CHAPTER 3

A BROAD OVERVIEW OF SESSION INITIATION PROTOCOL

3.1 INTRODUCTION

This chapter discusses the Session Initiation Protocol, which is the basic to the proposed Seamless Session Initiation Protocol (S-SIP) and Session Schedule Manager schemes (SSM).

3.2 FUNCTIONS OF SIP

SIP is limited to the setup, modification and termination of sessions. It serves four major purposes

- SIP allows for the establishment of user location (translating from a user's name to their current network address).
- SIP provides for feature negotiation so that all of the participants in a session can agree on the features to be supported among them.
- SIP is a mechanism for call management (adding, dropping, or transferring participants).
- SIP allows for changing the features of a session while it is in progress.
3.3 COMPONENTS OF SIP

An entity interacting in a SIP scenario is called as User Agents (UA). User Agents work in two different fashions:

- **User Agent Client (UAC):** It generates requests and sends the request to servers.
- **User Agent Server (UAS):** It gets requests, processes the requests and generates responses.

**Clients:** In general, the notion of clients is associated to the end users. The applications are running on the systems used by the people. It is the soft phone application running on a PC or a messaging device in an IP phone. It generates a request when you try to call another person over the network and sends the request to a server (generally a proxy server).

**Servers:** Servers are in general part of the network. They possess a predefined set of rules to handle the requests sent by clients. Servers are categorized of several types:

- **Proxy Server:** It is the most common type of server in a SIP environment. When a request is generated, the exact address of the recipient is not known in advance. Here, the client sends the request to a proxy server. The server on behalf of the client (as if giving a proxy for it) forwards the request to another proxy server or the recipient itself.

- **Redirect Server:** A redirect server redirects the request back to the client. It indicates that the client needs to get a different route to the recipient. It happens generally, when a recipient has moved from its original position either temporarily or permanently.
Registrar: The users have to register their locations to a Registrar server. Users have to refresh their locations from time to time by registering (sending a special type of message) to a Registrar server.

Location Server: The addresses registered to a Registrar server are stored in a Location Server.

3.4 COMMANDS OF SIP

- INVITE: Invites a user to a call
- ACK: Acknowledgement is used to facilitate reliable message exchange for INVITEs.
- BYE: Terminates a connection between users
- CANCEL: Terminates a request, or search, for a user. It is used if a client sends an INVITE and then changes its decision to call the recipient.
- OPTIONS: Solicits information about a server's capabilities.
- REGISTER: Registers a user's current location
- INFO: Used for mid-session signaling

3.5 RESPONSE TYPES OF SIP

The first digit of a status code defines the category of response. The response between 100 and 199 is termed as a "1xx" response and it is done for any other type. SIP/2.0 allows six types of responses. They are similar to HTTP.

- 1xx: Provisional -- request received, continuing to process the request.
- 2xx: Success -- the action was successfully received, understood, and accepted.
- 3xx: Redirection -- further action needs to be taken in order to complete the request.
- 4xx: Client Error -- the request contains bad syntax or cannot be fulfilled at this server.
- 5xx: Server Error -- the server failed to fulfill an apparently valid request.
- 6xx: Global Failure -- the request cannot be fulfilled at any server.

If a response is received with a status code of the form yxx which is not understood by the receiving party, it treats the response as a y00 response. If a client receives an unknown response 345, it treats as a 300 response. An unknown 1xx is treated as 183 (Session in Progress). So each UA must know how to react to 100, 183, 200, 300, 400, 500 and 600.

3.6 SIP SESSION SETUP

SIP signaling protocol follows the server-client paradigm and is used widely in the Internet by protocols like HTTP or SMTP. Figure 3.1 presents a typical exchange of requests and responses.

Before understanding the methods, first you should understand the pictorial diagram. User 1 uses his soft phone to reach the SIP phone of user2. Server1 and server2 help to setup the session on behalf of the users. The common arrangement of the proxies and the end-users is called as SIP Trapezoid as it is depicted by the dotted line. The messages appear vertically in the order they appear.

The message on top (INVITE M1) comes first followed by others. The direction of arrows shows the sender and recipient of each message. Each message contains a 3-digit-number followed by a name and each one is labeled by M and a serial number. The 3-digit-number is the numerical code
of the associated message comprehended easily by machines. Human users use the name to identify the message.

The transaction starts with user1 by initiating an INVITE request for user2. But, user1 does not know the exact location of user2 in the IP network. So it passes the request to server1. Server1 on behalf of user1 forwards an INVITE request for user2 to server2. It sends a TRYING response to user1 informing that it is trying to reach user2. Receiving INVITE M2 from server1, server2 works in a similar fashion as server1.

It forwards an INVITE request to user2. Here, server2 knows the location of user2. If it did not know the location, it would have forwarded it to another proxy server. So an INVITE request may travel through several proxies before reaching the recipient. After forwarding INVITE M3 server2 issues a TRYING response to server1.

The SIP phone, on receiving the INVITE request, starts ringing and informing that a call request has come for user2. It sends a RINGING response back to server2 which reaches user1 through server1. The user1 gets the feedback that user2 has received the INVITE request.

![Figure 3.1 SIP session setup](image)
User2 at this point has a choice to accept or decline the call. Let's assume that they decide to accept it. As soon as they accept the call, a 200 OK response is sent by the phone to the server2. Retracing the route of INVITE, it reaches user1.

The softphone of user1 sends an ACK message to confirm the setup of the call. This 3-way-handshaking (INVITE+OK+ACK) is used for reliable call setup. The ACK message is not only using the proxies to reach user2 but also now the user1 knows the exact location of user2.

Once the connection has been setup, media flows between the two endpoints. Media flow is controlled by using protocols different from SIP called as RTP.

When one party in the session decides to disconnect, it (user2 in this case) sends a BYE message to the other party. The other party sends a 200 OK message to confirm the termination of the session.

3.7 REQUEST AND RESPONSE MESSAGE

3.7.1 Request Message Format

In the previous SIP session it is noticed that the requests are sent by clients to the servers. The discussion takes place that what type of request actually contains. The following is the format of INVITE request as sent by user1.

INVITE sip:user2@server2.com SIP/2.0
Via: SIP/2.0/UDP pc33.server1.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: user2 <sip:user2@server2.com>
From: user1 <sip:user1@server1.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.server1.com
CSeq: 314159 INVITE
Contact: <sip:user1@pc33.server1.com>
Content-Type: application/sdp
Content-Length: 142

The first line of the text-encoded message is called as Request-Line. It identifies that the message is a request.

**Request-Line**

Method SP Request-URI SP SIP-Version CRLF [SP = single-space & CRLF=Carriage Return + Line Feed (i.e. the character inserted when you press the "Enter" or "Return" key of your computer)]. Here method is INVITE, request-uri is "user2@server2.com" and SIP version is 2.

The following lines are a set of header fields.

- **Via**
  
  It contains the local address of user1 i.e. pc33.server1.com Where it expects the responses to come.

- **Max-Forward**
  
  It is used to limit the number of hops that the request takes place before reaching the recipient. It is decreased by one at each hop. It is necessary to prevent the request from traveling forever in case, if it is trapped in a loop.

- **To**
  
  It contains a display name "user2" and a SIP or SIPS URI <user2@server2.com>
- **From**
  It also contains a display name "user1" and a SIP or SIPS URI <user1@server1.com>. It also contains a tag which is a pseudo-random sequence inserted by the SIP application. It works as an identifier of the caller in the dialog.

- **Call-ID**
  It is a globally unique identifier of the call generated as the combination of a pseudo-random string and the soft phone’s IP address. The Call-ID is unique for a call. A call may contain several dialogs. Each dialog is uniquely identified by a combination of From, To and Call-ID.

- **CSeq**
  It contains an integer and a method name. When a transaction starts, the first message is given a random CSeq. After that it is incremented by one with each and every new message. It is used to detect the non-delivery of a message or out-of-order delivery of messages.

- **Contact**
  It contains a SIP or SIPS URI that is a direct route to user1. It contains a username and a fully qualified domain name (FQDN). It may also have an IP address. Via field is used to send the response to the request. Contact field is used to send the future requests. That is why the 200 OK responses from user2 go to user1 through proxies. But when user2 generates a BYE request (a new request and not a response to INVITE), it goes directly to user1 bypassing the proxies.
- **Content-Type**
  It contains a description of the message body.

- **Content-Length**
  It is an octet (byte) count of the message body.

The header contains other header fields. However, those fields are optional. The body is used to convey information about the media session written in Session Description Protocol (SDP).

### 3.7.2 Response Message Format

Here, it initiates what the SIP response of user2 looks like.

SIP/2.0 200 OK
Via: SIP/2.0/UDP
site4.server2.com;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP
site3.server1.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP
pc33.server1.com;branch=z9hG4bK776asdhds;received=192.0.2.1
To: user2 <sip:user2@server2.com>;tag=a6c85cf
From: user1 <sip:user1@server1.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.server1.com
CSeq: 314159 INVITE
Contact: <sip:user2@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131

---- User2 Message Body Not Shown ----
**Status Line**

The first line in a response is called as Status line.

Here, SIP version is 2, Status-Code is 200 and Reason Phrase is OK.

The header fields that follow the status line are similar to those in a request. The differences are mentioned below:

- **Via**
  
  There are more than one via field. This is because each element through which the INVITE request has passed and added its identity in the Via field. Three Via fields are added by softphone of user1, server1 the first proxy and server2 the second proxy. The response retraces the path of INVITE using the Via fields. On its way back, each element removes the corresponding Via field before forwarding it back to the caller.

- **To**
  
  The field To contains a tag. This tag is used to represent the callee in a dialog.

- **Contact**
  
  It contains the exact address of user2. So user1 doesn't need to use the proxy servers to find user2 in the future.

It is a 2xx response. However responses are different depending upon the particular situations.
3.8 REQUIREMENTS OF USER

SIP is an off-the-shelf viable and reliable set of rules for managing three broad user requirements:

- Session setup: It negotiates the session parameters with the participants.
- Session management: It provides for cancelling sessions, adding new participants, modifying of session parameters, and invoking services.
- Mobility: It supports for service, session, device and personal mobility.

SIP organizes and sets up the media exchanges that follow as part of a well-defined session. The session definition normally includes the following:

- Identification: a session ID, subject, an indication of which end points will be used for data transfer and further requests.
- Routing: the path for data transmission.
- User availability: a mechanism to accept or turn down a request.
- Device capabilities: advertisement or requirement of device feature and media capabilities.
- Content: the content type, length, encoding and language used.

Because of the open nature of the IETF standards, the fact SIP is text based and shares many features with existing specifications and it has been readily understood, extended and implemented. An enormous number of functions have been tacked onto the initial standard, notably those for
presence (displaying the status like ‘away’, ‘online’ etc), event notification and subscription, messaging, privacy, routing, QoS, interworking with the public telephone system, gateway traversal and many more.

The reason for all the extra standards and practices are twofold.

- SIP confines itself to define how the sessions make setup, maintained and managed.
- It is not limited by usage scenario.