NEC Empowered by Innovation

Session Initiation Protocol (SIP) Basic Description Guide

Table of Contents:

Document Description

This manual will provide a basic description of Session Initiation Protocol (SIP) and the expected Requests and Responses. This manual is not intended to replace the applicable RFCs for SIP but only simplify the meanings. This document is not explicitly for any one NEC Product but more for SIP Itself.

Section 1 – Network Elements

1.1 User Agent

A SIP User Agent (UA) is a logical network end-point used to create or receive SIP messages and thereby manage a SIP session. A SIP UA can perform the role of a User Agent Client (UAC), which sends SIP requests, and the User Agent Server (UAS), which receives the requests and returns a SIP response. These roles of UAC and UAS only last for the duration of a SIP transaction.

1.2 Proxy server

An intermediary entity that acts as both a server (UAS) and a client (UAC) for the purpose of making requests on behalf of other clients. A proxy server primarily plays the role of routing, which means its job is to ensure that a request is sent to another entity "closer" to the targeted user.

1.3 Registrar

A server that accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles which registers one or more IP addresses to a certain SIP URI.
SIP registrars are logical elements, and are commonly co-located with SIP proxies. But it is also possible and often good for network scalability to place this location service with a redirect server.

1.4 Redirect server

A user agent server that generates 3xx (Redirection) responses to requests it receives, directing the client to contact an alternate set of URIs. The redirect server allows proxy servers to direct SIP session invitations to external domains.

1.5 Session border controller

Session border controllers serve as middle boxes between UA and SIP server for various types of functions, including network topology hiding, and assistance in NAT traversal.

1.6 Gateway

Gateways can be used to interface a SIP network to other networks, such as the public switched telephone network, which use different protocols or technologies.

1.7 Application Layer Gateway

ALG is a SIP aware monitoring device commonly contained in Routers and or Firewalls. SIP ALGs can have the capabilities of changing SIP Messages and should be disabled if any issues are experienced with SIP Calls.

Section 2 – SIP Messaging

2.1 SIP Requests

- REGISTER: Used by a User Agent (UA) to indicate 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.
- BYE: Terminates a session between two users in a conference.
- OPTIONS: Requests information about the capabilities of a caller, without setting up a call.
- PRACK (Provisional Response Acknowledgement): PRACK improves network reliability by adding an acknowledgement system to the provisional Responses (1xx). PRACK is sent in response to provisional response (1xx).
- SUBSCRIBE: Used to request current state and state updates from a remote node.
- NOTIFY: Sent to inform subscribers of changes in state to which the subscriber has a subscription.

2.2 SIP Responses

- 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 (typically by sender) to complete the request.
- Client Error (4xx): The request contains bad syntax or cannot be fulfilled at the server.
- Server Error (5xx): The server failed to fulfill an apparently valid request.
- Global Failure (6xx): The request cannot be fulfilled at any server.

2.3 Real Time Protocol

RTP protocol is used to carry the real-time multimedia data (including audio, video, and text), the protocol makes it possible to encode and split the data into packets and transport such packets over the Internet.

2.4 Session Description Protocol

Session Description Protocol is used to describe and encode the capabilities of session participants. Such a description is then used to negotiate the characteristics of the session so that all the devices can participate. That includes, for example, negotiation of Codec used to encode media so all the participants will be able to decode it, negotiation of transport protocol used and so on.

An INVITE w/ SDP Message will have SDP to determine the RTP connection information for the Calling Party. Although the most common method or call scenario will be SDP in the 200OK Message for the called party, some carriers will require Early Media which will have the SDP in the 18X RINGING Message. Early Media allows for Ring Back or other call tones to be played from the Carrier side before the called party answers.

Examples of SDP will be shown in Section 3.13.

Section 3 – SIP Messaging Headers

A valid SIP request formulated by a UAC MUST at a minimum contain the following header fields: To, From, CSeq, Call-ID, Max-Forwards, and Via; all of these header fields are mandatory in all SIP requests. These six header fields are the fundamental building blocks of a SIP message, as they jointly provide for most of the critical message routing services including the addressing of messages, the routing of responses, limiting message propagation, ordering of messages, and the unique identification of transactions. These header fields are in addition to the mandatory request line.

3.1 Request-URI or Request-Line

The initial Request-URI of the message SHOULD be set to the value of the URI in the To field. There are exceptions to this and it is not mandatory as noted by the SHOULD notation. The Request-URI or Request-Line contains the method, Request-URI, and SIP version. Full details of this Header are covered in RFC 3261.

Example of Request-URI:

Request-Line: INVITE sip:2142622000@Telcosipserver.com:5060 SIP/2.0
Method: INVITE
[Resent Packet: False]

3.2 VIA Header

The Via header field indicates the transport used for the transaction and identifies the location where the response is to be sent. A Via header field value is added only after the transport that will be used to reach the next hop has been selected.

When the UAC creates a request, it MUST insert a Via into that request. The protocol name and protocol version in the header field MUST be SIP and 2.0, respectively. The Via header field value MUST contain a branch parameter. This parameter is used to identify the transaction created by that request. This parameter is used by both the client and the server.

The branch parameter value MUST be unique across space and time for all requests sent by the UA. The exceptions to this rule are CANCEL and ACK for non-2xx responses. As discussed below, a CANCEL request will have the same value of the branch parameter as the request it cancels. An ACK for a non-2xx response will also have the same branch ID as the INVITE whose response it acknowledges

The branch ID inserted by an element compliant with this specification MUST always begin with the characters "z9hG4bK". These 7 characters are used as a magic cookie (7 is deemed sufficient to ensure that an older RFC 2543 implementation would not pick such a value), so that servers receiving the request can determine that the branch ID was constructed in the fashion described by this. Full details of this Header are covered in RFC 3261.

Example of Via Header:

Via: SIP/2.0/UDP 172.16.1.10:5060;branch=z9hG4bK-e14500690ed2-21
Transport: UDP (Packet Type)
Sent-by Address: 172.16.1.10 (Originating IP Address – Required field)
Sent-by port: 5060 (Originating Port – Required field)
Branch: z9hG4bK-e14500690ed2-21 (Transaction Identifier – Required field)

3.3 To Header

The To header field first and foremost specifies the desired "logical" recipient of the request, or the address-of-record of the user or resource that is the target of this request. In simple terms, the To header is the dialed or called number. Full details of this Header are covered in RFC 3261.

Example of To Header:

To: <sip:2142622000@Telcosipserver.com:5060>
SIP to address: sip:2142622000@Telcosipserver.com:5060

3.4 From Header

The From header field indicates the logical identity of the initiator of the request, possibly the user's address-of-record. Like the To header field, it contains a URI and optionally a display name. It is used by SIP elements to determine which processing rules to apply to a request (for example, automatic call rejection). As such, it is very important that the From URI not contain IP addresses or the Fully Qualified Domain Name of the host on which the UA is running, since these are not logical names. In simple terms, the From header is the calling party. Full details of this Header are covered in RFC 3261.

Example of From Header:

From: "Station 1000" <sip:2142621000@Telcosipserver.com:5060>;tag=18eb500690ed1-21
SIP Display info: "Station 1000"
SIP from address: sip:2142621000@Telcosipserver.com:5060
SIP tag: 18eb500690ed1-21

3.5 Call-ID Header

The Call-ID header field acts as a unique identifier to group together a series of messages. It MUST be the same for all requests and responses sent by either UA in a dialog. It SHOULD be the same in each registration from a UA. Full details of this Header are covered in RFC 3261.

Example of Call-ID Header:

Call-ID: bb-500690ed-0-21@172.16.1.10

3.6 CSeq Header

The CSeq header field serves as a way to identify and order transactions. It consists of a sequence number and a method. The method MUST match that of the request. For non-REGISTER requests outside of a dialog, the sequence number value is arbitrary. The sequence number value MUST be expressible as a 32-bit unsigned integer and MUST be less than 2**31. As long as it follows the above guidelines, a client may use any mechanism it would like to select CSeq header field values. Full details of this Header are covered in RFC 3261.

Example of Call-ID Header:

CSeq: 1 INVITE
Sequence Number: 1
Method: INVITE

3.7 Max-Forwards Header

The Max-Forwards header field serves to limit the number of hops a request can transit on the way to its destination. It consists of an integer that is decremented by one at each hop (Router). If the Max-Forwards value reaches 0 before the request reaches its destination, it will be rejected with a 483(Too Many Hops) error response. A UAC MUST insert a Max-Forwards header field into each request it originates with a value that SHOULD be 70. Full details of this Header are covered in RFC 3261.

Example of Max-Forwards Header:

Max-Forwards: 70

3.8 Contact Header

The Contact header field provides a SIP or SIPS URI that can be used to contact that specific instance of the UA for subsequent requests. The Contact header field MUST be present and contain exactly one SIP or SIPS URI in any request that can result in the establishment of a dialog. Full details of this Header are covered in RFC 3261.

Example of Contact Header:

Contact: <sip:2142621000@172.16.1.10:5060>
Contact Binding: <sip:2142621000@172.16.1.10:5060>
URI: <sip:2142621000@172.16.1.10:5060>
SIP contact address: sip:2142621000@172.16.1.10:5060

3.9 Allow Header (Optional Header)

An Allow header field SHOULD be present in the INVITE. It indicates what methods can be invoked within a dialog, on the UA sending the INVITE, for the duration of the dialog. For example, a UA capable of receiving INFO requests within a dialog SHOULD include an Allow header field listing the INFO method. Most Allow headers will state the SIP Request messages the UA supports. Full details of this Header are covered in RFC 3261.

Example of Allow Header:

Allow: INVITE, ACK, CANCEL, BYE, UPDATE, PRACK

3.10 Supported and Required Header (Optional Header)

If the UAC supports extensions to SIP that can be applied by the server to the response, the UAC SHOULD include a Supported header field in the request listing the option tags for those extensions. Full details of this Header are covered in RFC 3261.

Example of Supported and Required Headers:

Supported: 100 REL, timer

Required: timer

3.11 Min-SE Header (Optional Header)

The Min-SE header field establishes the lower bound for the session refresh interval; i.e., the fastest rate any proxy servicing this request will be allowed to require. The purpose of this header field is to prevent hostile proxies from setting arbitrarily short refresh intervals so that their neighbors are overloaded. Each proxy processing the request can raise this lower bound (increase the period between refreshes) but is not allowed to lower it. Full details of this Header are covered in RFC 3261.

Example of Min-SE Header:

Min-SE: 180 (180 represents seconds = 3 minutes)

3.12 Session-Expires Header (Optional Header)

The Session-Expires header field establishes the upper bound for the session refresh interval; i.e., the time period after processing a request for which any session-stateful proxy must retain its state for this session. Any proxy servicing this request can lower this value, but it is not allowed to decrease it below the value specified in the Min-SE header field. Session-Expires is active if the Supported or Required Header contains a “timer” option.

If the 2xx response did not contain a Session-Expires header field, there is no session expiration. In this case, no refreshes need to be sent. A 2xx without a Session-Expires can come for both initial and subsequent session refresh requests. This means that the session timer can be 'turned-off' in mid dialog by receiving a response without a Session-Expires header field.

The Session-Expires field can also indicate you will initiate the refresh by containing “refresher=uac” if the UAC wishes to perform the refresh.

Full details of this Header are covered in RFC 4028.

Example of Session-Expires Header:

Session-Expires: 1800 (1800 represents seconds = 30 minutes)

Example of Session-Expires Header with “refresher”:

Session-Expires: 1800;refresher=uac (The User Agent Client is responsible for generating a refresh of the session)

3.13 Message Body containing SDP (Session Description Protocol)

The Message body of the SIP Packet will contain the SDP if applicable. As described in Section 2.4, the SDP will control the Media portion of the SIP Session. When a SIP Request contains SDP a Content-Type or Accept Header will be included in the SIP Headers. Full details of this Protocol are covered in RFC 4566.

Example of Accept and Content-Type Headers:

Accept: application/sdp
Content-Type: application/sdp

3.13.1 Common Session Description Protocol Information Fields

13.13.1.1 Session Description Protocol Version (v):

The "v=" field gives the version of the Session Description Protocol. This memo defines version 0. There is no minor version number.

13.13.1.2 Owner/Creator, Session Id (o):

The "o=" field gives the originator of the session (username and the address of the user's host) plus a session identifier and version number.

13.13.1.3 Session Name (s):

The "s=" field is the textual session name. There MUST be one and only one "s=" field per session description. The "s=" field MUST NOT be empty and SHOULD contain ISO 10646 characters. If a Session has no meaningful name, the value "s= " SHOULD be used (i.e., a single space as the session name).

13.13.1.4 Connection Information (c):

The "c=" field contains connection data. A session description MUST contain either at least one "c=" field in each media description or a single "c=" field at the session level. It MAY contain a single session-level "c=" field and additional "c=" field(s) per media description.

13.13.1.5 Time Description, active time (t):

The "t=" lines specify the start and stop times for a session. Multiple "t=" lines MAY be used if a session is active at multiple irregularly spaced times; each additional "t=" line specifies an additional period of time for which the session will be active.

13.13.1.6 Media Description, name and address (m):

A session description may contain a number of media descriptions. Each media description starts with an "m=" field and is terminated by either the next "m=" field or by the end of the session description.
A media field has several sub-fields:
<media> is the media type. Currently defined media are "audio", "video", "text", "application", and "message", although this list may be extended in the future.

13.13.1.7 Media Attribute (a):

Attributes are the primary means for extending SDP. Attributes may be defined to be used as "session-level" attributes, "media-level" attributes, or both. A media description may have any number of attributes ("a=" fields) that are media specific. These are referred to as "media-level" attributes and add information about the media stream.
The most common Attributes used in VoIP are:

sendrecv = This specifies that the resources should be started in send and receive mode.
rtpmap = Media for RTP or DTMF encoding in the form of “<encoding name>/<clock rate>”

Examples:
rtpmap:0 PCMU/8000 (Represents G.711 encoding with a Clock Rate of 8000)
rtpmap:18 G729/8000 (Represents G.729 encoding with a Clock Rate of 8000)
rtpmap:2 G726/8000 (Represents G.726 encoding with a Clock Rate of 8000)
rtpmap:9 G722/8000 (Represents G.722 encoding with a Clock Rate of 8000)
rtpmap:101 telephone-event/8000 (Represents Out of Band DTMF 101 encoding with a Clock Rate of 8000)
Note: For Out of Band DTMF, please see RFC 2833 for further information. 101 and 110 encoding are common.

ptime = Media for Packet size

Examples:
ptime:20 (Represents a packet size of 20 milliseconds) Note: If no ptime is in the SDP, it is perceived to be 20 ms.
ptime:30 (Represents a packet size of 30 milliseconds)

Section 4 – SIP Messaging

4.1 Common SIP Request and Response messages

This section will cover some of the most common Request and Response SIP messages used in VoIP. The following will contain the basic meaning of these Requests and Responses. RFC3261 and RFC3265 can be referenced for more details.

4.1.1 REGISTER Request Message

REGISTER requests add, remove, and query bindings. A REGISTER request can add a new binding between an address-of-record and one or more contact addresses.

4.1.2 INVITE Request Message

The INVITE message is the initial request for a SIP Session. It will contain the required Request-URI, VIA, From, To, Call-ID, Max-Forward, Cseq Headers that will set the parameters of the SIP Session.

4.1.3 ACK Request Message

The ACK request constructed by the client transaction MUST contain values for the Call-ID, From, and Request-URI that are equal to the values of those header fields in the request passed to the transport by the client transaction (call this the "original request"). The To header field in the ACK MUST equal the To header field in the response being acknowledged, and therefore will usually differ from the To header field in the original request by the addition of the tag parameter. The ACK MUST contain a single Via header field, and this MUST be equal to the top Via header field of the original request. The CSeq header field in the ACK MUST contain the same value for the sequence number as was present in the original request, but the method parameter MUST be equal to "ACK".

4.1.4 BYE Request Message

A BYE Request is for termination of a Session and once a BYE Request is sent the UAC MUST consider the session terminated (and therefore stop sending or listening for media) as soon as the BYE request is passed to the client transaction. If the response for the BYE is a 481 (Call/Transaction Does Not Exist) or a 408 (Request Timeout) or no response at all is received for the BYE (that is, a timeout is returned by the client transaction), the UAC MUST consider the session and the dialog terminated.

4.1.5 SUBSCRIBE Request Message

The SUBSCRIBE method is used to request current state and state updates from a remote node.

4.1.6 NOTIFY Request Message

NOTIFY messages are sent to inform subscribers of changes in state to which the subscriber has a subscription. A NOTIFY does not terminate its corresponding subscription; in other words, a single SUBSCRIBE request may trigger several NOTIFY requests.

4.1.7 100 TRYING Response Message

This response indicates that the request has been received by the next-hop server and that some unspecified action is being taken on behalf of this call (for example, a database is being consulted). This response, like all other provisional responses, stops retransmissions of an INVITE by a UAC.

4.1.8 180 RINGING Response Message

The UA receiving the INVITE is trying to alert the user. This response MAY be used to initiate local ringback at the UA sending the INVITE.

4.1.9 181 Call is Being Forwarded Response Message

A server MAY use this status code to indicate that the call is being forwarded to a different set of destinations.

4.1.10 183 Session Progress Response Message

The 183 (Session Progress) response is used to convey information about the progress of the call that is not otherwise classified. The Reason-Phrase, header fields, or message body MAY be used to convey more details about the call progress.

4.1.11 200OK Response Message

The request has succeeded. The information returned with the response depends on the method used in the request.

4.1.12 401 Unauthorized Response Message

The request requires user authentication. This response is issued by UASs and registrars.

4.1.13 403 Forbidden Response Message

The server understood the request, but is refusing to fulfill it. Authorization will not help, and the request SHOULD NOT be repeated.

4.1.14 404 Not Found Response Message

The server has definitive information that the user does not exist at the domain specified in the Request-URI. This status is also returned if the domain in the Request-URI does not match any of the domains handled by the recipient of the request.

4.1.15 481 Call/Transaction Does Not Exists Response Message

This status indicates that the UAS received a request that does not match any existing dialog or transaction.

4.1.16 486 Busy Here Response Message

The called parties end system was contacted successfully, but the called party is currently not willing or able to take additional calls at this end system.

4.1.17 487 Request Terminated Response Message

The request was terminated by a BYE or CANCEL request. This response is never returned for a CANCEL request itself.

4.1.18 488 Not Acceptable Here Response Message

The response has the same meaning as 606 (Not Acceptable), but only applies to the specific resource addressed by the Request-URI and the request may succeed elsewhere.

4.1.19 500 Internal Server Error Response Message

The server encountered an unexpected condition that prevented it from fulfilling the request. The client MAY display the specific error condition and MAY retry the request after several seconds.

4.1.20 503 Service Unavailable Response Message

The server is temporarily unable to process the request due to a temporary overloading or maintenance of the server. The server MAY indicate when the client should retry the request in a Retry-After header field. If no Retry-After is given, the client MUST act as if it had received a 500 (Server Internal Error) response.

4.1.21 600 Busy Everywhere Response Message

The called party's end system was contacted successfully but the called party is busy and does not wish to take the call at this time. The response MAY indicate a better time to call in the Retry-After header field. If the called party does not wish to reveal the reason for declining the call, the called party uses status code 603 (Decline) instead. This status response is returned only if the client knows that no other end point (such as a voice mail system) will answer the request. Otherwise, 486 (Busy Here) should be returned.

4.1.22 603 Decline Response Message

The called party was successfully contacted but the user explicitly does not wish to or cannot participate. The response MAY indicate a better time to call in the Retry-After header field. This status response is returned only if the client knows that no other end point will answer the request.

Section 5 – SIP Registration and Event Messaging

5.1 SIP Registration

SIP Registration will be a 2 step or 4 step processes depending on whether the Server requires Authentication or not. If no Authentication is required, the original Registration will be accepted immediately and responded to with a 200OK response message. If Authentication is required, the original Registration request will be challenged with a 401 Unauthorized response requiring another Registration including information providing the identity of the registering UA. The Digest authentication scheme is the method of SIP Authentication. Registration with Authentication is the most common method of Registration but some Servers or Providers may not require Registration and the identity of the UA is stored in the Server database by IP Address. The example in this section will show Registration with Authentication.

5.2 Digest Authentication

Digest Authentication is a security protocol originally designed for HTTP security and was adopted for use in SIP. It uses a challenge-response mechanism for authentication. It is 4 way handshake using the initial request (Register), challenge (401 Unauthorized), response (Register) and allow (200OK). The 401 Unauthorized will contain the Digest Authentication information under the WWW-Authenticate response header and the re-try Register will contain the Digest responses in the Authorization Header. See RFC2617 for full details of Digest authentication.

5.2.1 WWW-Authenticate Response Header Fields of the 401 Unauthorized

5.2.1.1 Realm Field

A string that is displayed to users so they know which username and password to use for authentication. This string should contain at least the name of the host performing the authentication and might additionally indicate the collection of users who might have access. An example might be "registered_users@gotham.news.com".

5.2.1.2 Nounce Field

A server-specified data string which should be uniquely generated each time a 401 response is made. It is recommended that this string be base64 or hexadecimal data. Specifically, since the string is passed in the header lines as a quoted string, the double-quote character is not allowed.

5.2.1.3 Opaque Field

A string of data, specified by the server, which should be returned by the client unchanged in the Authorization header of subsequent requests with URIs in the same protection space. It is recommended that this string be base64 or hexadecimal data.

5.2.1.4 Stale Flag Field

A flag indicating that the previous request from the client was rejected because the nonce value was stale. If stale is TRUE (case-insensitive), the client may wish to simply retry the request with a new encrypted response, without re-prompting the user for a new username and password. The server should only set stale to TRUE if it receives a request for which the nonce is invalid but with a valid digest for that nonce (indicating that the client knows the correct username/password). If stale is FALSE, or anything other than TRUE, or the stale directive is not present, the username and/or password are invalid, and new values must be obtained.

5.2.1.5 Algorithm Field

A string indicating a pair of algorithms used to produce the digest and a checksum. If this is not present it is assumed to be "MD5". If the algorithm is not understood, the challenge should be ignored (and a different one used, if there is more than one).

5.2.1.6 QOP Field

This directive is optional, but is made so only for backward compatibility with RFC 2069; it SHOULD be used by all implementations compliant with this version of the Digest scheme. If present, it is a quoted string of one or more tokens indicating the "quality of protection" values supported by the server. The value "auth" indicates authentication; the value "auth-int" indicates authentication with integrity protection.

5.3.1 Authorization Response Header Fields of the Re-Try Register Request

5.3.1.1 Digest Authentication Response Field

A string of 32 hex digits computed as defined below, which proves that the user knows a password.

5.3.1.2 User Name Response Field

Users name in the specified realm.

5.3.1.3 Opaque Response Field

Must be the same as supplied in the WWW-Authenticate Header of the 401 Unauthorized message.

5.3.1.4 Algorithm Response Field

Must be the same as supplied in the WWW-Authenticate Header of the 401 Unauthorized message.

5.3.1.5 QOP Response Field

Indicates what "quality of protection" the client has applied to the message. If present, its value MUST be one of the alternatives the server indicated it supports in the WWW-Authenticate header of the 401 Unauthorized.

5.3.1.6 CNONCE Response Field

This MUST be specified if a qop directive is sent (see above), and MUST NOT be specified if the server did not send a qop directive in the WWW-Authenticate header field of the 401 Unauthorized.

5.3.1.7 NONCE COUNT Response Field

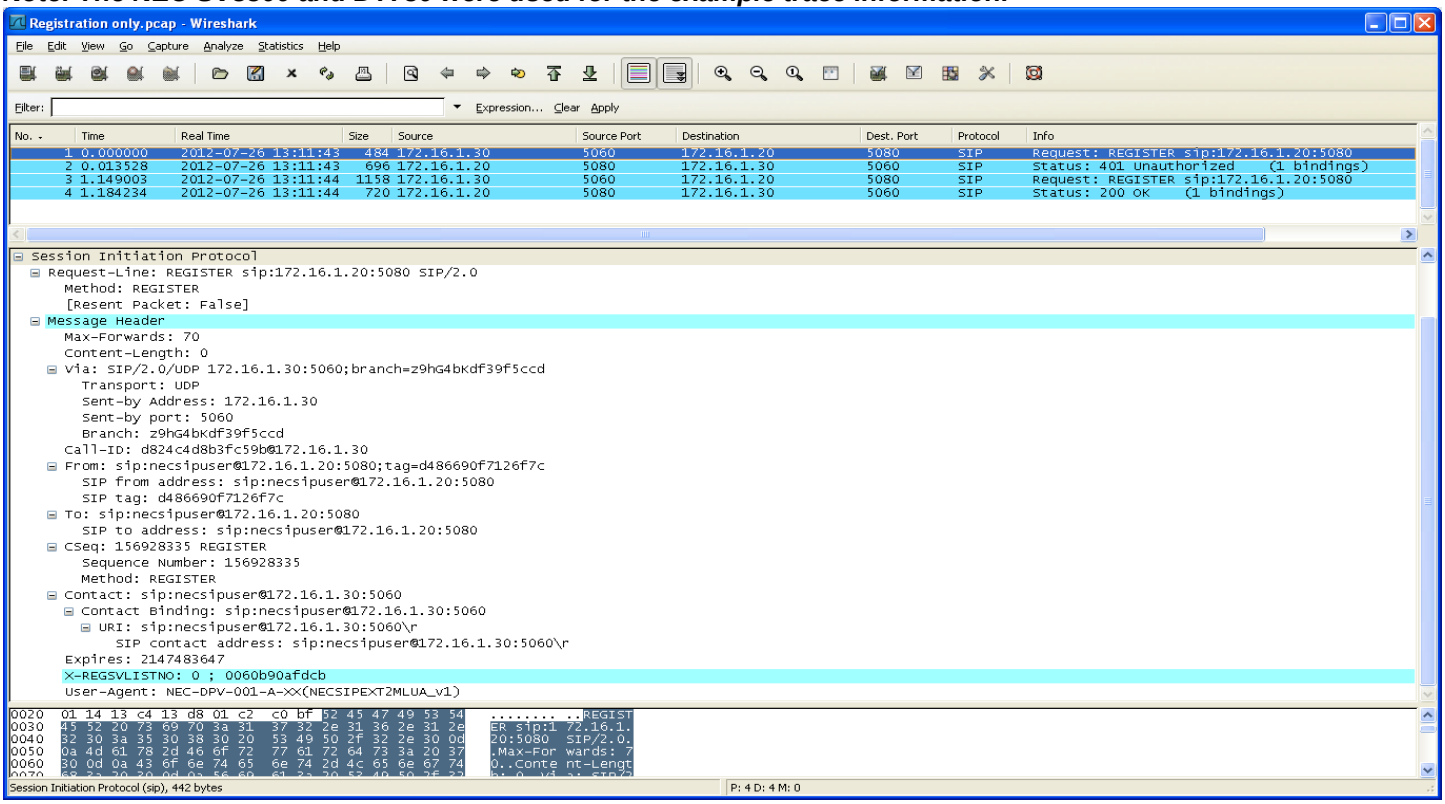
This MUST be specified if a qop directive is sent (see above), and MUST NOT be specified if the server did not send a qop directive in the WWW-Authenticate header field of the 401 Unauthorized. The nc-value is the hexadecimal count of the number of requests (including the current request) that the client has sent with the nonce value in this request. For example, in the first request sent in response to a given nonce value, the client sends "nc=00000001".

5.3.1.8 Authentication or Digest URI Response Field

The URI from Request-URI of the Request-Line; duplicated here because proxies are allowed to change the Request-Line in transit.

5.4.1 Initial REGISTRATION Request Message

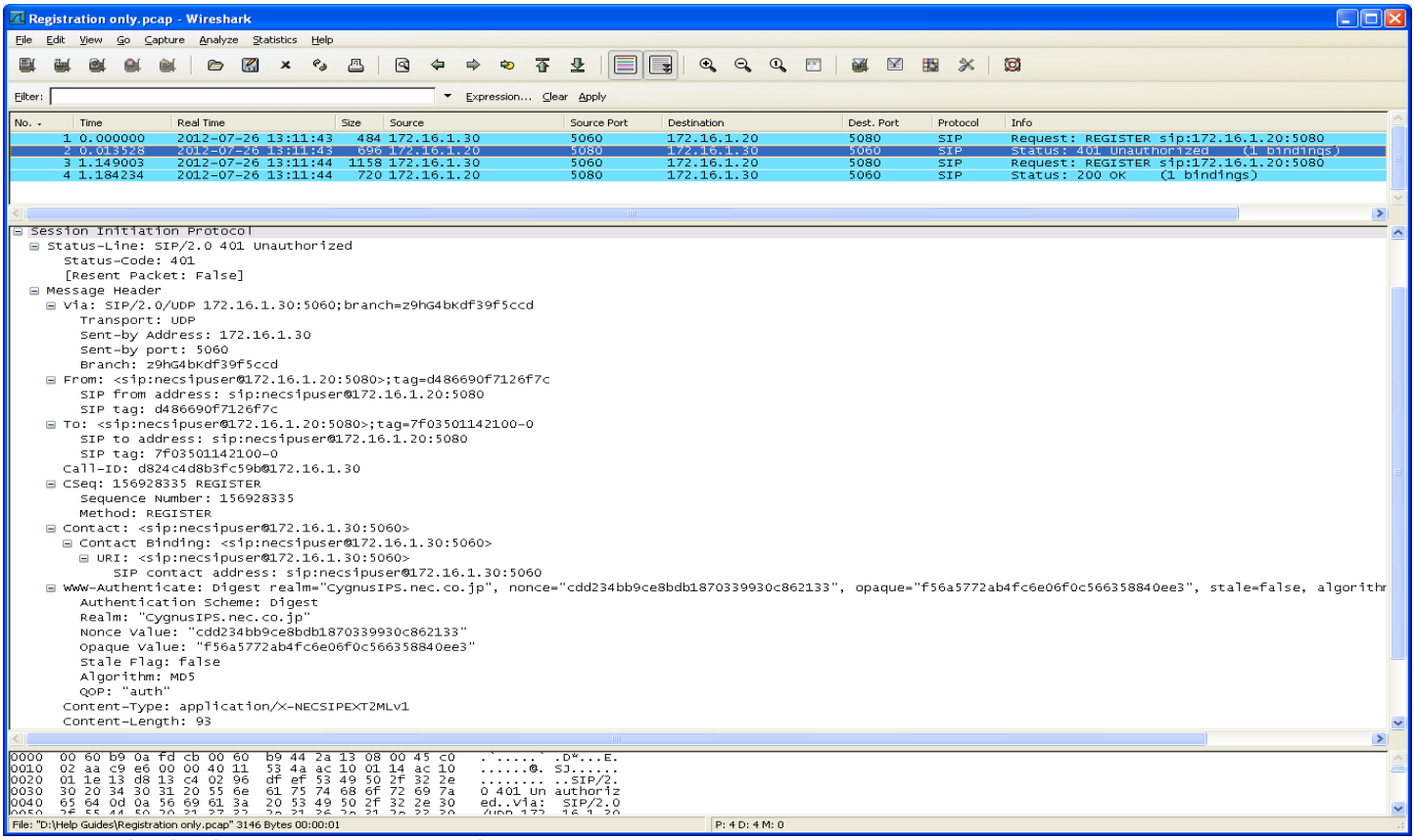
Note: The NEC SV8300 and DT730 were used for the example trace information.



Original REGISTRATION Request Message in Text Format

Session Initiation Protocol
Request-Line: REGISTER sip:172.16.1.20:5080 SIP/2.0 (Must contain the IP or Domain of the Server)
Method: REGISTER
[Resent Packet: False]
Message Header
Max-Forwards: 70
Content-Length: 0
Via: SIP/2.0/UDP 172.16.1.30:5060;branch=z9hG4bKdf39f5ccd
Transport: UDP
Sent-by Address: 172.16.1.30 (Originating IP Address)
Sent-by port: 5060 (Originating Port)
Branch: z9hG4bKdf39f5ccd (Transaction Identifier)
Call-ID: d824c4d8b3fc59b@172.16.1.30 (Call Identifier – will remain the same for every Packet of Registration)
From: sip:necsipuser@172.16.1.20:5080;tag=d486690f7126f7c (Registering Party information with tag information)
SIP from address: sip:necsipuser@172.16.1.20:5080
SIP tag: d486690f7126f7c
To: sip:necsipuser@172.16.1.20:5080 (Registrar information-tag information will be generated in the next response)
SIP to address: sip:necsipuser@172.16.1.20:5080
CSeq: 156928335 REGISTER
Sequence Number: 156928335 (Sequence Number of Registration process)
Method: REGISTER
Contact: sip:necsipuser@172.16.1.30:5060 (Contact information of the Registering Party)
Contact Binding: sip:necsipuser@172.16.1.30:5060

5.4.2 401 Unauthorized Response Message

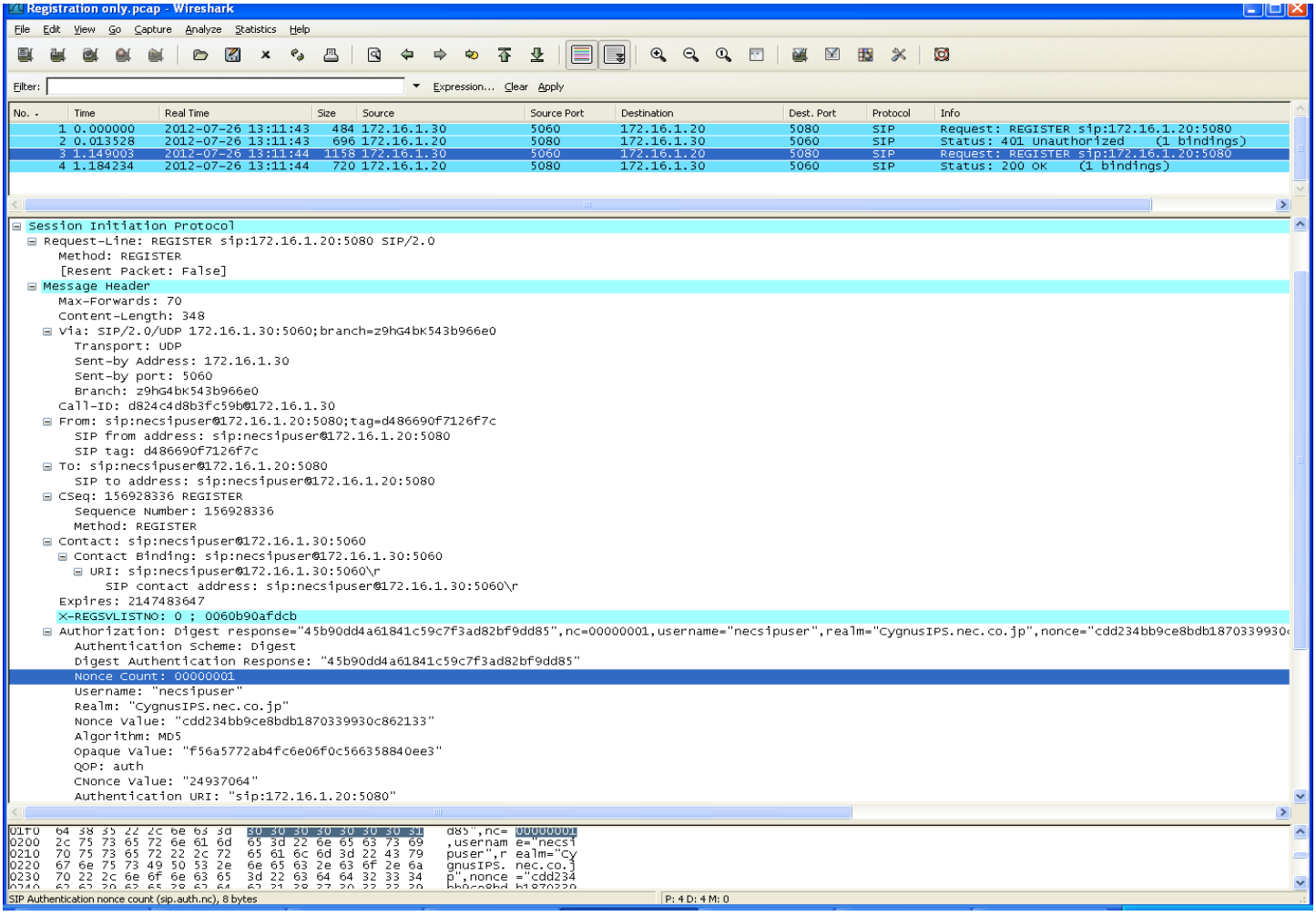


401 Unauthorized Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 401 Unauthorized
  - Status-Code: 401
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.30:5060;branch=z9hG4bKdf39f5ccd
    - Transport: UDP
    - Sent-by Address: 172.16.1.30 (Originating IP Address)
    - Sent-by port: 5060 (Originating Port)
    - Branch: z9hG4bKdf39f5ccd (Same as Register)
  - From: <sip:necsipuser@172.16.1.20:5080>;tag=d486690f7126f7c (Same as Register)
    - SIP from address: sip:necsipuser@172.16.1.20:5080
    - SIP tag: d486690f7126f7c
  - To: <sip:necsipuser@172.16.1.20:5080>;tag=7f03501142100-0 (Same as Register but now contains tag information)
    - SIP to address: sip:necsipuser@172.16.1.20:5080
    - SIP tag: 7f03501142100-0
  - Call-ID: d824c4d8b3fc59b@172.16.1.30 (Same as Register)
  - CSeq: 156928335 REGISTER (Same as Register)
    - Sequence Number: 156928335 (Same as Register)
    - Method: REGISTER
  - Contact: <sip:necsipuser@172.16.1.30:5060>
    - Contact Binding: <sip:necsipuser@172.16.1.30:5060>
      - URI: <sip:necsipuser@172.16.1.30:5060>
        - SIP contact address: sip:necsipuser@172.16.1.30:5060

WWW-Authenticate: Digest realm="CygnusIPS.nec.co.jp", nonce="cdd234bb9ce8bdb1870339930c862133",
opaque="f56a5772ab4fc6e06f0c566358840ee3", stale=false, algorithm=MD5, qop="auth"
Authentication Scheme: Digest (Authentication method)
Realm: "CygnusIPS.nec.co.jp"
Nonce Value: "cdd234bb9ce8bdb1870339930c862133"
Opaque Value: "f56a5772ab4fc6e06f0c566358840ee3"
Stale Flag: false
Algorithm: MD5 (Encryption method)
QOP: "auth"
Content-Type: application/X-NECSIPEXT2MLv1
Content-Length: 93

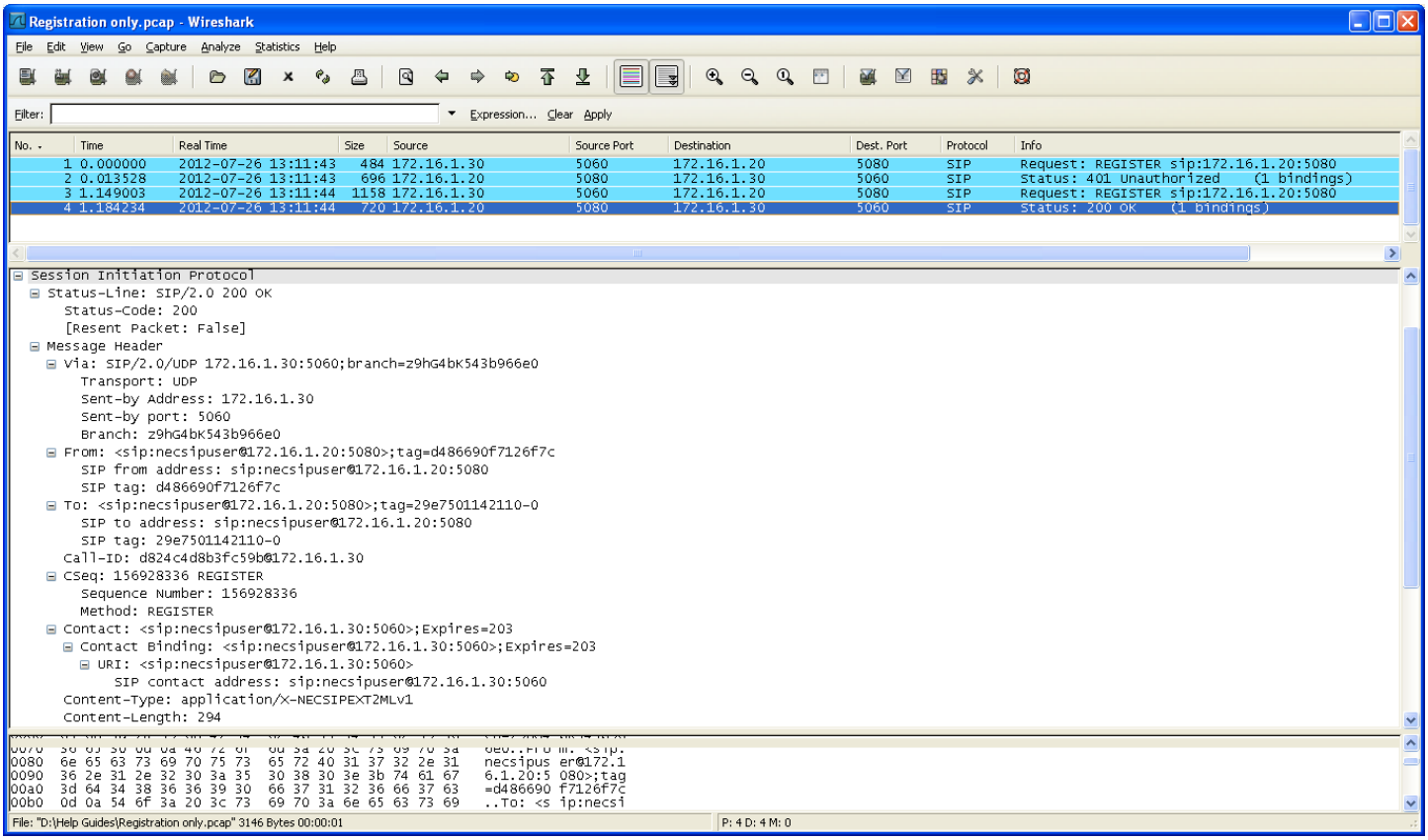
5.4.3 Re-Try REGISTER Request Message



Re-Try REGISTER Request Message in Text Format

Session Initiation Protocol
Request-Line: REGISTER sip:172.16.1.20:5080 SIP/2.0
Method: REGISTER
[Resent Packet: False]
Message Header
Max-Forwards: 70
Content-Length: 348
Via: SIP/2.0/UDP 172.16.1.30:5060;branch=z9hG4bK543b966e0
Transport: UDP
Sent-by Address: 172.16.1.30
Sent-by port: 5060
Branch: z9hG4bK543b966e0 (New Branch on re-try)
Call-ID: d824c4d8b3fc59b@172.16.1.30 (Same as Initial Register)
From: sip:necsipuser@172.16.1.20:5080;tag=d486690f7126f7c (Same as Initial Register and 401)
SIP from address: sip:necsipuser@172.16.1.20:5080
SIP tag: d486690f7126f7c
To: sip:necsipuser@172.16.1.20:5080
SIP to address: sip:necsipuser@172.16.1.20:5080
CSeq: 156928336 REGISTER
Sequence Number: 156928336 (Counted up by 1 from Initial Register)
Method: REGISTER
Contact: sip:necsipuser@172.16.1.30:5060
Contact Binding: sip:necsipuser@172.16.1.30:5060
URI: sip:necsipuser@172.16.1.30:5060\r
SIP contact address: sip:necsipuser@172.16.1.30:5060\r
Expires: 2147483647
X-REGSVLISTNO: 0 ; 0060b90afdcb
Authorization: Digest response="45b90dd4a61841c59c7f3ad82bf9dd85",nc=00000001,username="necsipuser",
realm="CygnusIPS.nec.co.jp",nonce="cdd234bb9ce8bdb1870339930c862133",algorithm=MD5,opaque="f56a5772ab4fc6
e06f0c566358840ee3",qop=auth,cnonc
Authentication Scheme: Digest (Same as 401 Unauthorized)
Digest Authentication Response: "45b90dd4a61841c59c7f3ad82bf9dd85" (Password Encrypted)
Nonce Count: 00000001 (Count as specified by spec)
Username: "necsipuser" (User name)
Realm: "CygnusIPS.nec.co.jp" (Same as 401 Unauthorized)
Nonce Value: "cdd234bb9ce8bdb1870339930c862133" (Same as 401 Unauthorized)
Algorithm: MD5 (Same as 401 Unauthorized)
Opaque Value: "f56a5772ab4fc6e06f0c566358840ee3" (Same as 401 Unauthorized)
QOP: auth (Same as 401 Unauthorized)
CNonce Value: "24937064"
Authentication URI: "sip:172.16.1.20:5080"

5.4.4 200OK Response Message allowing Registration



200OK Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 200 OK
  - Status-Code: 200
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.30:5060;branch=z9hG4bK543b966e0
    - Transport: UDP
    - Sent-by Address: 172.16.1.30
    - Sent-by port: 5060
    - Branch: z9hG4bK543b966e0 (Same as re-try Register)
  - From: <sip:necsipuser@172.16.1.20:5080>;tag=d486690f7126f7c (Same as Initial Register and 401)
    - SIP from address: sip:necsipuser@172.16.1.20:5080
    - SIP tag: d486690f7126f7c
  - To: <sip:necsipuser@172.16.1.20:5080>;tag=29e7501142110-0
    - SIP to address: sip:necsipuser@172.16.1.20:5080
    - SIP tag: 29e7501142110-0
  - Call-ID: d824c4d8b3fc59b@172.16.1.30 (Same as Initial Register)
  - CSeq: 156928336 REGISTER
    - Sequence Number: 156928336 (Same as re-try Register)
    - Method: REGISTER
  - Contact: <sip:necsipuser@172.16.1.30:5060>;Expires=203
    - Contact Binding: <sip:necsipuser@172.16.1.30:5060>;Expires=203 (Register expires in 203 seconds)
      - URI: <sip:necsipuser@172.16.1.30:5060>
        - SIP contact address: sip:necsipuser@172.16.1.30:5060

5.5 SIP Event Request Packets (SUBSCRIBE and NOTIFY)

In this section, the SUBSCRIBE and NOTIFY event request messaging. An event package is an additional specification which defines a set of state information to be reported by a Notifier to a Subscriber.

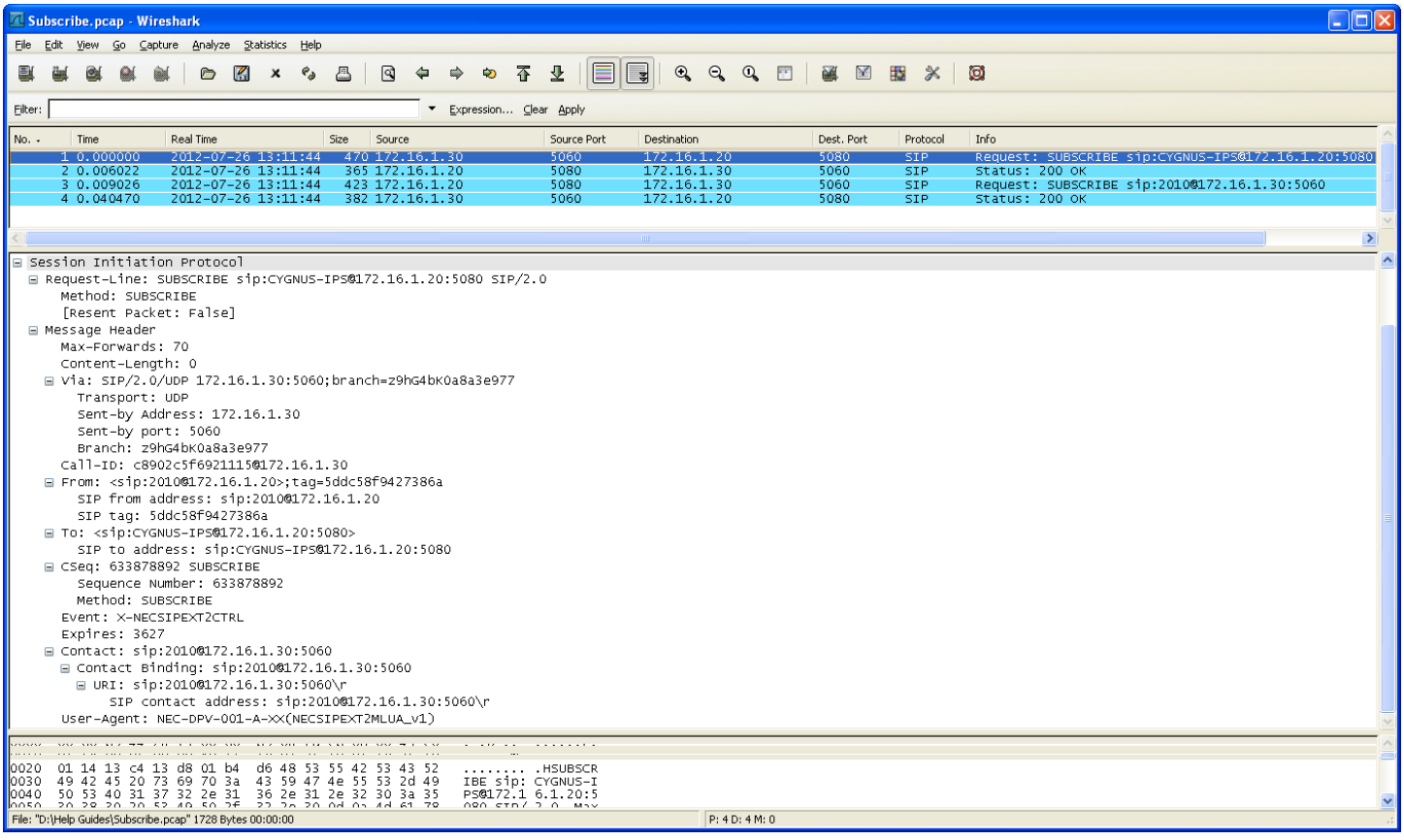
In the examples provided in this section, NEC SIP Messages will be used that were captured from a SIP IP Terminal. Although some of the information will be proprietary, it will convey the basic meaning of the messaging. SIP IP Terminal Subscribing will be Extension 2010. The terminal will Subscribe to the NEC SIP Server and the NEC SIP Server will Subscribe to the terminal, allowing Notifications to be sent and received from both devices.

Note: The NEC SV8300 and DT730 were used for the example trace information.

5.5.1 SUBSCRIBE Request Message

The SUBSCRIBE message is used to request current state and state updates from a remote node. In simple terms, it will allow the NOTIFY messages to be sent to the SUBSCRIBER. SUBSCRIBE Messages will need a 200OK response.

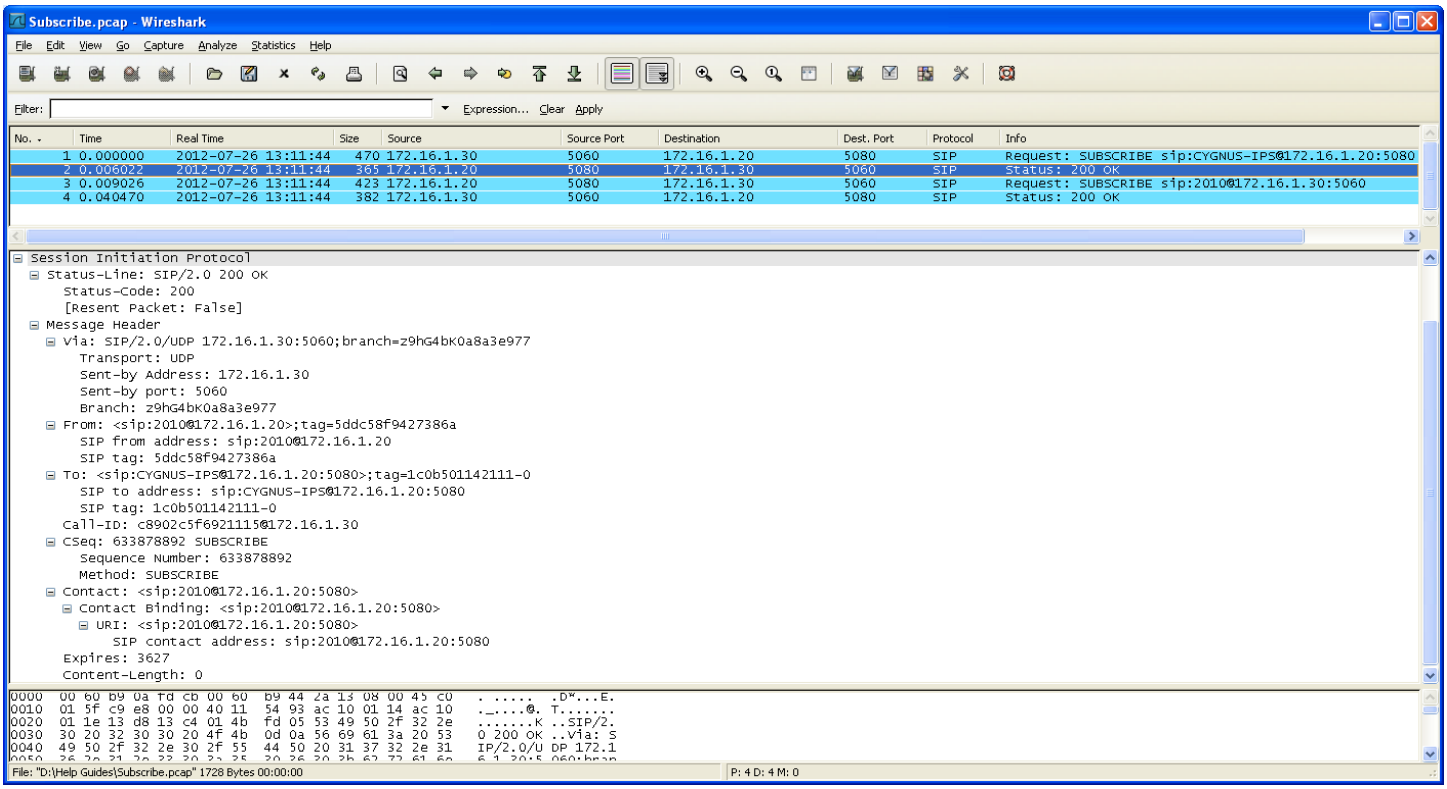
5.5.1.1 SUBSCRIBE Request Message from Terminal



SUBSCRIBE Request Message in Text Format

Session Initiation Protocol
  Request-Line: SUBSCRIBE sip:CYGNUS-IPS@172.16.1.20:5080 SIP/2.0
    Method: SUBSCRIBE
    [Resent Packet: False]
  Message Header
    Max-Forwards: 70
    Content-Length: 0
    Via: SIP/2.0/UDP 172.16.1.30:5060;branch=z9hG4bK0a8a3e977
      Transport: UDP
      Sent-by Address: 172.16.1.30 (Originating IP Address of the SIP IP Terminal)
      Sent-by port: 5060 (Originating Port)
      Branch: z9hG4bK0a8a3e977 (Transaction Identifier)
    Call-ID: c8902c5f6921115@172.16.1.30 (Call Identifier)
    From: <sip:2010@172.16.1.20>;tag=5ddc58f9427386a (SUBSCRIBER Information – Extension 2010)
      SIP from address: sip:2010@172.16.1.20
      SIP tag: 5ddc58f9427386a
    To: <sip:CYGNUS-IPS@172.16.1.20:5080> (NEC SIP Server)
      SIP to address: sip:CYGNUS-IPS@172.16.1.20:5080 (No tag – Unique tag will be generated in 200OK)
    CSeq: 633878892 SUBSCRIBE
      Sequence Number: 633878892
      Method: SUBSCRIBE
    Event: X-NECSIPEXT2CTRL (Event information allowing NEC SIP Station Control Messaging)
    Expires: 3627
    Contact: sip:2010@172.16.1.30:5060
      Contact Binding: sip:2010@172.16.1.30:5060
        URI: sip:2010@172.16.1.30:5060\r
          SIP contact address: sip:2010@172.16.1.30:5060\r
    User-Agent: NEC-DPV-001-A-XX(NECSIPEXT2MLUA_v1)

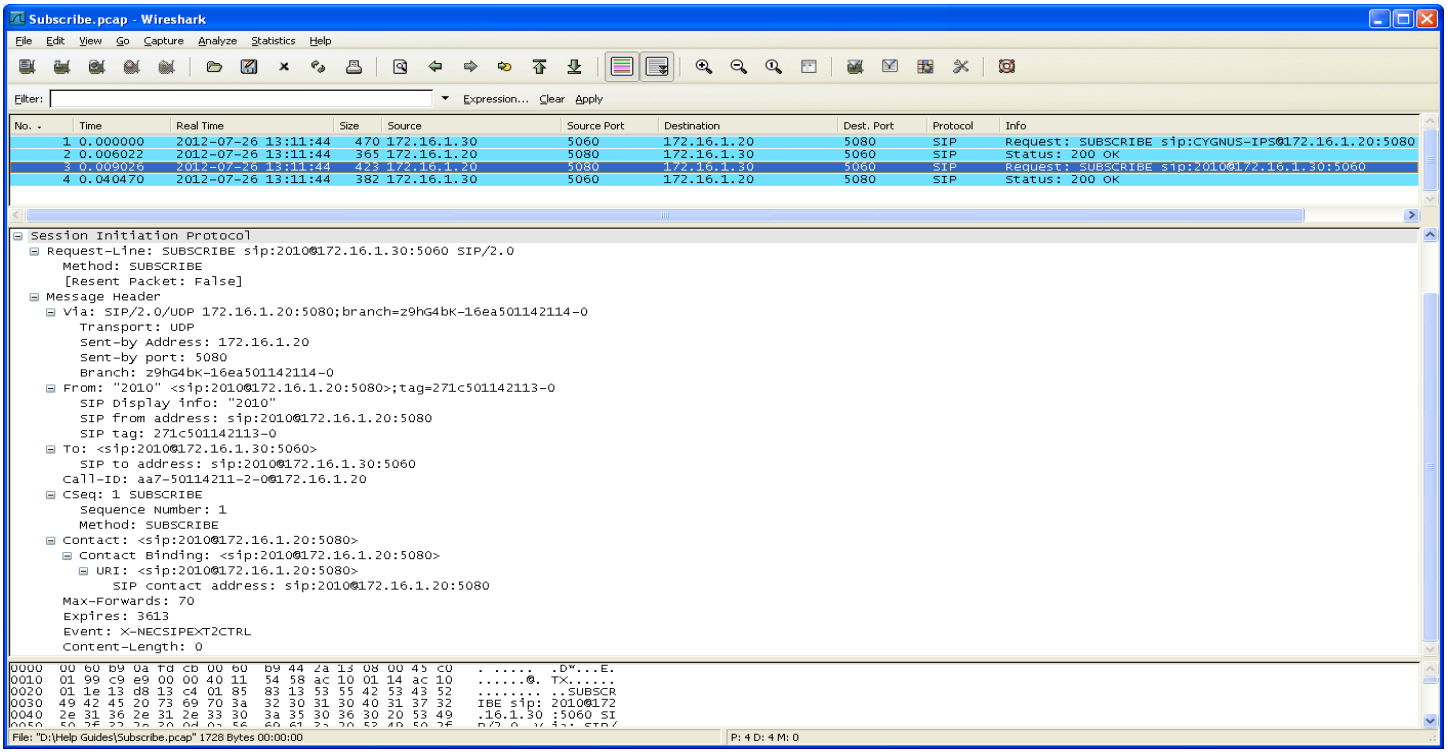
5.5.1.2 200OK Response Message to the SUBSCRIBE



200OK Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 200 OK
  - Status-Code: 200
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.30:5060;branch=z9hG4bK0a8a3e977
    - Transport: UDP
    - Sent-by Address: 172.16.1.30 (Originating IP Address of the SIP IP Terminal)
    - Sent-by port: 5060 (Originating Port of the SIP IP Terminal)
    - Branch: z9hG4bK0a8a3e977 (Transaction Identifier – Same as SUBSCRIBE)
  - From: <sip:2010@172.16.1.20>;tag=5ddc58f9427386a (Same as SUBSCRIBE)
    - SIP from address: sip:2010@172.16.1.20
    - SIP tag: 5ddc58f9427386a
  - To: <sip:CYGNUS-IPS@172.16.1.20:5080>;tag=1c0b501142111-0 (Same as SUBSCRIBE but with unique tag)
    - SIP to address: sip:CYGNUS-IPS@172.16.1.20:5080
    - SIP tag: 1c0b501142111-0
  - Call-ID: c8902c5f6921115@172.16.1.30 (Same as SUBSCRIBE)
  - CSeq: 633878892 SUBSCRIBE
    - Sequence Number: 633878892
    - Method: SUBSCRIBE
  - Contact: <sip:2010@172.16.1.20:5080>
    - Contact Binding: <sip:2010@172.16.1.20:5080>
      - URI: <sip:2010@172.16.1.20:5080>
        - SIP contact address: sip:2010@172.16.1.20:5080
  - Expires: 3627
  - Content-Length: 0

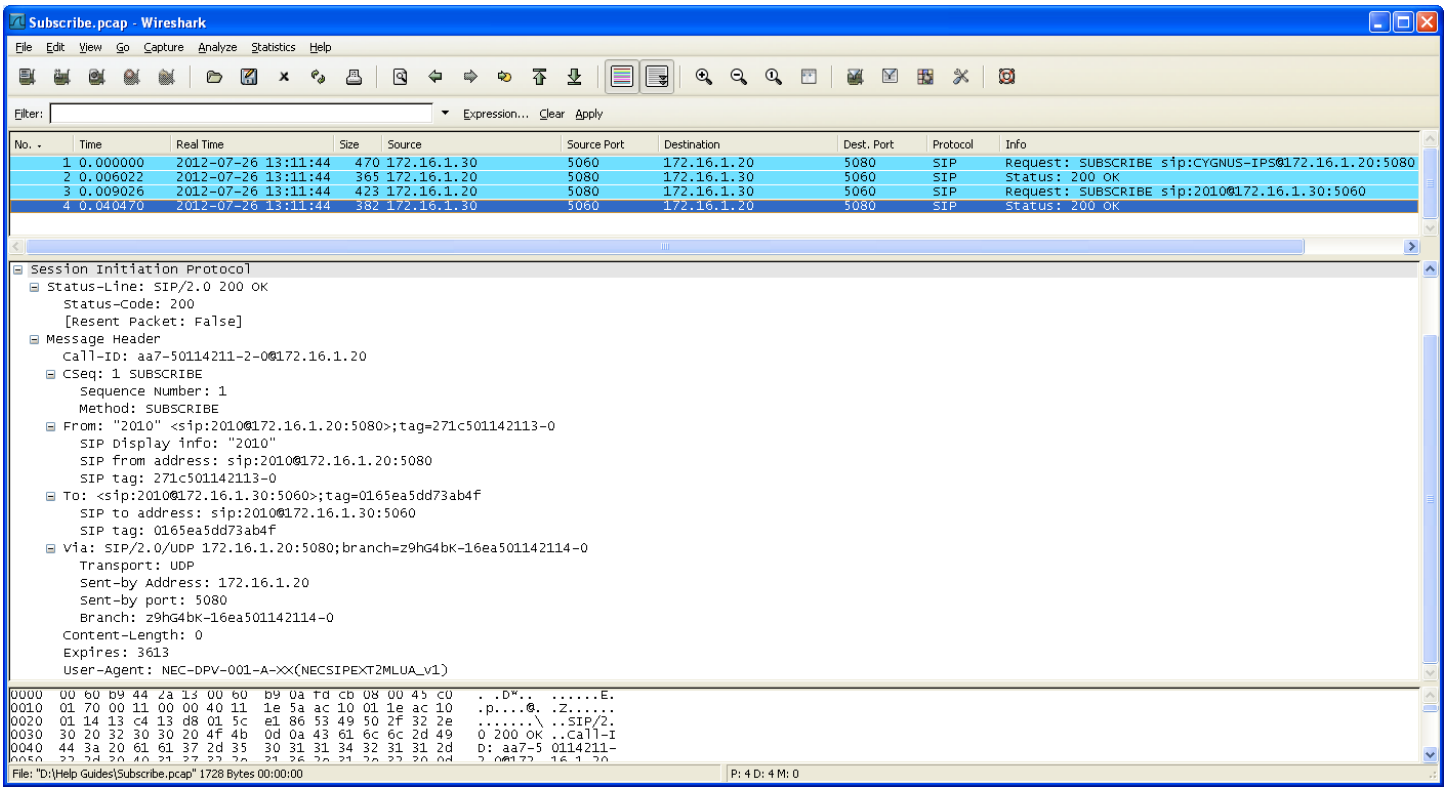
5.5.1.3 SUBSCRIBE Request Message from NEC SIP Server



SUBSCRIBE Request Message in Text Format

Session Initiation Protocol
- Request-Line: SUBSCRIBE sip:2010@172.16.1.30:5060 SIP/2.0
  - Method: SUBSCRIBE
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.20:5080;branch=z9hG4bK-16ea501142114-0
    - Transport: UDP
    - Sent-by Address: 172.16.1.20 (Originating IP Address of the NEC SIP Server)
    - Sent-by port: 5080 (Originating Port of the NEC SIP Server)
    - Branch: z9hG4bK-16ea501142114-0 (Transaction Identifier)
  - From: "2010" <sip:2010@172.16.1.20:5080>;tag=271c501142113-0 (Contains SIP IP Extension information)
    - SIP Display info: "2010"
    - SIP from address: sip:2010@172.16.1.20:5080
    - SIP tag: 271c501142113-0
  - To: <sip:2010@172.16.1.30:5060> (Contains SIP IP Extension information for subscription)
    - SIP to address: sip:2010@172.16.1.30:5060
  - Call-ID: aa7-50114211-2-0@172.16.1.20 (Call Identifier)
  - CSeq: 1 SUBSCRIBE
    - Sequence Number: 1
    - Method: SUBSCRIBE
  - Contact: <sip:2010@172.16.1.20:5080>
    - Contact Binding: <sip:2010@172.16.1.20:5080>
      - URI: <sip:2010@172.16.1.20:5080>
        - SIP contact address: sip:2010@172.16.1.20:5080
  - Max-Forwards: 70
  - Expires: 3613
  - Event: X-NECSIPEXT2CTRL (Event information allowing NEC SIP Station Control Messaging)
  - Content-Length: 0

5.5.1.4 200OK Response Message for SUBSCRIBE



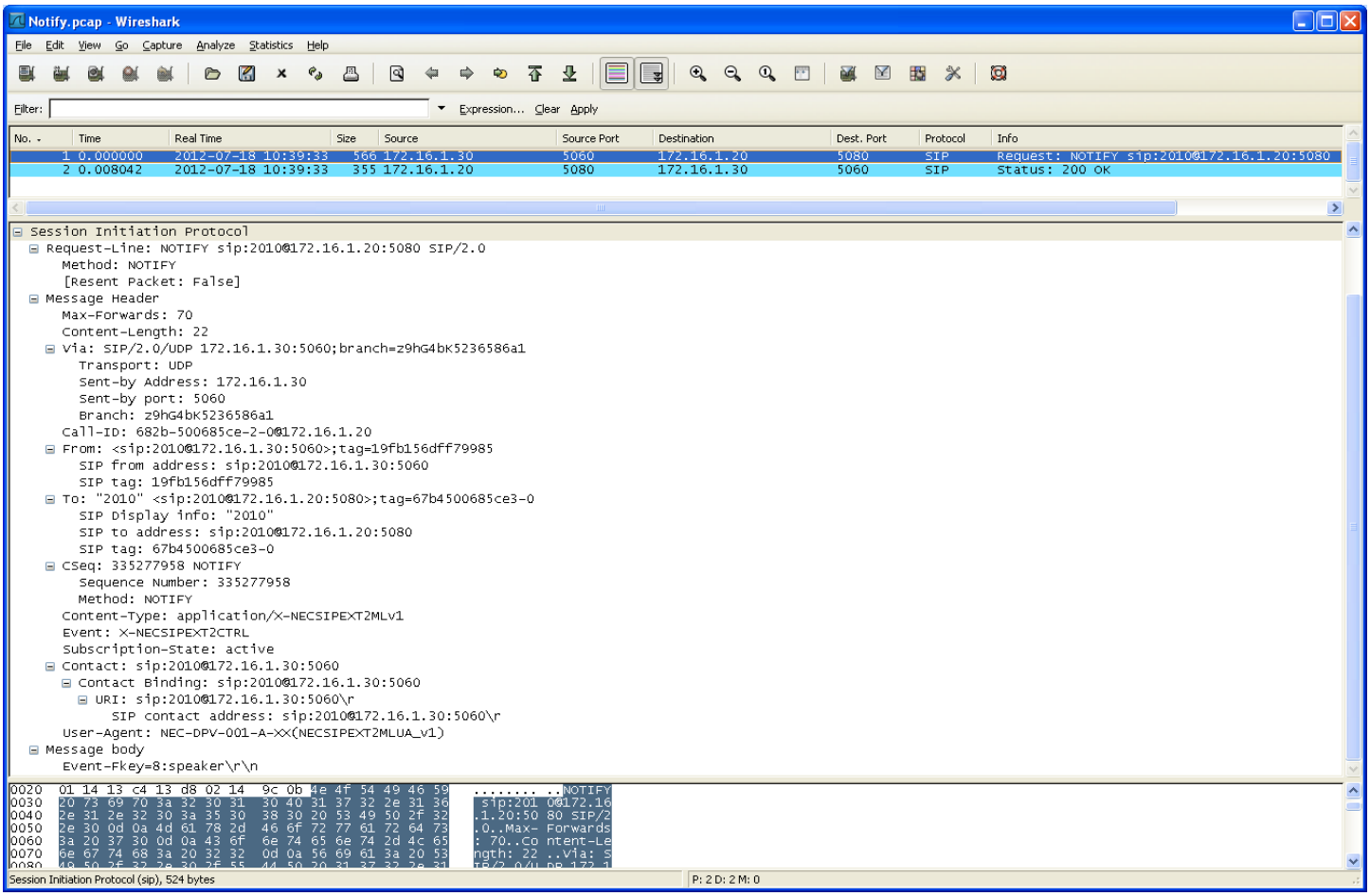
200OK Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 200 OK
  - Status-Code: 200
  - [Resent Packet: False]
- Message Header
  - Call-ID: aa7-50114211-2-0@172.16.1.20 (Same as SUBSCRIBE)
  - CSeq: 1 SUBSCRIBE
    - Sequence Number: 1
    - Method: SUBSCRIBE
  - From: "2010" <sip:2010@172.16.1.20:5080>;tag=271c501142113-0
    - SIP Display info: "2010"
    - SIP from address: sip:2010@172.16.1.20:5080
    - SIP tag: 271c501142113-0
  - To: <sip:2010@172.16.1.30:5060>;tag=0165ea5dd73ab4f
    - SIP to address: sip:2010@172.16.1.30:5060
    - SIP tag: 0165ea5dd73ab4f
  - Via: SIP/2.0/UDP 172.16.1.20:5080;branch=z9hG4bK-16ea501142114-0
    - Transport: UDP
    - Sent-by Address: 172.16.1.20
    - Sent-by port: 5080
    - Branch: z9hG4bK-16ea501142114-0
  - Content-Length: 0
  - Expires: 3613
  - User-Agent: NEC-DPV-001-A-XX(NECSIPEXT2MLUA_v1)

5.5.2 NOTIFY Request Message

NOTIFY messages are sent to inform subscribers of changes in state to which the subscriber has a subscription. A NOTIFY does not terminate its corresponding subscription; in other words, a single SUBSCRIBE request may trigger several NOTIFY requests. NOTIFY Messages will need a 200OK response.

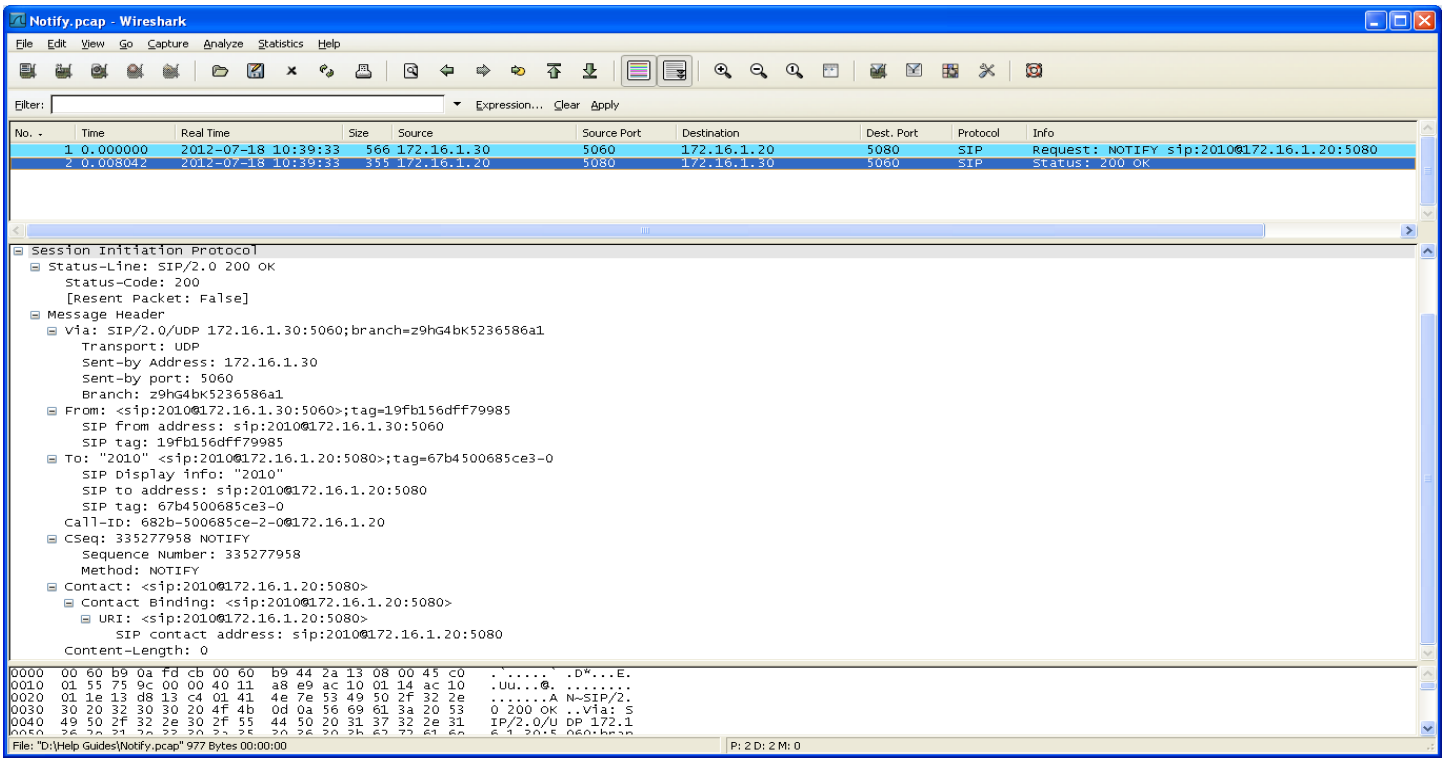
5.5.2.1 NOTIFY Request Message for Speaker Key Pressed



NOTIFY Request Message in Text Format

Session Initiation Protocol
  Request-Line: NOTIFY sip:2010@172.16.1.20:5080 SIP/2.0
    Method: NOTIFY
    [Resent Packet: False]
  Message Header
    Max-Forwards: 70
    Content-Length: 22
    Via: SIP/2.0/UDP 172.16.1.30:5060;branch=z9hG4bK5236586a1
      Transport: UDP
      Sent-by Address: 172.16.1.30
      Sent-by port: 5060
      Branch: z9hG4bK5236586a1
    Call-ID: 682b-500685ce-2-0@172.16.1.20 (Call Identifier)
    From: <sip:2010@172.16.1.30:5060>;tag=19fb156dff79985
      SIP from address: sip:2010@172.16.1.30:5060
      SIP tag: 19fb156dff79985
    To: "2010" <sip:2010@172.16.1.20:5080>;tag=67b4500685ce3-0
      SIP Display info: "2010"
      SIP to address: sip:2010@172.16.1.20:5080
      SIP tag: 67b4500685ce3-0
    CSeq: 335277958 NOTIFY
      Sequence Number: 335277958
      Method: NOTIFY
    Content-Type: application/X-NECSIPEXT2MLv1
    Event: X-NECSIPEXT2CTRL
    Subscription-State: active
    Contact: sip:2010@172.16.1.30:5060
      Contact Binding: sip:2010@172.16.1.30:5060
        URI: sip:2010@172.16.1.30:5060\r
          SIP contact address: sip:2010@172.16.1.30:5060\r
    User-Agent: NEC-DPV-001-A-XX(NECSIPEXT2MLUA_v1)
  Message body
    Event-Fkey=8:speaker\r\n (Event information of Speaker Key Press)

5.5.2.2 200OK Response Message for NOTIFY



200OK Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 200 OK
  - Status-Code: 200
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.30:5060;branch=z9hG4bK5236586a1
    - Transport: UDP
    - Sent-by Address: 172.16.1.30
    - Sent-by port: 5060
    - Branch: z9hG4bK5236586a1
  - From: <sip:2010@172.16.1.30:5060>;tag=19fb156dff79985
    - SIP from address: sip:2010@172.16.1.30:5060
    - SIP tag: 19fb156dff79985
  - To: "2010" <sip:2010@172.16.1.20:5080>;tag=67b4500685ce3-0
    - SIP Display info: "2010"
    - SIP to address: sip:2010@172.16.1.20:5080
    - SIP tag: 67b4500685ce3-0
  - Call-ID: 682b-500685ce-2-0@172.16.1.20 (Same as Notify)
  - CSeq: 335277958 NOTIFY
    - Sequence Number: 335277958
    - Method: NOTIFY
  - Contact: <sip:2010@172.16.1.20:5080>
    - Contact Binding: <sip:2010@172.16.1.20:5080>
      - URI: <sip:2010@172.16.1.20:5080>
        - SIP contact address: sip:2010@172.16.1.20:5080
  - Content-Length: 0

Section 6 – SIP Call Flow

6.1 SIP Call Flow

SIP will use a 3 way handshake call setup using the INVITE/200OK with SDP/ACK as the 3 messages to setup a SIP Call. For Early Media calls, the handshake will be INVITE/180 RINGING with SDP/PRACK. In the examples below, the basic call flow and Early Media are shown. Notice that RTP will begin to flow after the INVITE/200OK with SDP/ACK of the normal call and after the INVITE/180 RINGING with SDP/PRACK of the Early Media call.

Diagram 4.0 of basic SIP Call Flow

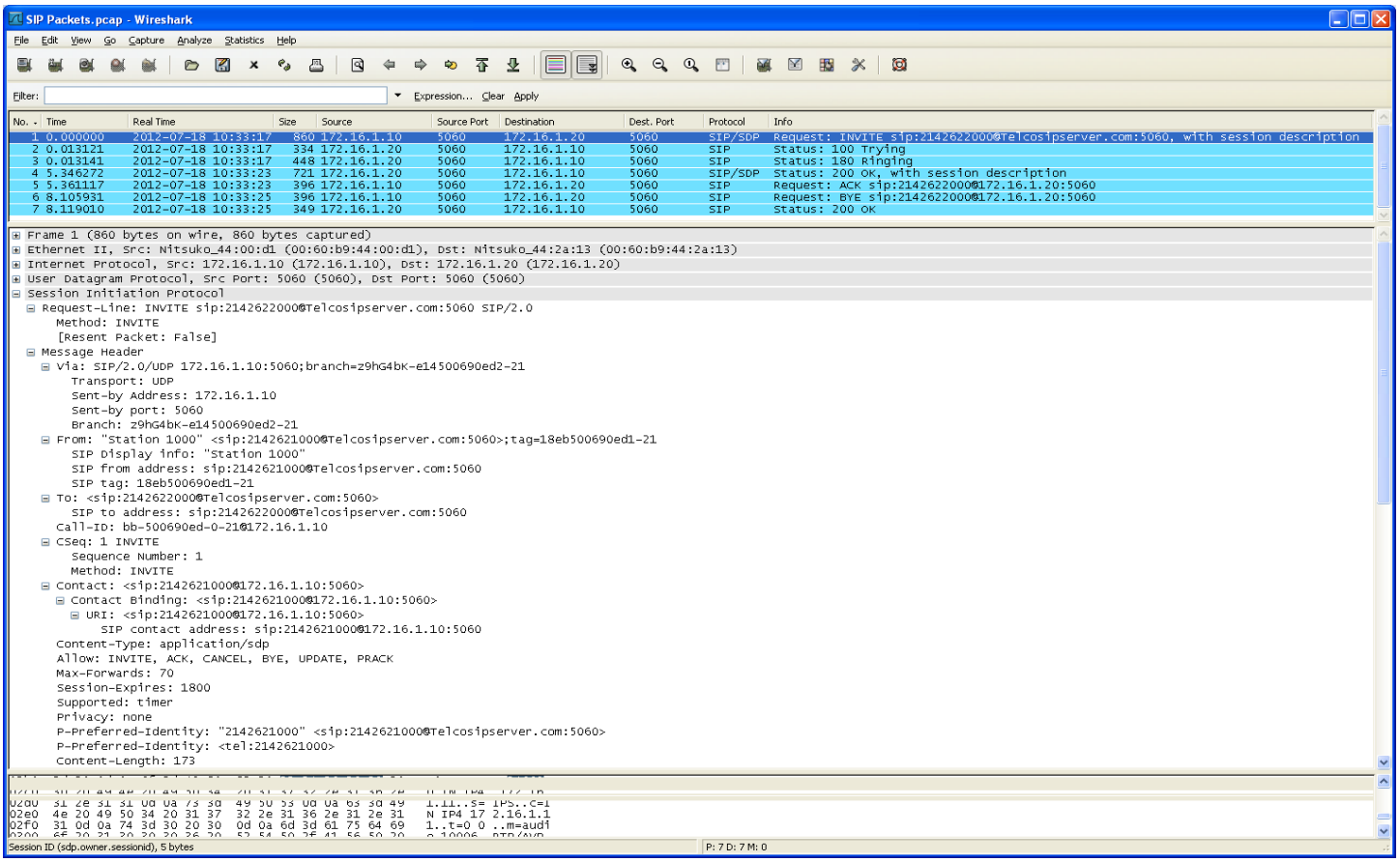
A normal call will flow as shown below:

Calling Side		**Called Side**
INVITE w/SDP	---------->	
	<----------	100 TRYING
	<----------	180 RINGING
	<----------	200OK w/SDP
ACK	---------->	
RTP	<---------->	RTP
BYE	---------->	
	<----------	200OK

An Early Media call will flow as shown below:

Calling Side		**Called Side**
INVITE w/SDP	---------->	
	<----------	100 TRYING
	<----------	180 RINGING w/SDP
PRACK	---------->	
RTP	<---------->	RTP
	<----------	200OK w/SDP
ACK	---------->	
BYE	---------->	
	<----------	200OK

6.2 INVITE Request Message

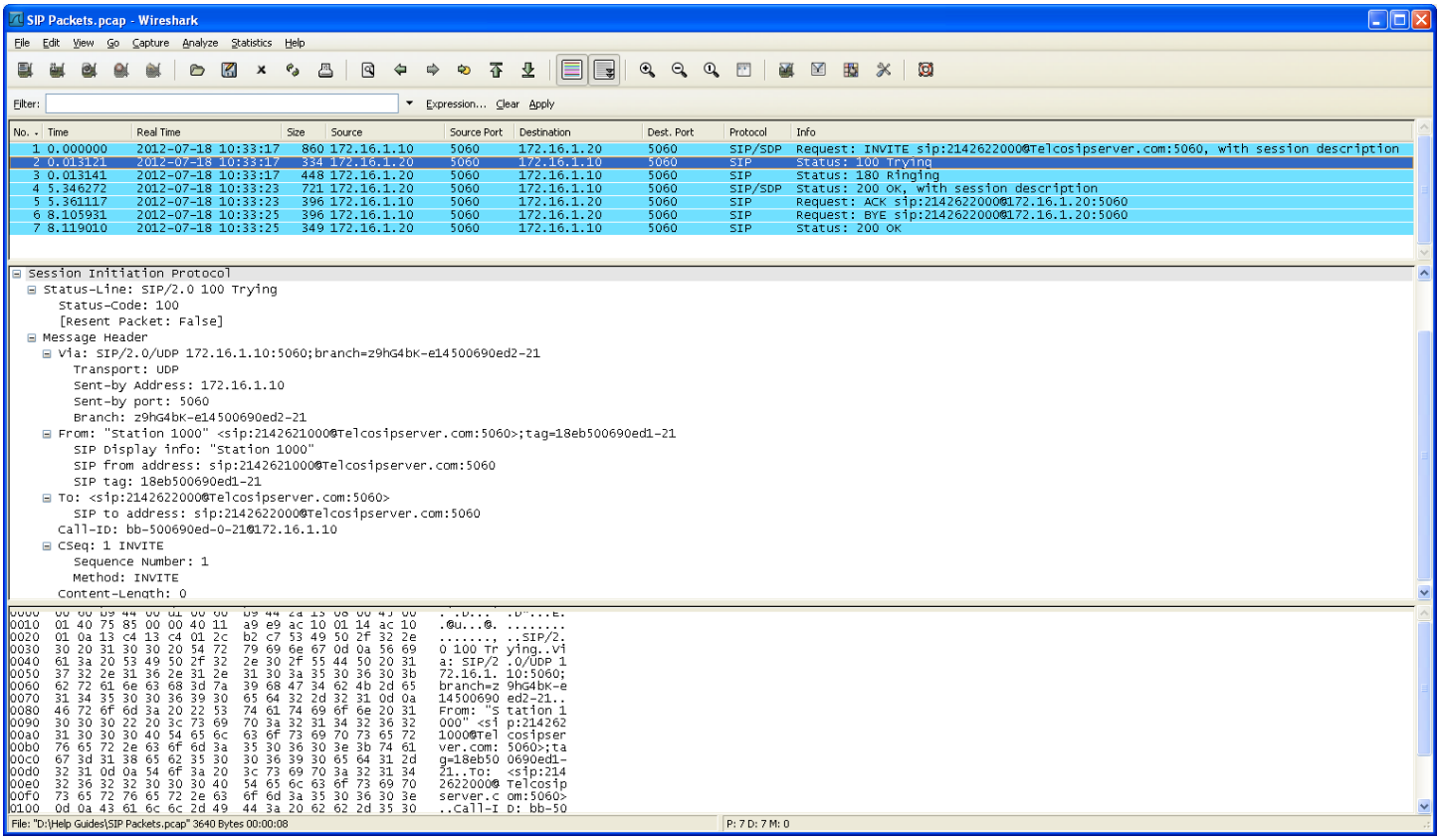


INVITE Request Message in Text Format

Session Initiation Protocol
Request-Line: INVITE sip:2142622000@Telcosipserver.com:5060 SIP/2.0 (Call to 214-262-2000)
Method: INVITE
[Resent Packet: False]
Message Header
Via: SIP/2.0/UDP 172.16.1.10:5060;branch=z9hG4bK-e14500690ed2-21
Transport: UDP
Sent-by Address: 172.16.1.10 (Originating IP Address)
Sent-by port: 5060 (Originating Port)
Branch: z9hG4bK-e14500690ed2-21 (Transaction Identifier)
From: "Station 1000" <sip:2142621000@Telcosipserver.com:5060>;tag=18eb500690ed1-21 (Calling party will generate a unique sip tag parameter in the From field and the called party will generate a sip tag in the To field of a 1XX response)
SIP Display info: "Station 1000" (Calling Name)
SIP from address: sip:2142621000@Telcosipserver.com:5060 (*Calling Number* 2142621000@*Domain Name* Telcosipserver.com: *Control Port* 5060) Note: Domain Name can be an IP Address.
SIP tag: 18eb500690ed1-21
To: <sip:2142622000@Telcosipserver.com:5060>
SIP to address: sip:2142622000@Telcosipserver.com:5060 (Called Number 214-262-2000)
Call-ID: bb-500690ed-0-21@172.16.1.10 (Call Identifier-Will remain the same for every control SIP packet of the call)
CSeq: 1 INVITE
Sequence Number: 1 (Identifying this message as the first sequence of the SIP Session)
Method: INVITE

Contact: <sip:2142621000@172.16.1.10:5060> (Contact information of the Calling Party)
Contact Binding: <sip:2142621000@172.16.1.10:5060>
URI: <sip:2142621000@172.16.1.10:5060>
SIP contact address: sip:2142621000@172.16.1.10:5060
Content-Type: application/sdp (Notification of SDP in the message)
Allow: INVITE, ACK, CANCEL, BYE, UPDATE, PRACK (Calling Party will allow these SIP Requests and Responses)
Max-Forwards: 70 (Max Forwards of 70 Hops)
Session-Expires: 1800 (Session will expire in 30 minutes)
Supported: timer (Allowing Session Timer)
Privacy: none (Allowing display of the Calling Party)
P-Preferred-Identity: "2142621000" <sip:2142621000@Telcosipserver.com:5060>
P-Preferred-Identity: <tel:2142621000> (Preferred Calling Party Display)
Content-Length: 173
Message body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): IPS 24669 0 IN IP4 172.16.1.11
Owner Username: IPS (Owner information of the Calling UAC)
Session ID: 24669
Session Version: 0
Owner Network Type: IN
Owner Address Type: IP4
Owner Address: 172.16.1.11 (Connection IP Address for the RTP)
Session Name (s): IPS
Connection Information (c): IN IP4 172.16.1.11
Connection Network Type: IN
Connection Address Type: IP4 (IP Version 4)
Connection Address: 172.16.1.11 (Connection IP Address for the RTP)
Time Description, active time (t): 0 0 (Stating there is no Session expiry for the RTP)
Session Start Time: 0
Session Stop Time: 0
Media Description, name and address (m): audio 10006 RTP/AVP 0 101
Media Type: audio
Media Port: 10006 (RTP Port of Calling Party)
Media Proto: RTP/AVP
Media Format: ITU-T G.711 PCMU (First choice for RTP Codec)
Media Format: ITU-T G.729 (Second choice for RTP Codec)
Media Format: 101 (Requesting RFC 2833 Out of Band DTMF with 101 encoding)
Media Attribute (a): rtpmap:0 PCMU/8000 (First choice for RTP Codec G.711)
Media Attribute Fieldname: rtpmap
Media Format: 0
MIME Type: PCMU
Media Attribute (a): ptime:20 (First choice G.711 for Packet Size 20 milliseconds attribute)
Media Attribute Fieldname: ptime
Media Attribute Value: 20
Media Attribute (a): rtpmap:2 G726-32/8000 (Second choice for RTP Codec G.729)
Media Attribute Fieldname: rtpmap
Media Format: 2
MIME Type: G726-32
Media Attribute (a): ptime:30 (Second choice G.729 for Packet Size 30 milliseconds attribute)
Media Attribute Fieldname: ptime
Media Attribute Value: 30
Media Attribute (a): rtpmap:101 telephone-event/8000 (First choice for DTMF Out of Band)
Media Attribute Fieldname: rtpmap
Media Format: 101 (Encoding 101 for Line Lockout Tone)
MIME Type: telephone-event

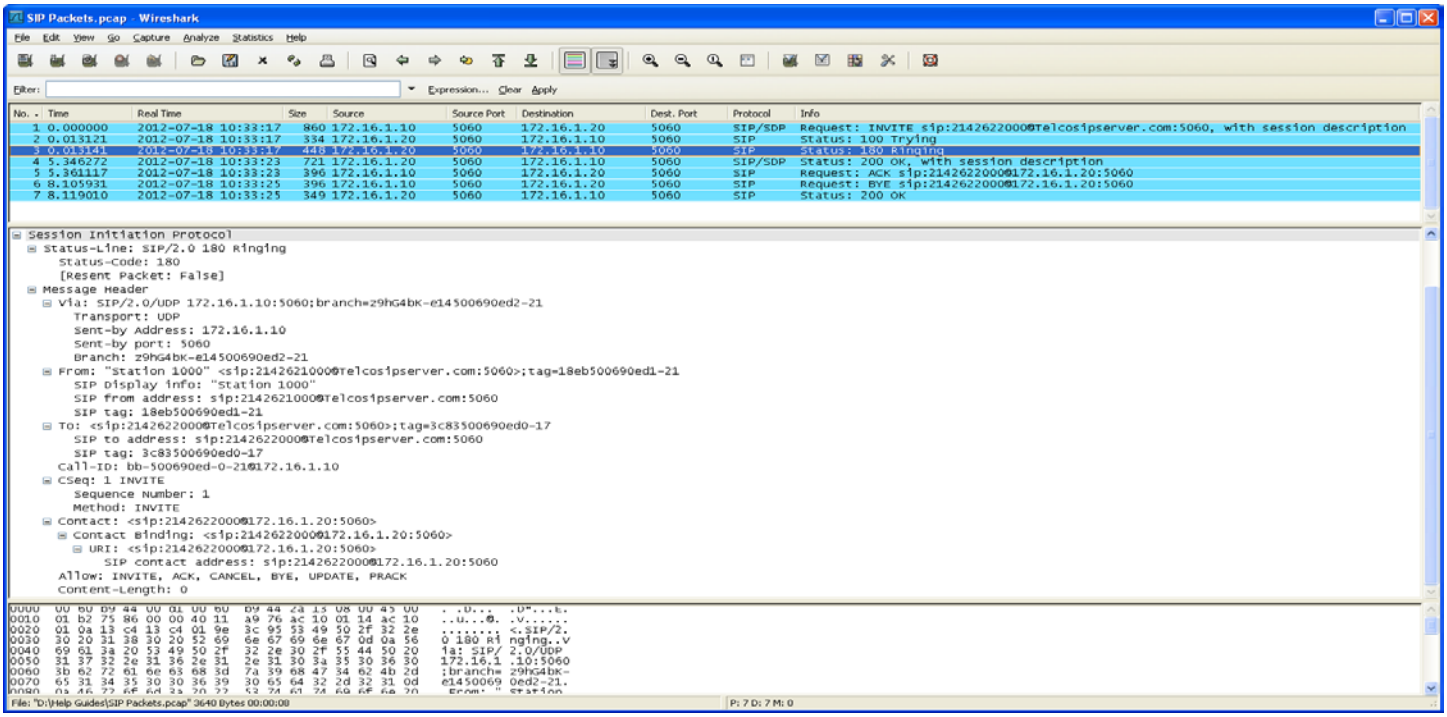
6.3 100 TRYING Response Message



100 TRYING Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 100 Trying
  - Status-Code: 100
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.10:5060;branch=z9hG4bK-e14500690ed2-21 (Same as INVITE)
    - Transport: UDP
    - Sent-by Address: 172.16.1.10
    - Sent-by port: 5060
    - Branch: z9hG4bK-e14500690ed2-21
  - From: "Station 1000" <sip:2142621000@Telcosipserver.com:5060>;tag=18eb500690ed1-21 (Same as INVITE)
    - SIP Display info: "Station 1000"
    - SIP from address: sip:2142621000@Telcosipserver.com:5060
    - SIP tag: 18eb500690ed1-21
  - To: <sip:2142622000@Telcosipserver.com:5060> (Same as INVITE)
    - SIP to address: sip:2142622000@Telcosipserver.com:5060
  - Call-ID: bb-500690ed-0-21@172.16.1.10 (Same as INVITE)
  - CSeq: 1 INVITE
    - Sequence Number: 1 (Identifying this message as a response to the INVITE of the SIP Session)
    - Method: INVITE
  - Content-Length: 0

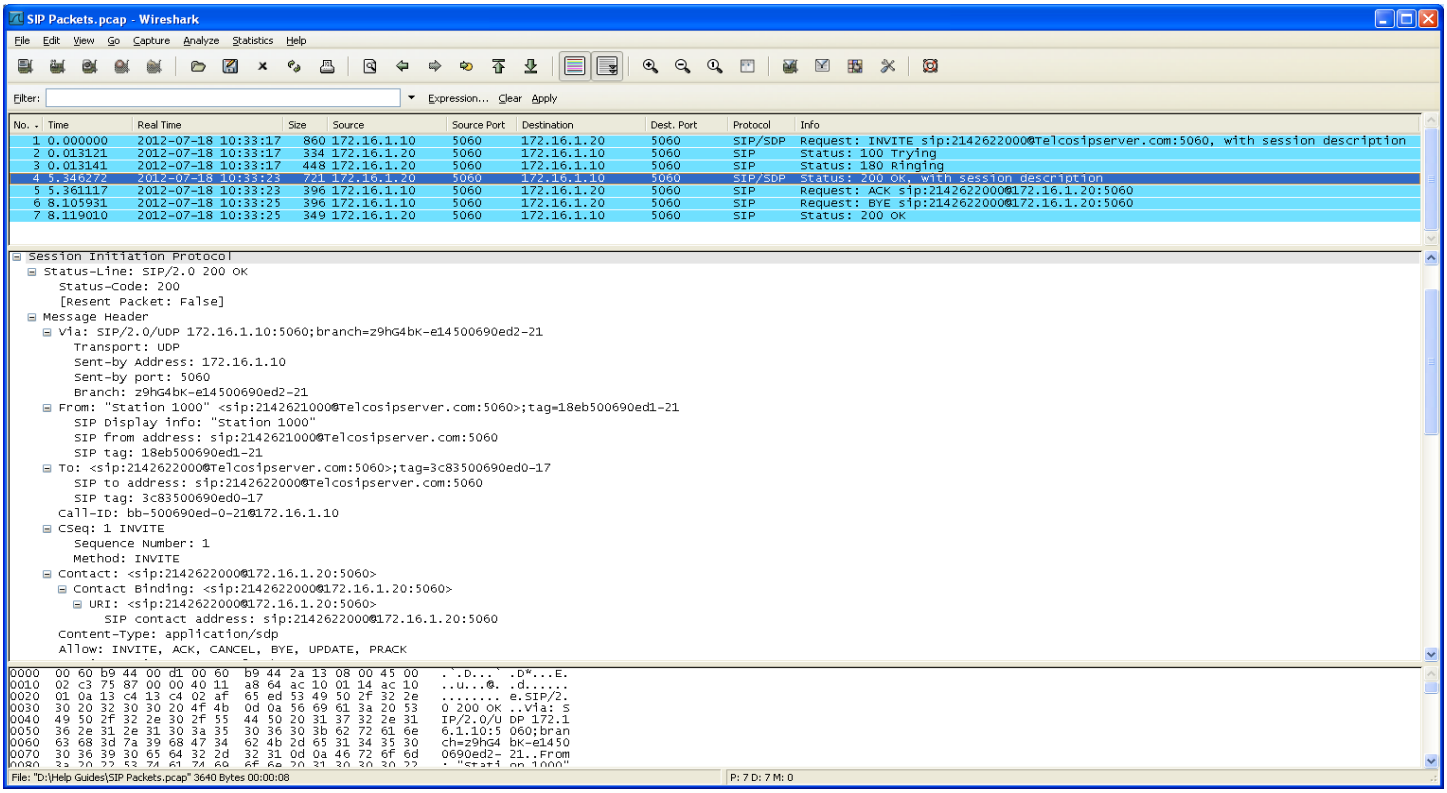
6.4 180 RINGING Response Message



180 RINGING Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 180 Ringing
  - Status-Code: 180
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.10:5060;branch=z9hG4bK-e14500690ed2-21 (Same as INVITE)
    - Transport: UDP
    - Sent-by Address: 172.16.1.10
    - Sent-by port: 5060
    - Branch: z9hG4bK-e14500690ed2-21
  - From: "Station 1000" <sip:2142621000@Telcosipserver.com:5060>;tag=18eb500690ed1-21 (Same as INVITE)
    - SIP Display info: "Station 1000"
    - SIP from address: sip:2142621000@Telcosipserver.com:5060
    - SIP tag: 18eb500690ed1-21
  - To: <sip:2142622000@Telcosipserver.com:5060>;tag=3c83500690ed0-17 (As with the INVITE the called UA now generates a unique sip tag parameter)
    - SIP to address: sip:2142622000@Telcosipserver.com:5060
    - SIP tag: 3c83500690ed0-17
  - Call-ID: bb-500690ed-0-21@172.16.1.10 (Same as INVITE)
  - CSeq: 1 INVITE
    - Sequence Number: 1 (Identifying this message as a response to the INVITE of the SIP Session)
    - Method: INVITE
  - Contact: <sip:2142622000@172.16.1.20:5060>
    - Contact Binding: <sip:2142622000@172.16.1.20:5060>
      - URI: <sip:2142622000@172.16.1.20:5060>
        - SIP contact address: sip:2142622000@172.16.1.20:5060
  - Allow: INVITE, ACK, CANCEL, BYE, UPDATE, PRACK
  - Content-Length: 0

6.5 200OK Response Message

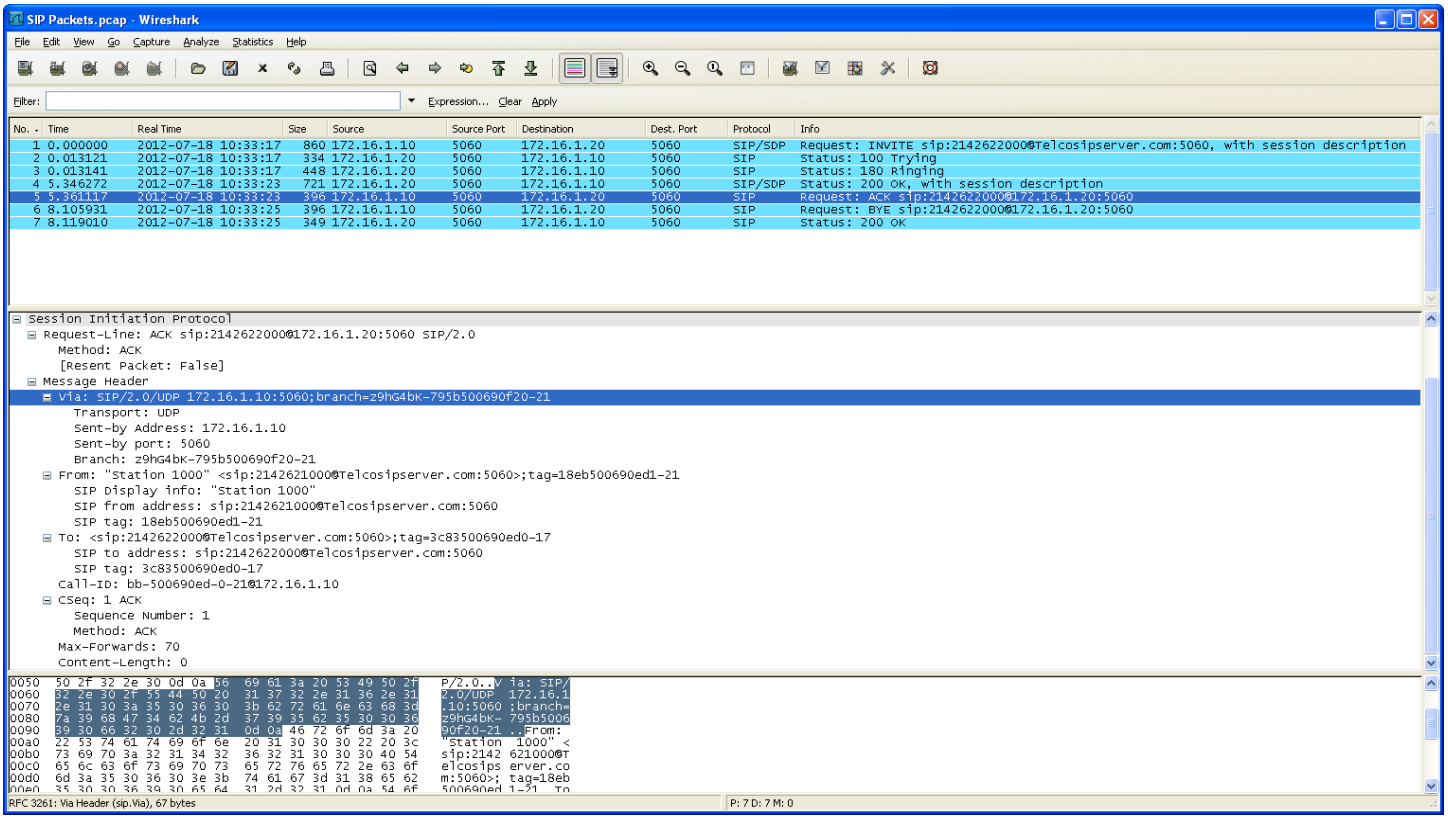


200OK Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 200 OK
  - Status-Code: 200
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.10:5060;branch=z9hG4bK-e14500690ed2-21 (Same as INVITE)
    - Transport: UDP
    - Sent-by Address: 172.16.1.10
    - Sent-by port: 5060
    - Branch: z9hG4bK-e14500690ed2-21
  - From: "Station 1000" <sip:2142621000@Telcosipserver.com:5060>;tag=18eb500690ed1-21 (Same as INVITE)
    - SIP Display info: "Station 1000"
    - SIP from address: sip:2142621000@Telcosipserver.com:5060
    - SIP tag: 18eb500690ed1-21
  - To: <sip:2142622000@Telcosipserver.com:5060>;tag=3c83500690ed0-17 (Same as the 180 RINGING)
    - SIP to address: sip:2142622000@Telcosipserver.com:5060
    - SIP tag: 3c83500690ed0-17
  - Call-ID: bb-500690ed-0-21@172.16.1.10 (Same as INVITE)
  - CSeq: 1 INVITE
    - Sequence Number: 1 (Identifying this message as a response to the INVITE of the SIP Session)
    - Method: INVITE
  - Contact: <sip:2142622000@172.16.1.20:5060>
    - Contact Binding: <sip:2142622000@172.16.1.20:5060>
      - URI: <sip:2142622000@172.16.1.20:5060>
        - SIP contact address: sip:2142622000@172.16.1.20:5060
  - Content-Type: application/sdp (Notification of SDP in the message)

Allow: INVITE, ACK, CANCEL, BYE, UPDATE, PRACK (Called Party will allow these SIP Requests and Responses)
Session-Expires: 1800;refresher=uac (Session will expire in 30 minutes and the UAC is responsible for refresh)
Supported: timer (Allowing Session Timer)
Require: timer (Requiring Session Timer)
Content-Length: 172
Message body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): IPS 5927 0 IN IP4 172.16.1.21
Owner Username: IPS
Session ID: 5927
Session Version: 0
Owner Network Type: IN
Owner Address Type: IP4
Owner Address: 172.16.1.21
Session Name (s): IPS
Connection Information (c): IN IP4 172.16.1.21
Connection Network Type: IN
Connection Address Type: IP4
Connection Address: 172.16.1.21 (Connection IP Address for the RTP)
Time Description, active time (t): 0 0
Session Start Time: 0
Session Stop Time: 0
Media Description, name and address (m): audio 10008 RTP/AVP 0 101
Media Type: audio
Media Port: 10008 (Connection Port for the RTP. INVITE stated Media Port 10008 so RTP will flow between these ports)
Media Proto: RTP/AVP
Media Format: ITU-T G.711 PCMU (Confirming G.711 will be the RTP Codec)
Media Format: 101 (Confirming Out of DTMF RFC2833 will be the DTMF Codec)
Media Attribute (a): rtpmap:0 PCMU/8000 (Confirming G.711 will be the RTP Codec)
Media Attribute Fieldname: rtpmap
Media Format: 0
MIME Type: PCMU
Media Attribute (a): ptime:20 (Confirming the payload will be 20ms packets)
Media Attribute Fieldname: ptime
Media Attribute Value: 20
Media Attribute (a): rtpmap:101 telephone-event/8000
Media Attribute Fieldname: rtpmap
Media Format: 101
MIME Type: telephone-event

6.6 ACK Request Message

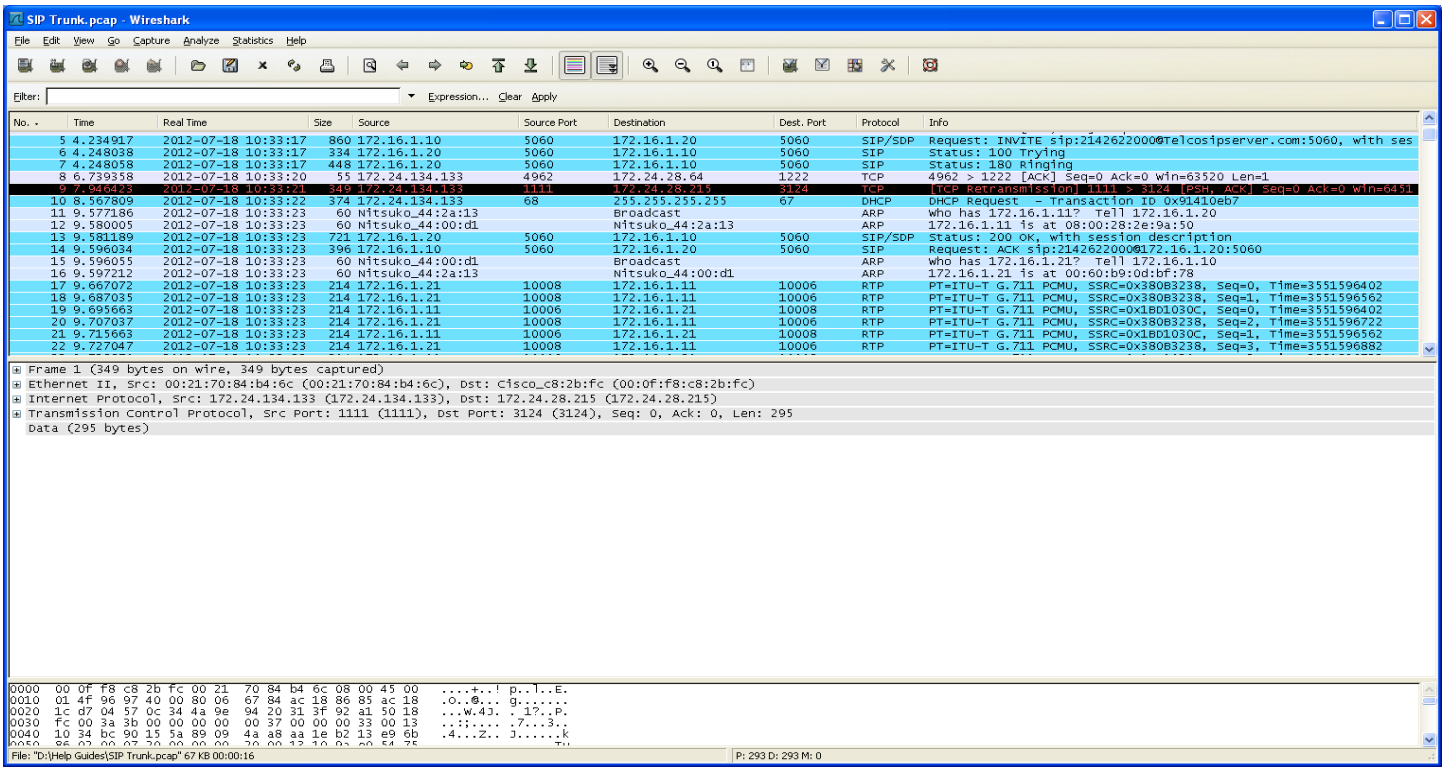


ACK Request Message in Text Format

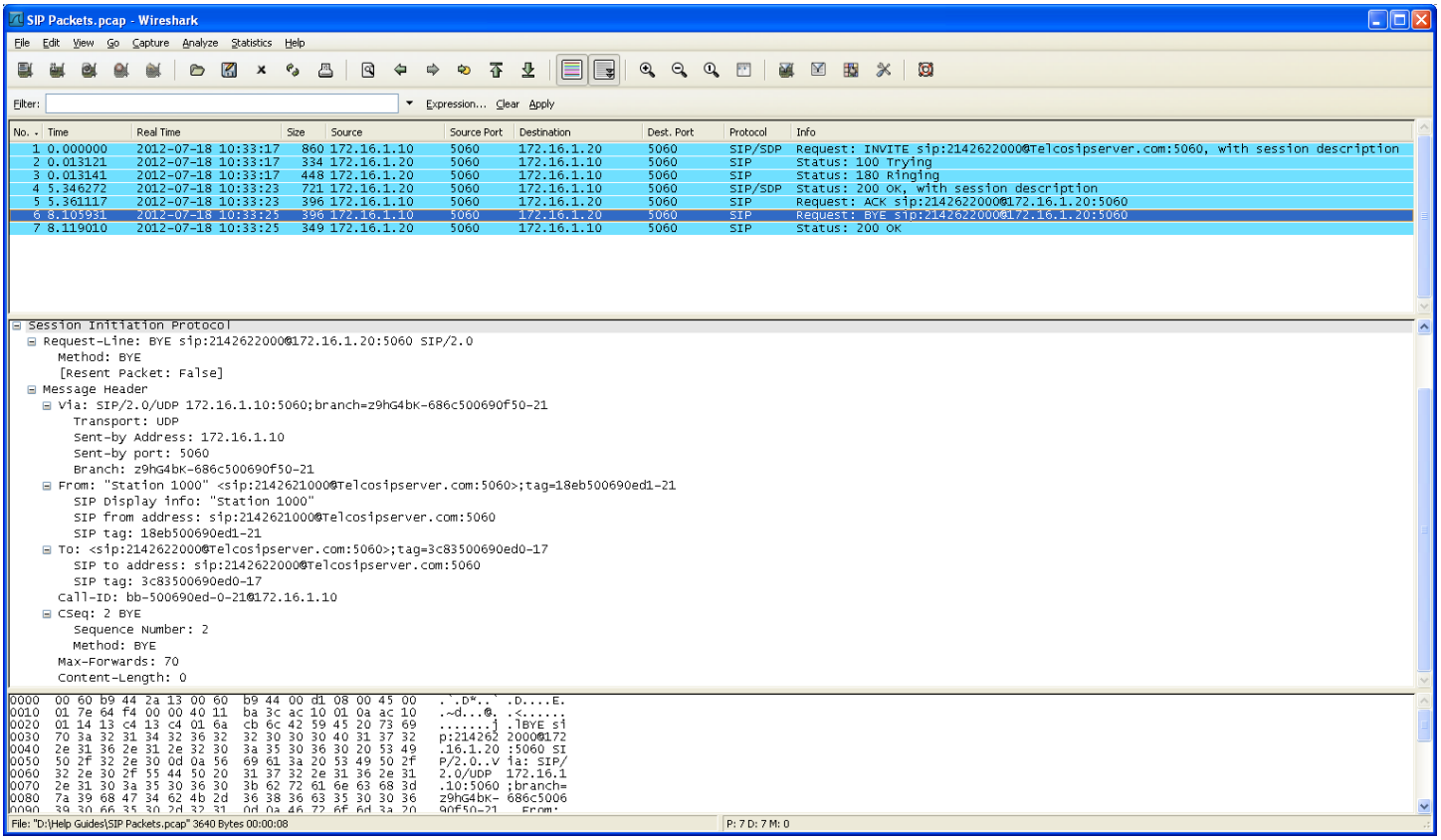
Session Initiation Protocol
- Request-Line: ACK sip:2142622000@172.16.1.20:5060 SIP/2.0 (Same as INVITE)
  - Method: ACK
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.10:5060;branch=z9hG4bK-795b500690f20-21 (New unique ID)
    - Transport: UDP
    - Sent-by Address: 172.16.1.10
    - Sent-by port: 5060
    - Branch: z9hG4bK-795b500690f20-21
  - From: "Station 1000" <sip:2142621000@Telcosipserver.com:5060>;tag=18eb500690ed1-21 (Same as INVITE)
    - SIP Display info: "Station 1000"
    - SIP from address: sip:2142621000@Telcosipserver.com:5060
    - SIP tag: 18eb500690ed1-21
  - To: <sip:2142622000@Telcosipserver.com:5060>;tag=3c83500690ed0-17 (Same as the 180 RINGING & 200OK)
    - SIP to address: sip:2142622000@Telcosipserver.com:5060
    - SIP tag: 3c83500690ed0-17
  - Call-ID: bb-500690ed-0-21@172.16.1.10 (Same as INVITE)
  - CSeq: 1 ACK
    - Sequence Number: 1 (The sequence number is still 1 but is now for the ACK)
    - Method: ACK
  - Max-Forwards: 70
  - Content-Length: 0

6.7 RTP Communication

In the screenshot below, a Wireshark capture is displaying the SIP Setup and RTP communication between IP Address 172.16.1.11 Port 10006 and IP Address 172.16.1.21 Port 10008 after receiving the ACK Request.



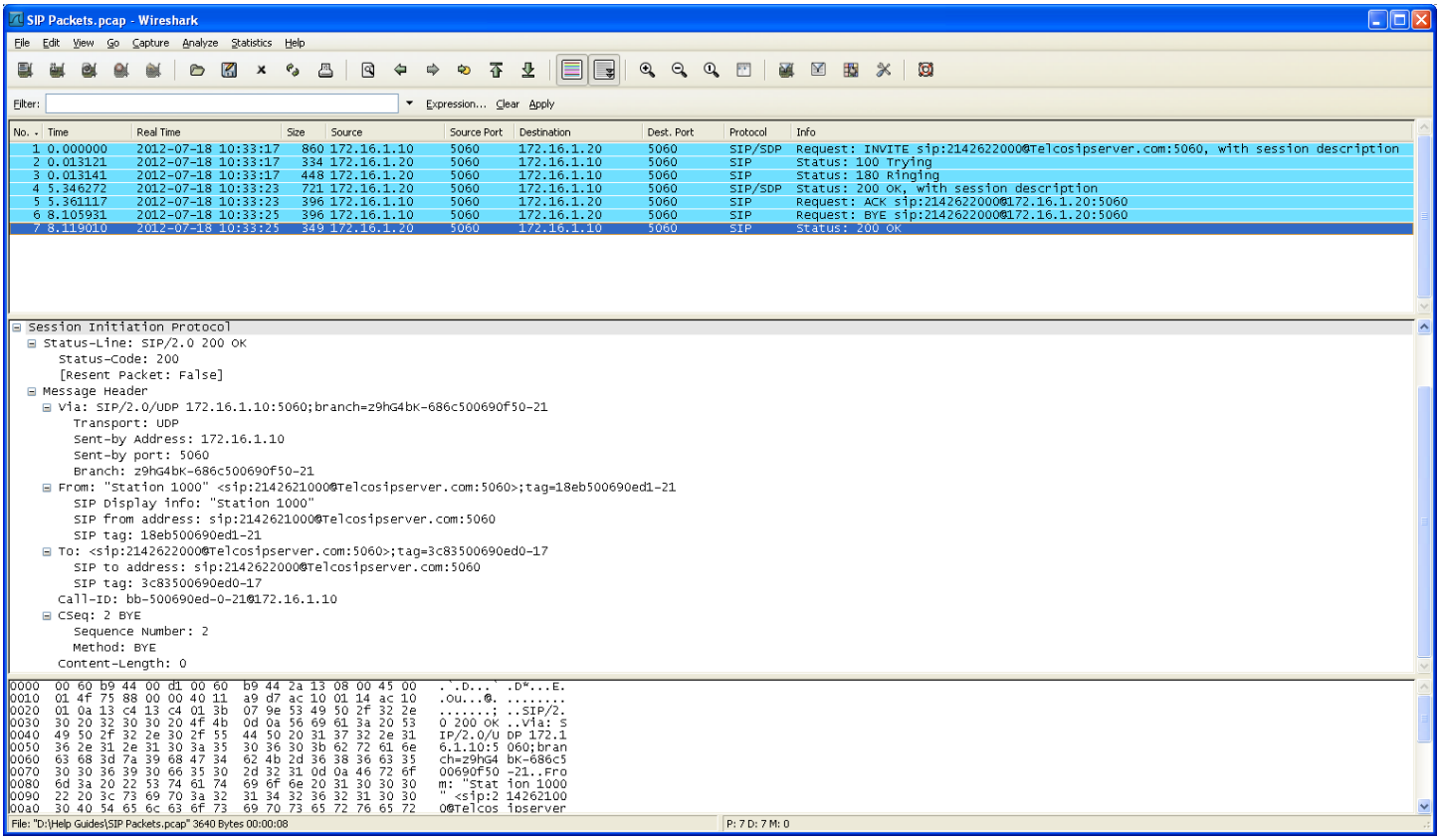
6.8 BYE Request Message



BYE Request Message in Text Format

Session Initiation Protocol
- Request-Line: BYE sip:2142622000@172.16.1.20:5060 SIP/2.0 (Same as INVITE)
  - Method: BYE
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.10:5060;branch=z9hG4bK-686c500690f50-21 (New unique ID)
    - Transport: UDP
    - Sent-by Address: 172.16.1.10
    - Sent-by port: 5060
    - Branch: z9hG4bK-686c500690f50-21
  - From: "Station 1000" <sip:2142621000@Telcosipserver.com:5060>;tag=18eb500690ed1-21 (Same as INVITE)
    - SIP Display info: "Station 1000"
    - SIP from address: sip:2142621000@Telcosipserver.com:5060
    - SIP tag: 18eb500690ed1-21
  - To: <sip:2142622000@Telcosipserver.com:5060>;tag=3c83500690ed0-17 (Same as 180 RINGING and 200 OK)
    - SIP to address: sip:2142622000@Telcosipserver.com:5060
    - SIP tag: 3c83500690ed0-17
  - Call-ID: bb-500690ed-0-21@172.16.1.10 (Same as INVITE)
  - CSeq: 2 BYE
    - Sequence Number: 2 (The sequence number is now 2 since it is the second Request message after the ACK)
    - Method: BYE
  - Max-Forwards: 70
  - Content-Length: 0

6.9 200OK Response Message



200OK Response Message in Text Format

Session Initiation Protocol
- Status-Line: SIP/2.0 200 OK
  - Status-Code: 200
  - [Resent Packet: False]
- Message Header
  - Via: SIP/2.0/UDP 172.16.1.10:5060;branch=z9hG4bK-686c500690f50-21 (Same as BYE)
    - Transport: UDP
    - Sent-by Address: 172.16.1.10
    - Sent-by port: 5060
    - Branch: z9hG4bK-686c500690f50-21
  - From: "Station 1000" <sip:2142621000@Telcosipserver.com:5060>;tag=18eb500690ed1-21 (Same as INVITE)
    - SIP Display info: "Station 1000"
    - SIP from address: sip:2142621000@Telcosipserver.com:5060
    - SIP tag: 18eb500690ed1-21
  - To: <sip:2142622000@Telcosipserver.com:5060>;tag=3c83500690ed0-17 (Same as 180 RINGING and 200OK)
    - SIP to address: sip:2142622000@Telcosipserver.com:5060
    - SIP tag: 3c83500690ed0-17
  - Call-ID: bb-500690ed-0-21@172.16.1.10 (Same as INVITE)
  - CSeq: 2 BYE
    - Sequence Number: 2 (Identified as an acknowledgement to the BYE)
    - Method: BYE
  - Content-Length: 0