SIP-based VoIP network and its interworking with the PSTN

by Yuan Zhang

The Session Initiation Protocol (SIP) is considered to be a future powerful alternative to the H.323 standard as the signalling system for the dominant Voice over IP (VoIP) communications. This paper provides an in-depth analysis of SIP by describing the SIP protocol stack, summarising the main features of the protocol, and illustrating its architecture, message and operation. The paper also explains the architecture and the two key aspects of signalling interworking when SIP is interconnected with the PSTN.

1 Introduction

Two different architectures are in use today in IP (Internet Protocol) telephony: H.323[1], developed by the International Telecommunication Union (ITU-T), and the Session Initiation Protocol (SIP)[2], developed by the Internet Engineering Task Force (IETF). H.323 was the first signalling protocol standard for VoIP (Voice over Internet Protocol), whereas SIP is gaining increasing popularity. Various vendors are now including SIP in their products without waiting for its full maturity.

From its beginning SIP designers have kept in mind certain critical consideration, such as modularity, integration with Internet services, extensibility and simplicity. However, SIP itself is not a vertically integrated communication system[3]; it is only part of the overall IP telephony architecture, which is composed of a range of packet-switched protocols. Fig. 1 shows the SIP-based IP telephony protocol stack. Note that SIP lies in the application layer and must be used in conjunction with other protocols in order to provide real-time services. For instance, when a SIP signalling session is successfully set up, an RTP[4] (Real-Time Transport Protocol) media session is opened and the virtual session begins.

From Fig. 1 it can be seen that a digitally encoded voice stream is carried over an IP network by using the Real-Time Transport Protocol[4]. The RTP Control Protocol (RTCP)[4] is a companion protocol of RTP and is mainly used for providing quality-related feedback. The Resource ReSerVation Protocol (RSVP)[5] is an optional protocol that will generally result in resources being reserved at each node along the data path to meet the quality-of-service requirements of particular application data streams. The Real-Time Streaming Protocol (RTSP)[6] functions as a network remote control for on-demand delivery of real-time data. The Media Gateway Control Protocol (MGCP)[7] or Megaco[8] is used for media gateway control when interoperating with traditional circuit-switched networks. The main signalling protocol is SIP, which plays the role of H.323 (see the panel) in an H.323-based VoIP network.

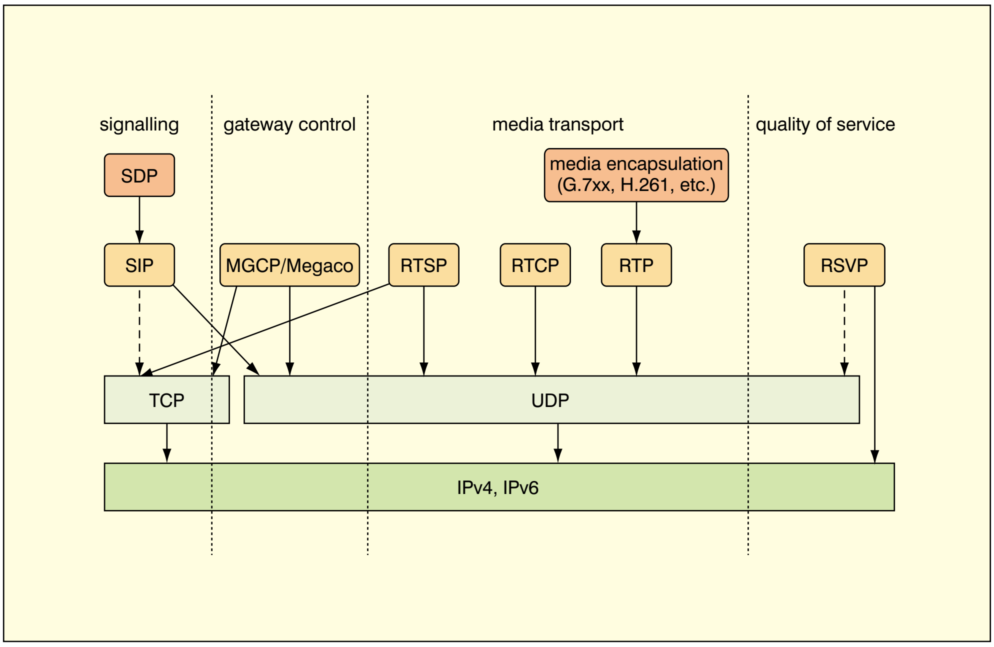


Fig. 1 SIP-based IP telephony protocol stack

The H.323 standard

The first version of H.323, which was intended for multimedia communications over local-area networks (LANs), appeared in 1996. Many found it to be lacking the functions needed for supporting VoIP in a broader environment. Consequently it was revised and H.323 version 2[1]—'Packet-based multimedia communications systems'—was released in 1998. This version of H.323 has received more support than its predecessor, particularly among those network operators and equipment vendors who have a background in more traditional telephony. H.323 is not an individual protocol; rather it is a complete, vertically integrated suite of protocols that defines every component of a VoIP network—terminals, gateways, gatekeepers, MCUs (Multipoint Control Units) and servers with other features. Amongst others, H.323 uses the following standards:

- Q.931 for call set-up
- H.225 for call signalling
- H.245 for exchanging information on terminal capabilities and creation of media channels
- H.245 for RAS-registration, admission and status (RAS) control
- RTP/RTCP for sequencing audio and video packets
- G.711/712, a codec specification
- T.120 for data conferencing.

All these protocols—involving dozens of back-and-forth messages—are called upon in setting up a simple point-to-point voice call. In contrast, SIP is a simple protocol that specifies only what it needs to. For example, SIP works with RTP but does not mandate it.

A major concern of the IETF in developing VoIP standards is to make the best use of existing protocols, with only a few adjustments to fit the new application environment. The hierarchical approach is a distinct difference between VoIP and the traditional circuit-switched telephony (PSTN, the public switched telephone network). In the latter, for example, functions such as call-establishment, billing, routing, and information-exchange are all integrated in the Signalling System No. 7 (SS7) ISDN User Part (ISUP[9]) signalling protocol. In the VoIP world, however, protocol functions are divided into layers; each protocol serves a particular function, allowing for better software modularity, as well as system flexibility and extensibility. End systems or network servers that only provide a specific service need only implement the corresponding protocol, without interoperability problems.

Today, all traditional telephony services depend on the ISDN SS7. However, no translation function for SS7 signalling messages is provided for SIP. The rest of this paper analyses SIP in general and examines some important issues surrounding interworking with the ISUP in some depth.

2 Overview of SIP

Introduction to SIP

SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions such as Internet telephony calls[2].

SIP has the following features:

- It is text-based. This allows easy implementation in object-oriented programming languages such as Java and Perl, allows easy debugging and, most importantly, makes SIP flexible and extensible.
- It involves less signalling[10]. SIP is designed to meet only the basic requirements (create, modify and terminate) of a call-signalling protocol so that the signalling is kept as simple as possible. This means that calls can be established faster than with H.323 call set-up. Rapid call set-up is crucial for high quality of service (QoS). Furthermore, a number of parameters used for the negotiation and establishment of media streams between call participants are encapsulated within the body of the SIP message (by using the Session Description Protocol, SDP[11]). This also speeds up the call set-up procedure.
- It is transport-layer-protocol neutral. Although SIP is designed to be independent of the transport-layer protocol, typically it runs over UDP (User Datagram Protocol) rather than over TCP (Transmission Control Protocol). TCP's connection set-up and acknowledgment routines introduce delays, which are annoying and must be avoided in voice transmission. By adopting UDP, however, the timing of messages and their retransmission can be controlled by the application layer. The destination can be located via multicast, without the need to specify a different TCP channel for each signalling connection.
- Parallel search is possible. A stateful SIP server has the ability to split or 'fork' an incoming call so that several extensions can be rung at once. The first extension to answer takes the call. This feature is handy if a user is working between two locations (a laboratory and an office, for example), or where someone is ringing both a boss and his secretary.

SIP supports five facets of establishing and terminating a call:

- user location—discovery of a user, wherever they are located
- user availability—determination of the willingness of the called party to engage in communications

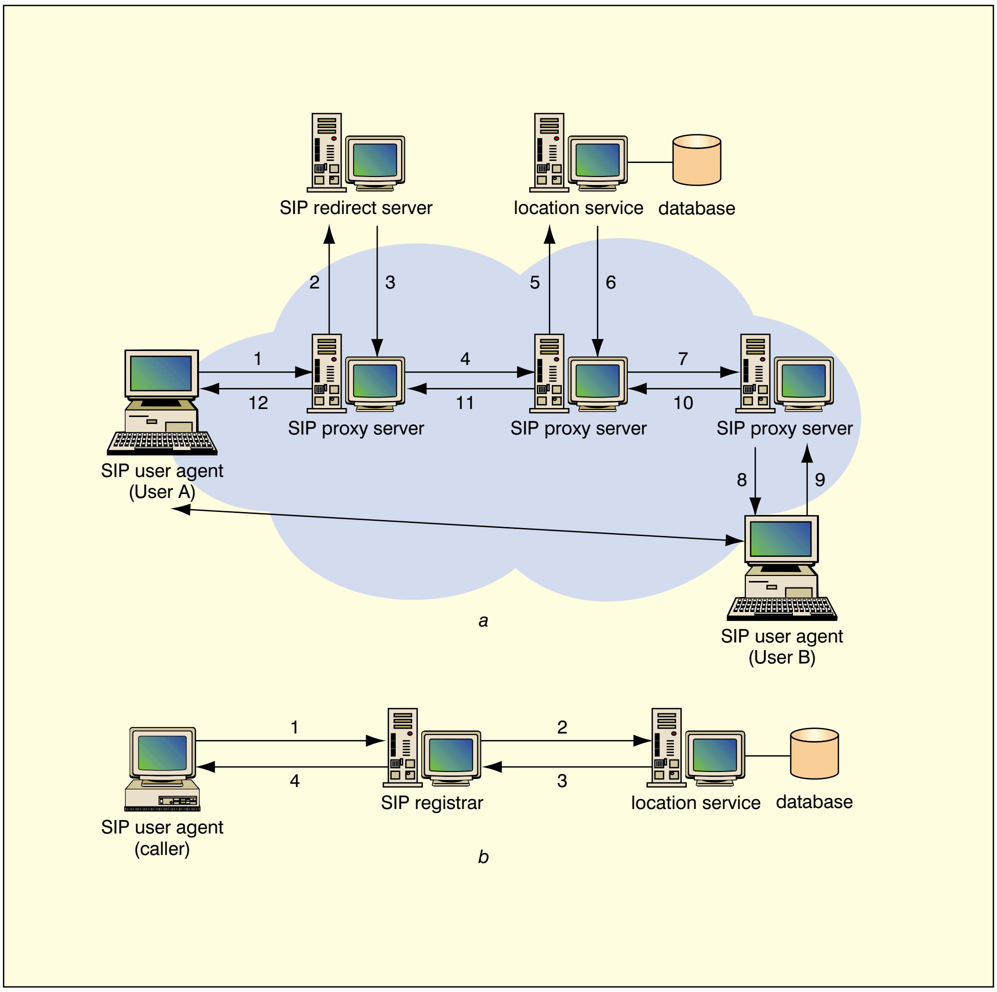


Fig. 2 SIP architecture

- user capabilities—negotiation and determination of the media formats to be used between the calling and called parties
- session set-up—a dialogue is established and media streams flow
- session handling, including transfer and termination of sessions, modifying session parameters and invoking services.

Network architecture

SIP defines two basic classes of network entities: clients and servers. A client is any network element, such as a PC with a headset attachment or a SIP phone, that sends SIP requests and receives SIP responses. A server is a network element that receives requests and sends back responses, which accept, reject or redirect the request. So SIP is a client–server protocol. Note that clients and servers are logical entities. Their roles last only for the duration of a certain transaction, which means that a client might also be found on the same platform as a server. For example, SIP enables the use of proxies, which act as both client and servers for the purpose of making requests on behalf of other clients.

Four different types of servers exist: proxies, user agent servers (UAS), redirect servers and registrars (see Fig. 2).

Proxy servers are application-layer routers that are responsible for receiving a request, determining where to send it based on knowledge of the location of the user, and then sending it there. To other entities it appears as if the message is coming from the proxy rather than from some entity hidden behind the proxy. A proxy must implement both the client and server requirements of one specification. It is also useful for enforcing policy and for traversing firewalls.

A UAS is a logical entity that generates a response to a SIP request and contacts the user. In reality, a SIP device (such as a SIP-enabled telephone) will function as both a user agent client (UAC) and as a UAS, which enables SIP to be used for peer-to-peer communication.

A redirect server is a server that accepts SIP requests, maps the destination address to a set of one or more addresses, and returns the new routing information to the originator of the request. Thereafter, the originator of the request can send a new request to the address(es) returned by the redirect server. A redirect server does not issue any SIP requests of its own.

A registrar acts as a front end to the location service for a domain, reading and writing mappings based on the contents of the REGISTER messages. SIP proxy servers, which are responsible for sending a request to the current host at which the callee is reachable, consult this location service.

Note that the distinction between SIP server types is logical only, not physical. Typically, a registrar is

Table 1: The SIP request methods

SIP message	Description
INVITE	Invites a user to a call
ACK	Used to facilitate reliable message exchange for INVITEs
OPTIONS	Solicits information about a server's capabilities
BYE	Terminates a connection between users or declines a call
CANCEL	Terminates a request, or search, for a user
REGISTER	Registers a user's current location
INFO	Used for mid-session signalling

combined with a proxy or redirect server in a real network.

Overview of the SIP message

As mentioned previously, SIP is text-based (it uses the ISO 10646 character set in UTF-8 [8 bit unicode transformation format] encoding) and uses a similar syntax to the Hypertext Transfer Protocol (HTTP). A SIP message, either a request from a client to a server or a response from a server to a client, is the basic unit of SIP communication. It contains a structured sequence of octets matching a defined syntax. Both Request and Response messages (see Fig. 3) consist of a start-line, one or more headers, an empty line (carriage-return line-feed, CRLF) indicating the end of the headers, and an optional message-body. A general message may look like this:

```
Generic-message =   start-line
                    *message-header
                    CRLF
                    [message-body]
     start-line =   Request-Line | Status-Line
```

Request for Comments RFC 2543 defines six methods that can be used in requests: INVITE, ACK, OPTIONS, BYE, CANCEL, and REGISTER. As there is no general-purpose mechanism to carry session control information during the session, the IETF has added an INFO[12] method to solve this problem. As will be seen in the next subsection, one such piece of session control information is the ISUP signalling message used to control telephony call services. Table 1 gives a brief description of the SIP methods.

The response message contains a status code (a three-digit number) indicating the outcome of an attempt to understand and satisfy the request. Table 2 gives a brief description of the different classes of status codes.

Message headers provide further information about a message and enable the message to be dealt with correctly. In this respect, some headers are akin to the message parameters of ISUP, making mapping between the two possible. RFC 2543 defines a set of different message headers.

The message-body provides a description of the session to be established. The type of media and codec, the sampling rate, etc. are included for negotiation with the called party. By default, SDP[11] is used for this purpose (see the panel on SDP). However, as SIP itself is independent of the characteristics of a session, the session description is delivered as an opaque body.

Session Description Protocol

SDP simply provides a format for describing session information, including session-level parameters and media-level parameters, to potential session participants. Because SDP does not provide a means for transporting or advertising the sessions to potential session participants, it must be used in conjunction with other protocols. For example, SIP carries SDP information within the SIP message body.

To set up an audio session, the caller must know the callee's address(es). In SIP, such addresses look similar to e-mail addresses and are known as SIP Uniform Resource Locators (URLs). Like all URLs, SIP URLs may be placed in Web pages, e-mail messages or printed literature. They contain sufficient information to initiate and maintain a communication session with the resource. SIP URLs also take the form 'user@host', with some parameters attached. Indeed, it is quite likely that users will reuse their e-mail address as their SIP address, avoiding the need to specify a different identifier for each means of communication. For interworking with a traditional circuit-switched network, SIP enables the user portion of the SIP address to be a telephone number. Thus, we could have a SIP address such as:

sip: 3344556789@telco.net

In a given network, such a SIP address could be used to

Table 2: Status code classes

Class	Description	Example
1xx	Informational: request received, continuing to process the request	100 Trying, 180 Ringing
2xx	Successful: the action was successfully received, understood and accepted	200 OK
3xx	Redirection: further action needs to be taken in order to complete the request	302 Moved Temporarily
4xx	Client Error: the request contains bad syntax or cannot be fulfilled at this server	404 Not Found
5xx	Server Error: the server failed to fulfil an apparently valid request	501 Not Implemented
6xx	Global Failure: the request cannot be fulfilled at any server	603 Decline

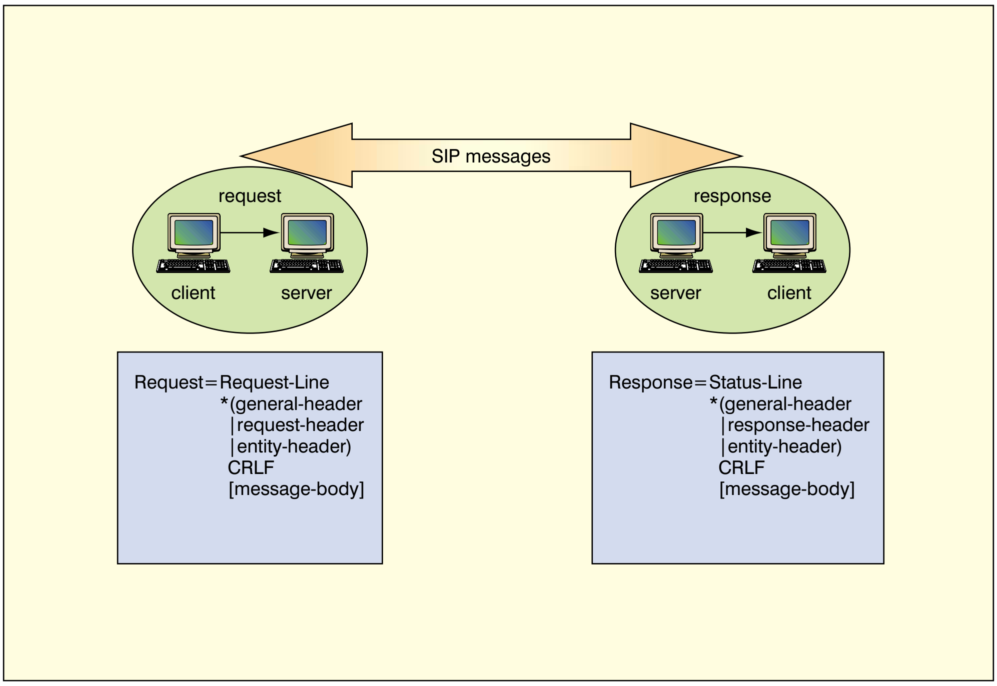


Fig. 3 SIP messages

cause routing of media to a gateway that interfaces with the traditional telephone company. To clearly indicate that a call is to a telephone number, the URL could be supplemented with the parameter user=phone. In such a case, the URL would have the following format:

sip: 3344556789@telco.net; user=phone.

Overview of operation

The INVITE request is the most fundamental and important SIP request in establishing a call. The following example (see Figs. 4 and 5) of a SIP message exchange between two users, Shirley and Dan, illustrates the basic functions of SIP.

In this example, Dan, who resides in the domain home.com, wants to call Shirley. Usually they reside within the same domain, so Dan may use a softphone (SIP-based) to send an INVITE for sip: shirley@home.com (Shirley's SIP URL) to a local proxy server, shown in the Fig. 5 as home.com proxy server. The INVITE (F1) request contains a number of header fields and it might look like this:

```
INVITE sip: shirley@home.com SIP/2.0
Via: SIP/2.0/UDP 202.194.1.1: 5060
From: Dan <sip: dan@home.com>; tag=740924
To: Shirley <sip: shirley@home.com>
Call-ID: 123456abcd@202.194.1.1
Cseq: 1 INVITE
Contact: <sip: dan@202.194.1.1>
Content-Type: application/sdp
Content-Length: 142
(message body omitted)
```

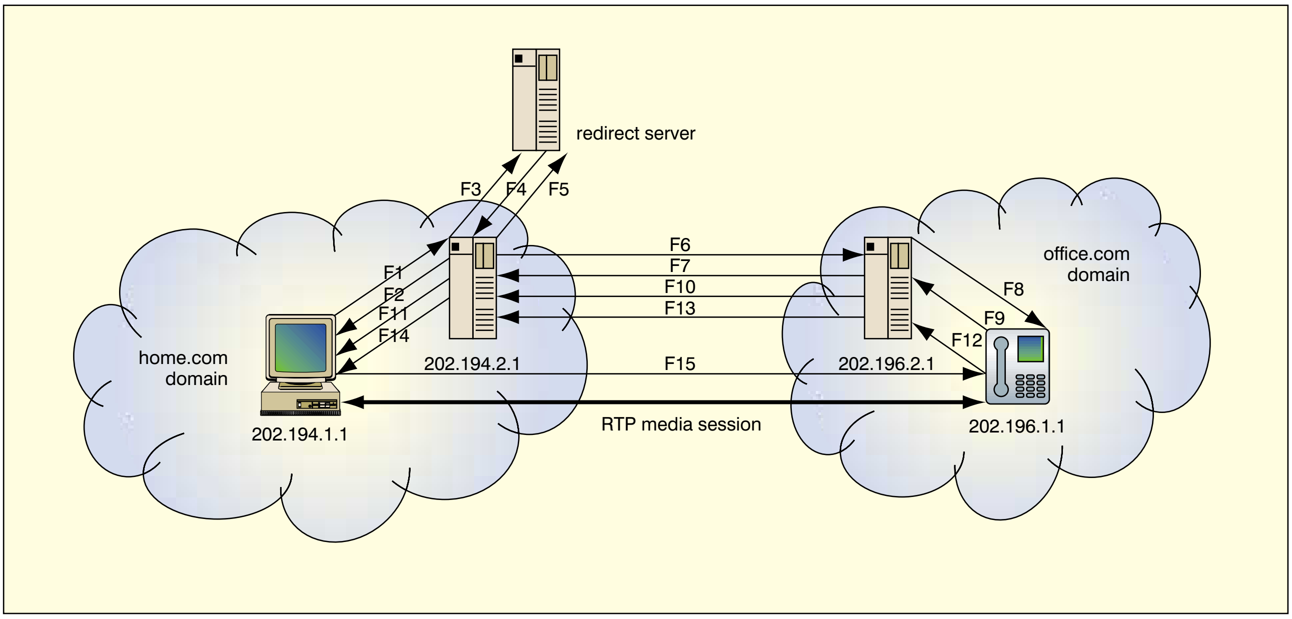
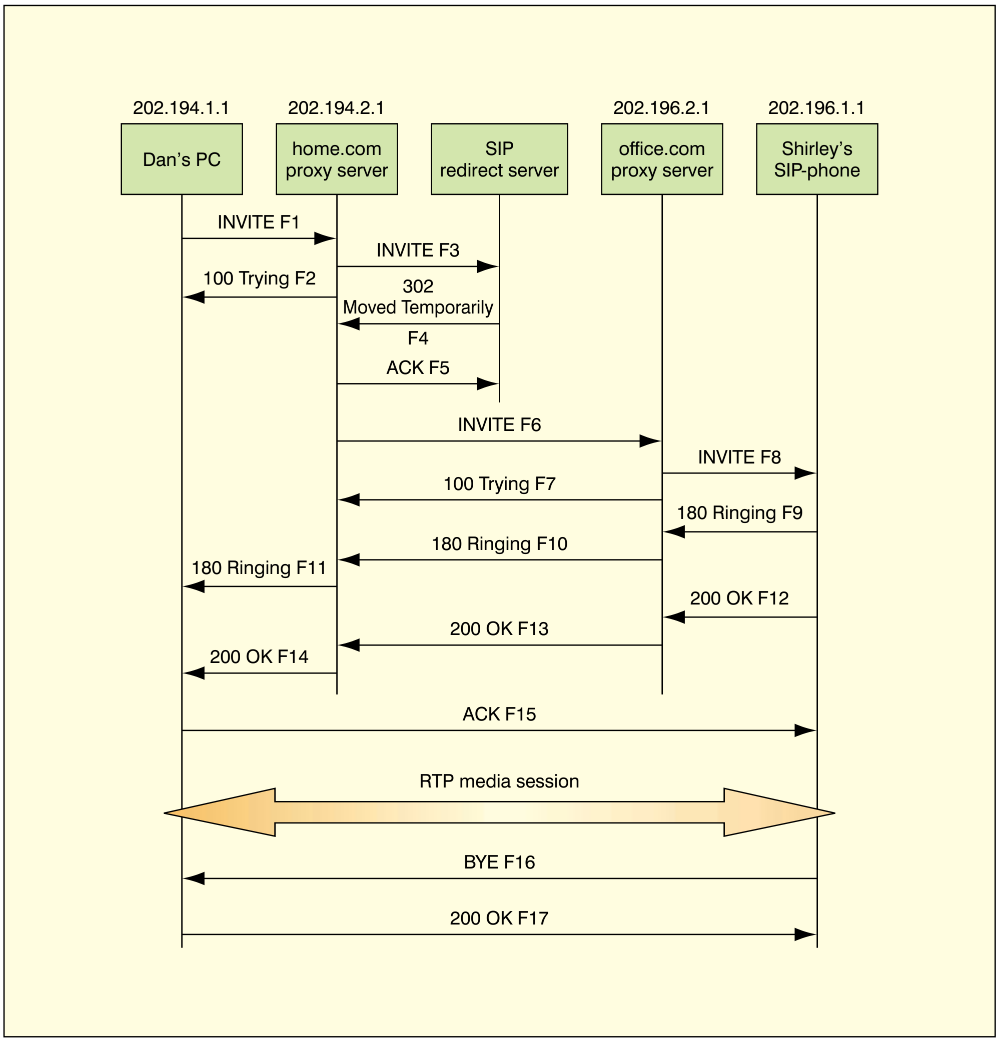


Fig. 4 SIP operation in redirect mode

Fig. 5 Example of a SIP session set-up



The home.com proxy server receives the INVITE request and generates a 100 Trying response, which is sent back to Dan's softphone to indicate the proxy's working state. During the course of locating Shirley, one SIP network server can proxy or redirect the call to additional servers until it arrives at one that definitely knows the IP address at which Shirley can be found. Here, for simplicity, suppose the home.com proxy server sends the INVITE to a redirect server to try to identify Shirley's current location.

The redirect server determines that Shirley does not presently reside within the domain home.com, but is reachable at office.com. (Shirley must have registered the current host on which she resides.) The redirect server returns this information to the home.com proxy server in a 302 Moved Temporarily response, which gives the new address for Shirley as sip: shirley@office.com. The 302 response message (F4) might look like this:

```
SIP/2.0 302 Moved Temporarily
Via: SIP/2.0/UDP 202.194.1.1: 5060
From: Dan <sip: dan@home.com>; tag=740924
To: Shirley <sip: shirley@home.com>
Call-ID: 123456abcd@202.194.1.1
Cseq: 1 INVITE
Contact: <sip: shirley@office.com>
Contact-Length: 0
```

As this represents a final response to the INVITE, the proxy server 'ACKs' this response. Then the home.com proxy server has a choice: it can either return the 302 response directly to Dan for him to try again, or it can try the suggested location itself on Dan's behalf. In this example, the home.com proxy server attempts to locate sip: shirley@office.com (possibly by performing a DNS [Domain Name Server] look-up) by forwarding the INVITE to the proxy server of the office.com domain. The new INVITE message (F6) might look like this:

```
INVITE sip: shirley@office.com SIP/2.0
Via: SIP/2.0/UDP 202.194.2.1: 5060
Via: SIP/2.0/UDP 202.194.1.1: 5060
From: Dan <dan@home.com>; tag=740924
To: Shirley <sip: shirley@office.com>
Call-ID: 123456abcd@202.194.1.1
Cseq: 2 INVITE
Contact: <sip: dan@202.194.1.1>
Content-Type: application/sdp
Content-Length: 142
(message body omitted)
```

The office.com proxy server, which controls the domain office.com, receives the INVITE and responds with a 100 Trying response back to the home.com proxy server to indicate that it has received the INVITE and is processing

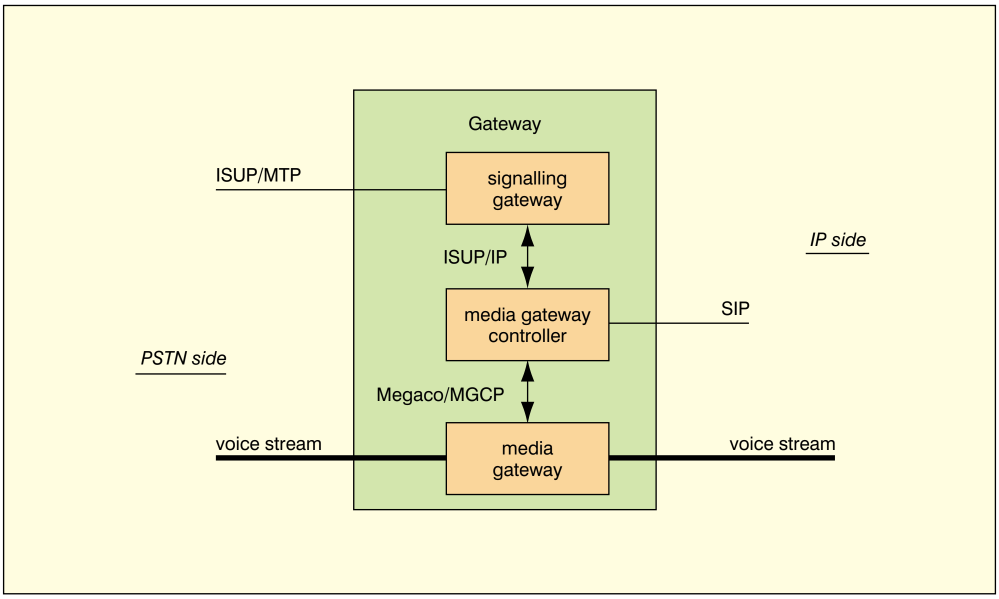


Fig. 6 A functional description of a SIP–PSTN gateway

the request. Of course, in real life there may be many more hops than this. The office.com proxy server then locates Shirley (by consulting a location service, e.g. DNS look-up) and completes the routing of the INVITE. Shirley's SIP phone rings and sends a 180 Ringing response back (through the two proxies) to Dan. Shirley then picks up the handset and a 200 OK response is sent to indicate that the call is accepted. The 200 OK message (F12) might look like this (note that there are three Via headers—one added by Dan's PC, one added by the home.com proxy, and one added by the office.com proxy):

```
SIP/2.0 200 OK
Via: sip/2.0/UDP 202.196.2.1: 5060
Via: sip/2.0/UDP 202.194.2.1: 5060
Via: sip/2.0/UDP 202.194.1.1: 5060
From: Dan <dan@home.com>; tag=740924
To: Shirley <sip: shirley@office.com>; tag=780101
Call-ID: 123456abcd@202.194.1.1
Cseq: 2 INVITE
Contact: <sip: shirley@202.196.1.1>
Content-Type: application/sdp
Content-Length: 132
(message body omitted)
```

Finally, an ACK is sent directly from Dan to Shirley, bypassing the two proxies, to confirm the reception of the 200 OK response. Then their media session begins. Generally, the end-to-end media packets will take a different path from the SIP signalling messages to the same destination, depending on varying network traffic conditions and other factors. At the end of the call, Shirley hangs up, which causes a BYE message to be sent. Dan sends a 200 OK to confirm receipt of the message and the call is over.

From this example, we can see how easy SIP is to implement. Its flexibility and power shouldn't be belied, however. As we shall see in the next section, SIP also fits well with those protocols that are used for media gate control and hence it forms part of the overall architecture known as softswitch.

3 SIP interworking with the SS7

SIP–PSTN gateway architecture

Although performing telephony call signalling and transporting the associated audio media over IP offers significant advantages, such as lower network implementation costs, lower bandwidth requirements, integration of voice and data applications and new features, it is not feasible to replace all existing circuit-switched telephony. Traditional circuit-switched networks (PSTN) and VoIP networks will coexist for a long time, making their seamless interworking a vital issue.

Interworking with PSTN usually involves three call set-up scenarios: PSTN to PSTN via intermediate SIP networks; PSTN to SIP terminator; and SIP terminator to PSTN. In all cases, a SIP gateway (GW) is involved in connecting the PSTN with the Internet.

As it is difficult to implement signalling and media transformation processing in a single monolithic structure, a gateway is decomposed into three functional components: a signalling gateway (SG), a media gateway (MG), and a media gateway controller (MGC)[13]. Fig. 6 shows the architecture of a distributed gateway:

- The signalling gateway receives and routes all ISUP messages for the media gateway. More specifically, the lower layers of SS7 (the Message Transfer Part [MTP], see Fig. 7) are replaced by IP. The upper layers of SS7 (the ISDN User Part [ISUP]) are encapsulated into TCP/IP headers and transmitted over an IP interface to a signalling gateway. It is the responsibility of the signalling gateway to translate the dialled number into an IP address before the call is routed over the IP network.
- The media gateway provides mapping or transcoding between media in the PSTN domain (e.g. PCM encoded voice) and media in the IP domain (e.g. media transported over RTP/UDP/IP).
- The media gateway controller accepts signalling from the PSTN in native format and converts it to the format

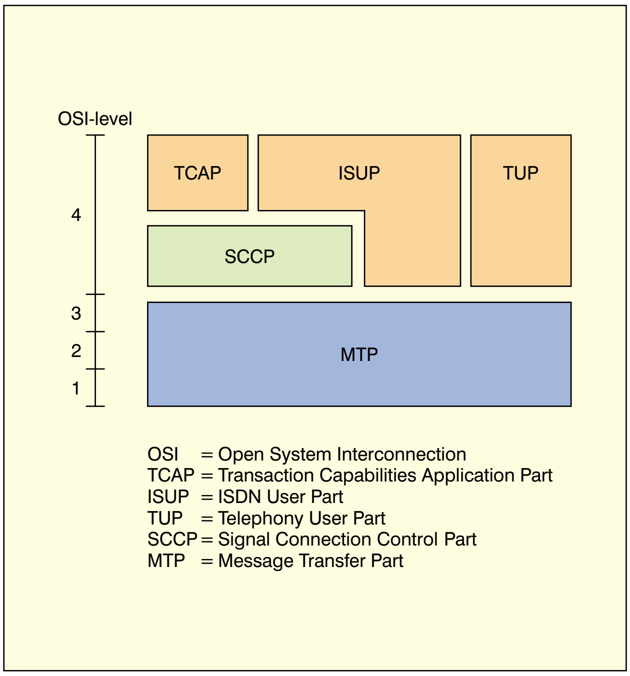

OSI-level
4
3
2
1
TCAP
ISUP
TUP
SCCP
MTP
OSI = Open System Interconnection
TCAP = Transaction Capabilities Application Part
ISUP = ISDN User Part
TUP = Telephony User Part
SCCP = Signal Connection Control Part
MTP = Message Transfer Part

Fig. 7 The SS7 protocol stack

that the IP network uses. It also controls the media gateway(s) by introducing Megaco/MGCP and performs 3A functions (authentication, authorisation and accounting). In reality, the media gateway and the media gateway controller are often merged together in one physical device.

The media conversion and transport can be regarded as a slave function, which is invoked and manipulated to meet the needs dictated by signalling. So signalling conversion is of major concern. SIP is a signalling protocol and in most cases the corresponding signalling protocol used in PSTN is the ISDN User Part (ISUP) of SS7. A media gateway controller has logical interfaces to both networks, the network carrying ISUP and the network carrying SIP. It is used to form a bridge between SIP and ISUP networks so that calls originating in the PSTN can reach IP telephone end points and vice versa.

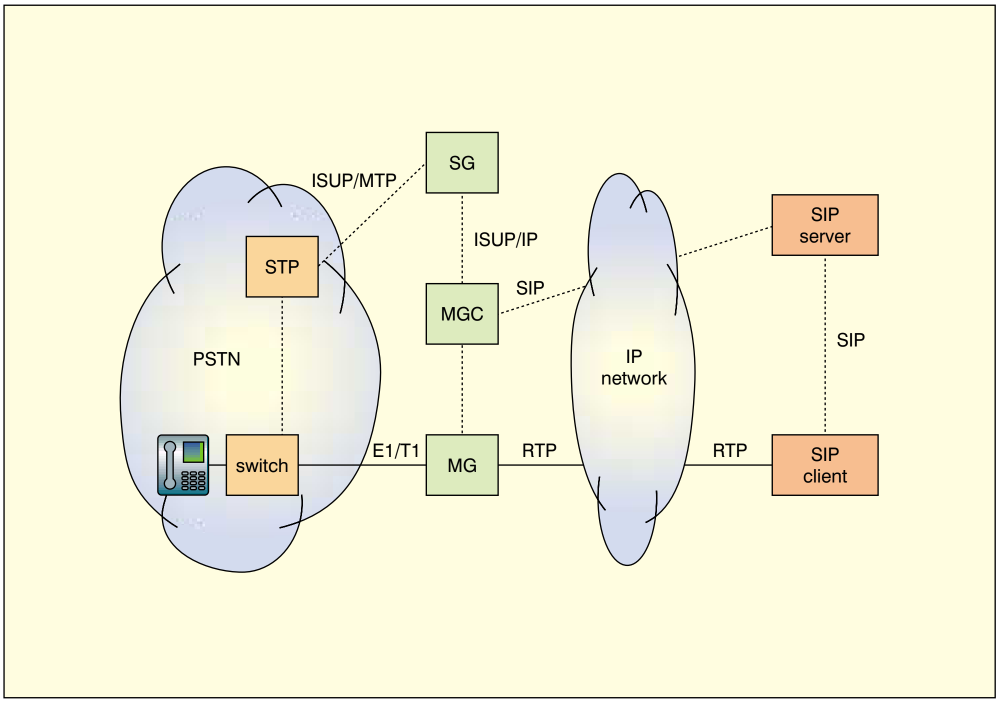
Signalling interworking

First developed in the late 1960s, Signalling System Number 7 has gradually become the backbone of today's communication networks. It performs out-of-band signalling in support of the call-establishment, billing, routing, and information-exchange functions of the PSTN. In order to seamlessly integrate the IP network with the PSTN, it is important to retain the SS7 information (ISUP) at the points of interconnection and to use this information for the purpose of call establishment.

As already mentioned, the media gateway controller is the entity that implements the mapping between SIP and ISUP: it speaks ISUP to the PSTN and SIP to the IP network and converts between the two. Usually an MGC preserves the ISUP information received from the PSTN by encapsulating it in the SIP message for forwarding on. Thus, transparency of ISUP features not otherwise supported in SIP can be ensured[14]. SS7 information is available without any loss to the SIP network across the PSTN–IP interface. The MGC might package both SDP and ISUP elements into the same SIP message by using the MIME (Multipurpose Internet Mail Extensions)[15] multipart format. Relevant problems, such as the mapping between SIP and ISUP, the framework of SIP for telephony, etc., are under discussion within the IETF SIPPING working group[16].

On the other hand, certain information is translated from an SS7 ISUP message to an SIP message in order to allow SIP elements, such as proxy servers, to make appropriate routing decisions. This issue focuses on ISUP parameter-SIP header mapping. For instance, the

Fig. 8 General ISUP–SIP conversion



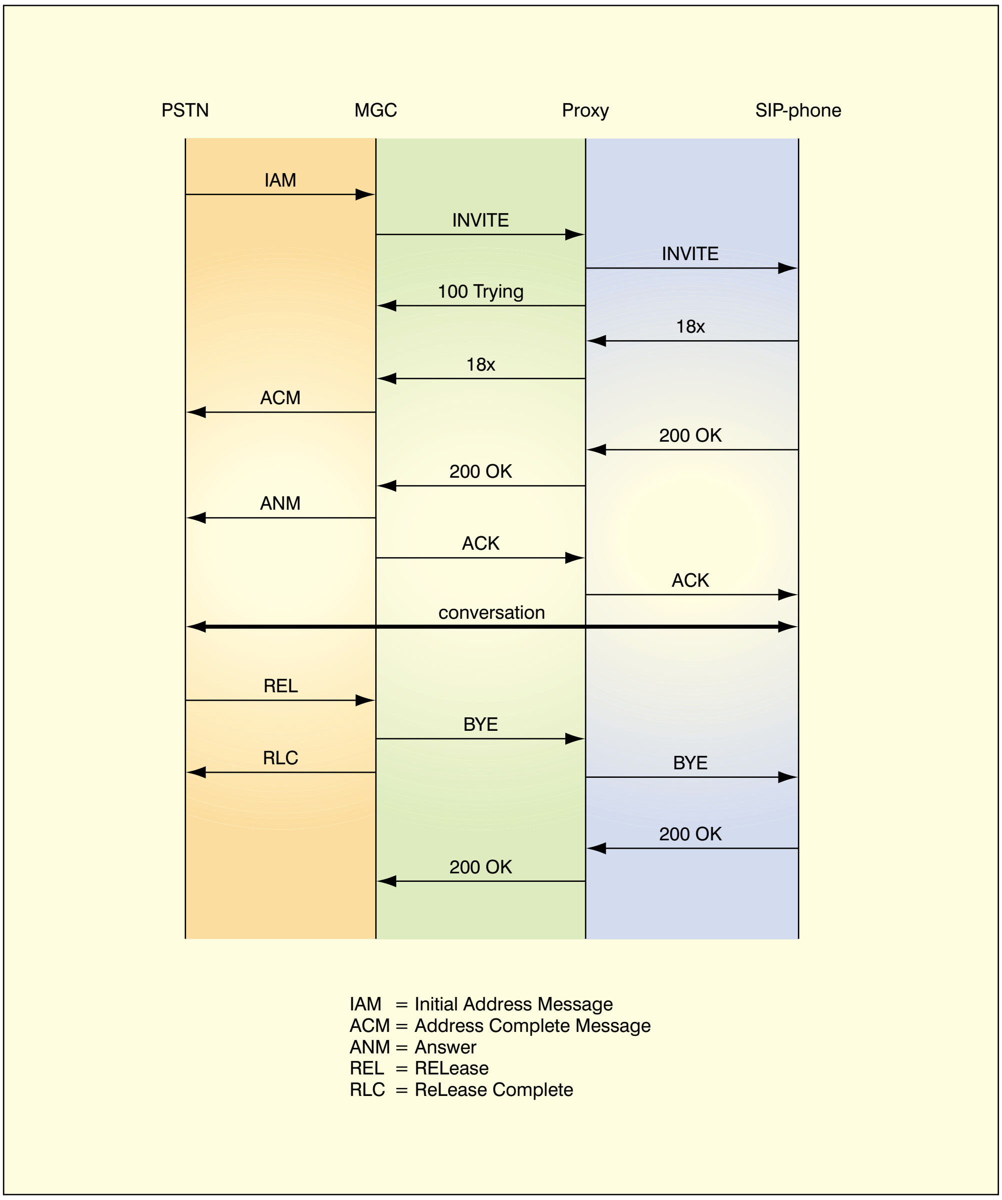


Fig. 9 A PSTN–IP call flow

mapping between the ISUP Initial Address Message (IAM) parameter and the SIP INVITE message headers is of great importance. Once an INVITE has been sent for a particular session, headers such as the To and From fields essentially become fixed, and no further translation will be required during subsequent signalling, which is routed in accordance with Via and Route headers. It is necessary to specify the rules that govern the mapping between ISUP and SIP messages. Further information on this topic is given in Reference 17.

In the case of PSTN terminations, the media gateway controller at the egress generates the ISUP message, sometimes after modification, either from the SIP message body or from the SIP headers.

A signalling interworking example

Fig. 8 shows a call that originates from the PSTN and terminates at a SIP phone. Fig. 9 shows the flow for a simple successful call, involving a general ISUP–SIP conversion for a PSTN-originated call terminating in IP.

When a PSTN user wishes to begin a session with a SIP user, they just dial the user part of the callee's SIP URL. Then the PSTN network generates an initial address message (IAM) and routes it towards an appropriate media gateway controller. The latter is the point of ingress for message flows over the IP network for this call. Upon receipt of the IAM, the media gateway controller takes the steps needed to preserve the ISUP information. It forms the SIP INVITE from the ISUP through encapsulation and translation. This might, for instance, involve setting the 'To' field in the INVITE to the Called Party Number of the ISUP IAM. The media gateway controller then encapsulates the ISUP IAM into the SIP INVITE and sends it to the appropriate SIP node. The following steps in Fig. 9 illustrate how the SIP message is handled and the generation of an ISUP message by the media gateway controller according to the response (18x, 200, etc.) it receives. Note that the terminator (SIP-phone) has no use for the encapsulated ISUP and disregards it. Finally the calling party hangs up and generates a release message, upon receipt of which the media gateway controller will send a BYE message towards the SIP network. The media gateway controller also frees the PSTN circuit and indicates that it is available for reuse by sending an release complete (RLC) message. Finally a 200 OK response is sent by the SIP-phone and the call is over. This call-flow provides a clear understanding of the signalling system of the real VoIP world.

4 Future work and conclusion

Although SIP has a bright future, it is still very much in the development phase. A review of the solutions offered by different vendors reveals that there is more support for H.323 than SIP. However, the rate at which SIP voice gateways are being developed is tremendously high.

Potential problems exist when the softswitch network provides a transit function between PSTN and SIP-based VoIP networks. Since different organisations design protocols in different ways, there is seldom a direct match between the messages and parameters of one protocol and those of another. In addition, interworking has to address the differences between state machines, timers, etc. The result, although workable, is often less than perfect. Consequently, there are many issues associated with ISUP–SIP interworking that need resolution, including the following:

- There are several flavours of ISUP, which lead to different message flows, i.e. to different exchanges of ISUP messages during a call. When a SIP network connects two PSTNs, the egress media gateway controller should follow specific rules to deal with this problem (e.g. suspend and resume for terminal portability).
- The ingress media gateway controller might package both SDP and ISUP elements into a SIP message using the MIME multipart format. The terminator device, however, may not support a multipart payload or the ISUP MIME type, with the result that the SIP request will be rejected.
- For a PSTN-originated call, a SIP message is produced at the ingress media gateway controller as a result of ISUP encapsulation and translation. Sometimes this message (especially if it is an INVITE) may undergo transformation at the hands of an application server. Normally only the SIP headers will be modified, with the result that the header information does not agree with the encapsulated ISUP—a violation of the rules. Also part of the encapsulated ISUP may be rendered irrelevant and obsolete. Rules that delineate the preferred behaviour of the entities in question (whether originating or terminating) under the specific circumstances surrounding each such case need to be formulated.
- European phone numbers do not have a fixed length so that the ingress media gateway controller cannot know when the number is complete.
- The transmission of ISUP in SIP bodies may provide opportunities for abuse and fraud, especially by SIP-phones, and this raises security problems.

In view of the above, it can be seen that it is still too early to consider SIP as a replacement for H.323 in the near future. Much work needs to be done before SIP will provide carrier-grade services. The market will finally determine which protocol becomes the acknowledged leader. However, if technical aspects and the introduction of novel integrated services are the governing factors then SIP's chances of success are high.

Yuan Zhang obtained his Bachelor's degree in Electrical Engineering from Nankai University in 1996. He is currently pursuing a Master's degree in Computer Engineering from Shandong University. His research interests include Internet real-time and multimedia services and protocols, modelling and analysis of computer communication networks, and broadband networking.

Address: School of Information Science and Engineering,
Shandong University, P. R. China
E-mail: zy_dandan@sdu.edu.cn

References

1 ITU-T Recommendation H.323: 'Packet-based multimedia communications systems', February 1998

2 HANDLEY, M., SCHULZRINNE, H., SCHOOLER, E., and ROSENBERG, J.: 'SIP: Session Initiation Protocol'. RFC 2543, IETF, March 1999

3 ROSENBERG, J., *et al.*: 'SIP: Session Initiation Protocol'. RFC3261, IETF, June 2002

4 SCHULZRINNE, H., CASNER, S., FREDERICK, R., JACOBSON, V.: 'RTP: a transport protocol for realtime applications'. RFC 1889, IETF, January 1996

5 BRADEN, R. (Ed.), *et al.*: 'Resource ReSerVation Protocol (RSVP)—Version 1. Functional Specification'. RFC 2205, IETF, September 1997

6 SCHULZRINNE, H., RAO, A., and LANPHIER, R.: 'Real Time Streaming Protocol (RTSP)'. RFC 2326, IETF, April 1998

7 ARANGO, M., *et. al.*: 'Media Gateway Control Protocol (MGCP) Version 1.0'. RFC 2705, IETF, October 1999

8 CUERVO, F., *et al.*: 'Megaco Protocol Version 1.0'. RFC 3015, IETF, November 2000

9 'Signaling System No. 7; ISDN user part signaling procedures'. ITU-T Q.764 Recommendation, September 1997

10 COLLINS, D.: 'Carrier grade voice over IP' (McGraw-Hill, 2000), p.169

11 HANDLEY, M., and JACOBSON, V.: 'SDP: Session Description Protocol'. RFC 2327, IETF, April 1998

12 DONOVAN, S.: 'The SIP INFO method'. RFC 2976, IETF, October 2000

13 ONG, L., *et. al.*: 'Framework Architecture for Signaling Transport'. RFC 2719, IETF, October 1999

14 VEMURI, A., and PETERSON, J.: 'S Initiation Protocol for Telephones (SIP-T): Context and architectures'. RFC 3372, IETF, September 2002

15 ZIMMERER, E., et al.: 'MIME media types for ISUP and QSIG objects'. RFC 3204, IETF, December 2001

16 See http://www.ietf.org/html.charters/sipping-charter.html

17 CAMARILLO, G., *et al.*: 'ISUP to SIP mapping'. Internet Draft, <draft-ietf-sipping-isup-06>, IETF, August 2002 (work in progress)

First received 13th April and in revised form 14th August 2002.