Real-Time Traffic

- The widespread deployment of high-speed LANs and WANs and the increase in the line capacity on the Internet and other internets has opened up the possibility of using IP-based networks for the transport of real-time traffic.

- Requirements of real-time traffic differ from those of high-speed but non-real-time traffic.

- With traditional internet applications, such as file transfer, electronic mail, and client/server applications including the Web, the performance metrics of interest are generally throughput and delay.

- There is also a concern with reliability, and mechanisms are used to make sure that no data are lost, corrupted, or misordered during transit.

- By contrast, real-time applications are more concerned with timing issues.

- In most cases, there is a requirement that data be delivered at a constant rate equal to the sending rate.

- In other cases, a deadline is associated with each block of data, such that the data are not usable after the deadline has expired.
Real-Time Traffic Example

Figure 24.5 Real-Time Traffic
Real-Time Traffic Profiles

- **Continuous data source:**
  Fixed-size packets are generated at fixed intervals. This characterizes applications that are constantly generating data, have few redundancies, and that are too important to compress in a lossy way. Examples are air traffic control radar and real-time simulations.

- **On/off source:**
  The source alternates between periods when fixed-size packets are generated at fixed intervals and periods of inactivity. A voice source, such as in telephony or audio conferencing, fits this profile.

- **Variable packet size:**
  The source generates variable-length packets at uniform intervals. An example is digitized video in which different frames may experience different compression ratios for the same output quality level.
Real-Time Traffic Requirements

- Low jitter
- Low latency
- Ability to easily integrate non-real-time and real-time services
- Adaptable to dynamically changing network and traffic conditions
- Good performance for large networks and large numbers of connections
- Modest buffer requirements within the network
- High effective capacity utilization
- Low overhead in header bits per packet
- Low processing overhead per packet within the network and at the end system
Hard vs. Soft Real-Time Applications

• **Soft real-time applications:**
  - can tolerate the loss of some portion of the communicated data.
  - impose fewer requirements on the network
  - focus on maximizing network utilization, even at the cost of some lost or misordered packets.

• **Hard real-time applications:**
  - have zero loss tolerance.
  - a deterministic upper bound on jitter and high reliability take precedence over network utilization considerations.
Session Initiation Protocol (SIP)

- The Session Initiation Protocol (SIP), is an application-level control protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network.

- The key driving force behind SIP is to enable Internet telephony, also referred to as Voice over IP (VoIP).

- SIP can support any type of single media or multimedia session, including teleconferencing.

- SIP supports five facets of establishing and terminating multimedia communications:
  - User location:
    Users can move to other locations and access their telephony or other application features from remote locations.
  - User availability:
    Determination of the willingness of the called party to engage in communications.
  - User capabilities:
    Determination of the media and media parameters to be used.
  - Session setup:
    Setup up point-to-point and multiparty calls, with agreed session parameters.
  - Session management:
    Including transfer and termination of sessions, modifying session parameters, and invoking services.
SIP Design Elements

- SIP employs design elements developed for earlier protocols.

- SIP is based on an HTTP-like request/response transaction model.

- Each transaction consists of a client request that invokes a particular method, or function, on the server and at least one response.

- SIP uses most of the header fields, encoding rules, and status codes of HTTP. This provides a readable text-based format for displaying information.

- SIP also uses concepts similar to the recursive and iterative searches of DNS.

- SIP incorporates the use of a Session Description Protocol (SDP), which defines session content using a set of types similar to those used in MIME.
SIP Components and Protocols
SIP Components

• **User Agent:**
  Resides in every SIP end station.
  - User Agent Client (UAC):
    issues SIP requests
  - User Agent Server (UAS):
    Receives SIP requests and generates a SIP response that accepts, rejects, or redirects the request.

• **Redirect Server:**
  Used during session initiation to determine the address of the called device, which it returns to the calling device, directing the UAC to contact an alternate URI.
  This is analogous to iterative searches in DNS.

• **Proxy Server:**
  An intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients.
  A proxy server primarily ensures that a request is sent to another entity closer to the targeted user.
  Proxies are also useful for enforcing policy.
  A proxy interprets, and, if necessary, rewrites specific parts of a request message before forwarding it.

• **Registrar:**
  Accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles.

• **Location Service:**
  Is used by a SIP redirect or proxy server to obtain information about a callee's possible location(s).
  For this purpose, the location service maintains a database of SIP-address/IP-address mappings.
SIP Servers and Protocols

• The various servers are defined as logical devices.

• They may be implemented as separate servers configured on the Internet or they may be combined into a single application that resides in a physical server.

• A user agent acting as a client (UAC Alice) uses SIP to set up a session with a user agent that will act as a server (UAS Bob).

• The session initiation dialogue uses SIP and involves one or more proxy servers to forward requests and responses between the two user agents.

• The user agents make use of the Session Description Protocol (SDP), which is used to describe the media session.

• The proxy servers may also act as redirect servers as needed. If redirection is done, a proxy server will need to consult the location service database, which may be collocated with a proxy server or not. The communication between the proxy server and the location service is beyond the scope of the SIP standard.

• DNS is also an important part of SIP operation. Typically, a UAC will make a request using the domain name of the UAS, rather than an IP address. A proxy server will need to consult a DNS server to find a proxy server for the target domain.
• SIP typically runs on top of UDP for performance reasons, and provides its own reliability mechanisms, but may also use TCP.

• If a secure, encrypted transport mechanism is desired, SIP messages may alternatively be carried over the Transport Layer Security (TLS) protocol.

• Associated with SIP is the Session Description Protocol (SDP).

• SIP is used to invite one or more participants to a session, while the SDP-encoded body of the SIP message contains information about what media encodings (e.g., voice, video) the parties can and will use.

• Once this information is exchanged and acknowledged, all participants are aware of the participants' IP addresses, available transmission capacity, and media type.

• At this point, data transmission begins, using an appropriate transport protocol. Typically, the Real-Time Transport Protocol (RTP) is used.

• Throughout the session, participants can make changes to session parameters, such as new media types or new parties to the session, using SIP messages.
SIP Protocol Stack and Messages

(a) SIP Protocol Stack

(b) SIP Request/Response Signaling Message Types

<table>
<thead>
<tr>
<th>SIP message</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE</td>
<td>Invites a user to join a call/session</td>
</tr>
<tr>
<td>ACK</td>
<td>Used to acknowledge receipt of an INVITE response message</td>
</tr>
<tr>
<td>REGISTER</td>
<td>Used to inform a SIP redirect server of the current location of a user</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>Used to request the capabilities of a host device</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Terminates a search for a user</td>
</tr>
<tr>
<td>BYE</td>
<td>Inform the other user(s) that the user is leaving a call/session</td>
</tr>
</tbody>
</table>

Internet telephony: (a) example host device protocol stack; (b) SIP request/response signaling message types.
SIP Uniform Resource Identifier (URI)

- A resource within a SIP network is identified by a Uniform Resource Identifier (URI).

- Examples of communications resources include the following:
  - A user of an online service;
  - An appearance on a multiline phone;
  - A mailbox on a messaging system;
  - A telephone number at a gateway service;
  - A group (such as "sales" or "helpdesk") in an organization.

- SIP URIs have a format based on email address formats, namely user@domain.

- An ordinary SIP URI is of the form:
  sip:bob@biloxi.com

- The URI may also include a password, port number, and related parameters.

- If secure transmission is required, "sip:" is replaced by "sips:". In this case, SIP messages are transported over TLS.
SIP Call Setup Attempt Scenario
SIP Presence Example
SIP Registration and Notification Example

1. REGISTER
2. Update Database
3. OK
4. NOTIFY
5. NOTIFY

6. NOTIFY
   <Signed In>
7. NOTIFY
8. 200 OK
9. 200 OK
10. 200 OK

User Agent alice

User Agent bob (1.2.3.4)
SIP Successful Call Setup
SIP Messages

• SIP is a text-based protocol with a syntax similar to that of HTTP.

• There are two different types of SIP messages, requests and responses.

• The first line of a request has a method, defining the nature of the request and a Request-URI, indicating where the request should be sent.

• The first line of a response has a response code.

• All messages include a header, consisting of a number of lines, each line beginning with a header label.

• A message can also contain a body, such as an SDP media description.
SIP Request Methods

• **REGISTER:**
  Used by a user agent to notify a SIP network of its current IP address and the URLs for which it would like to receive calls

• **INVITE:**
  Used to establish a media session between user agents

• **ACK:**
  Confirms reliable message exchanges

• **CANCEL:**
  Terminates a pending request, but does not undo a completed call

• **BYE:**
  Terminates a session between two users in a conference

• **OPTIONS:**
  Solicits information about the capabilities of the callee, but does not set up a call
SIP Request Message Example

(Message (1) in Successful Call Setup Example)

INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP 12.26.17.91:5060
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@12.26.17.91
CSeq: 314159 INVITE
Contact: <sip:alice@atlanta.com>
Content-Type: application/sdp
Content-Length: 142
SIP Request Message Headers

• **Via:**
  show the path the request has taken in the SIP network and is used to route responses back along the same path. It contains the IP address (12.26.17.91), port number (5060), and transport protocol (UDP) that Alice wants Bob to use in his response.

• **Max-Forwards:**
  limits the number of hops a request can make on the way to its destination. It consists of an integer that is decremented by one by each proxy that forwards the request. If it reaches 0 before the request reaches its destination, it will be rejected with a 483 (Too Many Hops) error response.

• **To:**
  contains a display name (Bob) and a SIP/SIPS URI toward which the request was originally directed.

• **From:**
  contains a display name (Alice) and a SIP/SIPS URI that indicate the originator of the request. It also has a tag with a random string added to the URI by the UAC to identify the session.

• **Call-ID:**
  contains a globally unique identifier for this call.

• **CSeq or Command Sequence:**
  contains an integer and a method name. Used to distinguish a retransmission from a new request.

• **Contact:**
  contains a SIP URI for direct communications between UAs.

• **Content-Type:**
  indicates the type of the message body.

• **Content-Length:**
  gives the length in octets of the message body.
SIP Response Types

• **Provisional (1xx):**
  Request received and being processed.

• **Success (2xx):**
  The action was successfully received, understood, and accepted.

• **Redirection (3xx):**
  Further action needs to be taken in order to complete the request.

• **Client Error (4xx):**
  The request contains bad syntax or cannot be fulfilled at this server.

• **Server Error (5xx):**
  The server failed to fulfill an apparently valid request.

• **Global Failure (6xx):**
  The request cannot be fulfilled at any server.
SIP Response Message Example

(Message (11) in Successful Call Setup Example)

SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
Via: SIP/2.0/UDP 12.26.17.91:5060
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@12.26.17.91
CSeq: 314159 INVITE
Contact: <sip:bob@biloxi.com>
Content-Type: application/sdp
Content-Length: 131
Session Description Protocol (SDP)

The Session Description Protocol (SDP) describes the content of sessions, including telephony, Internet radio, and multimedia applications. SDP includes information about the following:

- **Media streams:**
  A session can include multiple streams of differing content. SDP currently defines audio, video, data, control, and application as stream types, similar to the MIME types used for Internet mail.

- **Addresses:**
  Indicates the destination addresses, which may be a multicast address, for a media stream.

- **Ports:**
  For each stream, the UDP port numbers for sending and receiving are specified.

- **Payload types:**
  For each media stream type in use (e.g., telephony), the payload type indicates the media formats that can be used during the session.

- **Start and stop times:**
  These apply to broadcast sessions, like a television or radio program. The start, stop, and repeat times of the session are indicated.

- **Originator:**
  For broadcast sessions, the originator is specified, with contact information. This may be useful if a receiver encounters technical difficulties.
Session Description

- The information is described in each SIP message body using a textual format (<type> = <value>).

- SDP Session Descriptor Types:
  
  v =  (protocol version)
  o =  (owner/creator and session identifier).
  s =  (session name)
  i = * (session information)
  u = * (URI of description)
  e = * (email address)
  p = * (phone number)
  c = * (connection information - not required if included in all media)
  b = * (bandwidth information)
  One or more time descriptions (see below)
  z = * (time zone adjustments)
  k = * (encryption key)
  a = * (zero or more session attribute lines)
  Zero or more media descriptions (see below)

Time description
  t =  (time the session is active)
  r = * (zero or more repeat times)

Media description
  m =  (media name and transport address)
  i = * (media title)
  c = * (connection information - optional if included at session-level)
  b = * (bandwidth information)
  k = * (encryption key)
  a = * (zero or more media attribute lines)
SIP Call Set-Up Procedure

- Example: called party present at given SIP address
  - Calling party: sip:tom.C@university.edu
  - Called party: sip:karen.S@company.com

  - Each access network has a **proxy server** to which all SIP INVITE messages are sent.

  - Each host knows the IP address of its local proxy server.

  - The proxy server knows the SIP name of each user currently logged in at the site and the IP address of the user's host device.

  - Upon receiving an INVITE request message from the calling party, the proxy server obtains the name of the called party proxy server (company.com) using DNS.

  - The calling proxy server then sends an INVITE message to the called proxy server.

  - The called proxy server determines that the called party is logged in and also the IP address of the called host.

  - If the called party can accept the call, an INVITE response message is returned.

  - On receipt of the response, the SIP in the calling host returns an ACK message.

  - The two users/hosts can now exchange information.
SIP Proxy Message Routing

(a)

Calling host

PS-A

PS-B

Called host

Access network (university.edu)

Global Internet

Access network (company.com)

SIP = session initiation protocol

PS = (SIP) proxy server

(b)

Access network B

(company.com)

Access network A

Access network C

Called host

PS-B

RS-B

Calling host

PS-A

Global Internet

[university.edu]

(organization.org)

RS = redirect server

SIP message routing examples: (a) direct using proxy servers; (b) indirect using a redirect server.
SIP Call Set-Up Procedure cont.

- Example: called party not present at given SIP address

  - Calling party: sip:tom.C@university.edu
  - Called party: sip:karen.S@company.com

  - Called party currently at: sip:karen.S@organization.org

  - On receipt of the INVITE request from the calling proxy server, the called proxy server determines that karen.S is not currently logged in at this location.

  - The INVITE request is forwarded to the redirect server.

  - The redirect server has the list of alternate SIP addresses for the called party. This list is returned to the calling proxy server (using the Contact: field of the INVITE response message).

  - The calling proxy server then proceeds as before using an alternative SIP address for the called party.
User B initiates a new SIP session with the SIP Server (i.e. the user "logs on to" the SIP server).

- User B sends a SIP REGISTER request to the SIP server. The request includes the user’s contact list.
- The SIP server provides a challenge to User B.
- User B enters her/his valid user ID and password.
- The SIP server validates the user's credentials.
- The SIP server registers the user in its contact database and returns a response (200 OK) to User B's SIP client. The response includes the user's current contact list in Contact headers.
SIP Registration: Message Details

F1 REGISTER B -> SIP Server

REGISTER sip:ss2.wcom.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 123456789@there.com
CSeq: 1 REGISTER
Contact: <sip:UserB@110.111.112.113>
Contact: <sip: +1-972-555-2222@gw1.wcom.com;user=phone>
Contact: tel:+1-972-555-2222
Content-Length: 0

F2 401 Unauthorized SIP Server -> User B

SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 123456789@there.com
CSeq: 1 REGISTER
WWW-Authenticate: Digest realm="MCI WorldCom SIP",
    domain="sip:ss2.wcom.com",
    nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="",
    stale="FALSE", algorithm="MD5"
Content-Length: 0
SIP Registration: Message Details cont.

F3 REGISTER B -> SIP Server

REGISTER sip:ss2.wcom.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 123456789@there.com
CSeq: 2 REGISTER
Contact: <sip:UserB@110.111.112.113>
Contact: <sip:+1-972-555-2222@gw1.wcom.com;user=phone>
Contact: tel:+1-972-555-2222
Authorization:Digest username="UserB",
   realm="MCI WorldComSIP",
   nonce="ea9c8e88df84f1cec4341ae6cbe5a359",
   opaque="", uri="sip:ss2.wcom.com",
   response="dfe56131d1958046689cd83306477ecc"
Content-Length: 0

F4 200 OK SIP Server -> B

SIP/2.0 200 OK
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 123456789@there.com
CSeq: 2 REGISTER
Contact: <sip:UserB@110.111.112.113>;expires=3600
Contact: <sip:+1-972-555-2222@gw1.wcom.com;user=phone>; expires=3600
Contact: tel:+1-972-555-2222;expires=3600
Content-Length: 0
Example: Simple SIP to SIP Call

<table>
<thead>
<tr>
<th>User A</th>
<th>User B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>INVITE F1 &gt; (100 Trying) F2</td>
<td></td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
</tr>
<tr>
<td>180 Ringing F3</td>
<td></td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
</tr>
<tr>
<td>200 OK F4</td>
<td></td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
</tr>
<tr>
<td>ACK F5</td>
<td></td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
</tr>
<tr>
<td>Two-Way RTP Media &lt; = = = = = = &gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
</tr>
<tr>
<td>BYE F6</td>
<td></td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
</tr>
<tr>
<td>200 OK F7</td>
<td></td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
</tr>
</tbody>
</table>

- In this scenario, User A completes a call to User B directly.
SIP-to-SIP Call: Message Details

F1 INVITE User A -> User B

INVITE sip:UserB@there.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345601@here.com
CSeq: 1 INVITE
Contact: <sip:UserA@100.101.102.103>
Content-Type: application/sdp
Content-Length: 147

v = 0
o = UserA 2890844526 2890844526 IN IP4 here.com
s = Session SDP
c = IN IP4 100.101.102.103
t = 0 0
m = audio 49172 RTP/AVP 0
a = rtpmap:0 PCMU/8000

F2 (100 Trying) User B -> User A

SIP/2.0 100 Trying
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345601@here.com
CSeq: 1 INVITE
Content-Length: 0
SIP-to-SIP Call: Message Details cont.

F3 180 Ringing User B -> User A

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>;tag=8321234356
Call-ID: 12345601@here.com
CSeq: 1 INVITE
Content-Length: 0

F4 200 OK User B -> User A

SIP/2.0 200 OK
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>;tag=832123456
Call-ID: 12345601@here.com
CSeq: 1 INVITE
Contact: <sip:UserB@110.111.112.113>
Content-Type: application/sdp
Content-Length: 147

v = 0
o = UserB 2890844527 2890844527 IN IP4 there.com
s = Session SDP
c = IN IP4 110.111.112.113
t = 0 0
m = audio 3456 RTP/AVP 0
a = rtpmap:0 PCMU/8000
SIP-to-SIP Call: Message Details cont.

F5 ACK User A -> User B

ACK sip:UserB@there.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>;tag=8321234356
Call-ID: 12345601@here.com
CSeq: 1 ACK
Content-Length: 0

/* RTP streams are established between A and B */

/* User B Hangs Up with User A. */

F6 BYE User B -> User A

BYE sip: UserA@here.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>;tag=8321234356
To: BigGuy <sip:UserA@here.com>
Call-ID: 12345601@here.com
CSeq: 1 BYE
Content-Length: 0

F7 200 OK User A -> User B

SIP/2.0 200 OK
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>;tag=8321234356
To: BigGuy <sip:UserA@here.com>
Call-ID: 12345601@here.com
CSeq: 1 BYE
Content-Length: 0
Example: SIP to SIP Call through 2 Proxies

- User A completes a call to User B using two proxies.
- The initial INVITE (F1) does not contain the Authorization credentials Proxy 1 requires, so a 407 Proxy Authorization response is sent containing the challenge information.
- A new INVITE (F4) is then sent containing the correct credentials and the call proceeds.
- The call terminates when User B disconnects by initiating a BYE message.
### Example: Unsuccessful SIP to SIP Call - No Answer

<table>
<thead>
<tr>
<th>User A</th>
<th>Proxy 1</th>
<th>Proxy 2</th>
<th>User B</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE F1</td>
<td>INVITE F2</td>
<td>INVITE F4</td>
<td></td>
</tr>
<tr>
<td>(100) F3</td>
<td>(100) F5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>180 F8</td>
<td>180 F7</td>
<td></td>
<td>180 F6</td>
</tr>
<tr>
<td>CANCEL F9</td>
<td>CANCEL F11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200 F10</td>
<td>200 F12</td>
<td></td>
<td>200 F14</td>
</tr>
<tr>
<td>CANCEL F13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>487 F17</td>
<td>ACK F18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>487 F19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACK F20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- In this scenario, User A gives up on the call before User B answers (sends a 200 OK response).
- User A sends a CANCEL (F9) since no final response had been received from User B.
- If a 200 OK to the INVITE had crossed with the CANCEL, User A would have sent an ACK then a BYE to User B in order to properly terminate the call.
- Note that the CANCEL message is acknowledged with a 200 OK on a hop by hop basis.
Example: Unsuccessful SIP to SIP Call - Busy

- In this scenario, User B is busy and sends a 486 Busy Here response to User A's INVITE.

- Note that the 4xx response is ACKed at each signaling leg.
Example: Unsuccessful SIP to SIP Call - No Response

- In this example, there is no response from User B to User A's INVITE messages being re-transmitted by Proxy 2.

- After the sixth retransmission, Proxy 2 gives up and sends a CANCEL to User B and a 480 No Response to User A.

- Note that the CANCEL would also be retransmitted six times, as governed by SIP timer T1.
Interworking between IP and PSTN/ISDN Hosts

- Location Servers (LS) contain a database of the IP address of the LS that should be used to reach all of the gateways in other regional/national networks and also the PSTN/ISDN codes associated with each of these gateways.

- The Gateway Location Protocol (GLP) is used to construct and update this database.
User A dials the globalized E.164 number +1-972-555-2222 to reach User B. It is assumed that the SIP User Agent Client converts the digits into a global number and puts them into a SIP URL.

- Network Gateway (NGW 1) is the interface to User B.
- In this scenario, User B answers the call then User A disconnects the call.
- Signaling between NGW 1 and User B's telephone switch is ANSI ISUP.
Example: ISUP PSTN to SIP Call

- User A from the PSTN calls User B through a Network Gateway NGW1 and Proxy Server Proxy 1.
- When User B answers the call, the media path is setup end-to-end.
- Call terminates when User A hangs up the call, with User A's telephone switch sending an ISUP RELease message which is mapped to a BYE by NGW 1.
### Example: ISUP PSTN to ISUP PSTN via SIP Call

<table>
<thead>
<tr>
<th>User A</th>
<th>NGW 1</th>
<th>Proxy 1</th>
<th>GW 2</th>
<th>User C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAM F1</td>
<td>INVITE F2</td>
<td>INVITE F3</td>
<td>IAM F4</td>
<td>ACM F5</td>
</tr>
<tr>
<td>ACM F8</td>
<td>183 F7</td>
<td>183 F6</td>
<td>ACM F8</td>
<td></td>
</tr>
</tbody>
</table>

One Way Voice

<table>
<thead>
<tr>
<th>One Way Voice</th>
<th>One Way RTP Media</th>
<th>One Way Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM F8</td>
<td>183 F7</td>
<td>ACM F8</td>
</tr>
<tr>
<td>183 - Session Progress message</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Both Way Voice

<table>
<thead>
<tr>
<th>Both Way Voice</th>
<th>Both Way RTP Media</th>
<th>Both Way Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANM F12</td>
<td>200 F11</td>
<td>ANM F9</td>
</tr>
<tr>
<td>200 F10</td>
<td>ACK F13</td>
<td>Both Way Voice</td>
</tr>
<tr>
<td>ACK F14</td>
<td></td>
<td>REL F15</td>
</tr>
<tr>
<td>200 F19</td>
<td></td>
<td>RLC F17</td>
</tr>
</tbody>
</table>

Both Way Voice

<table>
<thead>
<tr>
<th>Both Way Voice</th>
<th>Both Way RTP Media</th>
<th>Both Way Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANM F12</td>
<td>200 F11</td>
<td>ANM F9</td>
</tr>
<tr>
<td>200 F10</td>
<td>ACK F13</td>
<td>Both Way Voice</td>
</tr>
<tr>
<td>ACK F14</td>
<td></td>
<td>REL F15</td>
</tr>
<tr>
<td>200 F19</td>
<td></td>
<td>RLC F17</td>
</tr>
</tbody>
</table>

Both Way Voice

<table>
<thead>
<tr>
<th>Both Way Voice</th>
<th>Both Way RTP Media</th>
<th>Both Way Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANM F12</td>
<td>200 F11</td>
<td>ANM F9</td>
</tr>
<tr>
<td>200 F10</td>
<td>ACK F13</td>
<td>Both Way Voice</td>
</tr>
<tr>
<td>ACK F14</td>
<td></td>
<td>REL F15</td>
</tr>
<tr>
<td>200 F19</td>
<td></td>
<td>RLC F17</td>
</tr>
</tbody>
</table>

Both Way Voice

<table>
<thead>
<tr>
<th>Both Way Voice</th>
<th>Both Way RTP Media</th>
<th>Both Way Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANM F12</td>
<td>200 F11</td>
<td>ANM F9</td>
</tr>
<tr>
<td>200 F10</td>
<td>ACK F13</td>
<td>Both Way Voice</td>
</tr>
<tr>
<td>ACK F14</td>
<td></td>
<td>REL F15</td>
</tr>
<tr>
<td>200 F19</td>
<td></td>
<td>RLC F17</td>
</tr>
</tbody>
</table>

Both Way Voice

<table>
<thead>
<tr>
<th>Both Way Voice</th>
<th>Both Way RTP Media</th>
<th>Both Way Voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANM F12</td>
<td>200 F11</td>
<td>ANM F9</td>
</tr>
<tr>
<td>200 F10</td>
<td>ACK F13</td>
<td>Both Way Voice</td>
</tr>
<tr>
<td>ACK F14</td>
<td></td>
<td>REL F15</td>
</tr>
<tr>
<td>200 F19</td>
<td></td>
<td>RLC F17</td>
</tr>
</tbody>
</table>
Example: ISUP PSTN to ISUP PSTN via SIP Call cont.

- In this scenario, User A in the PSTN calls User C who is an extension on a PBX.

- User A's telephone switch signals via SS7 to the Network Gateway NGW 1, while User C's PBX signals via SS7 with the Enterprise Gateway GW 2.

- The CdPN and CgPN are mapped into SIP URLs and placed in the To and From headers.

- Proxy 1 looks up the dialed digits in the Request-URI and maps the digits to the PBX extension of User C served by GW 2.

- The Request-URI in F3 uses the phone-context tag to identify what private dialing plan is being referenced.

- The INVITE is then forwarded to GW 2 for call completion.

- An early media path is established end-to-end so that User A can hear the ringing tone generated by PBX C.

- User C answers the call and the media path is cut through in both directions.

- User C hangs up terminating the call.