CHAPTER – 5
INITIALIZATION OF SIP IN IMS

5.1 Introduction to SIP
5.2 SIP Functionality
5.3 Sip Elements
5.3.1 Different roles of a SIP server
5.3.1.1 Proxy server:
5.3.1.2 Redirect server:
5.3.1.3 Registrar server:
5.4 SIP Pre-Setup Procedures in IMS Environment
5.4.1 GPRS Attach
5.4.2 PDP Context Activation
5.4.3 CSCF Discovery
5.4.4 Service Registration of SIP in IMS
5.5 Overview of SIP Session Flow Procedures in IMS
5.5.1 Session Setup Procedures
5.5.1.1 Origination Procedures
5.5.1.2 S-CSCF to S-CSCF Procedures
5.5.1.3 Mobile termination procedures
5.5.1.4 Summary of The Session Setup Procedures
5.5.2 Session Release Procedures
5.6 SIP Message Structure
5.6.1 SIP request
5.6.2 SIP method extensions
5.6.3 SIP Response
5.6.4 SIP Headers
5.7 A Simple SIP Example
5.8 Tools to read SIP Messages
5.8.1 SIP Logger
5.8.2 SIP Parser

CHAPTER – 5
INITIALIZATION OF SIP IN IMS

5.1 Introduction to SIP

The Session Initiation Protocol (SIP) is a new signaling protocol developed to set up, modify, and tear down multimedia sessions over the Internet [103]. This chapter covers some background for the understanding of the protocol. SIP was developed by the Internet Engineering Task Force (IETF) as part of the Internet Multimedia Conferencing Architecture, and was designed to dovetail with other Internet protocols such as TCP, UDP, IP, DNS, and others.

IMS relies on the session initiation protocol (SIP) for the development of applications and services. SIP is a signaling protocol specifically designed for multimedia. It offers advantages over signaling system 7 (SS7), which is used throughout the public switched telephone network (PSTN) and was designed specifically for voice services. Unlike SS7, SIP was designed to support voice, data, and multimedia services.

SIP is focused on session control—establishing, changing and terminating sessions—and it supports dynamic modification of multimedia streams for any given session. Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions can contain any combination of media (voice, data, video, audio files, anything), and can be modified at any time to add new parties or to change the nature of the session. SIP has been chosen as the signaling protocol for establishing multimedia sessions in IMS Release5. This chapter demonstrates the operations defined in IMS for establishing multimedia sessions.

5.2 SIP Functionality

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls. SIP can also invite participants to already existing sessions, such as multicast conferences. Media can be added to (and removed from) an existing session. SIP transparently supports name mapping and redirection services, which supports personal mobility - users can maintain a single externally visible identifier regardless of their network location.

SIP supports five facets of establishing and terminating multimedia communications:

1. **User location:** Determination of the end system to be used for communication.
2. **User availability:** Determination of the willingness of the called party to engage in communications.
3. **User capabilities:** Determination of the media and media parameter to be used.

4. **Session setup:** "ringing", establishment of session parameters at both called and calling party.
5. **Session management:** Including transfer and termination of sessions, modifying session parameters, and invoking services.

SIP is not a vertically integrated communications system. SIP is rather a component that can be used with other IETF protocols to build complete multimedia architecture. Typically, these architectures will include protocols such as the Real-time Transport Protocol (RTP) [104] for transporting real-time data and providing QoS feedback, the Real-Time streaming protocol (RTSP) for controlling delivery of streaming media, the Media Gateway Control Protocol (MEGACO) for controlling gateways to the Public Switched Telephone Network (PSTN), and the Session Description Protocol (SDP) for describing multimedia sessions. Therefore, SIP should be used in conjunction with other protocols in order to provide complete services to the users. However, the basic functionality and operation of SIP does not depend on any of these protocols.

SIP does not provide services. Rather, SIP provides primitives that can be used to implement different services. For example, SIP can locate a user and deliver an opaque object to his current location. If this primitive is used to deliver a session description written in SDP, for instance, the endpoints can agree on the parameters of a session. If the same primitive is used to deliver a photo of the caller as well as the session description, a "caller ID" service can be easily implemented.

As this example shows, a single primitive is typically used to provide several different services. SIP can be used to initiate a session that uses some other conference control protocol. Since SIP messages and the sessions they establish can pass through entirely different networks, SIP cannot, and does not, provide any kind of network resource reservation capabilities. The nature of the services provided make security particularly important. To that end, SIP provides a suite of security services, which include denial off service prevention, authentication (both user to user and proxy to user), integrity protection, and encryption and privacy services. SIP works with both IPv4 and IPv6.

5.3 SIP Elements

SIP is an application-layer control protocol that handles the setup, modification, and teardown of multimedia sessions. SIP is used in combination with other protocols to describe the session characteristics to potential session participants. SIP is based on a request and response transaction model similar to HTTP. Each transaction consists of a request that invokes a particular method or a function on the server and at least one response.

SIP is generally considered to be a Agent–server protocol. At a high level there are two types of SIP elements:

1. User Agents
2. Servers.

User agents: User Agents are endpoints in a SIP network: they originate and terminate calls. Examples of User Agents (UA) [105] include: SIP phones (hard sets), laptops or PDA with a SIP client (e.g., soft phone), Media gateway (e.g. T1/E1 gateway), access gateway (e.g., FAX gateway), conferencing systems, etc. All these devices also initiate and terminate the media session (voice, video, FAX, etc.). A UA is itself comprised of two entities (software):

- UAC (initiates call by sending INVITE with E.164 or URI dialing)
- UAS (receives call requests).

Server: A server generally responds to a request sent by an agent. A server can be a software application, such as Live Communications Server 2003, or a hardware device. There are several types of servers in a SIP network including

- Proxy server
- Redirect server
- SIP registrar.

5.3.1 Different roles of a SIP server

SIP servers have different roles, such as:

5.3.1.1 Proxy server:

A Proxy server performs signaling and relay. In other words, it determines where to send signaling messages and forward requests on behalf of the UA. To do so, it consults databases (DNS, location servers, etc.) [106]. It is important to remember that Proxy servers have no media capabilities; they are in the control path only. Proxy servers must pass unrecognized SIP messages through unchanged. Thus new features do not require changes to proxy servers used in an infrastructure.

This principle enables new features to be deployed in a network by only upgrading the end devices. The routing function can be configured (programmed) according to user preferences, type of call (e.g., 911), least-GW-cost, or other criteria. Note that the proxy server is not the only “place” where service can be programmed. In fact, service programmability can reside in end-devices as well, such as for visual caller ID, distinctive ringing or possible Call Forwarding. Proxy servers can try several destinations sequentially or in parallel, this capability called forking enables multiple devices to be associated with the same address.

There are three types of Proxy servers according to the type of state information they keep:

1) A stateless proxy keeps no state
2) A transaction stateful proxy only keeps state on pending transactions.

3) A call stateful proxy keeps state for the entire duration of a SIP session.

Most implementations are stateful proxy-based as this is useful for implementing such services as “forward on no reply” and also to implement forking. Stateless proxies are easier to scale (especially under heavy load scenarios) and can act as an application-layer load distributor (used in the core of a network). Redundancy designs are easier to achieve with stateless proxies.

5.3.1.2 Redirect server:

A SIP redirect server accepts a SIP request and conveys to the originating client the way to route the call. Redirect servers are servers that redirect SIP requests to another device. A redirect server responds to the request with the address to which the request should be redirected (e.g., a request for nic@mitel.com can be redirected to nic@home.com). SIP does not specify any implementation models –for example, all above servers can reside on the same hardware platform. The underlying OS can be Windows, Solaris, Linux or any embedded real time OS. For example, VOCAL is an open-source VoIP [107] software from Vovida.org [120]. VOCAL software suite is a robust implementation of the SIP protocol and its various entities and is used widely. It is important to note that the above servers (proxy, redirect and registrar) are all optional SIP components.

In fact, a UA may issue an INVITE directly to a targeted endpoint and many telephony features may be implemented directly on the UA. The SIP model is based on intelligent endpoints that can act without other intelligence from the network infrastructure (refer to section below on peer-to-peer vs. centralized model).

5.3.1.3 Registrar server:

A SIP registrar server accepts registration requests and maps agent’s address to a user’s sign-in name, or SIP URI [108]. Typically, a registrar is combined with a proxy or redirect server. A SIP registrar accepts registration requests from users (e.g., I am now at 192.168.0.10) and maintains user location information in a database. Mobility is thus achieved by the use of a REGISTER message (from UA) and by keeping a location database updated.

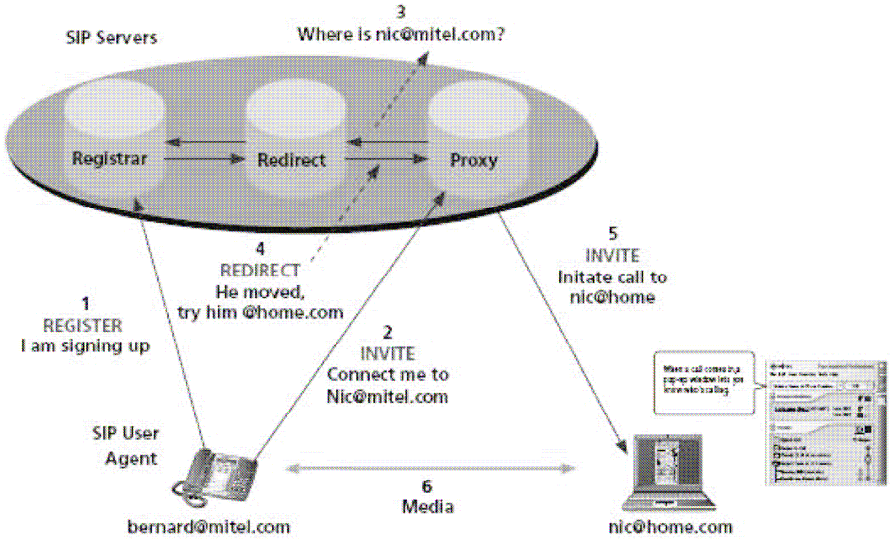

Figure - 5.1: Example of user mobility using registers and redirects messages [108]

5.4 SIP Pre-Setup Procedures in IMS Environment

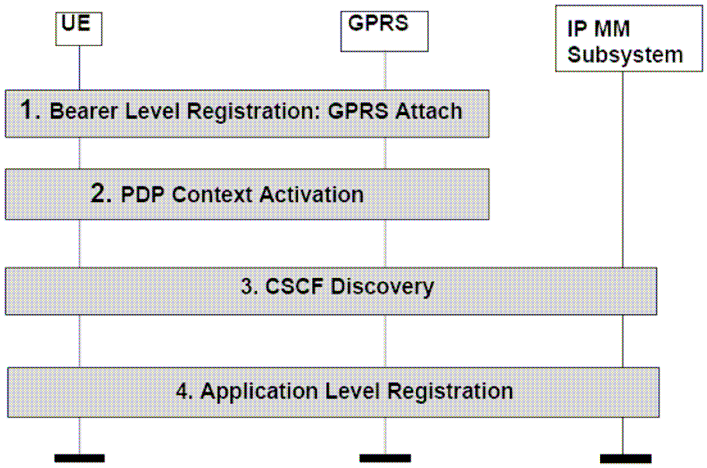

Figure - 5.2: Procedures Before SIP Sessions

When a UE is powered on and locked on to the IMS system, it must take several critical steps before communicating SIP signaling messages required to establish a data session.

The key steps are as following:

1. GPRS Attach: to establish Mobility Management Contexts at UE and SGSN.
2. PDP context Activation: to establish GGSN connectivity.
3. CSCF discovery: to obtain the address of P-CSCF, the first contact point within the IMS subsystem.
4. Service Registration: to send subscriber profile to a S-CSCF in its home network to obtain IMS services.

In summary, it must create a path toward the proxy CSCF, and performs the service registration to the Serving CSCF in its home network through the P-CSCF for SIP services. will introduce each step in details.

5.4.1 GPRS Attach

IP Multimedia Subsystem uses packet domain of the Core Network to transfer data and signaling in an efficient manner. A common packet domain Core Network (PS-CN) is used for both the GERAN and the UTRAN. This common Core Network is designed to support several Qos levels to allow efficient transfer of non real-time traffic (e.g. intermittent and bursty data transfers, occasional transmission of large volumes of data) and real-time traffic (e.g. voice, video). The Serving GPRS Support Node (SGSN) keeps track of the location of an individual mobile and performs security functions and access control. The Gateway GPRS Support Node (GGSN) provides inter-working with packet data networks, and is connected with the SGSNs via the PLMN IP backbone.

In order to get access to packet domain service with the IMS network, a UE shall first make its presence known to the network by performing a GPRS attach (or called PS attach). At attach, the SGSN establishes a mobility management context containing information pertaining to e.g. mobility and security for the UE, and the authentication procedure is performed in association with the establishment of the mobility management context.

The IMS UE sends its International Mobile Subscriber Identifier (IMSI) to the SGSN in the Attach message. The SGSN uses the IMSI to send a request to the UE’s HSS for the authentication parameters. The HSS provides the authentication information to the SGSN, enabling the SGSN to authenticate the subscriber’s IMSI [109, 110].

The successful completion of authentication procedure triggers the SGSN to send a location update to the HSS and this triggers the subscriber’s profile to be downloaded to the SGSN. This includes information such as the subscribed services, the QoS profile, any static IP addresses allocated and so on. Then the SGSN completes the Attach procedure by sending an Attach Complete message to the UE [109].

By GPRS Attach, the location of the mobile is known within the IMS network, and a logical association is now established between the UE and the SGSN, this logical connection is maintained as the UE moves within the coverage area controlled by that SGSN [109]. However this is only the first step toward packet data service. Before the UE can request IM services, a PDP context must be activated to carry IM subsystem related signaling.

5.4.2 PDP Context Activation

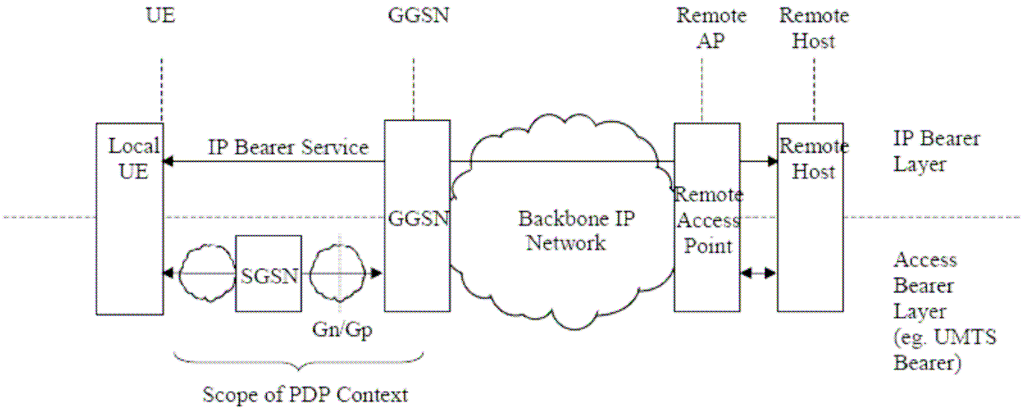

Figure - 5.3: Scope of PDP Context [22]

A UE subscribed to the IMS packet domain service is allocated one or more PDP (Packet Data Protocol) addresses either by the wireless operator statically, or by GGSN dynamically during the PDP context activation. Each PDP address is an element of a PDP context. Every PDP context exists independently in one of two states indicating whether data transfer is enabled for that PDP address or not. The Inactive state means the data service for a certain PDP address of the UE is not activated, and the PDP context contains no routing or mapping information to process traffic related to that PDP address. In Active state, the PDP context for the PDP address in use is activated in the UE, SGSN and GGSN, and the PDP context contains mapping and routing information for transferring data for that particular PDP address between the UE and the GGSN [111].

So after a UE is attached to an SGSN, it must activate a PDP context to begin the packet data communication by initiating the PDP Context Activation procedure. This operation negotiates an active PDP address (in this case, IP address) for the UE and sets up an association between the UE's current SGSN and a corresponding GGSN that anchors the PDP address, thus creates a SGSN-GGSN path for the UE toward the packet data service. User data is encapsulated with GPRS-specific protocol information and transferred transparently between the UE and the GGSN.

In the case of a SIP service, the first PDP context must be activated for all SIP related signaling traffic. This is referred to as primary PDP Context. The UE may also send a secondary PDP Context Activation, which uses the same PDP address as the Primary Context with distinctly different QoS requirements. The SGSN chooses the appropriate GGSN for different contexts and services. The choice of the GGSN by the SGSN is independent of the radio resource allocations. A mobile may initiate secondary PDP context and may be connected to more than one GGSN [109].

5.4.3 CSCF Discovery

The P-CSCF is the first contact point in the IMS subsystem for the UE. The discovery of the IP address of the P-CSCF shall be performed after or as part of a successful activation of a PDP context for IMS signaling using one of the following mechanisms [112]:

1. Use of DHCP to provide the UE with the domain name of a Proxy-CSCF and the address of a Domain Name Server (DNS) that is capable of resolving the P-CSCF name. The GGSN acts as a DHCP Relay Agent, relaying DHCP messages between UE and the DHCP server.

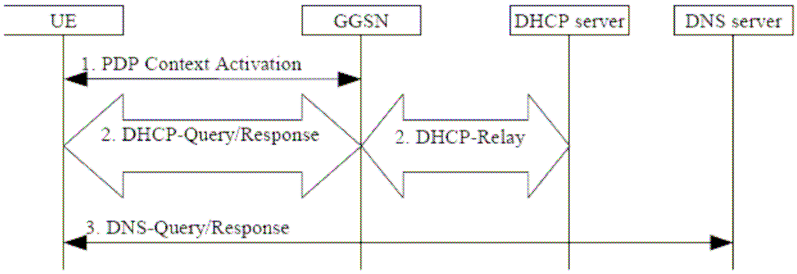

Figure - 5.4: P-CSCF Discovery Using DHCP and DNS [112]

2. The UE requests the P-CSCF address from the GGSN when activating the PDP context. The GGSN sends the P-CSCF address to the UE when accepting the PDP context activation. Both the P-CSCF address request and the P-CSCF address shall be sent transparently through the SGSN.

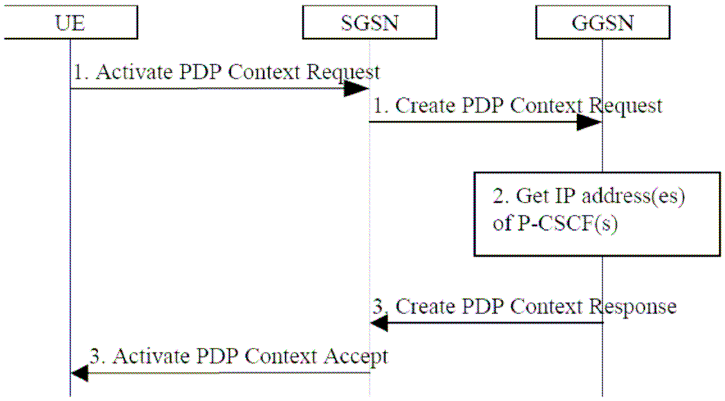

Figure - 5.5: P-CSCF Discovery Using PDP Context Activation Signaling [112]

After reception of IP address of a P-CSCF the UE may initiate communication towards the IP Multimedia subsystem.

5.4.4 Service Registration of SIP in IMS

A UE needs to perform IMS service registration before it can set up a session. Through a successful registration the UE will be assigned a suitable S-CSCF in its home network to obtain the IMS services.

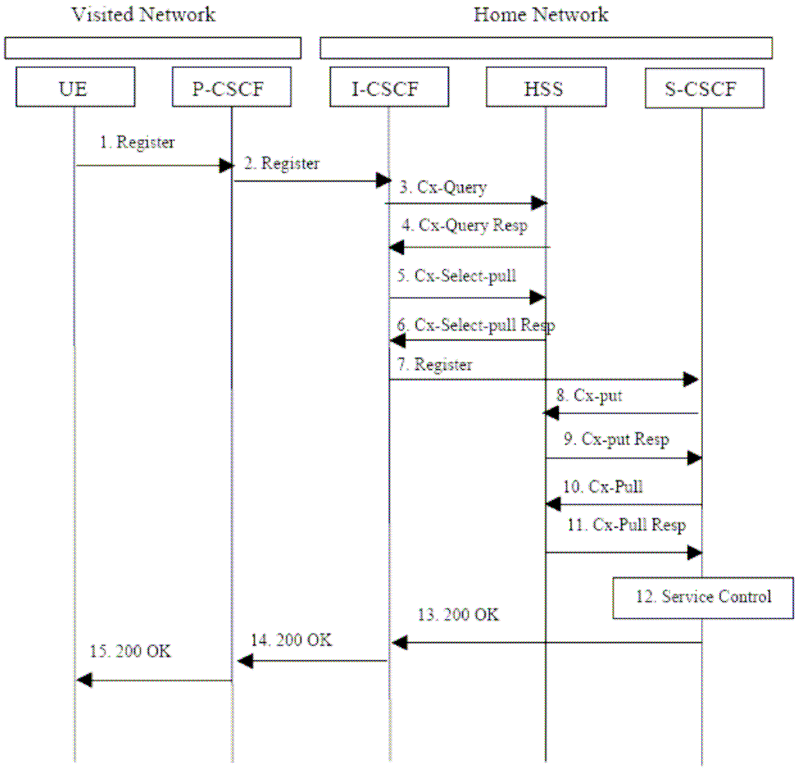

Figure - 5.6: Registration Procedure for Un-registered User [112]

For a user roaming at a visited network, a detailed information flow is shown in Fig 5. For users located in their home network, the home network shall perform the role of the visited network element and the home network elements. The procedures are the same [112]:

1. The UE sends the Register information flow to the P-CSCF. The information includes the subscriber identity and home networks domain name.
2. Upon receipt of the register information flow, the P-CSCF shall examine the “home domain name” to discover the entry point to the home network (i.e. the I-CSCF). The proxy sends the Register information flow to the I-CSCF with the P-CSCF address/name, P-CSCF network identifier (e.g., domain name of the P-CSCF network), and subscriber’s identity, etc. The main job of I-CSCF is to query the HSS and find the location of the S-CSCF.
3. The I-CSCF sends a IMS proprietary message, Cx-Query information flow to the HSS with the subscriber’s identity, P-CSCF network identifier. The HSS then checks whether the user is registered already. The HSS shall indicate whether the user is allowed to register in that P-CSCF

network according to the User subscription and operator limitations/restrictions if any.

4. Cx-Query Resp is sent from the HSS to the I-CSCF. If the checking in HSS was not successful the Cx-Query Resp shall reject the registration attempt. Otherwise the message will contain the S-CSCF name, if it is known by the HSS, or the S-CSCF capabilities, if it is necessary to select a new S-CSCF.

5. If the I-CSCF has not been provided with the name of the S-CSCF then the I-CSCF will send Cx-Select-Pull to the HSS to request the information related to the required S-CSCF capabilities that shall be input into the S-CSCF selection function.

6. On receipt of the Cx-Select-Pull, the HSS shall send Cx-Select-Pull Resp (required S-CSCF capabilities) to the I-CSCF.

7. The I-CSCF, using the name of the S-CSCF, shall determine the address of the S-CSCF through a name-address resolution mechanism. The I-CSCF also determines the name of a suitable home network contact point, possibly based on information received from the HSS. The home network contact point may either be the S-CSCF itself, or a suitable I-CSCF (THIG) in case network configuration hiding is desired. If an I-CSCF (THIG) is chosen as the home network contact point for implementing network configuration hiding, it may be distinct from the I-CSCF that appears in this registration flow, and it shall be capable of deriving the S-CSCF name from the home contact information. I-CSCF shall then send the register information flow to the selected S-CSCF. The flow includes P-CSCF address/name, subscriber's identity, P-CSCF network identifier, UE IP address, and the home network contact point. The home network contact point will be used by the P-CSCF to forward session initiation signaling to the home network.

8. The S-CSCF sends Cx-Put with subscriber's identity and S-CSCF name to the HSS. The HSS stores the S-CSCF name for that user.

9. The HSS sends Cx-Put Resp to the S-CSCF to acknowledge the sending of Cx-Put.

10. On receipt of the Cx-Put Resp information flow, the S-CSCF shall send the Cx-Pull information flow with subscriber's identity to the HSS in order to be able to download the relevant information from the user profile to the S-CSCF. The S-CSCF shall store the P-CSCF address/name, which represents the address/name that the home network forwards the subsequent terminating session signaling to for the UE

11. The HSS shall return the information flow Cx-Pull Resp with user information to the S-CSCF. The user information passed from the HSS to the S-CSCF shall include one or more names/addresses information, which can be used to access the platform(s) used for service control while the user is registered at this S-CSCF. The S-CSCF shall store the information for the indicated user. In addition to the names/addresses information, security information may also be sent for use within the S-CSCF.

12. Based on the filter criteria, the S-CSCF shall send register information to the service control platform and perform whatever service control procedures are appropriate.
13. The S-CSCF returns the 200 OK information flow with home network contact information to the I-CSCF. If an I-CSCF is chosen as the home network contact point for implementing network configuration hiding, the I-CSCF shall encrypt the S-CSCF address in the home network contact information.
14. The I-CSCF sends information flow 200 OK flow to the P-CSCF. The I-CSCF shall release all registration information after sending information flow 200 OK.
15. The P-CSCF stores the home network contact information, and sends information flow 200 OK to the UE.

5.5 Overview of SIP Session Flow Procedures in IMS

5.5.1 Session Setup Procedures

For an IP Multimedia Subsystem session, the session flow consists three types of procedures: mobile origination (MO), S-CSCF to S-CSCF, and mobile termination (MT). A large number of end-to-end session flows are built from combinations of origination, serving to serving and termination procedures.

The original sequence may be one of the following:

MO#1: Mobile Origination, a mobile roaming at a visit network initiates a session setup;

MO#2: Mobile Origination, a mobile located at home network initiates a session setup;

PSTN-O: PSTN origination;

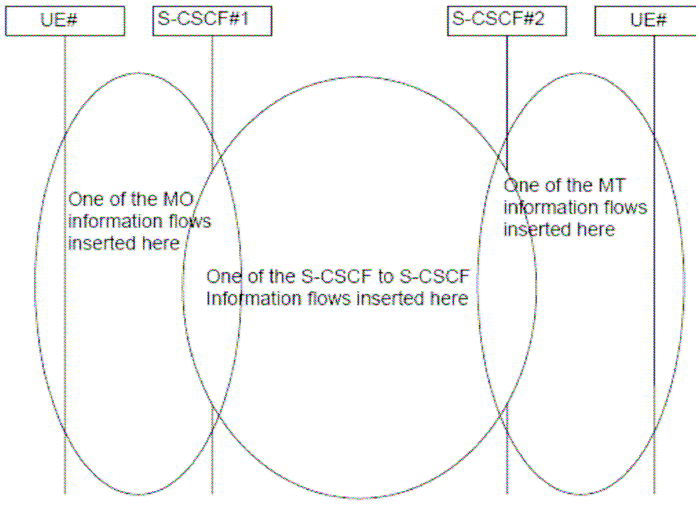

Figure - 5.7: Overview of Session Flow Sections

For the termination sequence:

MT#1: Mobile Termination, the called mobile is roaming at a visit network;

MT#2: Mobile Termination, the called mobile is at its home network;

MT#3: The called party is unregistered for IMS services, for ex, users of the legacy wireless networks.

PSTN-T: PSTN termination;

For Serving-CSCF to Serving CSCF:

S-S#1: The S-CSCF serving the calling party and the S-CSCF serving the called party are in different networks.

S-S#2: The S-CSCF serving the calling party and the S-CSCF serving the called party are in the same network.

S-S#3: Session origination with PSTN termination in the same network as the S-CSCF.

S-S#4: Session origination with PSTN termination in a different network to the S-CSCF.

5.5.1.1 Origination Procedures

UE always has a P-CSCF associated with it determined by the CSCF discovery process. This P-CSCF is located in the same network as the GGSN, performs resource authorization, and may have additional functions in handling of emergency sessions. And as the result of the registration procedure, the P-CSCF determines the next hop toward the S-CSCF (possibly through an I-CSCF to hide the network configuration). Thus a signaling path between the UE and the S-CSCF that is assigned to perform the service is determined at the time of UE registration and will remain fixed for the life of the registration. The UE is now capable of initiating a session setup with the signaling path.

Further present a detailed description of the MO #1 process [112]. The detailed information flow of MO#2 will not be described here. The procedures are no much different with MO#1 except the P-CSCF and S-CSCF involved are in the same network.

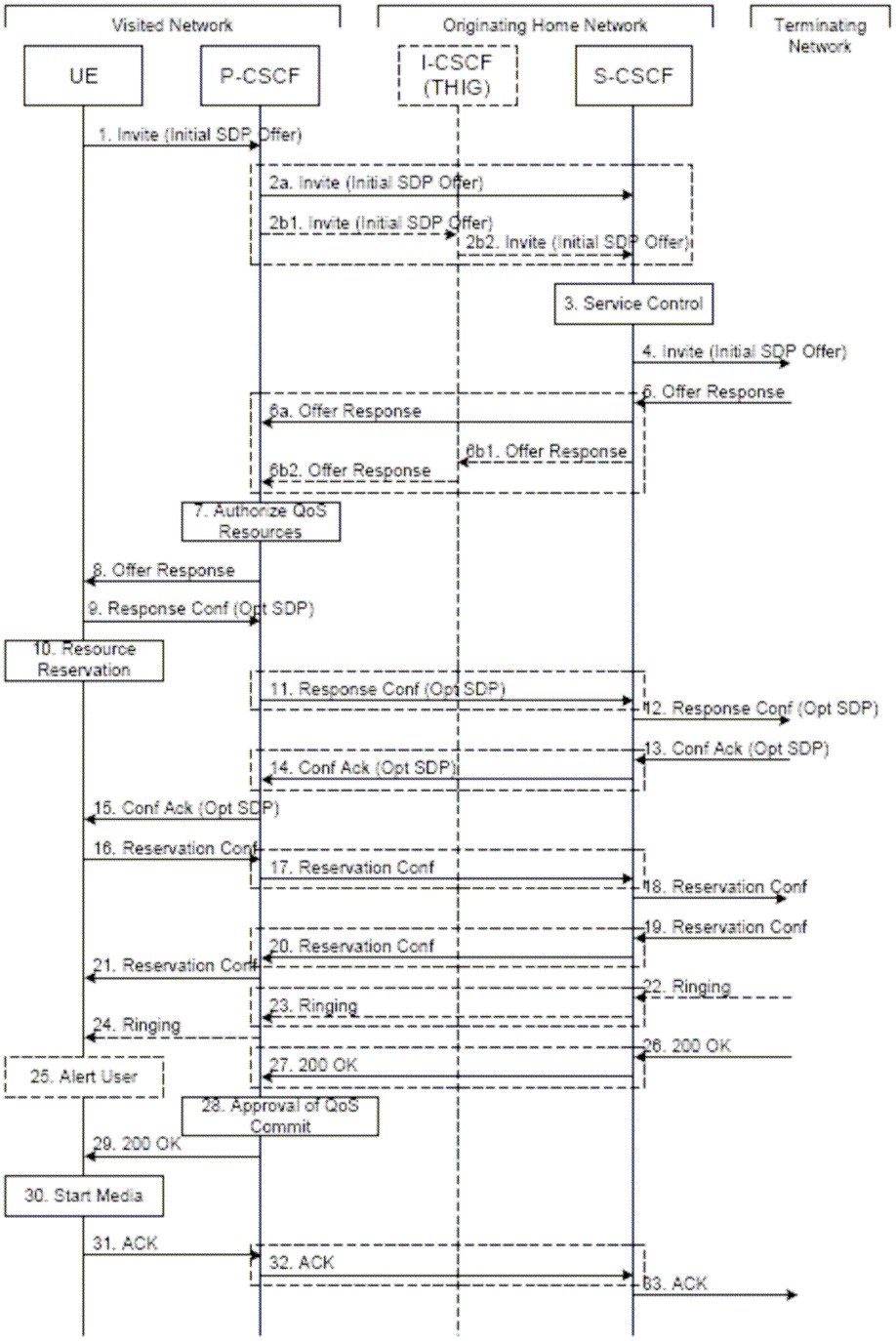

Figure - 5.8: Mobile origination procedure – Roaming [112]

1. UE sends the SIP INVITE request, containing an initial SDP, to the P-CSCF determined via the CSCF discovery mechanism. The initial SDP may represent one or more media for a multi-media session.
2. P-CSCF remembers the next hop CSCF for this UE from the registration procedure. If the home network operator does not desire to keep their network configuration hidden, the name/address of the S-CSCF was provided during registration, and the INVITE request is

forwarded directly to the S-CSCF. If the home network operator chooses to keep their network configuration hidden, the name/address of an I-CSCF (THIG) in the home network was provided during registration, and the INVITE request is forwarded through this I-CSCF (THIG) to the S-CSCF.

3. S-CSCF validates the service profile, and invokes any origination service logic required for this user. This includes authorization of the requested SDP based on the user's subscription for multi-media services.

4. S-CSCF forwards the request, as specified by the S-S procedures.

5. The media stream capabilities of the destination are returned along the signaling path, via the S-S procedures.

6. S-CSCF forwards the Offer Response message to P-CSCF. Based on the choice made in step #2 above, this may be sent directly to P-CSCF (6a) or may be sent through I-CSCF (THIG) (6b1 and 6b2).

7. P-CSCF authorizes the resources necessary for this session. The Authorization-Token is generated by the PDF (Policy Decision Function), a logical entity of the P-CSCF.

8. The Authorization-Token is included in the Offer Response message. P-CSCF forwards the message to the originating endpoint

9. UE decides the offered set of media streams for this session, and confirms receipt of the Offer Response by sending a Response Confirmation to the P-CSCF. The Response Confirmation may also contain SDP. This may be the same SDP as in the Offer Response received in Step 8 or a subset. If new media are defined by this SDP, P-CSCF (PDF) will perform a new authorization as in Step 7 following Step 14. The originating UE is free to continue to offer new media on this operation or on subsequent exchanges using the Update method. Each offer/answer exchange will cause the P-CSCF (PDF) to repeat the Authorization step (Step 7) again.

10. After determining the needed resources in step 8, UE initiates the reservation procedures for the resources needed for this session.

11. P-CSCF forwards the Response Confirmation to S-CSCF. This may possibly be routed through the I-CSCF depending on operator configuration of the I-CSCF. Step 11 may be similar to Step 2 depending on whether or not configuration hiding is used.

12. S-CSCF forwards this message to the terminating endpoint, via the S-S procedure.

13-15. The terminating end point responds to the originating end with an acknowledgement. If Optional SDP is contained in the Response Confirmation, the Confirmation Acknowledge will also contain an SDP response. If the SDP has changed, the P-CSCF authorizes the resources again. Step 14 may be similar to Step 6 depending on whether or not configuration hiding is used.

16-18. When the resource reservation is completed, UE sends the successful Resource Reservation message to the terminating endpoint, via the signaling path established by the INVITE message. The message is sent first to P-CSCF. Step 17 may be similar to Step 2 depending on whether or not configuration hiding is used.

19-21. The terminating endpoint responds to the originating end when successful resource reservation has occurred. If the SDP has changed, the P-CSCF performs the authorization again.

22-24. The Terminating endpoint may generate ringing and it is then forwarded via the session path to the UE.

25. UE indicates to the originating user that the destination is ringing

26-27. When the destination party answers, the terminating endpoint sends a SIP 200-OK final response to the originating end, as specified by the termination procedures and the S-S procedures, to P-CSCF.

28. P-CSCF indicates the resources reserved for this session should now be approved for use.

29. P-CSCF sends a SIP 200-OK final response to the session originator

30. UE starts the media flow(s) for this session

31-33. UE responds to the 200 OK with a SIP ACK message sent along the signalling path. Step 32 may be similar to Step 2 depending on whether or not configuration hiding is used.

5.5.1.2 S-CSCF to S-CSCF Procedures

The S-CSCF to S-CSCF procedures specify the signaling path between the serving CSCF that handles session origination on behalf of the caller, and the serving CSCF that handles session termination on behalf of the called party.

The S-CSCF handling session origination performs an analysis of the destination address, and determines whether it is a subscriber of the same network operator or a different operator. If the analysis of the destination address determined that it belongs to a subscriber of a different operator, the request is forwarded (optionally through an I-CSCF (THIG) within the originating operator's network) to a well-known entry point in the destination operator's network, the I-CSCF. The I-CSCF queries the HSS for current location information. The I-CSCF then forwards the request to the S-CSCF. If the analysis of the destination address determines that it belongs to a subscriber of the same operator, the S-CSCF passes the request to a local I-CSCF, who queries the HSS for current location information. The I-CSCF then forwards the request to the S-CSCF serving the destination user.

Here describe the information flow between two S-CSCFs belonging to different operators [112].

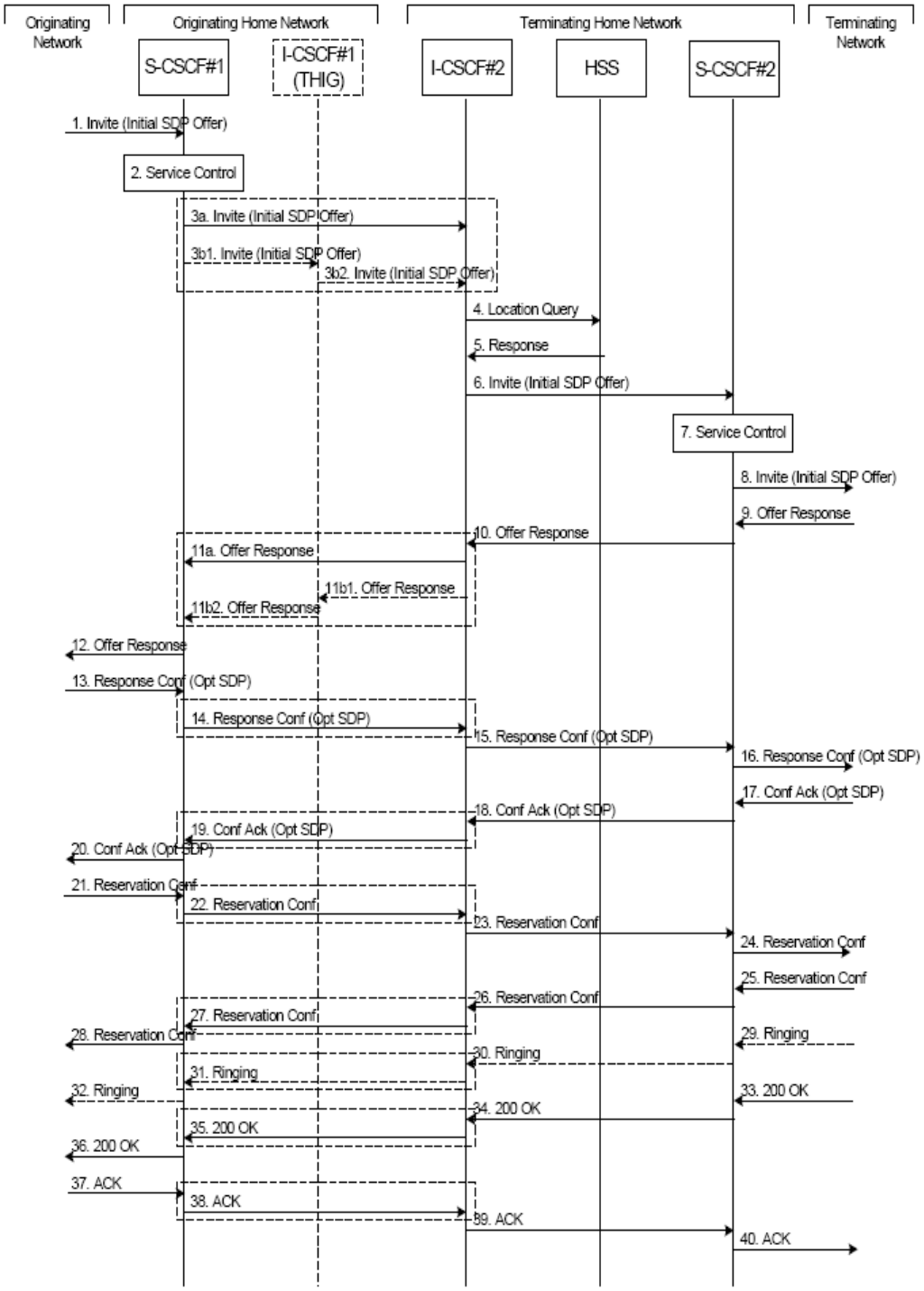

Figure - 5.9: S-CSCF to S-CSCF Procedure – Different Operators [112]

1. The SIP INVITE request is sent from the UE to S-CSCF#1 by the procedures of the originating flow. This message should contain the initial media description offer in the SDP.
2. S-CSCF#1 invokes whatever service logic is appropriate for this session attempt.

3. S-CSCF#1 performs an analysis of the destination address, and determines the network operator to whom the subscriber belongs. For S-S#1, this flow is an inter-operator message to the I-CSCF entry point for the terminating user. If the originating operator desires to keep their internal configuration hidden, then S-CSCF#1 forwards the INVITE request through I-CSCF (THIG)#1 (choice b); otherwise S-CSCF#1 forwards the INVITE request directly to I-CSCF#2, the well-known entry point into the terminating user's network (choice a).
4. I-CSCF#2 (at the border of the terminating user's network) may query the HSS for current location information.
5. HSS responds with the address of the current Serving-CSCF for the terminating user.
6. I-CSCF#2 forwards the INVITE request to the S-CSCF #2 that will handle the session termination.
7. S-CSCF#2 invokes whatever service logic is appropriate for this session set up attempt.
8. The sequence continues with the message flows determined by the termination procedure.
9. The media stream capabilities of the destination are returned along the signaling path, as per the termination procedure.
10. S-CSCF#2 forwards the SDP to I-CSCF#2
11. I-CSCF#2 forwards the SDP to S-CSCF#1. Based on the choice made in step #3 above, this may be sent directly to S-CSCF#1 (11a) or may be sent through I-CSCF (THIG)#1 (11b1 and 11b2)
12. S-CSCF#1 forwards the SDP to the originator, as per the originating procedure.
13. The originator decides on the offered set of media streams, confirms receipt of the Offer Response with a Response Confirmation, and forwards this information to S-CSCF#1 by the origination procedures. The Response Confirmation may also contain SDP. This may be the same SDP as in the Offer Response received in Step 12 or a subset.

14-15 S-CSCF#1 forwards the offered SDP to S-CSCF#2. Step 14 may be similar to Step 3 depending on whether or not configuration hiding is being used.

16. S-CSCF#2 forwards the offered SDP to the terminating endpoint, via the termination procedure introduce at 3.2.1.3.

17-20 The terminating end point acknowledges the offer with answered SDP and passes through the session path to the originating end point. Step 19 may be similar to Step 11 depending on whether or not configuration hiding is being used.

21-24. Originating endpoint acknowledges successful resource reservation and the message is forwarded to the terminating end point. Step 22 may be similar to Step 3 depending on whether or not configuration hiding is used.

25-28. Terminating endpoint acknowledges the response and this message is sent to the originating end point through the established session path. Step 27 may be similar to Step 11 depending on whether or not configuration hiding is being used.

29-32. Terminating end point then generates ringing and this message is sent to the originating end point through the established session path. Step 31 may be similar to Step 11 depending on whether or not configuration hiding is being used.

33-36. Terminating end point then sends 200 OK via the established session path to the originating end point. Step 35 may be similar to Step 11 depending on whether or not configuration hiding is being used.

37-40. Originating end point acknowledges the establishment of the session and sends to the terminating end point via the established session path. Step 38 may be similar to Step 3 depending on whether or not configuration hiding is being used.

The detailed information flow of S-S#2 will not be described here. The procedures are no much different except the CSCFs involved (S-CSCF#1&2, I-CSCF#2) are in same network.

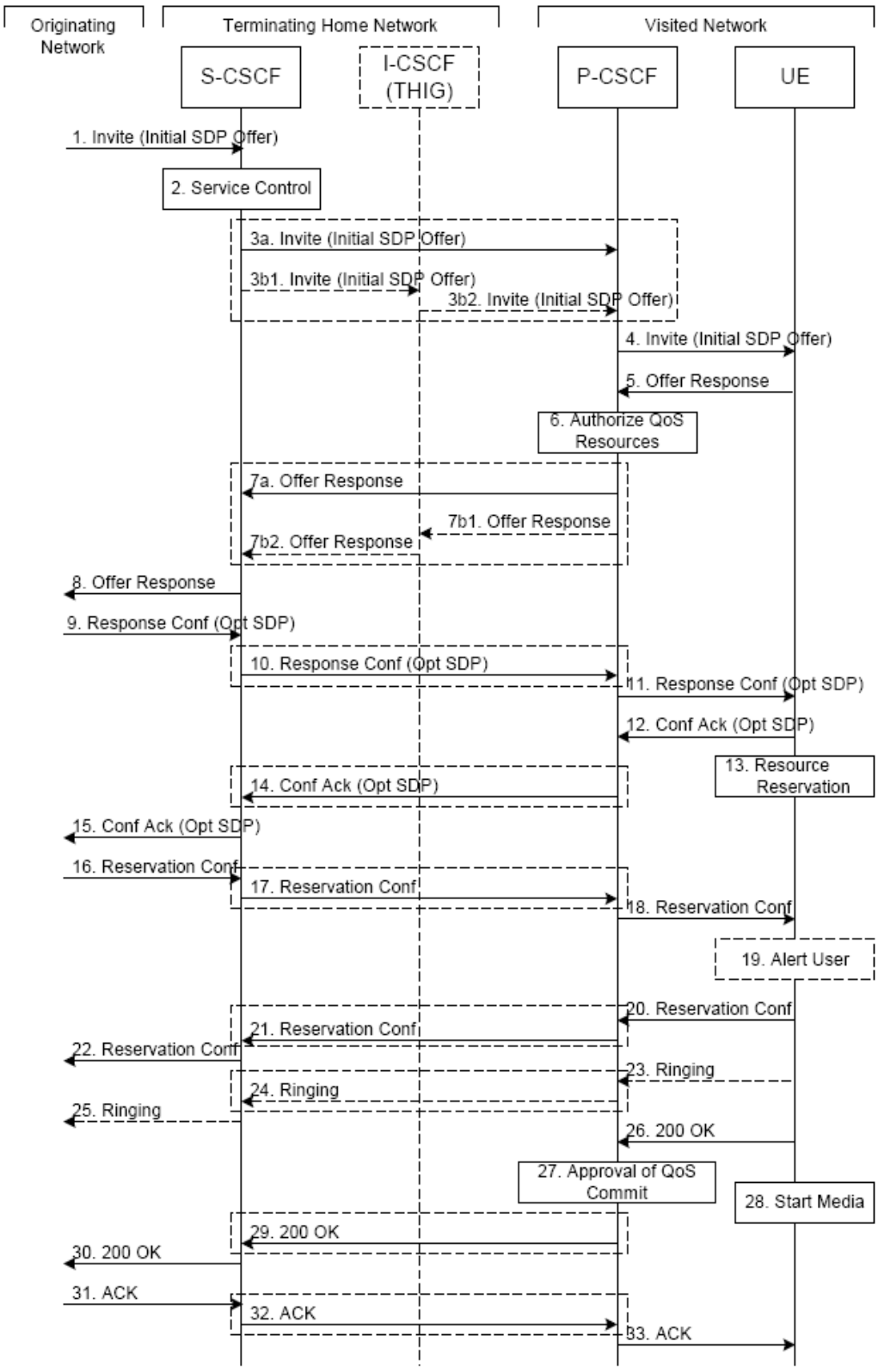

Figure - 5.10: Mobile termination procedure – roaming [112]

5.5.1.3 Mobile termination procedures

The session termination procedures specify the signaling path between the Serving CSCF assigned to perform the session termination service and the UE. Same as discussed in the origination procedures, this path is determined at the time of UE registration. However the signaling flows are in the reverse direction of the session-initiation signaling flows.

Procedure MT#1 is as following [112]:

1. The originating party sends the SIP INVITE request, containing an initial SDP, via one of the origination procedures, and via one of the Inter-Serving procedures, to the Serving-CSCF for the terminating users.
2. S-CSCF validates the service profile, and invokes any termination service logic required for this user. This includes authorization of the requested SDP based on the user's subscription for multi-media services.
3. S-CSCF remembers (from the registration procedure) the next hop CSCF for this UE. If the home network operator does not desire to keep their network configuration hidden, the INVITE request is forwarded directly to the P-CSCF (choice a). If the home network operator desires to keep their network configuration hidden, the INVITE request is forwarded through an I-CSCF (THIG) to the P-CSCF (choice b).
4. The PDF generates the Authorization-Token and includes it in the INVITE message. P-CSCF remembers the UE address from the registration procedure, and forwards the INVITE to the UE.
5. UE determines the subset of the media flows proposed by the originating endpoint that it supports, and responds with an Offer Response message back to the originator. The SDP may represent one or more media for a multi-media session. This response is sent to P-CSCF.
6. P-CSCF authorizes the resources necessary for this session.
7. P-CSCF forwards the Offer Response message to S-CSCF. Based on the choice made in step #3 above, this may be sent directly to S-CSCF (7a) or may be sent through I-CSCF (THIG) (7b1 and 7b2).
8. S-CSCF forwards the Offer Response message to the originator, per the S-S procedure.
9. The originating endpoint sends a Response Confirmation via the S-S procedure, to S-CSCF. The Response Confirmation may also contain SDP. This may be the same SDP as in the Offer Response sent in Step 8 or a subset. If new media are defined by this SDP, the P-CSCF (PDF) following Step 12 will do a new authorization (as in Step 6). The originating UE is free to continue to offer new media on this operation or on subsequent exchanges using the Update method. Each

offer/answer exchange will cause the P-CSCF (PDF) to repeat the Authorization step (Step 6) again.

10. S-CSCF forwards the Response Confirmation to P-CSCF. This may possibly be routed through the I-CSCF depending on operator configuration of the I-CSCF.

11. P-CSCF forwards the Response Confirmation to UE.

12. UE responds to the Response Confirmation with an acknowledgement. If Optional SDP is contained in the Response Confirmation, the Confirmation Ack will also contain an SDP response. If the SDP has changed, the P-CSCF authorizes the resources again.

13. UE initiates the reservation procedures for the resources needed for this session.

14-15. P-CSCF forwards the Confirmation Ack to the S-CSCF and then to the originating end point via session path. Step 14 may be similar to Step 7 depending on whether or not configuration hiding is used.

16-18. When the originating endpoint has completed its resource reservation, it sends the successful Resource Reservation message to S-CSCF, via the S-S procedures. The S-CSCF forwards the message toward the terminating endpoint along the signaling path. Step 17 may be similar to Step 3 depending on whether or not configuration hiding is used.

19. UE#2 alerts the destination user of an incoming session set up attempt.

20-22. UE#2 responds to the successful resource reservation towards the originating end point. Step 21 may be similar to Step 7 depending on whether or not configuration hiding is used.

23-25. UE may alert the user and wait for an indication from the user before completing the session set up. If so, it indicates this to the originating party by a provisional response indicating Ringing. This message is sent to P-CSCF and along the signaling path to the originating end. Step 24 may be similar to Step 7 depending on whether or not configuration hiding is used. Indicates the resources reserved for this session should now be committed.

28. UE starts the media flow(s) for this session

29-30. P-CSCF sends a SIP 200-OK final response along the signaling path back to the S-CSCF. Step 29 may be similar to Step 7 depending on whether or not configuration hiding is used.

31-33. The originating party responds to the 200-OK final response with a SIP ACK message that is sent to S-CSCF via the S-S procedure and forwarded to the terminating end along the signaling path. Step 32 may be similar to Step 3 depending on whether or not configuration hiding is used.

The detailed information flow of MT#2 will not be described here. The procedures are no much different except the P-CSCF and S-CSCF involved are in the same network.

5.5.1.4 Summary of The Session Setup Procedures

If group the CSCFs according to the UE they are serving but not he networks they are in, and assume no topology hiding (I-CSCF (THIG)) is utilized so the P-CSCF knows the address of the S-CSCF, will get a common session flow between two mobiles as following no matter how the session is built by different combination of the origination, S-CSCF to S-CSCF and termination procedures.

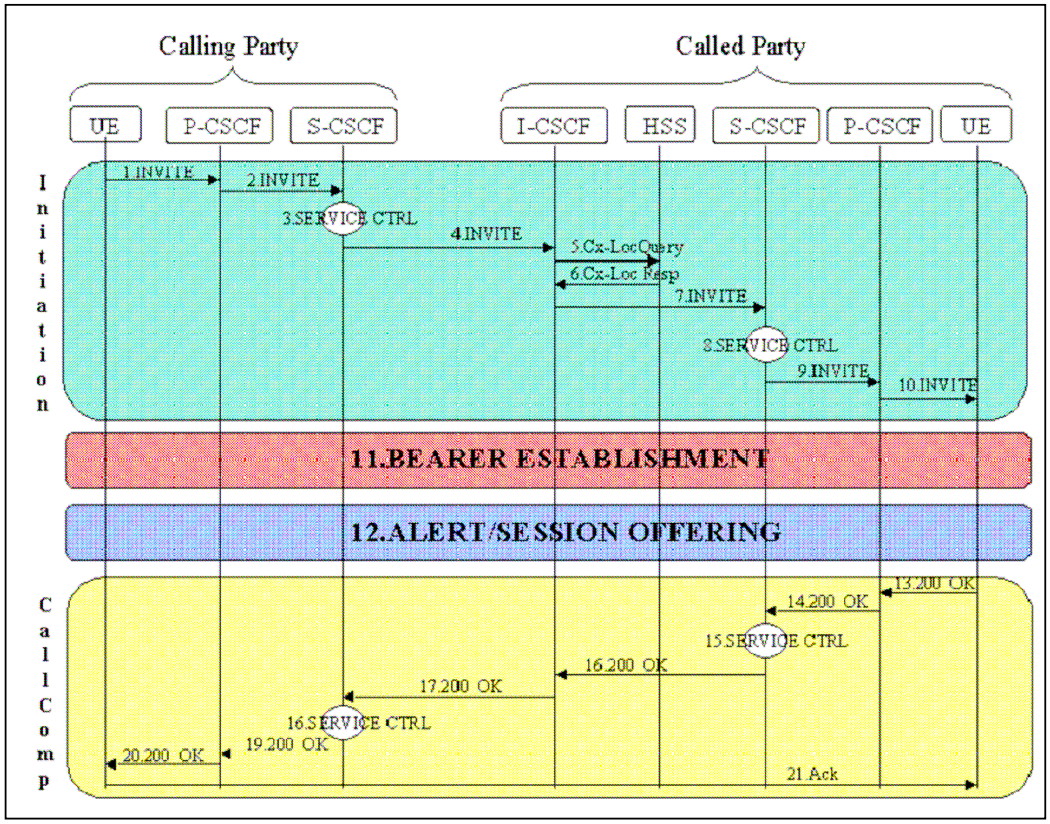

Figure – 5.11: Simplified Mobile-to-Mobile Call flow

In short, the calling party sends the INVITE message through P-CSCF to the S-CSCF that is already known by registration. The message is then sent to the I-CSCF, which is the first contact point of the home network of the called party for incoming network signaling. Then the message is routed to serving and proxy CSCFs of the called party. Subsequently the bearer is established and the called user is alerted. When the called user answers the call the OK message is routed via CSCFs used. Calling party acknowledges to the called user and call establishment is complete.

In the figure, configuration hiding is not applied. The CSCFs involved may or may not be in same network, depending on the different scenarios.

5.5.2 Session Release Procedures

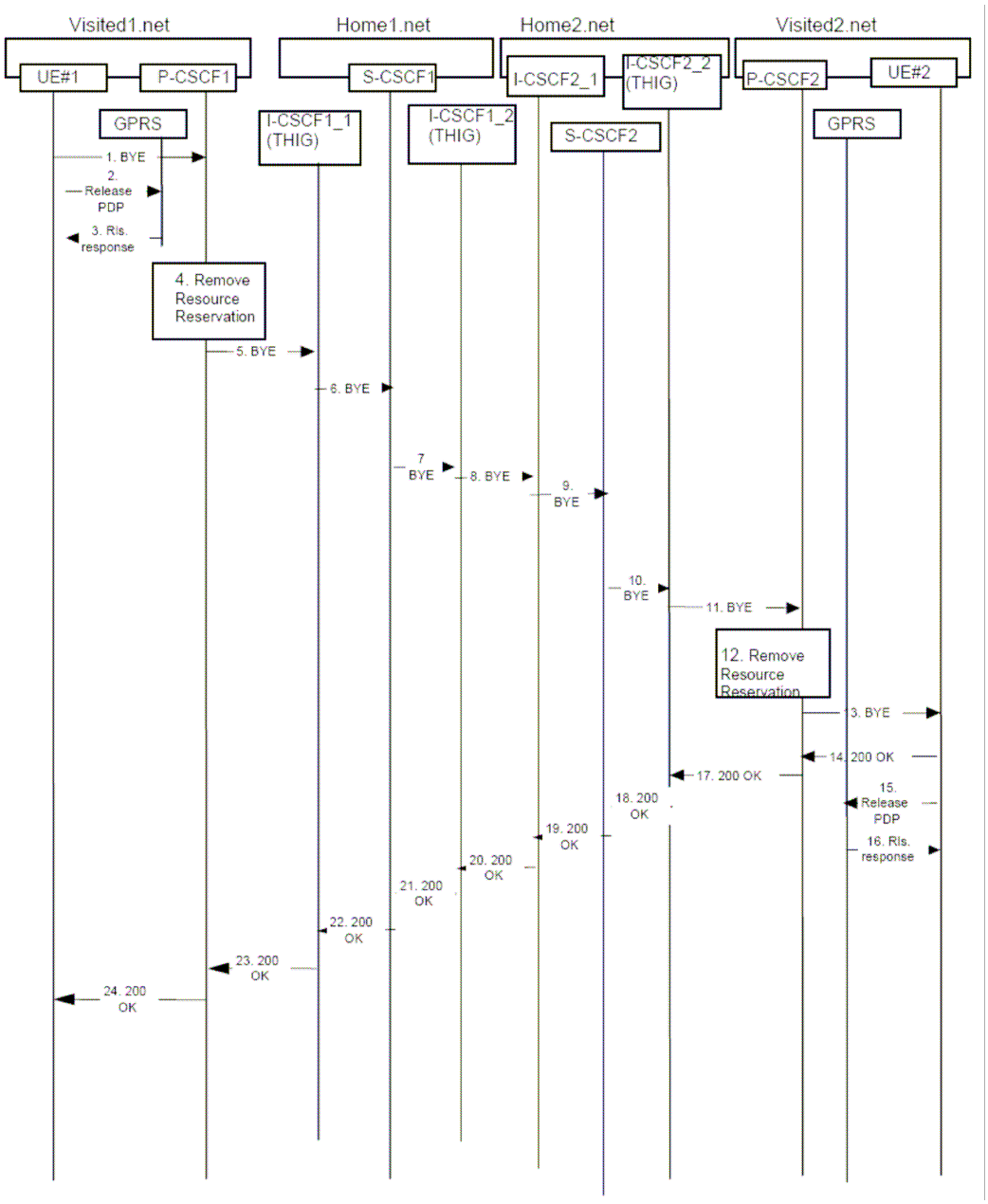

Figure – 5.12: Mobile Initiated Session Release [113]

The above flow shows a mobile terminal initiated SIP session release. It is assumed that the session is active and that the bearer was established directly between the two visited networks. Here the visited networks could be the Home network in either or both cases, and the use of I-CSCF (THIG) are optional.

1. One mobile party hangs up, which generates a SIP BYE request from the UE to the P-CSCF.

2. Steps 2 and 3 may take place before or after Step 1 and in parallel with Step 4. The UE initiates the release of the bearer PDP context. The GPRS subsystem releases the PDP context. The IP network resources that had been reserved for the message receive path to the mobile for this session are now released. This is initiated from the GGSN. If RSVP was used to allocated resources, then the appropriate release messages for that protocol would invoked here.
3. The GPRS subsystem responds to the UE.
4. The P-CSCF/PDF removes the authorization for resources that had previously been issued for this endpoint for this session. This step will also result in a release indication to the GPRS subsystem to confirm that the IP bearers associated with the session have been deleted
5. The P-CSCF sends a SIP BYE request to the I-CSCF (THIG) hiding the S-CSCF of the releasing party.
6. The I-CSCF (THIG) sends a SIP BYE request to the S-CSCF of the releasing party.
7. The SIP BYE request is sent from the S-CSCF to the I-CSCF (THIG).
8. The SIP BYE request is sent from the I-CSCF (THIG) to the I-CSCF of the network of the other party.
9. The SIP BYE request is forwarded from the I-CSCF that was used to determine the location of S-CSCF of the other party.
10. The SIP BYE request is forwarded to the I-CSCF (THIG).
11. The I-CSCF (THIG) forwards the SIP BYE request to the P-CSCF.
12. The P-CSCF removes the authorization for resources that had previously been issued for this endpoint for this session. This step also results in a release indication to the GPRS subsystem to confirm that the IP bearers associated with the UE#2 session have been deleted.
13. The P-CSCF forwards the SIP BYE request on to the UE.
14. The mobile responds with a 200 OK response, which is sent back to the P-CSCF.
15. Steps 15 and 16 may be done in parallel with step 14. The Mobile initiates the release of the bearer PDP context.
16. The GPRS subsystem releases the PDP context. The IP network resources that were reserved for the message receive path to the mobile for this session are now released. This is initiated from the GGSN. If RSVP was used to allocated resources, then the appropriate release messages for that protocol would invoked here.
17. The P-CSCF sends the 200 OK to the I-CSCF (THIG).
18. The I-CSCF (THIG) sends the 200 OK to the S-CSCF.
19. The S-CSCF of the other party forwards the 200 OK to its selecting I-CSCF.

20. The selecting I-CSCF forwards the 200 OK to the I-CSCF (THIG).
21. The I-CSCF (THIG) forwards the 200 OK to the S-CSCF.
22. The S-CSCF of the releasing party forwards the 200 OK to the I-CSCF (THIG).
23. The I-CSCF (THIG) forwards the 200 OK to the P-CSCF of the releasing party.
24. The P-CSCF of the releasing party forwards the 200 OK to the UE.

5.6 SIP Message Structure

SIP is a text-based protocol that is similar to HTTP, which makes it easy read and understand. A SIP message is either a request from a client to a server or a response from a server to a client. Both the request and the response contain a start-line followed by one or more headers and a message body. For example:

message = start-line

*message header

CRLF

[message-body]

The request line specifies the type of request being issued, while the response line indicates the success or failure of a request. If a request is not executed, the status line indicates the type of failure or the reason for the failure.

• SIP messages are either a request or a response

• Structure of a SIP message includes:

Start line

Header(s)

Body

• Sample structure

message = start-line

*message header

CRLF

Example An overview of SIP message structure

5.6.1 SIP request

A SIP request consists of a method token, a request URI, and the SIP version. A method token is used to identify the request. The request URI is the address of the device where the request is being sent.

Methods for handling different kinds of requests

- INVITE
- ACK
- BYE
- CANCEL
- OPTIONS
- REGISTER

SIP method extensions include:

- SUBSCRIBE
- NOTIFY
- MESSAGE
- INFO
- SERVICE
- REFER
- NEGOTIATE

SIP request methods

The original SIP RFC 3261 [121] defines six methods, which are used for different types of requests. The following table describes these methods.

Table - 5.1: SIP Method with Description

Method Name	Description
INVITE	It initiates a session. This method includes information about the calling and called users and the type of media that is to be exchanged.
ACK	Sent by the client who sends the INVITE.ACK is sent to confirm that the session is established. Media can then be exchanged.
BYE	Terminate a session. This method can be sent be either user.
CANCEL	Terminates a pending request, such as an outstanding INVITE. After a session is established, a BYE method needs to be used to terminate the session.
OPTIONS	Queries the capabilities of the server or the other devices. It can be used to check media capabilities before issuing an INVITE
REGISTER	Used by a client to login and register its address with a SIP registrar server.

5.6.2 SIP method extensions

A number of extensions and enhancements have been made to the original SIP RFC 2543. This includes the addition of the following new methods to SIP, which can be used for event notification, instant messaging and call control:

- **SUBSCRIBE:** The SUBSCRIBE method enables a user to subscribe to certain events. This means that the user should be informed when such events occur.

- **NOTIFY**: The NOTIFY method is used to inform the user that a subscribed event has occurred. Windows Messenger uses the SUBSCRIBE method to request contacts, groups, and allow and block lists from the server and to get the presence of contacts in a group. Live Communications Server 2003 uses the NOTIFY method to deliver the data obtained by the SUBSCRIBE method to the client.

- **MESSAGE**: SIP can also be used for Instant Messaging. A user sends an instant message to another user by sending a request that includes the MESSAGE method. This request carries the actual text in a body of a SIP packet.

- **INFO**: The INFO method is used for transferring information during a session, such as user activity. For example, Windows Messenger 5.0 uses the INFO method to inform the called user that Bob, the calling user, is typing on the keyboard. As a result, in the conversation UI, the called user sees a dialog, “bob is typing.”

- **SERVICE**: The SERVICE method can carry a Simple Object Access Protocol (SOAP) message as its payload. Windows Messenger 5.0 uses the SERIVCE method to add contacts and groups on the server. This method is also used to search for contacts in the SIP domain.

- **NEGOTIATE**: The NEGOTIATE method is used to negotiate various kinds of parameters, such as security mechanisms and algorithms. Live Communications Server 2003 uses the NEGOTIATE method to provide compression between clients and servers.

- **REFER**: A REFER request enables the sender of the request to instruct the receiver to contact a third party using the contact details provided in the request. Call Transfer is a commonly used application of the REFER method.

5.6.3 SIP Response

SIP response contains:

• Status code, three-digit number indicating the outcome of the request

• Reason phrase, provides a textual description of the outcome

Different classes of a response:

- 1xx: provisional
- 2xx: success
- 6xx: global failure
- 3xx: redirection
- 4xx: client server
- 5xx: server error
- 6xx: global failure

A SIP response contains a status code, which is a three-digit number that indicates the outcome of the request. The response also contains a reason phrase, which provides a textual description of the outcome of the request. The reason code is interpreted and acted upon by the client software. The reason phrase helps the user understand the response. Status codes defined in SIP have values between 100 and 699 and the first digit of the reason code indicates the response class. For example, all the status codes between 100 and 199 belong to one class.

Table 5.2: Different classes in SIP

Class name	Description
1xx: Provisional	Request received, continuing to process the request. for example, 180 indicates that the phone of the called user is ringing.
2xx: Success	Action was successfully received, understood, and accepted. Only 200 OK and 202 ACCEPTED have been defined in this class
3xx: Redirection	Further action needs to be taken to complete the request. For example, a front-end server 302 to redirect the client to a home server.
4xx: Client Error	Request contains bad syntax or cannot be fulfilled at this server. For example a home server sends a response, 401 Unauthorized, if the client needs to provide credentials.
5xx: Server Error	Server failed to fulfill a valid request. For example a server sends a response, 504 timeout, if the MTLS has not been configured between the home servers.
6xx:Global Failure	Request cannot be fulfilled at any server. This is a new class defined for SIP, but is not currently used with live communication server 2003

5.6.4 SIP Headers

SIP includes a number of message headers in a SIP message. These headers contain information that enables the receiver to understand the message better or handle the message properly. Some headers make sense only in certain requests or responses. In some cases, the presence of a particular header depends on the context. The presence of a particular header in a response might be reasonable only if the response is issued to a specific request.

General headers:

- Used in both requests and responses.
- Contains basic information needed for the handling of requests and responses
- Examples: the To and From header fields

Request header:

- Apply only to SIP requests.
- Provide additional information to the server regarding the request itself or regarding the client.
- Examples: the Subject and Priority header fields

Response Header:

- Apply only to response (status) messages.
- Provide further information about the response that cannot be included in the status line.
- Examples: unsupported and Retry after header fields

5.7 A Simple SIP Example

Figure 4.13 shows the SIP message exchange between two SIP-enabled devices. The two devices could be SIP phones, hand-held, palmtops, or cell phones. It is assumed that both devices are connected to an IP network such as the Internet and know each other's IP address [117].

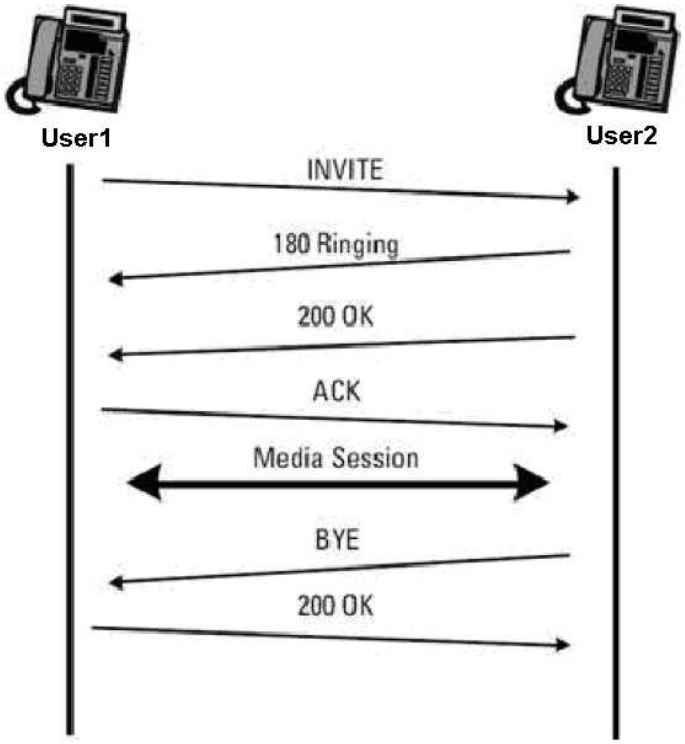

Figure – 5.13: A simple SIP example

The calling party, Tesla, begins the message exchange by sending a SIP INVITE message to the called party, User1. The INVITE contains the details of the type of session or call that is requested. It could be a simple voice (audio) session, a multimedia session such as a videoconference, or it could be a gaming session.

The INVITE message contains the following fields:

INVITE sip:user1@imstestbed.net SIP/2.0

Via: SIP/2.0/UDP lab.imstestbed.net:5060

To: User1 <sip:User1@imstestbed.net>

From: User2 <sip:user2@imstestbed.net>

Call-ID: 123456789@lab.imstestbed.net

CSeq: 1 INVITE

Subject: About That Power Outage...

Contact: sip:user2@imstestbed.net

Content-Type: application/sdp

Content-Length: 158

v=0

o=Tesla 2890844526 2890844526 IN IP4 lab.imstestbed.net

s=Phone Call

c=IN IP4 100.101.102.103

t=0 0

m=audio 49170 RTP/AVP 0

a=rtpmap:0 PCMU/8000

The fields listed in the INVITE message are called headers. They have the form Header: Value CRLF. The first line of the request message, called the start line, lists the method, which is INVITE, the Request-URI (Uniform Resource Indicator), then the SIP version number (2.0), all separated by spaces. Each line of a SIP message is terminated by a CRLF [122]. The Request-URI is a special form of SIP URL and indicates the resource to which the request is being sent. SIP URLs.

The first header following the start line is a via header. Each SIP device that originates or forwards a SIP message stamps its own address in a via header, usually written as a host name that can be resolved into an IP address using a DNS query. The Via header contains the SIP Version number (2.0), a "/", then UDP for UDP transport, a space, then the hostname or address, a colon, then a port number, in this example the "well-known" SIP port number 5060.

The next headers are the To and From headers, which show the originator and destination of the SIP request. When a name label is used, as in this example, the SIP URL is enclosed in brackets and used for routing the request. The name can be displayed during alerting. The Call-ID header has the same form as an e-mail address but is actually an identifier used to keep track of a particular SIP session. The originator of the request creates a locally unique string, then usually adds an "@" and its host name to make it globally unique. The combination of the local address (From header), remote address (To header), and Call-ID identifies the "call leg." Both parties to identify this call because they could have multiple calls set up between them use the call leg. Subsequent requests for this call will refer to this call leg.

The next header shown is the CSeq, [119] or command sequence. It contains a number, followed by the method name, INVITE in this case. This number is incremented for each new request sent. In this example, the command sequence number is initialized to 1, but it could start at another value. The Via headers plus the To, From, Call-ID, and CSeq headers represent the minimum required header set in any SIP message. Other headers can be included as optional additional information, or information needed for a specific request type.

A Contact header is included in this message, which contains the SIP URL of Tesla; this URL can be used to route messages directly to Tesla. The optional Subject header is present in this example. It is not used by the protocol, but could be displayed during alerting to aid the called party in deciding whether to accept the call. The same sort of useful prioritization and screening all routinely do using the Subject and From headers in an e-mail message is also possible with a SIP INVITE request. Additional headers are present in this

INVITE message, which contain the media information necessary to set up the call.

The Content-Type and Content-Length headers indicate that the message body is Session Description Protocol (SDP) [116] and contains 158 octets of data. A blank line separates message body from the header list, which ends with the Content-Length header.

In this case, there are seven lines of SDP data describing the media attributes that the caller Tesla desires for the call. This media information is needed because SIP makes no assumptions about the type of media session to be established—the caller must specify exactly what type of session (audio, video, gaming) that he wishes to establish. The SDP field names are listed in Table 5.3. A quick review of the lines shows the basic information necessary to establish a session. This includes the:

- Connection IP address (100.101.102.103);
- Media format (audio);
- Port number (49170);
- Media transport protocol (RTP);
- Media encoding (PCM μ Law);
- Sampling rate (8000 Hz).

Table - 5.3: SDP data

SDP parameter	**Parameter Name**
v=0	Version number
o= IP4 lab.imstestbed.net	
Origin containing name	
s=Phone Call	Subject
c=IN IP4 100.101.102.103	Connection
t=0 0	Time
m=audio	49170 RTP/AVP 0 Media
a=rtpmap:0 PCMU/8000	Attributes

INVITE is an example of a SIP request message. There are five other methods or types of SIP requests currently defined in the SIP specification. The next message in Figure 5.6 is a 180 Ringing message sent in response to the INVITE. This message indicates that the called party User1 has received the INVITE and that alerting is taking place. The alerting could be ringing a phone, flashing a message on a screen, or any other method of attracting the attention of the called party, User1.

The 180 Ringing is an example of a SIP response message. Responses are numerical and are classified by the first digit of the number. A 180 response is

an "informational class" response, identified by the first digit being a 1. Informational responses are used to convey non-critical information about the progress of the call. SIP response codes were based on HTTP version 1.1 response codes with some extensions and additions. Anyone who has ever browsed the World Wide Web has likely received a "404 Not Found" response from a web server when a requested page was not found. 404 Not Found is also a valid SIP "client error class" response in a call to an unknown user.

Response code number in SIP alone determines the way the response is interpreted by the server or the user. The reason phrase, Ringing in this case, is suggested in the standard, but any text can be used to convey more information. For instance, 180 Hello! is a perfectly valid SIP response.

The 180 Ringing response has the following structure:

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP lab.imstestbed.net:5060

To: User1 <sip:user1@imstestbed.net>

From: User2 <sip:user2@imstestbed.net>

Call-ID: 123456789@imstestbed.net

CSeq: 1 INVITE

Content-Length: 0

The message was created by copying many of the headers from the INVITE message, including the Via, To, From, Call-ID, and CSeq, then adding a response start line containing the SIP version number, the response code, and the reason phrase. This approach simplifies the message processing for responses.

Note that the To and From headers are not reversed in the response message as one might expect them to be. Even though this message is sent to User1 from Tesla, the headers read the opposite. This is because the To and From headers in SIP are defined to indicate the direction of the request, not the direction of the message. Since Tesla initiated this request, all messages will read To: User1 From: Tesla.

When the called party decides to accept the call (i.e., the phone is answered), a 200 OK response is sent. This response also indicates that the type of media session proposed by the caller is acceptable. The 200 OK is an example of a "success class" response. The 200 OK message body contains User1's media information:

SIP/2.0 200 OK

Via: SIP/2.0/UDP lab.imstestbed.net:5060

To: User1 <sip:user1@imstestbed.net>

From: User2 <sip:user2@imstestbed.net>

Call-ID: 123456789@lab.imstestbed.net

CSeq: 1 INVITE

Contact: sip:user1@imstestbed.net

Content-Type: application/sdp

Content-Length: 155

v=0

o=User1 2890844526 2890844526 IN IP4 tower.radio.org

s=Phone Call

c=IN IP4 200.201.202.203

t=0 0

m=audio 60000 RTP/AVP 0

a=rtpmap:0 PCMU/8000

This response is constructed the same way as the 180 Ringing responses. The media capabilities, however, must be communicated in a SDP message body added to the response. From the same SDP fields as Table 4.3, the SDP contains:

- End-point IP address (200.201.202.203);
- Media format (audio);
- Port number (60000);
- Media transport protocol (RTP);
- Media encoding (PCM μ Law);
- Sampling rate (8000 Hz).

The final step is to confirm the media session with an "acknowledgment" request. The confirmation means that Tesla can support the media session proposed by User1.

This exchange of media information allows the media session to be established using another protocol, RTP in this example.

ACK sip:user1@imstestbed.net SIP/2.0

Via: SIP/2.0/UDP lab.imstestbed.net:5060

To: User1 <sip:user1@imstestbed.net>

From: User2 <sip:user2@imstestbed.net>

Call-ID: 123456789@lab.imstestbed.net

CSeq: 1 ACK

Content-Length: 0

The command sequence, CSeq, has the same number as the INVITE, but the method is set to ACK. At this point, the media session begins using the media information carried in the SIP messages. The media session takes place using another protocol, typically RTP. This message exchange shows that SIP is an end-to-end signaling protocol. A SIP network or SIP server is not required for the protocol to be used. Two end-points running a SIP protocol stack and knowing each other's IP addresses can use SIP to set up a media session between them.

Although less obvious, this example also shows the client-server nature of the SIP protocol. When Tesla originates the INVITE request, he is acting as a SIP client. When User1 responds to the request, he is acting as a SIP server. After the media session is established, the BYE request and acts as the SIP client, while Tesla acts as the SIP server when he responds. This is why a SIP-enabled device must contain both SIP server and SIP client software—during a typical session, both are needed. This is quite different from other client-server Internet protocols such as HTTP or FTP. The web browser is always an HTTP client, and the web server is always an HTTP server, and similarly for FTP. In SIP, an endpoint will switch back and forth during a session between being a client and a server.

In Figure 5.13, a BYE request is sent by User1 to terminate the media session:

BYE sip:user2@imstestbed.net SIP/2.0

Via: SIP/2.0/UDP tower.radio.org:5060

To: User2 <sip:user2@imstestbed.net>

From: User1 <sip:user1@imstestbed.net>

Call-ID: 123456789@lab.imstestbed.net

CSeq: 1 BYE

Content-Length: 0

The Via header in this example is populated with User1's host address. The To and From headers reflect that this request is originated by User1, as they are reversed from the messages in the previous transaction. Tesla, however, is able to identify the call leg and tear down the correct media session.

The confirmation response to the BYE is a 200 OK:

SIP/2.0 200 OK

Via: SIP/2.0/UDP tower.radio.org:5060

To: User2 <sip:user2@imstestbed.net>

From: User1 <sip:user1@imstestbed.net>

Call-ID: 123456789@lab.imstestbed.net

CSeq: 1 BYE

Content-Length: 0

The response echoes the CSeq of the original request: 1 BYE.

5.8 Tools to read SIP Messages

The SIP Logger and SIP Parser tools can be used to read a SIP message.

5.8.1 SIP Logger:

SIP Logger is an executable file with a read me file, LoggerReadme.htm that contains information about the installation and use of SIP Logger. SIP Logger allows the Live Communications Server administrator to log SIP traffic on the server before it is encrypted. SIP Logger helps the administrator debug communication issues between clients and servers and between multiple servers. Messages are sent to a text file that is selected during SIP Logger startup [43]. The default maximum size of the file is 100 MB. Note that SIP Logger is not intended to replace the Live Communications Server IM Archiving Server.

5.8.2 SIP Parser:

SIP Parser is a dynamic-link library (DLL) that works with Network Monitor to enable parsing of the SIP signaling information exchanged between clients and servers. SIP Parser captures only SIP messages sent over TCP or UDP. TLS is not supported because of encryption. SIP Parser requires the installation of Network Monitor on the computer where SIP messages need to be parsed. A readme file, SipParserReadme.html, contains information about the installation and use of SIP Parser.