SIP-Based VoIP Network And Its Interworking With The PSTN

Zhang Yuan
(Faculty of Information Engineering and Technology, Shandong University, P.R.China, 250100)

Abstract

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

Key words  SIP,  Interwork,  ISUP

Introduction

Today in IP telephony industry, there are two different architectures: H.323, developed by ITU-T, and the Session Initiation Protocol (SIP) [1], developed by IETF. H.323 is the first standardized signaling protocol for VoIP, while SIP is gaining more and more popularity. Various vendors are including SIP in their products without waiting for its full maturity.

SIP designers have kept some crucial aspects, such as modularity, integration with Internet services, extensibility and simplicity, in their minds from the early beginning. However, SIP itself is not a vertically integrated communication system [2]. It is only part of the overall IP telephony architecture, which is composed of a range of packet-switched protocols. Figure 1 shows the SIP-based IP telephony protocol stack. Note that SIP lies in the application-layer and should be used in conjunction with other protocols in order to provide real-time services.

In figure 1, we see that digitally encoded voice stream is carried over an IP network by using the real-time transport protocol (RTP) [3]. The RTP control protocol (RTCP) [3] is a companion protocol of RTP and is mainly used for providing quality-related feedback. The resource reservation protocol (RSVP) [4] is an optional protocol that will generally result in resources being reserved in each node along the data path to meet the specific quality of service for particular application data streams. The real-time streaming protocol (RTSP) [5] functions as a network remote control over on-demand delivery of real-time data. The media gateway control protocol (MGCP) [6] or Megaco [7] is used for media gateway control when interoperating with traditional circuit-switched networks. The main signaling protocol is SIP, which plays the role of H.323 in an H.323-based VoIP network.

Thus, we can see that one of IETF’s major concerns of developing VoIP standards is to make the best of existing protocols, with only a few adjustment to fit the new application environment. The hierarchical approach is a distinct difference between VoIP and the traditional circuit-switched telephony (PSTN). For example, functions, such as call-establishment, billing, routing, and information-exchange, are all integrated in the SS7 ISDN User Part (ISUP [9]) signaling protocol. While in the VoIP world, 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 that particular protocol, without interoperability problems.

Today, the entire traditional telephony services depend on the ISDN SS7. While for SIP, no translation function for SS7 signaling messages is provided. In the following sections, the article analyzes the SIP in general and examines some important issues surrounding interworking with the ISUP in some depth.

**Survey of SIP**

1. Introduction to SIP

SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions such as Internet telephony calls [1].

SIP has the following features:

- **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.

- **less signaling** [15]: SIP is designed to meet the basic requirements (create, modify and terminate) of a call-signaling protocol and only goes a bit far in order to keep the signaling as simple as possible. This means that calls can be established faster, and rapid call setup is crucial to high QoS. Further more, a number of parameters, used for the negotiation and establishment of media stream, between call participants, are encapsulated within the SIP message body (by using session description protocol [8]). This also speeds the call set up procedure.

- **transport-layer-protocol neutral**: Although SIP is designed to be independent of the transport-layer protocol, typically it runs over UDP rather than over TCP. TCP’s
connection-setup and acknowledgement 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 signaling connection.

- parallel search: 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 lab and an office, for example), or where someone if ringing both a boss and their secretary.

SIP supports five facets of establishing and terminating a call.

- User location: discovery of a user wherever located;
- User availability: determination of the called party on whether to join or not;
- User capabilities: negotiation and determination of the media formats to be used between the calling and called party;
- Session setup: a dialog is established and audio streams flow;
- Session handling: including transfer and termination of sessions, modifying session parameters and invoking services.

2. Network Architecture

SIP defines two basic classes of network entities: client and server. A client is any network element, such as a PC with a headset attachment or a SIP phone, which 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 client and server are logical entities. Their roles last only for the duration of a certain transaction, which means a client might also be found within 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: proxy, user agent server (UAS), redirect server and registrar.

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 those 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 requirement of one specification. It is also useful for enforcing policy and for firewall traversal.

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. Then SIP proxy servers, which are responsible for sending a request to the current host that the callee is reachable at, will consult this
location service.

Do note that the distinction between types of SIP servers is only logical, not physical. Typically, a registrar is combined with a proxy or redirect server in a real network.

3. Overview of SIP Message

As mentioned previously, SIP is text-based (uses the ISO 10646 character set in UTF-8 encoding) and has 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 consist of a start-line, one or more headers, an empty line indicating the end of the headers, and an optional message-body. A general message may look like this:

\[
\text{Generic-message = start-line} \\
\quad \text{*message-header} \\
\quad \text{CRLF} \\
\quad \text{[message-body]} \\
\text{start-line = Request-Line / Status-Line}
\]

RFC2543 defines six methods that can be used in requests: INVITE, ACK, OPTIONS, BYE, CANCEL, and REGISTER. Seeing that there is no general-purpose mechanism to carry session control information during the session, IETF adds an INFO [10] method to solve this problem. We can see in the next section that one such session control information is ISUP signaling message used to control telephony call services. Table 1 gives a brief description of SIP methods.

<table>
<thead>
<tr>
<th>SIP Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE</td>
<td>Invites a user to a call</td>
</tr>
<tr>
<td>ACK</td>
<td>Used to facilitates reliable message change for INVITEs</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>Solicits information about a server’s capabilities</td>
</tr>
<tr>
<td>BYE</td>
<td>Terminates a connection between users or declines a call</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Terminates a request, or search, for a user</td>
</tr>
<tr>
<td>REGISTER</td>
<td>Registers a user’s current location</td>
</tr>
<tr>
<td>INFO</td>
<td>Used for mid-session signaling</td>
</tr>
</tbody>
</table>

Table 1

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

Message headers provide further information about a message and enable the right dealing with the message. In that respect, some headers are akin to the message parameters of ISUP, making mapping between the two possible. RFC2543 defines a serial of different message headers. Please refer to it for more details.

The message body describes a description of the session to be established. The type of media, codec, sampling rate, etc are included for negotiation with the called party. By default, SDP is used for this purpose. But SIP itself is independent of the characteristics of a session, such that the session description is delivered as an opaque body.

To setup 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, email messages or printed literature. They contain sufficient information to initiate and maintain a communication session with the resource. SIP URLs also take the form of user@host, with some parameters attached. Indeed, it is quite likely that one will reuse his e-mail address as his SIP address, avoiding the need to specify a different identifier for each means of communication.

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

Table 2

4. Overview of Operation

To establish a call, the INVITE request is the most fundamental and important SIP request. The following example of a SIP message exchange between two users, Shirley an Dan, shows the basic functions of SIP. Hope it will facilitate the understanding of the procedure of services offered by 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 figure as home.com Proxy Server. The INVITE 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: xxx
(Dan’s SDP omitted)
```

The home.com Proxy Server receives the INVITE request and generates a 100 Trying response, which is sent back to Dan’s softphone, indicating 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 know the IP address where 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.

Figure 3: SIP session setup example

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 home.com Proxy Server in a 302 Moved Temporarily response which gives the new address to locate Shirley as sip: shirley@office.com. The 302 response message 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>
As this represents a final response to the INVITE, the Proxy Server ACKs this response. Then 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, home.com Proxy attempts to locate sip: shirley@office.com (possibly by performing a DNS lookup) by forwarding it to the proxy server of office.com.

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@202.196.1.1>
Content-Type: application/sdp
Content-Length: xxx
```

(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 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) and completes the routing of the INVITE. Shirley’s SIP home 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 might look like this:

```
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: xxx
```

(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 signaling messages. 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 can see in the next section, SIP also fits well with those protocols that are used for media gate control—and as such, it forms part of the overall architecture known as softswitch.

**SIP Interworking with the SS7**

1. SIP-PSTN Gateway Architecture

   Although performing telephony call signaling and transporting the associated audio media over IP offer significant advantages, such as lower cost of network implementation, lower bandwidth requirements, integration of voice and data applications and new features, it is not feasible that all existing circuit-switched telephony should be replaced. The fact is that 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 concerns three call setup 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.

   In view of the deficiency of monolithic packaging of signaling and media transformation into one box, a GW is decomposed into three functional components: signaling gateway (SW), media gateway (MG), and media gateway controller (MGC) [11]. Figure 4 shows the architecture of a distributed GW.

   ![Figure 4: A functional description of SIP-PSTN Gateway](image)

   • SG: The SG receives and routes all ISUP messages for the MG. More specifically the lower layers of SS7 (the MTP) are replaced by IP. The upper layers of SS7 (the ISUP) are encapsulated into TCP/IP headers and transmitted over an IP interface to a SG. So it is the responsibility of the SG to translate the dialed number into an IP address before the call could be routed over the IP network.

   • MG: The MG maps or transcodes the media in the PSTN domain (e.g., PCM encoded voice) and media in the IP domain (e.g., media transported over RTP/UDP/IP).

   • MGC: The MGC accepts Signaling from PSTN in native format and converts it to the format that the IP network uses. MGC also controls the MG(s) by introducing Megaco/MGCP and performs the functions of 3A (Authentication, Authorization and
Accounting. In reality, the MG and the MGC are often merged together in one physical device.

The media conversion and transport can be considered a slave function, invoked and manipulated to meet the needs dictated by signaling. So signaling conversion is of major concern in this article. SIP is a signaling protocol. The corresponding signaling protocol used in PSTN, in most cases, is the ISDN User Part (ISUP) of SS7. An MGC has logical interfaces facing both networks, the network carrying ISUP and the network carrying SIP. It is used to bridge SIP and ISUP networks so that calls originating in the PSTN can reach IP telephone endpoints and vice versa.

2. Signaling Interworking

First developed in the late 1960s, SS7 gradually becomes the backbone of today’s communication network. It performs out-of-band signaling 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 inter-connection and use this information for the purpose of call establishment.

As mentioned early, MGC is the entity that implements the mapping between SIP and ISUP. It speaks ISUP to 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 further progressing. Thus, transparency of ISUP features not otherwise supported in SIP can be ensured [12]. SS7 information is available without any loss to the SIP network across the PSTN-IP interface. MGC might package both SDP and ISUP elements into the same SIP message by using the MIME [14] multipart format. Relevant problems are under discussing within the IETF SIPPING working group.

![Diagram](image_url)

Figure 5: General ISUP-SIP conversion

On the other hand, certain information is translated from an SS7 ISUP message to 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 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, such headers as the To and From field become essentially fixed, and no further translation will be required during subsequent signaling, 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. Information in this field can be learned from [13].

While in case of PSTN terminations, the MGC at the egress tries to degenerate the ISUP, sometimes after modifying, either from the SIP message body or from the SIP headers.

3. A Signaling Interworking Example

Figure 5 shows a call that originates from the PSTN and terminates at a SIP-phone. It is for the purpose of facilitating the understanding of the previous discussion.

A simple successful call-flow depicting the general ISUP-SIP conversion for a PSTN-originated call terminating in IP is follows:

When a PSTN user wishes to begin a session with a SIP user, the PSTN network generates an IAM message towards the MGC. This MGC is the point of ingress for message flows over the IP network for this call. Upon receipt of the IAM message, the MGC takes necessary steps to preserve the ISUP information. It formulates 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 MGC then encapsulates the ISUP
IAM into the SIP INVITE and ships it out to a proper SIP node. The following steps describe how SIP message is dealt with and what ISUP message is generated by the MGC according to a certain response (18x, 200, etc) it receives. Note here the terminator (SIP-phone) has no use for the encapsulated ISUP and disregards it. Finally the calling party hangs up and generates a REL message. Upon receipt of a REL message the MGC will send a BYE towards the SIP network. The MGC also frees the PSTN circuit and indicates that it is available for reuse by sending an RLC message. A 200 OK response will be sent by the SIP-phone, and the call is over. From this call-flow depiction, we can get a clear understanding of the signaling system of the practical VoIP world.

Future Work and Conclusion

Although SIP has a bright future, it is still very much in the development phase. If we track the progress of different vendors’ solutions, we can be aware that there are more H.323 supporters. Meantime the rate at which SIP voice gateways are being developed is tremendously high.

Potential problems exist, however, when the softswitch network provides a transit function between PSTN and SIP-based VoIP networks. Since protocols are designed by different organizations in different ways, seldom will we find 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. Only a few of them are presented below.

1. There are several flavors of ISUP, which leads to different message flows. That is, ISUP messages exchange during a call. In the situation when a SIP network connects two PSTNs, the egress-MGC should follow specific rules to deal with this problem.

2. The ingress-MGC might package both SDP and ISUP elements into a SIP message by using the MIME multipart format. But the terminator device may not support a multipart payload or the ISUP MIME type thus that SIP request will be rejected.

3. For a PSTN-originated call, a SIP message is produced at the ingress-MGC as a result of ISUP encapsulation and translation. Sometimes it (especially 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 information in the SIP headers does not agree with the encapsulated ISUP and this is a violation. Also part of the encapsulated ISUP may be rendered irrelevant and obsolete. Rules that delineate the preferred behavior of the entities in question (whether originating or terminating) and under the specific circumstances surrounding each such case need to be outlined.

4. European phone numbers does not have a fixed length so that the ingress-MGC cannot know when the number is complete.

5. The transit of ISUP in SIP bodies may provide opportunities for abuse and fraud, especially by SIP-phones, and this puts forward security problems.

Given all the information above, we can see it is still too early to place SIP as a replacement for H.323 in the near future. There is a long and laborious way for SIP to go to provide carrier grade services. The market will finally determine which one will be the acknowledged leader.
References


List of captions:

Title

Abstract

Introduction

Survey of SIP

1. Introduction to SIP
2. Network Architecture
3. Overview of SIP Message
4. Overview of Operation
SIP Interworking with the SS7
1. SIP-PSTN Gateway Architecture
2. Signaling Interworking
3. A Signaling Interworking Example
Future Work and Conclusion