Server accepts the invitation and contacts his location server for a more precise location (2). The location server returns one location of *vinod*, which is at host *vinodworkstation* (3). These steps are the same for both proxy and redirect server.

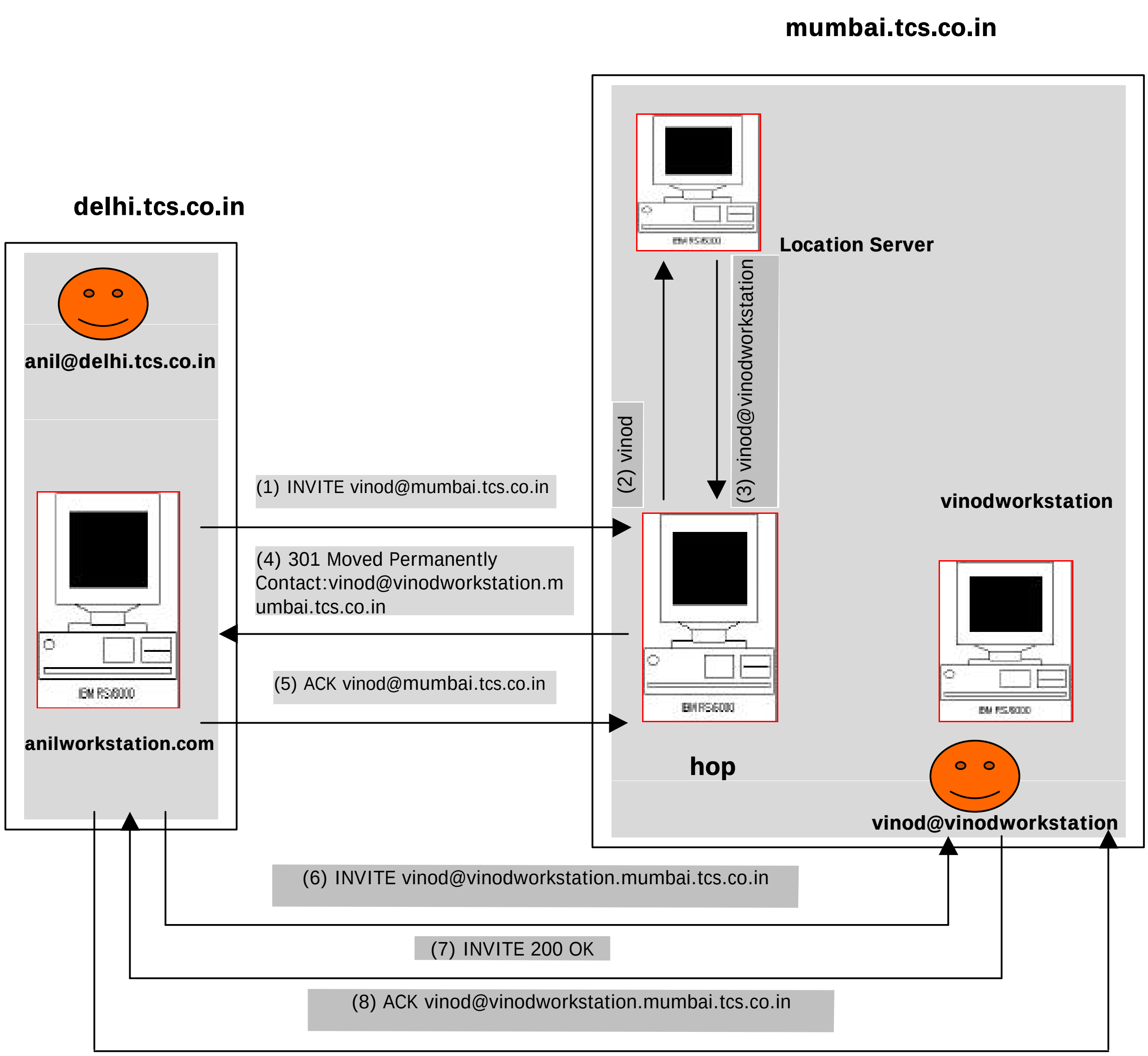


(Figure 2: Handling of INVITE Method by SIP Proxy)

In the *proxy case* (figure 2) the server then issues an INVITE request to the address given by the location server (4). The user agent server at *vinodworkstation* alerts the user (5), who is willing to accept the call. The acceptance is returned to the proxy server, by a 200 status response (6). A success response is then sent by the proxy to the original caller (7). This message is confirmed by caller, with an ACK request (8). The ACK request is then forwarded to the callee (9).

In the redirect case (figure 3) the server returns a redirection response of class 300 giving the address to contact in the Contact header field (4). The response sent is "301: Moved Permanently". The caller acknowledges the response with an ACK request to the server (5).

Based on the URL as specified in the Contact header, the caller issues a new INVITE request with the same Call-ID but a higher CSeq number. This is sent to the address given by the server (6). In this case the call succeeds and a response indicating this is sent to the caller (7). The signaling is completed with an ACK from the caller (8).



(Figure 3: SIP Redirect Server)

As shown in the above two examples, after a successful invitation the necessary parameters for a session are determined. In the case of RTP (Real Time Transport Protocol) audio communication these are the other side's IP-address and port, and the audio codecs to be used. When the RTP connection is set up it is possible to have a conversation. During this conference, either party can change the session by re-inviting the other party or for example invite a third party to the conference.

SIP Mobility

With a PSTN network, *Local Number Portability (LNP)* poses an implementation challenge. However it is a trivial application for SIP services if the user has a domain name, and address such as vinod@tcs.com. With their own domain name, users can actually have service portability by choosing the service provider, for example when on relocation, to host their service. The caller may always use the same address, phone number or URL, but will be redirected transparently to the network, location or device of choice of the called party. Mobility in an IP environment is classified as:

- *Personal mobility* - different terminals, same personal identity (address).
- *Terminal mobility* - the ability to maintain communications when moving a single end system from one subnet to another.
- *Service mobility* - keep same services while mobile.

SIP has been chosen for call control for the 3rd generation wireless network by the 3GPP (Third Generation Partnership Project) initiative.

SIP Interoperability

In the era of network convergence, a key challenge for the network operators and service providers is how to ensure interoperability between different communication protocols. SIP has been widely accepted by service providers because it can deliver enhanced services over next-generation networks. SIP supports interoperability with H.323 and ISUP (ISDN User Part) -- key protocols from both the IP and SS7 environments and hence gives service providers an advantage to offer new SIP services that can go well beyond VoIP. SIP interoperability has been demonstrated in SIP bakeoffs. The purpose of the bake-off is to test for interoperability of SIP implementations, determine the source of incompatibilities, and if the specification is at fault, prepare a "fix" for the SIP draft revision. So far 6 bakeoffs have taken place and leading SIP-products vendors have participated. The number of companies joining these bakeoffs has increased tremendously since the first SIP bakeoff of April 1999.

SIP and H.323

H.323 protocol is an ITU (International Telecommunication Union) standard while SIP is from IETF. SIP and H.323 protocols are architecturally different (although share few similarities) and hence are difficult to interwork. However, many soft-switches (endpoint devices) already support SIP to H.323 translations and interoperability between SIP and H.323 has been demonstrated in bakeoffs and other testing grounds. IETF is working on to develop guidelines for such interworking. Service providers with H.323 networks can use SIP/H.323 conversion boxes, or gateways, that will let them leverage their existing H.323 network. SIP application servers deployed in H.323 networks via conversion boxes will have a huge benefit for the H.323 operators. It will help them to move up to a SIP-based infrastructure and will also protect their existing investments while gaining access to the better scalability, openness, and rich features SIP can deliver.

SIP and ISUP

ISUP protocol is used in SS7 networks to set up and manage basic call connections. Consider a typical phone-to-phone call scenario, in which PSTN calls travels onto the Internet over a gateway and then travels back off to the PSTN. While SIP is used to route calls over the Internet; an extension of SIP called SIP-T (SIP for telephony) helps to preserve the needed ISUP information as the call is carried through the Internet. The ISUP information is carried, byte for byte, in the body of the SIP messages. SIP-T is backward compatible so that it will operate with all existing SIP phones. In IETF, SIP working group looks after the requirements of SIP-T.

SIP Extensions

SIP provides a simple but powerful platform to get its services and features extended. These extensions will help SIP to cope-with the changes of the Internet Telephony industry. This level of flexibility is critical to the rapidly moving VoIP field. SIP enhancements tend to be for specialized services, such as ISUP interworking, QoS (Quality of Service) negotiation, liveness detection, caller preferences or presence/instant messaging. All of these are backward compatible with the basic protocol, with extensions negotiated if both sides support them. Basic calls will succeed without the extensions.

Summary

SIP is a powerful tool for call control and signaling that is gaining tremendous support among service providers and vendors. SIP turns out to be an ideal protocol for providing truly converged applications. This is primarily because it borrows so heavily from other Internet protocols, and in particular, HTTP and SMTP. SIP supports features like MIME (Multipurpose Internet Mail Extensions), URL (Universal Resource Locators) and DNS (Domain Name System), which renders SIP ideal for converged services. SIP supports CPL (Call Processing Language) which enables the users to upload their location information through CPL scripts and then SIP server can take decision based on the CPL script. With the features like mobility (personal, terminal and service) and interoperability, SIP promises to bring a revolution in the internal telephony industry and has surely made a big impact in the network convergence. SIP stack is available commercially in the form of SIP products by leading vendors like dynamicsoft.

References

1. SIP: Session Initiation Protocol, Request for Comments 2543 by Handley, Schulzrinne, Schooler, Rosenberg.

2. The SIP INFO Method, Request for Comments 2976 by S. Donavan, October 2000.

3. The future of converged networks by Jonathan Rosenberg, Communications Solutions, March 2001.

4. SIP: Designed For Interoperability by Jonathan Rosenberg, Internet Telephony – Special Focus, October 2000

5. The Session Initiation Protocol (SIP): A Key Component for Internet Telephony by Jonathan Rosenberg and Richard Shockey, Computer Telephony, June 2000 (Vol. 8, Issue 6).

6. Internet telephony protocols by Linden deCarmo, Dr. Dobbs Journal, July 1999.

7. The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet by Henning Schulzrinne and Jonathan Rosenberg, Bell Labs Technical Journal, Vol. 3, November/December 1998.

8. Signaling for Internet Telephony by Henning Schulzrinne and Jonathan Rosenberg, Columbia University Technical Report CUCS-005-98; submitted for publication, January 1998.

9. Personal mobility for multimedia services in the Internet by Henning Schulzrinne, European Workshop on Interactive Distributed Multimedia Systems and Services, (Berlin, Germany), Mar. 1996.

10. What is SIP, RADVision white paper.

11. Web site http://www.cs.columbia.edu/~hgs/sip/ maintained by Henning Schulzrinne.

12. Web site of Internet Engineering Task Force http://www.ietf.org.