A Guide to
Session Initiation Protocol (SIP)

November 2010

Tom Jenkins
Senior Director, Xpress™ Products
# Table of Contents

- History ................................................................................................. 3
- What does SIP do? ............................................................................... 5
- SIP Network ......................................................................................... 6
- SIP Network Elements ....................................................................... 6
- SIP Messages ......................................................................................... 8
  - Methods .............................................................................................. 8
  - Responses ......................................................................................... 8
  - SIP Addressing ................................................................................. 9
  - SIP Message Structure .................................................................... 9
- Request Message Format ..................................................................... 10
- Request Line ......................................................................................... 10
- Header Section ..................................................................................... 10
- Message Body ...................................................................................... 10
- Response Message Format ................................................................. 11
  - Status Line ........................................................................................ 12
  - Header Section ................................................................................ 12
  - Message Body .................................................................................. 12
- Call Flow Example ............................................................................... 13
- Call Flow ............................................................................................... 14
- SIP Request/Methods ....................................................................... 16
- SIP Responses .................................................................................... 17
- SIP Headers .......................................................................................... 21
- Session Description Protocol (SDP) .................................................... 24
- SIP Registration Process ................................................................. 25
- SIP Call flow (Detailed Example) ....................................................... 29
- Conclusion ........................................................................................... 40
- Reference List ..................................................................................... 41
- About the Author ............................................................................... 44
- About PT ............................................................................................... 45
History
Session Initiation Protocol (SIP) was originally developed by Internet Engineering Task Force (IETF) Multi-Party Multimedia Session Control Working Group (MMUSIC) in 1997 and released as version 1. Significant changes were made and the version was changed to version 2 and published as Request For Comments 2543 (RFC 2543) in April of 1999. Due to the increased interest in SIP, the SIP working group within the IETF was formed in September of 1999. Updates and bug fixes were developed and published as RFC 2543 “bis.” RFC 2543 and the associated RFC 2543 “bis.” were replaced with the baseline SIP specification in use today – RFC 3261 published in June of 2002.

The ever increasing interest in the SIP Protocol drove the IETF to form several new working groups to handle the increased workload and take responsibility for specialized facets of SIP. The Working Group Table lists the working groups and their responsibility within the broader SIP environment.

*Oh, enough about history, if you would like to learn more, please reference the books listed at the end of this tutorial. Now, to get started with the information you actually wanted from this tutorial.*
<table>
<thead>
<tr>
<th>Working Group Acronym</th>
<th>Working Group Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCON</td>
<td>Centralized Conferencing</td>
</tr>
<tr>
<td>DRINKS</td>
<td>Data for Reachability of Inter/tra-Network SIP</td>
</tr>
<tr>
<td>BLISS</td>
<td>Basic Level of Interoperability for SIP Services</td>
</tr>
<tr>
<td>IMPP</td>
<td>Instant Messaging and Presence Protocol</td>
</tr>
<tr>
<td>XMPP</td>
<td>XML-based protocol for near real-time extensible messaging and presence</td>
</tr>
<tr>
<td>VPIM</td>
<td>Voice Profile for Internet Mail</td>
</tr>
<tr>
<td>AVT</td>
<td>Audio/Video Transport (RTP)</td>
</tr>
<tr>
<td>IPTEL</td>
<td>IP Telephony (CPL, GW location, TRIP)</td>
</tr>
<tr>
<td>MMUSIC</td>
<td>Multiparty Multimedia Session Control (SIP, SDP, conferencing)</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SIPPING</td>
<td>Session Initiation Proposal Investigation</td>
</tr>
<tr>
<td>ENUM</td>
<td>Telephone Number Mapping</td>
</tr>
<tr>
<td>MEGACO</td>
<td>Media Gateway Control (IP telephony gateways)</td>
</tr>
<tr>
<td>PINT</td>
<td>PSTN and Internet Internetworking (mixed services)</td>
</tr>
<tr>
<td>SIGTRAN</td>
<td>Signaling Transport (PSTN signaling over IP)</td>
</tr>
<tr>
<td>SPIRITS</td>
<td>Service in the PSTN/IN Requesting Internet Service</td>
</tr>
<tr>
<td>MEDIACTRL</td>
<td>Media Server Control</td>
</tr>
<tr>
<td>SPEECHSC</td>
<td>Speech Services Control</td>
</tr>
<tr>
<td>GEOPRIV</td>
<td>Geographic Location/Privacy</td>
</tr>
<tr>
<td>ROHC</td>
<td>Robust Header Compression (SigComp)</td>
</tr>
<tr>
<td>MIDCOM</td>
<td>Middlebox Communication (NAT, IPV4-IPV6)</td>
</tr>
<tr>
<td>BEHAVE</td>
<td>Behavior Engineering for Hindrance Avoidance</td>
</tr>
<tr>
<td>DIFFSERV</td>
<td>Differentiated Services (QoS in backbone)</td>
</tr>
<tr>
<td>INTSERV</td>
<td>Integrated Services (end-to-end QoS)</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol Label Switching</td>
</tr>
<tr>
<td>RSVP</td>
<td>Resource Reservation Setup Protocol</td>
</tr>
<tr>
<td>P2PSIP</td>
<td>Peer-to-Peer Session Initiation Protocol</td>
</tr>
<tr>
<td>SPEERMINT</td>
<td>Session PEERing for Multimedia INTerconnect</td>
</tr>
<tr>
<td>MSEC</td>
<td>Multicast Security</td>
</tr>
<tr>
<td>ECRIT</td>
<td>Emergency Context Resolution with Internet Technologies (E911)</td>
</tr>
</tbody>
</table>

Table 1: Working Group Table
What Does SIP Do?
Briefly stated SIP is a text based protocol responsible for the establishment, management and tearing down of media sessions in an Internet Protocol (IP) environment. Unlike other such protocols (ISDN, SS7 or H323) SIP only performs these functions and relies heavily on other protocols to describe the media sessions, transport the media, provide quality of service (QOS) or, in fact, transport the SIP protocol itself. SIP and its relationship to other protocols, is shown in Figure 1.

![Figure 1: Relationship to other Protocols](image)

**Acronyms**

- **DHCP**: Dynamic Host Control Protocol
- **DNS**: Domain Name System
- **IP**: Internet Protocol
- **PPP**: Point to Point Protocol
- **RTP**: Real Time Protocol
- **RTSP**: Real Time Streaming Protocol
- **SCTP**: Stream Control Transmission Protocol
- **SDP**: Session Description Protocol
- **SIP**: Session Initiation Protocol
- **TCP**: Transmission Control Protocol
- **UDP**: User Datagram Protocol
SIP Network
SIP, its associated transport protocols, and the media sessions it controls are carried over an IP network. There are specialized devices in a SIP network used to perform media establishment, routing, addressing, number translation, etc. The SIP network including SIP network elements is shown in Figure 2.

Figure 2: IP Network

SIP Network Elements
User Agents
SIP uses 2 broad categories of components – User Agents and Servers to provide the necessary functionality within the network.

User Agents – are entities within the SIP network, in a client server mode, that act on the behalf of users.
These entities must:

- Be able to establish media sessions
- Maintain call states of calls it initiates or participates in
- Support TCP and UDP
- Support SDP

Since UAs operate in a client / server mode there are User Agent Servers (UAS) and User Agent Clients (UAC).

UAS generate request and send them to servers – typically Proxies or Registrars.

UAC receive request, operate on them and send responses.

In SIP a UA can be either a UAS or UAC dependent on which UA initiated the request for the media session. It is also possible that a UA could be operating as both a UAS and UAC especially during a three-way call or call transfer.

**Servers**

The formal definition of a sip server is; “A server that uses SIP (Session Initiation Protocol) to manage real-time communication among SIP clients.” More succinctly, a SIP server makes up the core of a SIP network and contains a rules base for acting on request sent to it by UAs or other servers.

**Proxy Server** – receive SIP requests from a UA or another proxy, uses a database, location register or registrar to determine the routing of the SIP request to another UA or proxy. Typically, the proxy is not allowed to generate request, only forward them – the exception to this rule is the Cancel and ACK request which will be explained later in this tutorial. Proxy Servers can be either stateful or stateless. A stateful Proxy maintains call state or transaction information for the duration of the call or transaction; while a stateless proxy does not keep any state information.

**Redirect Server** – unlike a proxy Redirect Server receives a SIP request but rather than forwarding it – it responds to it – usually with a 3XX (redirection class) response which will be covered later in this tutorial. In general, a Redirect Server sends the 3XX response to a UA or proxy indicating the call or session initiation should be tried at a different location; the intended UA has moved either temporarily or permanently.
Registrar Server – as the name implies the Registrar Server (also known as Registration Servers) are used to gain information about location from a UA. In SIP, UAs are required to periodically register with the Registration Server. These servers usually require the UA be authenticated to circumvent calls from being hijacked by unauthorized users. Oh, what we have to go through to stop fraud!

Location Server – is used to store addresses obtained in the registration process.

It must also be noted that any or all of the servers described may either be deployed on a stand-alone basis or their functionalities may be combined into a physical entity.

Application Server – are specialized servers providing enhanced services in a SIP environment.

So now we have discussed the SIP network and the components that make up the network or I guess we have built a watch, when we only wanted to know, “what time it was.” Now we will move on to discuss the SIP protocol and more specifically the types of SIP messages.

SIP Messages
Since SIP was designed using a request/response model there are 2 types of SIP messages – request (also called Methods) and SIP responses.

Methods
As defined by RFC 3261, the baseline SIP specification, a request (Method) is “A SIP message sent from a client to a server, for the purpose of invoking a particular operation. There are 6 methods defined by RFC 3261 – Invite, Register, Bye, ACK, Cancel and Options other RFCs have defined the following methods: Refer, Subscribe, Notify, Publish, Message Update, Info and PRACK.

Each of these methods will be discussed just a bit later in this tutorial – be patient.

Responses
RFC 3261 defines a Response as, “A SIP message sent from a server to a client, to indicate the status of a request sent from the client to the server.” Responses are segmented into 6 classes indicating the general category of the response. The response classes are provided in the response table and will be discussed later.
<table>
<thead>
<tr>
<th>Response Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1XX</td>
<td>Informational</td>
</tr>
<tr>
<td>2XX</td>
<td>Success</td>
</tr>
<tr>
<td>3XX</td>
<td>Redirection</td>
</tr>
<tr>
<td>4XX</td>
<td>Client Error</td>
</tr>
<tr>
<td>5XX</td>
<td>Server Error</td>
</tr>
<tr>
<td>6XX</td>
<td>Global Error</td>
</tr>
</tbody>
</table>

Table 2: Response Table

SIP Addressing
SIP entities are identified by their SIP Uniform Resource Identifier (URI). This SIP URI can be thought of as the user’s Telephone number or address within the SIP network. These URIs are similar in format to email address such as:

Mail to: j.smith@nowhere.com.
Examples of SIP Addressing:
sip:jane.doe@192.168.1.102
sip:support@help.me.com
sip:22444032@mynumber.is.com

SIP Message Structure
As previously stated, SIP is a text-based protocol developed by the IETF. Some aspects of other text-based protocols were used in the design of SIP. The two protocols that come to mind are HyperText Transport Protocol (HTTP) – SIP used a similar addressing scheme. Simple Mail Transport Protocol (SMTP) – SIP uses the concept of headers some of which are identical – To:, From:, Subject: and Date:, just to mention a few.

Now on to our detailed discussion of Request Message format.
Request Message Format
Refer to Request Message Table for the following discussion of the format of SIP request messages. Request messages consist of three major sections the Request line, the Header section and the Message body.

Request Line
The Request line is composed of the request type, the SIP URI of the destination or next hop, and the version of SIP being used. From the example in Table 3, it can be seen that the request type is an INVITE request, with the SIP URI addressed to sip:9103682854@192.168.16.140 and the SIP version is 2.0 which is the current and only version at this time.

Header Section
Next is the header section containing multiple headers; each carrying its own well defined information. Each Header is terminated by a carriage return, line feed (<cr><lf>) at the end the header. A description of the major headers can be found in the header section of this tutorial.

Message Body
The final section of the Request message – Message Body – is optional depending on the type of message and where it falls within the establishment process. The boundary between the Header section and the message body is defined by a blank line. The message shown in Table 3 contains a message body of Session Description Protocol. This SDP message body is 322 characters long. This information is listed in the Content-Type header field (application/sdp) and the Content Length header field (322). The SDP message body contained in this INVITE message defines the media information necessary for this session. SDP will be discussed later within this tutorial.
Response Message Format
Refer to Table 4 for the following discussion of the format of SIP response messages. Response messages consist of three major sections the Status line, the Header section and the Message body. In some cases the Response message is quite a bit larger than the Request message because it carries a reason header indicating why the response is being sent.

Table 3: Request Message Table
Status Line
The Status Line is comprised of three elements; the protocol version, the status code and the reason phrase. Only the first two elements of the status line are processed by any SIP network element. The reason phrase makes it easy for us humans who may be examining the message. In the case of the example in Table 4 the protocol version is 2.0 – again the current and only version of SIP. The status code is 100, indicating that this response is part of the 1XX response class (Informational) and specifically that this is a 100 response. A 100 Response is used to indicate to the receiver that another device has received the request and it may take a bit longer for this media session to be established. The final portion of the Status line is the reason phrase of “Trying.”

Header Section
Next is the header section containing multiple headers; each carrying its own well defined information. Each Header is terminated by a carriage return, line feed (<cr><lf>) at the end the header. A description of the major headers can be found in the header section of this tutorial.

Message Body
The final section of the Request message – Message Body – is optional depending on the type of message and where it falls within the establishment process. The boundary between the Header section and the message body is defined by a blank line. The message shown in Response Message Table contains no message body as indicated by the Content-Length Header, in the Header Section, being set to 0.

<table>
<thead>
<tr>
<th>Status Line</th>
<th>SIP/2.0 100 Trying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td>Via: SIP/2.0/UDP 192.168.16.140:5060;branch=z9hG4bK-85z25d58d7461b9</td>
</tr>
<tr>
<td></td>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a></td>
</tr>
<tr>
<td></td>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
</tr>
<tr>
<td></td>
<td>Call-ID: Yzk2ZjQ5YzJhOWVvNTJmNjk5MWXmYmXmMjINzQwOTg</td>
</tr>
<tr>
<td></td>
<td>CSeq: 2 INVITE</td>
</tr>
<tr>
<td></td>
<td>Content-Length: 0</td>
</tr>
<tr>
<td>Empty Line</td>
<td></td>
</tr>
<tr>
<td>Message Body</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Response Message Table
Call Flow Example

Before we get started with our discussion about the SIP message call flow, I would first like to explain a bit about the network and users depicted in Figure 3. This is a single domain call so there is only one Proxy involved and no DNS lookup is required for this call. John Smith has a telephone number of 9103683957 and the IP address of his User Agent is 192.168.16.105. The proxy (IP address 192.168.16.140) used in the scenario is a stateful proxy, so all SIP messages are forced through the proxy to facilitate its ability to keep track of all call states. Finally, the other subscriber is Jane Doe with a telephone number of 9103682854 and an IP address of 192.168.16.102, for her User Agent. In this discussion no detailed messages will be used – only a discussion of message flow and usage. This same scenario will be used after we have a detailed discussion on messages and headers – at that time we will look at the individual messages involved in this call.

Figure 3: Network and Users
Call Flow

Invite #1
John Smith wishes to call Jane Doe. He dials 9103682854 on his User Agent. The User Agent is configured to route all outgoing request through the proxy at IP address 192.168.16.140. John's User Agent formulates an Invite message with all pertinent information regarding this call and sends it to the proxy.

Invite #2
The proxy receives the Invite request and analyzes it to determine where the request should be sent. The proxy knows that Jane Doe is currently active at IP – 192.168.16.102. The proxy appends its own IP address to the request and sends to Jane Doe's User Agent.

100 Trying 3#
Jane Doe's User Agent receives and analyzes the Invite Request from the proxy. The User Agent sends a 100 Trying Response regarding this call to the proxy.

100 Trying #4
The proxy receives the 100 Trying from Jane Doe's User Agent and forwards it the John Smith's User Agent indicating that the call will take a little longer to set up. John Smith's User Agent receives the 100 Trying.

180 Ringing #5
Jane's User Agent starts alerting Jane of an incoming call. The user agent sends a 180 Ringing response to the proxy indicating the alerting has begun.

180 Ringing #6
The Proxy receives the 180 Ringing response and forwards it to John’s User Agent. John's User Agents receives the 180 Ringing Response and uses it to trigger the Ringback tone from User Agent to John's handset.

200 OK #7
Jane and her associated User Agent have decided that the call from John should be accepted (Call Answered). A 200 OK response is sent to the proxy. The 200 OK contains an SDP Message Body listing the media parameters required by Jane's User Agent for establishing the media session.
200 OK #8
The 200 Ok responses is received by the proxy and forwarded to John's User Agent. John's User Agent receives the 200 OK response and starts preparing the ports, codexes and other capabilities for the media session.

ACK #9
John's User Agent sends a User Agent an ACK to the proxy regarding this call. As the ACK is sent the media session is established using the RTP Protocol and is addressed directly from John's User Agent to Jane’s User Agent.

ACK #10
The Proxy receives the ACK from John’s User Agent and forwards it to Jane's User Agent. Jane’s User Agent receives the ACK and insures the Media Session is established according to the SDP received in the Invite Request. The media from Jane’s User Agent toward John’s User Agent uses RTP and is addressed directly to John’s User Agent.

BYE #11
After the conversation has been completed, John hangs up the phone – his User Agent recognizes this condition and sends a Bye Request to the proxy regarding this call – tearing down the media session.

BYE #12
The proxy receives the Bye from John's User Agent and forwards it to Jane’s User Agent. Jane's User Agent receives the Bye and tears down the media session.

200 OK #13
Jane’s User Agent sends a 200 OK to the proxy confirming the Bye and the disconnection of the media session. The proxy receives the 200 OK and forwards it to John’s User Agent.

200 OK #14
John’s User Agent receives the 200 OK from the proxy and releases any resources associated with the call.
I hope you got that as I am not sure I could go through that again – oops, must do it again with the actual messages! This should give you a good basis for our upcoming discussion on Request, Responses and Headers.

**SIP Request/Methods**

As you remember *(I hope, I hope)*, from our earlier discussion of SIP request, a request (Method) is: “A SIP message sent from a client to a server,” for the purpose of invoking a particular operation. There were the original 6 request defined by RFC 3261– Invite, Register, Bye, ACK, Cancel and Options with the additional ones (Refer, Subscribe, Notify, Publish, Message Update, Info and PRACK) being defined in other RFCs.

**ACK (RFC 3261)**
The ACK method is used to acknowledge the final response to an INVITE method. Final responses are response classes 2xx, 3xx, 4xx, and 5xx.

**Bye (RFC 3261)**
The Bye method is used to terminate an established media session.

**Cancel (RFC 3261)**
The Cancel method is used to terminate a session before the session is established.

**Info (RFC 3261)**
The INFO method is used to carry call signaling information from a user agent to another user agent, with which it has an established media session.

**Invite (RFC 3261)**
The Invite method is used by a user agent to request the establishment of a session to another user agent.

**Message (RFC 3428)**
The Message method is used to transfer Instant Message information in SIP.

**Notify (RFC 3265)**
The Notify method is used to provide the updated event and status information requested in the Subscribe Method.
Options (RFC 3261)
The Options method is used to query a user agent or server about its current capabilities and availability.

PRACK (RFC 3262)
The PRACK method is used to acknowledge reliability transported provisional responses (1XX Class responses).

Publish (RFC 3903)
The Publish method is used by a user agent to send/publish event state information to a SIP Server known as an Event State Compositor (ESC).

Refer (RFC 3515)
The Refer method is used by a user agent to request another user agent access a particular SIP URI. This method is also used to perform a call transfer.

Register (RFC 3261)
The Register method is used by a SIP User Agent to notify the SIP Network of its current Contact URI (IP Address) and the URI that should have request routed to this Contact.

Subscribe (RFC 3265)
The Subscribe method is used to request event and status updates from a remote device.

Update (RFC 3311)
The Update method is used to modify the state of a session without changing the state of the existing dialog.

SIP Responses
SIP responses are generated by a UAS in response to a request sent by a UAC. These responses are categorized by number from the 100s to the 600s and also contain a reason phrase. The integer numbers are arranged in classes of 1XX, 2XX, 3XX, 4XX, 5XX, and 6XX. Each class of responses pertains to a particular set of conditions within the SIP network. Individual SIP entities only process or understand the specific response numbers – the reason phrase is meant to make this process more understandable for us humans. The 1XX responses are considered provisional
while all other classes are final responses. Information on each individual response can be found in the reference list provided at the end of this tutorial. A list of the response classes is provided in the Response Class Table.

<table>
<thead>
<tr>
<th>Response Class</th>
<th>Response Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1XX</td>
<td>Provisional</td>
</tr>
<tr>
<td>2XX</td>
<td>Success</td>
</tr>
<tr>
<td>3XX</td>
<td>Redirect</td>
</tr>
<tr>
<td>4XX</td>
<td>Client Error</td>
</tr>
<tr>
<td>5XX</td>
<td>Server Error</td>
</tr>
<tr>
<td>6XX</td>
<td>Global Error</td>
</tr>
</tbody>
</table>

**Table 5: Response Class Table**

1XX Response Class (Request received and being processed)
The 1XX class of responses is used to indicate call processing and more specifically the stage of call processing. The initial 1XX received, by a UAC, is understood by to mean that the Invite has been received by the UAS and stops any resending of the Invite by the UAC. The individual 1XX responses are listed in the 1XX Response Table.

<table>
<thead>
<tr>
<th>Response Class</th>
<th>Reason Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Trying</td>
</tr>
<tr>
<td>180</td>
<td>Ringing</td>
</tr>
<tr>
<td>181</td>
<td>Call is Being Forwarded</td>
</tr>
<tr>
<td>182</td>
<td>Call Queued</td>
</tr>
<tr>
<td>183</td>
<td>Session Progress</td>
</tr>
</tbody>
</table>

**Table 6: 1XX Response Table**

2XX Response Class (The action was successfully received, understood, and accepted)
The 2XX class of responses is used to indicate that UAS received, accepted and processed the request sent by the UAC.
Redirection (3xx): Further action needs to be taken (typically by sender) to complete the request. The 3XX class of responses is used to redirect to caller to another location due to the called having moved either temporarily or permanently. The redirection response is usually sent by proxy servers. These responses provide the address of other proxy server or the current address of the called party.

<table>
<thead>
<tr>
<th>Response Class</th>
<th>Reason Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Trying</td>
</tr>
<tr>
<td>180</td>
<td>Ringing</td>
</tr>
<tr>
<td>181</td>
<td>Call is Being Forwarded</td>
</tr>
<tr>
<td>182</td>
<td>Call Queued</td>
</tr>
<tr>
<td>183</td>
<td>Session Progress</td>
</tr>
</tbody>
</table>

Table 8: 3XX Response Table

Client Error (4xx): The request contains bad syntax or cannot be fulfilled at the server. The 4XX class responses are used by servers or UAS to indicate that the request cannot fulfilled. The headers contained in the 4XX class responses indicate to the UAC the nature of the error and how the request can be reformulated.
<table>
<thead>
<tr>
<th>Response Code</th>
<th>Reason Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Bad Request</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
</tr>
<tr>
<td>402</td>
<td>Payment Required</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
</tr>
<tr>
<td>404</td>
<td>Not Found</td>
</tr>
<tr>
<td>405</td>
<td>Method Not Allowed</td>
</tr>
<tr>
<td>406</td>
<td>Not Acceptable</td>
</tr>
<tr>
<td>407</td>
<td>Proxy Authentication Required</td>
</tr>
<tr>
<td>408</td>
<td>Request Timeout</td>
</tr>
<tr>
<td>409</td>
<td>Conflict</td>
</tr>
<tr>
<td>410</td>
<td>Gone</td>
</tr>
<tr>
<td>411</td>
<td>Length Required</td>
</tr>
<tr>
<td>412</td>
<td>Conditional Request Failed</td>
</tr>
<tr>
<td>413</td>
<td>Request Entity Too Large</td>
</tr>
<tr>
<td>414</td>
<td>Request-URI Too Long</td>
</tr>
<tr>
<td>415</td>
<td>Unsupported Media Type</td>
</tr>
<tr>
<td>416</td>
<td>Unsupported URI Scheme</td>
</tr>
<tr>
<td>417</td>
<td>Unknown Resource Priority</td>
</tr>
<tr>
<td>420</td>
<td>Bad Extension</td>
</tr>
<tr>
<td>421</td>
<td>Extension Required</td>
</tr>
<tr>
<td>422</td>
<td>Session Interval Too Short</td>
</tr>
<tr>
<td>423</td>
<td>Interval Too Brief</td>
</tr>
<tr>
<td>428</td>
<td>Use Identity Header</td>
</tr>
<tr>
<td>429</td>
<td>Provide Referrer Identity</td>
</tr>
<tr>
<td>430</td>
<td>Flow Failed</td>
</tr>
<tr>
<td>433</td>
<td>Anonymity Disallowed</td>
</tr>
<tr>
<td>436</td>
<td>Bad Identity-Info Header</td>
</tr>
<tr>
<td>437</td>
<td>Unsupported Certificate</td>
</tr>
<tr>
<td>438</td>
<td>Invalid Identity Header</td>
</tr>
<tr>
<td>439</td>
<td>First Hop Lacks Outbound Support</td>
</tr>
<tr>
<td>440</td>
<td>Max-Breadth Exceeded</td>
</tr>
<tr>
<td>470</td>
<td>Consent Needed</td>
</tr>
<tr>
<td>480</td>
<td>Temporarily Unavailable</td>
</tr>
<tr>
<td>481</td>
<td>Dialog/Transaction Does Not Exist</td>
</tr>
<tr>
<td>482</td>
<td>Loop Detected</td>
</tr>
<tr>
<td>483</td>
<td>Too Many Hops</td>
</tr>
<tr>
<td>484</td>
<td>Address Incomplete</td>
</tr>
<tr>
<td>485</td>
<td>Ambiguous</td>
</tr>
<tr>
<td>486</td>
<td>Busy Here</td>
</tr>
<tr>
<td>487</td>
<td>Request Terminated</td>
</tr>
<tr>
<td>488</td>
<td>Not Acceptable Here</td>
</tr>
<tr>
<td>489</td>
<td>Bad Event</td>
</tr>
<tr>
<td>491</td>
<td>Request Pending</td>
</tr>
<tr>
<td>493</td>
<td>Request Undecipherable</td>
</tr>
<tr>
<td>494</td>
<td>Security Agreement Required</td>
</tr>
</tbody>
</table>

**Table 9:** 4XX Response Table
Server Error (5xx): The server failed to fulfill an apparently valid request.
The 5XX response class is used to indicate that the request cannot be fulfilled due to a server error. These responses may contain a Retry-After: header indicating that the request may be retried after the specified amount of time.

<table>
<thead>
<tr>
<th>Response Class</th>
<th>Reason Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>Server Internal Error</td>
</tr>
<tr>
<td>501</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>502</td>
<td>Bad Gateway</td>
</tr>
<tr>
<td>503</td>
<td>Service Unavailable</td>
</tr>
<tr>
<td>504</td>
<td>Gateway Timeout</td>
</tr>
<tr>
<td>505</td>
<td>Version Not Supported</td>
</tr>
<tr>
<td>513</td>
<td>Message Too Large</td>
</tr>
<tr>
<td>580</td>
<td>Preconditions Failure</td>
</tr>
</tbody>
</table>

**Table 10: 5XX Response Table**

Global Failure (6xx): The request cannot be fulfilled at any server.
6XX response is sent by a server who has definitive information about the request URI that indicated that the request cannot be completed. These responses may contain a Retry-After: header indicating that the request may be retried after the specified amount of time.

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Reason Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>Busy Everywhere</td>
</tr>
<tr>
<td>603</td>
<td>Decline</td>
</tr>
<tr>
<td>604</td>
<td>Does Not Exist Anywhere</td>
</tr>
<tr>
<td>606</td>
<td>Not Acceptable</td>
</tr>
</tbody>
</table>

**Table 11: 6XX Response Table**

**SIP Headers**
SIP headers are used to convey attributes associated with the specific message types. These headers follow a format similar to Hyper Text Transport Protocol (HTTP) and contain the header type or name and then the header value. A header can span multiple lines and some headers may be seen multiple times within the same message. Headers can be seen in the Message Header Table and will be discussed following the table.
### Table 12: Message Header Table

**Via:**
The Via: header contains the logical address of the originator of the request or the address of the device that is expecting responses to the request.

**Max-Forwards:**
The Max-Forwards: header is used to limit the amount of intermediate nodes or devices (Hops) a message goes through while maintaining its validity.

<table>
<thead>
<tr>
<th>Request Line</th>
<th>INVITE sip:9103682854@192.168.16.140 SIP/2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z</td>
</tr>
<tr>
<td></td>
<td>Instant Messaging and Presence ProtocolMax-Forwards: 70</td>
</tr>
<tr>
<td></td>
<td>Contact: &lt;sip:9103683957@192.168.16.105:44646</td>
</tr>
<tr>
<td></td>
<td>To: &quot;Jane Doe&quot;&lt;sip:9103682854@192.168.16.140</td>
</tr>
<tr>
<td></td>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
</tr>
<tr>
<td></td>
<td>Call-ID: Yzk2ZjQ5YzJhOWVvNTJmNjk5MWMxYmMxMmJiNzQwOTg</td>
</tr>
<tr>
<td></td>
<td>CSeq: 2 INVITE</td>
</tr>
<tr>
<td></td>
<td>Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO</td>
</tr>
<tr>
<td></td>
<td>Record-Route: <a href="">sip:192.168.16.140;lr</a></td>
</tr>
<tr>
<td></td>
<td>Content-Type: application/sdp</td>
</tr>
<tr>
<td></td>
<td>Content-Length: 322</td>
</tr>
<tr>
<td>Empty Line</td>
<td>v=0</td>
</tr>
<tr>
<td></td>
<td>o=- 3 2 IN IP4 192.168.16.105</td>
</tr>
<tr>
<td></td>
<td>s=CounterPath X-Lite 3.0</td>
</tr>
<tr>
<td></td>
<td>c=IN IP4 192.168.16.105</td>
</tr>
<tr>
<td></td>
<td>t=0 0</td>
</tr>
<tr>
<td>Message Body</td>
<td>m=audio 32854 RTP/AVP 107 0 8 101</td>
</tr>
<tr>
<td></td>
<td>a=alt:1 2 : oND7cBBb qhQukSDp 10.100.100.250 32854</td>
</tr>
<tr>
<td></td>
<td>a=alt:2 1 : napYhQOC H731+iWt 192.168.16.105 32854</td>
</tr>
<tr>
<td></td>
<td>a=fmtp:101 0-15</td>
</tr>
<tr>
<td></td>
<td>a=rtpmap:107 BV32/16000</td>
</tr>
<tr>
<td></td>
<td>a=rtpmap:101 telephone-event/8000</td>
</tr>
<tr>
<td></td>
<td>a=sendrecv</td>
</tr>
</tbody>
</table>
Contact:
The Contact: header contains the SIP address that is the direct address of the originator of the message.

To:
The To: header contains the display name and the address of the UA representing the called party.

From:
The From: header contains the display name and the address of the UA that sent the Invite.

Call-ID:
The Call-ID: header is mandatory in all requests and responses. It is used to identify messages relating to a call between 2 users.

CSeq:
The CSeq: header is comprised of a random number and the Method name. It is used to determine non-delivery of messages or out of order messages.

Allow:
The Allow: header is used to indicate all methods supported by the sender of the message.

Record-Route:
The Record-Route: header is used by a proxy to force all subsequent routing of messages within a session through the proxy. This header contains the SIP address of the proxy requiring message routing.

Content-Type:
The Content-Type: header indicates what type of information is being carried in the message body.

Content-Length:
The Content-Length: header indicates the length of the message body in octets.

There are many other headers in the SIP environment – information of these headers can be found in their associated RFCs at the IETF website.
Session Description Protocol (SDP)
The Session Description Protocol is carried in the message bodies of both SIP request and responses. Rather than being a protocol, SDP is more of a description methodology for describing the type of media required for a session. SDP does NOT deliver media – only describes its parameters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>Protocol Version Number</td>
<td>m</td>
</tr>
<tr>
<td>o</td>
<td>Owner/Creator and Session Identifier</td>
<td>m</td>
</tr>
<tr>
<td>s</td>
<td>Session Name</td>
<td>m</td>
</tr>
<tr>
<td>i</td>
<td>Session Information</td>
<td>o</td>
</tr>
<tr>
<td>u</td>
<td>Uniform Resource Identifier</td>
<td>o</td>
</tr>
<tr>
<td>e</td>
<td>Email Address</td>
<td>o</td>
</tr>
<tr>
<td>p</td>
<td>Phone Number</td>
<td>o</td>
</tr>
<tr>
<td>c</td>
<td>Connection Information</td>
<td>m</td>
</tr>
<tr>
<td>b</td>
<td>Bandwidth Information</td>
<td>o</td>
</tr>
<tr>
<td>t</td>
<td>Time Sessions Starts and Stops</td>
<td>m</td>
</tr>
<tr>
<td>r</td>
<td>Repeat Times</td>
<td>o</td>
</tr>
<tr>
<td>z</td>
<td>Timezone Corrections</td>
<td>o</td>
</tr>
<tr>
<td>k</td>
<td>Encryption Key</td>
<td>o</td>
</tr>
<tr>
<td>a</td>
<td>Attribute Line</td>
<td>o</td>
</tr>
<tr>
<td>m</td>
<td>Media Information</td>
<td>o</td>
</tr>
<tr>
<td>a</td>
<td>Attribute Line</td>
<td>o</td>
</tr>
</tbody>
</table>

Table 13: SDP Field Table

Table 14: SDP Field Format Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>v=0</td>
</tr>
<tr>
<td>o</td>
<td>&lt;username&gt; &lt;session id&gt; &lt;version&gt; &lt;network type&gt; &lt;address type&gt; &lt;address&gt;</td>
</tr>
<tr>
<td>s</td>
<td>&lt;session name&gt;</td>
</tr>
<tr>
<td>i</td>
<td>&lt;textual description&gt;</td>
</tr>
<tr>
<td>u</td>
<td>&lt;uri&gt;</td>
</tr>
<tr>
<td>e</td>
<td>&lt;email-address&gt;</td>
</tr>
<tr>
<td>p</td>
<td>&lt;phone-number&gt;</td>
</tr>
<tr>
<td>c</td>
<td>&lt;network type&gt; &lt;address type&gt; &lt;connection address&gt;</td>
</tr>
<tr>
<td>b</td>
<td>&lt;bwtype&gt;:&lt;bandwidth&gt;</td>
</tr>
<tr>
<td>t</td>
<td>&lt; time&gt; &lt;stop time&gt;</td>
</tr>
<tr>
<td>r</td>
<td>&lt;repeat interval&gt; &lt;active duration&gt; &lt;offsets from start-time&gt;</td>
</tr>
<tr>
<td>z</td>
<td>&lt;adjustment time&gt; &lt;offset&gt; &lt;adjustment time&gt; &lt;offset&gt;</td>
</tr>
<tr>
<td>k</td>
<td>&lt;method&gt;:&lt;encryption key&gt;</td>
</tr>
<tr>
<td>a</td>
<td>&lt;attribute&gt;&lt;value&gt;</td>
</tr>
<tr>
<td>m</td>
<td>&lt;media&gt; &lt;port&gt; &lt;transport&gt; &lt;format list&gt;</td>
</tr>
<tr>
<td>a</td>
<td>&lt;attribute&gt;&lt;value&gt;</td>
</tr>
</tbody>
</table>

SIP Registration Process

The registration process is used by a UA to bind its current location to the user to facilitate location and routing within the SIP network. This is accomplished by the UA sending a register message to a registrar. The registrar maintains the binding relationship for an amount of time specified in the Expires header. If a UA wants to maintain the address / IP address relation then it must periodically re-register. Upon registration the registrar transfers the information to a location register. The terms for registrar and location register may be only logical, as the functionality can reside in the same physical device. In fact, Proxy Servers, Registrar and location registers combinations are quite common in small networks. Figure 4 depicts a registration process where the registrar challenges the register message sent by the UA.
**Figure 4:** Registration Process

<table>
<thead>
<tr>
<th>Request Line</th>
<th>REGISTER sip:192.168.16.140 SIP/2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Via</td>
<td>SIP/2.0/UDP 192.168.16.102:10658;branch=z9hG4bK-d8754z-6f64d606d85428835-1---d8754z;rport</td>
</tr>
<tr>
<td>Max-Forwards</td>
<td>70</td>
</tr>
<tr>
<td>Contact</td>
<td><a href="">sip:9103682854@192.168.16.102:10658;rinstance=c1bbc08e764e1a59</a></td>
</tr>
<tr>
<td>To</td>
<td>&quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a></td>
</tr>
<tr>
<td>From</td>
<td>&quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=74215226</td>
</tr>
<tr>
<td>Call-ID</td>
<td>MjNmYjBlZjhiODJiZmI4MjdhMzViNmlyMDFlNzk5MzQ.</td>
</tr>
<tr>
<td>CSeq</td>
<td>1 REGISTER</td>
</tr>
<tr>
<td>Expires</td>
<td>3600</td>
</tr>
<tr>
<td>Allow</td>
<td>INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO</td>
</tr>
<tr>
<td>User-Agent</td>
<td>X-Lite release 1104o stamp 56125</td>
</tr>
<tr>
<td>Content-Length</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 15:** Register #1
**Table 16:** 401 Unauthorized #2

<table>
<thead>
<tr>
<th>Request Line</th>
<th>REGISTER sip:192.168.16.140 SIP/2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Headers</strong></td>
<td></td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.102:10658;branch=z9hG4bK-d8754z-8c733834e54a867d-1---d8754z;;rport</td>
<td></td>
</tr>
<tr>
<td>Max-Forwards: 70</td>
<td></td>
</tr>
<tr>
<td>Contact: <a href="">sip:9103682854@192.168.16.102:10658;rinstance=c1bbc08e764e1a59</a></td>
<td></td>
</tr>
<tr>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a></td>
<td></td>
</tr>
<tr>
<td>From: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=74215226</td>
<td></td>
</tr>
<tr>
<td>Call-ID: MjNmYjBlZjhiODJiZmI4MjdhMzViNmlyMDFkNzk5MzQ.</td>
<td></td>
</tr>
<tr>
<td>CSeq: 2 REGISTER</td>
<td></td>
</tr>
<tr>
<td>Expires: 3600</td>
<td></td>
</tr>
<tr>
<td>Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO</td>
<td></td>
</tr>
<tr>
<td>User-Agent: X-Lite release 1104o stamp 56125</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 0</td>
<td></td>
</tr>
</tbody>
</table>

**Table 17:** Register #3
## Table 18: 200 OK #4

<table>
<thead>
<tr>
<th>Request Line</th>
<th>SIP/2.0 200 OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td></td>
</tr>
<tr>
<td>Via</td>
<td>Via: SIP/2.0/UDP 192.168.16.102:10658;branch=z9hG4bK-d8754z-8c733834e54a867d-1-d8754z-;received=192.168.16.102;rport=10658</td>
</tr>
<tr>
<td>Max-Forwards</td>
<td>Max-Forwards: 69</td>
</tr>
<tr>
<td>Contact</td>
<td>Contact: &lt;sip:9103682854@192.168.16.102:10658; rinstance=c1bbc08e764e1a59&gt;</td>
</tr>
<tr>
<td>To</td>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=43a8aa5e</td>
</tr>
<tr>
<td>From</td>
<td>From: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=74215226</td>
</tr>
<tr>
<td>Call-ID</td>
<td>Call-ID: MjNmYjBiZjiODjiZml4MjdhMzViNmlyMDFkNzk5MzQ.</td>
</tr>
<tr>
<td>CSeq</td>
<td>CSeq: 2 REGISTER</td>
</tr>
<tr>
<td>Allow</td>
<td>Allow: INVITE, BYE, CANCEL, OPTIONS, ACK, REGISTER, SUBSCRIBE, PUBLISH</td>
</tr>
<tr>
<td>Server</td>
<td>Server: TekSIP/v2.7</td>
</tr>
<tr>
<td>Date</td>
<td>Date: Fri, 30 Jul 2010 15:19:23 GMT</td>
</tr>
<tr>
<td>Allow-Events</td>
<td>Allow-Events: presence, presence.winfo, message-summary</td>
</tr>
<tr>
<td>Expires</td>
<td>Expires: 3600</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Content-Length: 0</td>
</tr>
</tbody>
</table>
SIP Call Flow (Detailed Example)

Figure 5 depict the network and call flow used in the discussion of the detailed call flow. Individual messages are included.

 Invite #1

John Smith wishes to call Jane Doe. He dials 9103682854 on his User Agent. The User Agent is configured to route all outgoing requests through the proxy at IP address 192.168.16.140. John's User Agent formulates an Invite message with all pertinent information regarding this call and sends it to the proxy, as demonstrated in message description Table 19: Invite #1.
<table>
<thead>
<tr>
<th>Request Line</th>
<th>INVITE sip:9103682854@192.168.16.140 SIP/2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-2651db598629a27e-1---d8754z-;rport</td>
</tr>
<tr>
<td></td>
<td>Max-Forwards: 70</td>
</tr>
<tr>
<td></td>
<td>Contact: &lt;sip:9103683957@192.168.16.105:44646</td>
</tr>
<tr>
<td></td>
<td>To: &quot;Jane Doe&quot;&lt;sip:9103682854@192.168.16.140</td>
</tr>
<tr>
<td></td>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
</tr>
<tr>
<td></td>
<td>Call-ID: Yzk2ZjQ5YzJhOWVvNjImNjk5MWxYmMxMmJiNzOwOTg.</td>
</tr>
<tr>
<td></td>
<td>CSeq: 2 INVITE</td>
</tr>
<tr>
<td></td>
<td>Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO</td>
</tr>
<tr>
<td></td>
<td>Content-Type: application/sdp</td>
</tr>
<tr>
<td></td>
<td>Proxy-Authorization: Digest username=&quot;9103683957&quot;,realm=&quot;192.168.16.140&quot;,nonce=&quot;2b028e9e929f845f097aa41196e511ac&quot;,uri=&quot;sip:9103682854@192.168.16.140&quot;,response=&quot;e714b87b94f9f5d4ca3b288984963fb5&quot;,cnonce=&quot;d58e9e10d0a63b19d65b1380324a101&quot;,nc=</td>
</tr>
<tr>
<td></td>
<td>User-Agent: X-Lite release 1104o stamp 56125</td>
</tr>
<tr>
<td></td>
<td>Content-Length: 322</td>
</tr>
<tr>
<td>Empty Line</td>
<td>v=0</td>
</tr>
<tr>
<td></td>
<td>o=- 3 2 IN IP4 192.168.16.105</td>
</tr>
<tr>
<td></td>
<td>s=CounterPath X-Lite 3.0</td>
</tr>
<tr>
<td></td>
<td>c=IN IP4 192.168.16.105</td>
</tr>
<tr>
<td></td>
<td>t=0 0</td>
</tr>
<tr>
<td></td>
<td>m=audio 32854 RTP/AVP 107 0 8 101</td>
</tr>
<tr>
<td></td>
<td>a=alt:1 2 : oND7cBBb qhQukSDp 10.100.100.250 32854</td>
</tr>
<tr>
<td></td>
<td>a=alt:2 1 : napYhQOC H731+IWT 192.168.16.105 32854</td>
</tr>
<tr>
<td></td>
<td>a=fmtp:101 0-15</td>
</tr>
<tr>
<td></td>
<td>a=rtpmap:107 BV32/16000</td>
</tr>
<tr>
<td></td>
<td>a=rtpmap:101 telephone-event/8000</td>
</tr>
<tr>
<td></td>
<td>a=sendrecv</td>
</tr>
</tbody>
</table>

**Table 19:** Invite #1
Invite #2

The proxy receives the Invite request and analyzes it to determine where the request should be sent. The proxy knows that Jane Doe is currently active at IP – 192.168.16.102. The proxy appends its own IP address, in a Via Header, to the request and sends to Jane Doe’s User Agent, as demonstrated in message description Table 20: Invite #2.

<table>
<thead>
<tr>
<th>Request Line</th>
<th>INVITE sip:9103682854@192.168.16.102:56912 SIP/2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.140:5060;branch=z9hG4bK-85z25d58d7461b9</td>
<td></td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-2651db598629a27e-1---d8754z-;received=192.168.16.105;rport=44646</td>
<td></td>
</tr>
<tr>
<td>Max-Forwards: 69</td>
<td></td>
</tr>
<tr>
<td>Contact: <a href="">sip:9103683957@192.168.16.105:44646</a></td>
<td></td>
</tr>
<tr>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a></td>
<td></td>
</tr>
<tr>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
<td></td>
</tr>
<tr>
<td>Call-ID: Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMyMxMmJiNzQwOTg.</td>
<td></td>
</tr>
<tr>
<td>CSeq: 2 INVITE</td>
<td></td>
</tr>
<tr>
<td>Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO</td>
<td></td>
</tr>
<tr>
<td>Record-Route: <a href="">sip:192.168.16.140;lr</a></td>
<td></td>
</tr>
<tr>
<td>User-Agent: X-Lite release 1104o stamp 56125</td>
<td></td>
</tr>
<tr>
<td>Content-Type: application/sdp</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 322</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Empty Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>v=0</td>
</tr>
<tr>
<td>o=- 3 2 IN IP4 192.168.16.105</td>
</tr>
<tr>
<td>s=CounterPath X-Lite 3.0</td>
</tr>
<tr>
<td>c=IN IP4 192.168.16.105</td>
</tr>
<tr>
<td>t=0 0</td>
</tr>
<tr>
<td>m=audio 32854 RTP/AVP 107 0 8 101</td>
</tr>
<tr>
<td>a=alt:2 1 : oND7cBBb qhQukSDp 10.100.100.250 32854</td>
</tr>
<tr>
<td>a=alt:2 1 : napYhQOC H731+tWt 192.168.16.105 32854</td>
</tr>
<tr>
<td>a=fmtp:101 0-15</td>
</tr>
<tr>
<td>a=rtpmap:107 BV32/16000</td>
</tr>
</tbody>
</table>

Table 20: Invite #2
100 Trying 3#
Jane Doe’s User Agent receives and analyses the Invite Request from the proxy. The User Agent sends a 100 Trying Response regarding this call to the proxy, as demonstrated in message description Table 21: 100 Trying #3.

<table>
<thead>
<tr>
<th>Request Line</th>
<th>SIP/2.0 100 Trying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td></td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.140:5060;branch=z9hG4bK-85z25d58d7461b9</td>
<td></td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-2651db598629a27e-1---d8754z- ;received=192.168.16.105;rport=44646</td>
<td></td>
</tr>
<tr>
<td>To: &quot;Jane Doe&quot;&lt;sip:9103682854@192.168.16.140</td>
<td></td>
</tr>
<tr>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
<td></td>
</tr>
<tr>
<td>Call-ID: Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMxYmMxMmJInzQwOTg.</td>
<td></td>
</tr>
<tr>
<td>CSeq: 2 INVITE</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 0</td>
<td></td>
</tr>
</tbody>
</table>

Table 21: 100 Trying 3

100 Trying #4
The proxy receives the 100 Trying from Jane Doe’s User Agent and forwards it the John Smith’s User Agent indicating that the call will take a little longer to set up. John Smith’s User Agent receives the 100 Trying, as demonstrated in message description Table 22: 100 Trying #4.

<table>
<thead>
<tr>
<th>Request Line</th>
<th>SIP/2.0 100 Trying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td></td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-2651db598629a27e-1---d8754z- ;received=192.168.16.105;rport=44646</td>
<td></td>
</tr>
<tr>
<td>To: &quot;Jane Doe&quot;&lt;sip:9103682854@192.168.16.140</td>
<td></td>
</tr>
<tr>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
<td></td>
</tr>
<tr>
<td>Call-ID: Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMxYmMxMmJInzQwOTg.</td>
<td></td>
</tr>
<tr>
<td>CSeq: 2 INVITE</td>
<td></td>
</tr>
<tr>
<td>Max-Forwards: 70</td>
<td></td>
</tr>
<tr>
<td>Record-Route: <a href="">sip:192.168.16.140;lr</a></td>
<td></td>
</tr>
<tr>
<td>Content-Length: 0</td>
<td></td>
</tr>
</tbody>
</table>

Table 22: 100 Trying 4
180 Ringing #5
Jane's User Agent starts alerting Jane of an incoming call. The user agent sends a 180 Ringing response to the proxy indicating the alerting has begun, as demonstrated in message description Table 23: 180 Ringing #5.

<table>
<thead>
<tr>
<th>Request Line</th>
<th>SIP/2.0 180 Ringing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.140:5060;branch=z9hG4bK-85z25d58d7461b9</td>
<td></td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-2651db598629a27e-1---d8754z- ;received=192.168.16.105;rport=44646</td>
<td></td>
</tr>
<tr>
<td>Record-Route: &lt;sip:192.168.16.140;lr</td>
<td></td>
</tr>
<tr>
<td>Contact: &lt;sip:9103682854@192.168.16.102:56912</td>
<td></td>
</tr>
<tr>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=f63e876c</td>
<td></td>
</tr>
<tr>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
<td></td>
</tr>
<tr>
<td>Call-ID: Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMxYmMxMmJiNzQwOTg.</td>
<td></td>
</tr>
<tr>
<td>CSeq: 2 INVITE</td>
<td></td>
</tr>
<tr>
<td>User-Agent: X-Lite release 1104o stamp 56125</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 0</td>
<td></td>
</tr>
</tbody>
</table>

Table 23: 180 Ringing 5

180 Ringing #6
The Proxy receives the 180 Ringing response and forwards it to John's User Agent. John's User Agents receives the 180 Ringing Response and uses it to trigger the Ringback tone from User Agent to John's Handset, as demonstrated in message description Table 24: 180 Ringing #6.

<table>
<thead>
<tr>
<th>Request Line</th>
<th>SIP/2.0 180 Ringing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-2651db598629a27e-1---d8754z- ;received=192.168.16.105;rport=44646</td>
<td></td>
</tr>
<tr>
<td>Contact: &lt;sip:9103682854@192.168.16.102:56912</td>
<td></td>
</tr>
<tr>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=f63e876c</td>
<td></td>
</tr>
<tr>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
<td></td>
</tr>
<tr>
<td>Call-ID: Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMxYmMxMmJiNzQwOTg.</td>
<td></td>
</tr>
<tr>
<td>CSeq: 2 INVITE</td>
<td></td>
</tr>
<tr>
<td>Max-Forwards: 70</td>
<td></td>
</tr>
<tr>
<td>Record-Route: &lt;sip:192.168.16.140;lr</td>
<td></td>
</tr>
<tr>
<td>User-Agent: X-Lite release 1104o stamp 5612</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 0</td>
<td></td>
</tr>
</tbody>
</table>

Table 24: 180 Ringing 6
**200 OK #7**

Jane and her associated User Agent have decided that the call from John should be accepted (Call Answered). A 200 OK response is sent to the proxy. The 200 OK contains an SDP Message Body listing the media parameters required by Jane’s User Agent for establishing the media session, as demonstrated in message description Table 25: 200 OK #7.

![Table 25: 200 OK #7](image)
200 OK #8
The 200 OK response is received by the proxy and forwarded to John’s User Agent. John’s User Agent receives the 200 OK response and starts preparing the ports, codexes and other capabilities for the media session, as demonstrated in message description Table 26: 200 OK #8.

```
<table>
<thead>
<tr>
<th>Request Line</th>
<th>SIP/2.0 200 OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Via:</td>
<td>SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bk-d8754z-2651db598629a27e-1---d8754z;received=192.168.16.105;router=44646</td>
</tr>
<tr>
<td>Contact:</td>
<td>&lt;sip:9103682854@192.168.16.102:56912</td>
</tr>
<tr>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=f63e876c</td>
<td></td>
</tr>
<tr>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
<td></td>
</tr>
<tr>
<td>Call-ID:</td>
<td>Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMyYmMmJiJmQwOTg</td>
</tr>
<tr>
<td>CSeq: 2 INVITE</td>
<td></td>
</tr>
<tr>
<td>Allow:</td>
<td>INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO</td>
</tr>
<tr>
<td>Max-Forwards: 70</td>
<td></td>
</tr>
<tr>
<td>Record-Route: &lt;sip:192.168.16.140;lr</td>
<td></td>
</tr>
<tr>
<td>User-Agent: X-Lite release 1104o stamp 56125</td>
<td></td>
</tr>
<tr>
<td>Content-Type: application/sdp</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 270</td>
<td></td>
</tr>
</tbody>
</table>

| Empty Line | v=0 |
| Message Body SDP | o=- 8 2 IN IP4 192.168.16.102 |
|             | s=CounterPath X-Lite 3.0 |
|             | c=IN IP4 192.168.16.102 |
|             | t=0 0 |
|             | m=audio 10574 RTP/AVP 107 0 8 101 |
|             | a=alt:1 1 : EBcwd5ZA XQt5DSSm 192.168.16.102 10574 |
|             | a=fmtp:101 0-15 |
|             | a=rtpmap:107 BV32/16000 |
|             | a=rtpmap:101 telephone-event/8000 |
|             | a=sendrecv |
```

Table 26: 200 OK 8
ACK #9
John's User Agent sends an ACK to the proxy regarding this call. As the ACK is sent, the media session is established using the RTP Protocol and is addressed directly from John's User Agent to Jane's User Agent, as demonstrated in message description Table 27: ACK #9.

<table>
<thead>
<tr>
<th>Request Line</th>
<th>ACK sip:9103682854@192.168.16.102:56912 SIP/2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-cd63ad64065fb73c-1---d8754z-rport</td>
</tr>
<tr>
<td></td>
<td>Max-Forwards: 70</td>
</tr>
<tr>
<td></td>
<td>Route: &lt;sip:192.168.16.140;lr</td>
</tr>
<tr>
<td></td>
<td>Contact: <a href="">sip:9103683957@192.168.16.140</a>;tag=f63e876c</td>
</tr>
<tr>
<td></td>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=f63e876c</td>
</tr>
<tr>
<td></td>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
</tr>
<tr>
<td></td>
<td>Call-ID: Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMyMmJlNzQwOTg.</td>
</tr>
<tr>
<td></td>
<td>CSeq: 2 ACK</td>
</tr>
<tr>
<td></td>
<td>Proxy-Authorization: Digest</td>
</tr>
<tr>
<td></td>
<td>username=&quot;9103683957&quot;,realm=&quot;192.168.16.140&quot;,nonce=&quot;2b028e9e929f845f097aa41196e511ac&quot;,uri=&quot;sip:9103682854@192.168.16.140&quot;,response=&quot;e714b87b94f9f5d4ca3b288984963fb5&quot;,cnonce=&quot;dc58e9e10d0a63b19d65b1380324a101&quot;,nc=</td>
</tr>
<tr>
<td></td>
<td>User-Agent: X-Lite release 1104o stamp 56125</td>
</tr>
<tr>
<td></td>
<td>Content-Length: 0</td>
</tr>
</tbody>
</table>

Table 27: ACK 9

ACK #10
The Proxy receives the ACK from John's User Agent and forwards it to Jane's User Agent. Jane's User Agent receives the ACK and insures the Media Session is established according to the SDP received in the Invite Request. The media from Jane's User Agent toward John's User Agent uses RTP and is addressed directly to John's User Agent, as demonstrated in message description Table 28: ACK #10.
After the conversation has been completed John hangs up the phone – his User Agent recognizes this condition and sends a Bye Request to the proxy regarding this call tearing down the media session, as demonstrated in message description Table 29: BYE #11.

Table 28: ACK 10

Table 29: BYE 11
BYE #12
The proxy receives the BYE from John’s User Agent and forwards it to Jane’s User Agent. Jane’s User Agent receives the Bye and tears down the media session, as demonstrated in message description Table 30: BYE #12.

<table>
<thead>
<tr>
<th>Request Line</th>
<th>BYE sip:9103682854@192.168.16.102:56912 SIP/2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Via: SIP/2.0/UDP 192.168.16.140:5060;branch=z9hG4bK-85zf7828d7482d1</td>
</tr>
<tr>
<td></td>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-f8728d218024a502-1---d8754z;received=192.168.16.105;rport=44646</td>
</tr>
<tr>
<td></td>
<td>Max-Forwards: 69</td>
</tr>
<tr>
<td></td>
<td>Contact: &lt;sip:9103683957@192.168.16.105:44646</td>
</tr>
<tr>
<td></td>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=f63e876c</td>
</tr>
<tr>
<td></td>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
</tr>
<tr>
<td></td>
<td>Call-ID: Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMyMmJiNzQwOTg.</td>
</tr>
<tr>
<td></td>
<td>CSeq: 3 BYE</td>
</tr>
<tr>
<td></td>
<td>Reason: SIP;description=&quot;User Hung Up&quot;</td>
</tr>
<tr>
<td></td>
<td>Route: &lt;sip:192.168.16.140;lri</td>
</tr>
<tr>
<td></td>
<td>Record-Route: &lt;sip:192.168.16.140;lri</td>
</tr>
<tr>
<td></td>
<td>User-Agent: X-Lite release 1104o stamp 56125</td>
</tr>
<tr>
<td></td>
<td>Content-Length: 0</td>
</tr>
</tbody>
</table>

Table 30: BYE 12

200 OK #13
Jane’s User Agent sends a 200 OK to the proxy confirming the Bye and the disconnection of the media session. The proxy receives the 200 OK and forwards it to John’s User Agent, as demonstrated in message description Table 31: 200 OK #13.
200 OK #14

John's User Agent receives the 200 OK from the proxy and releases any resources associated with the call, as demonstrated in message description Table 32: 200 OK #14.

Table 32: 200-OK 14

<table>
<thead>
<tr>
<th>Request Line</th>
<th>SIP/2.0 200 OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
<td></td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP 192.168.16.105:44646;branch=z9hG4bK-d8754z-f8728d218024a502-1---d8754z-;received=192.168.16.105;rport=44646</td>
<td></td>
</tr>
<tr>
<td>Contact: &lt;sip:9103682854@192.168.16.102:56912</td>
<td></td>
</tr>
<tr>
<td>To: &quot;Jane Doe&quot;<a href="">sip:9103682854@192.168.16.140</a>;tag=f63e876c</td>
<td></td>
</tr>
<tr>
<td>From: &quot;John Smith&quot;<a href="">sip:9103683957@192.168.16.140</a>;tag=5c06d71d</td>
<td></td>
</tr>
<tr>
<td>Call-ID: Yzk2ZjQ5YzJhOWVmNTJmNjk5MWMxYmMxMmJiNzQwOTg.</td>
<td></td>
</tr>
<tr>
<td>CSeq: 3 BYE</td>
<td></td>
</tr>
<tr>
<td>Max-Forwards: 70</td>
<td></td>
</tr>
<tr>
<td>Record-Route: &lt;sip:192.168.16.140;lr</td>
<td></td>
</tr>
<tr>
<td>User-Agent: X-Lite release 1104o stamp 56125</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 0</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion
The SIP protocol is a powerful element in the future of communications – telecommunications and any other type of communications requiring the setup and control of sessions to transmit a wide variety of information. SIP is the foundation of all types of Voice over IP (VoIP) communications in the enterprise, wireline and next generation wireless market spaces. Its power comes from the fact that is focused only on the establishment and the modification of media sessions. It relies heavily on other protocols to perform the transport of SIP itself and the actual media it controls.

*I hope this tutorial has been helpful in the start of your journey toward an understanding of SIP.*
Reference List

<table>
<thead>
<tr>
<th>RFC 2976</th>
<th>The SIP INFO Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 3261</td>
<td>SIP: Session Initiation Protocol</td>
</tr>
<tr>
<td>RFC 3262</td>
<td>Reliability of Provisional Responses in Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>RFC 3265</td>
<td>Session Initiation Protocol (SIP) - Specific Event Notification</td>
</tr>
<tr>
<td>RFC 3311</td>
<td>The Session Initiation Protocol (SIP) UPDATE Method</td>
</tr>
<tr>
<td>RFC 3312</td>
<td>Integration of Resource Management and Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>RFC 3313</td>
<td>Private Session Initiation Protocol (SIP) Extensions for Media Authorization</td>
</tr>
<tr>
<td>RFC 3323</td>
<td>A Privacy Mechanism for the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>RFC 3325</td>
<td>Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks</td>
</tr>
<tr>
<td>RFC 3326</td>
<td>The Reason Header Field for the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>RFC 3327</td>
<td>Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contracts</td>
</tr>
<tr>
<td>RFC 3329</td>
<td>Security Mechanism Agreement for the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>RFC 3372</td>
<td>Session Initiation Protocol for Telephones (SIP-T): Context and Architectures</td>
</tr>
<tr>
<td>RFC 3398</td>
<td>Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping</td>
</tr>
<tr>
<td>RFC 3428</td>
<td>Session Initiation Protocol (SIP) Extension for Instant Messaging</td>
</tr>
<tr>
<td>RFC 3455</td>
<td>Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)</td>
</tr>
<tr>
<td>RFC 3515</td>
<td>The Session Initiation Protocol (SIP) Refer Method</td>
</tr>
<tr>
<td>RFC 3578</td>
<td>Mapping of Integrated Services Digital Network (ISDN) User Part (ISUP) Overlap Signaling to the Session Initiation Protocol (SIP)</td>
</tr>
</tbody>
</table>
### Reference List (cont.)

<table>
<thead>
<tr>
<th>RFC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3588</td>
<td>Diameter Base Protocol</td>
</tr>
<tr>
<td>3603</td>
<td>Private Session Initiation Protocol (SIP) Proxy-to-Proxy Extensions for</td>
</tr>
<tr>
<td></td>
<td>Supporting the PacketCable Distributed Call Signaling Architecture</td>
</tr>
<tr>
<td>3605</td>
<td>Real Time Control Protocol (RTCP) Attribute in Session Description Protocol</td>
</tr>
<tr>
<td>3608</td>
<td>Session Initiation Protocol (SIP) Extension Header Field for Service Route</td>
</tr>
<tr>
<td></td>
<td>Discovery During Registration</td>
</tr>
<tr>
<td>3841</td>
<td>Caller Preferences for the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>3891</td>
<td>The Session Initiation Protocol (SIP) Replaces Header</td>
</tr>
<tr>
<td>3892</td>
<td>The Session Initiation Protocol (SIP) Referred-By Mechanism</td>
</tr>
<tr>
<td></td>
<td>Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>3903</td>
<td>Extension for Event State Publication</td>
</tr>
<tr>
<td>3911</td>
<td>The Session Initiation Protocol (SIP) Join Header</td>
</tr>
<tr>
<td>4028</td>
<td>Session Timers in the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>4244</td>
<td>An Extension to the Session Initiation Protocol (SIP) for Request History</td>
</tr>
<tr>
<td></td>
<td>Information</td>
</tr>
<tr>
<td>4354</td>
<td>A Session Initiation Protocol (SIP) Event Package and Data Format for Various</td>
</tr>
<tr>
<td></td>
<td>Settings in Support for the Push-to-Talk over Cellular (PoC) Service</td>
</tr>
<tr>
<td>4412</td>
<td>Communications Resource Priority for the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>4457</td>
<td>The Session Initiation Protocol (SIP) P-User-Database Private-Header (P-Header)</td>
</tr>
<tr>
<td>4474</td>
<td>Enhancements for Authenticated Identify Management in the Session</td>
</tr>
<tr>
<td></td>
<td>Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>4488</td>
<td>Suppression of Session Initiation Protocol (SIP) REFER Method Implicit</td>
</tr>
<tr>
<td></td>
<td>Subscription</td>
</tr>
</tbody>
</table>

© Copyright 2010 Performance Technologies, Inc. All Rights Reserved. 0910
### Reference List (cont.)

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4538</td>
<td>Request Authorization through Dialog Identification in the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>4566</td>
<td>SDP: Session Description Protocol</td>
</tr>
<tr>
<td>4964</td>
<td>The P-Answer-State Header Extension to the Session Initiation Protocol for the Open Mobile Alliance Push-to-Talk over Cellular</td>
</tr>
<tr>
<td>5002</td>
<td>The Session Intiation Protocol (SIP) P-Profile-Key Private Header (P-Header)</td>
</tr>
<tr>
<td>5009</td>
<td>Private Header (P-Header) Extension to the Session Initiation Protocol (SIP) for Authorization of Early Media</td>
</tr>
<tr>
<td>5079</td>
<td>Rejecting Anonymous Requests in the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>5360</td>
<td>A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>5373</td>
<td>Requesting Answering Modes for the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>5503</td>
<td>Private Session Initiation Protocol (SIP) Proxy-to-Proxy Extensions for Supporting the PacketCable Distributed Call Signaling Architecture</td>
</tr>
<tr>
<td>5603</td>
<td>The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>5876</td>
<td>Updates to Asserted Identify in the Session Initiation Protocol (SIP)</td>
</tr>
</tbody>
</table>

The Internet Engineering Task Force:  [www.ietf.org](http://www.ietf.org)


About the Author

Tom Jenkins has over 40 years experience in telecommunications. During his career, he has held positions related to SS7 Signaling including: Technical Support Manager, Manager of Product Management for STPs, International Sales Director for SS7 Test Equipment, and Vice President Sales and Marketing for SS7 Test Equipment. In 1997 Tom started Center Point Consulting, Inc., providing SS7, SigTran, and SIP training to over 2500 students worldwide. Tom has been actively involved with telecommunications signaling including SS7, SigTran, and SIP for 26 years. Today, Tom is the Senior Director of Xpress™ products for PT's Next-Generation applications. You can contact Tom at twj@pt.com.
About PT

Company Overview
PT is a global supplier of advanced network communications solutions to carrier, government, and OEM markets. The company's award-winning products include IP-centric network elements and applications designed for high availability, scalability, and long life cycle deployments. Over the years, PT has developed an impressive portfolio of communications offerings, through organic expansion as well as acquisitions.

Fully Integrated Product Portfolios
All products from PT reflect a significant body of real-world network experience and offer carriers, equipment manufacturers, and integrators carefully architected solution sets for communications network deployments.

The company's entire line of offerings is anchored by IPnexus®, PT's own IP-native, highly integrated platforms and element management systems. OEMs and application developers, including PT itself, leverage the robust carrier grade Linux® development environment and rich suite of communications protocols (PT’s NexusWare®) of IPnexus Application-Ready Systems as a cornerstone component of their end product value proposition.

SEGway™ Signaling Solutions, PT’s premier suite of IP-centric STPs, gateways, edges, and network applications, permits service providers to cost-effectively deliver revenue generating features in next-generation networks. SEGway solutions have been deployed in numerous world-class carrier networks around the globe because of their high density and unparalleled price-to-performance ratio. This, together with SEGway’s flexibility in edge-to-core deployments, small footprint, and ease of use, has earned the portfolio its reputation as “Simply Smarter Signaling.”

PT’s Xpress™, the company’s newest line, combines a powerful, customizable software application with a robust, hardware platform that is specifically engineered to deliver SIP-based solutions such as calling card and Pre-paid for next-generation networks. Based on a pure IP implementation, new service offerings can be quickly developed and deployed network-wide on IMS-enabled, converged TDM/IP and VoIP networks. The Xpress family of NGN solutions provides dynamic opportunities to build-out next-generation network solutions and value-added services.

Single Vendor Solution for Products and Services
Customers around the globe are turning to PT as a single source provider of high value proposition network communications solutions coupled with world-class support and maintenance programs.

Over 28 Years of Innovation and Industry Experience
PT has proven success in defining new levels of performance and adapting its products and services to a constantly changing marketplace. This expertise has enabled the company to remain a leader in its industry markets.

Today, backed by an innovative team of experts, PT offers a rich value proposition that enable its customers to meet their most pressing business and technology challenges, and thereby compete more successfully.

PT is headquartered in Rochester, NY and maintains sales and engineering offices around the world.