ATIS STANDARD

# ATIS-1000032

## SIP USER-NETWORK INTERFACE TESTING FRAMEWORK

TECHNICAL REPORT

ATIS is the leading technical planning and standards development organization committed to the rapid development of global, market-driven standards for the information, entertainment and communications industry. More than 200 companies actively formulate standards in ATIS' Committees, covering issues including: IPTV, Cloud Services, Energy Efficiency, IP-Based and Wireless Technologies, Quality of Service, Billing and Operational Support, Emergency Services, Architectural Platforms and Emerging Networks. In addition, numerous Incubators, Focus and Exploratory Groups address evolving industry priorities including Smart Grid, Machine-to-Machine, Networked Car, IP Downloadable Security, Policy Management and Network Optimization.

ATIS is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), a member and major U.S. contributor to the International Telecommunication Union (ITU) Radio and Telecommunications' Sectors, and a member of the Inter-American Telecommunication Commission (CITEL). ATIS is accredited by the American National Standards Institute (ANSI). For more information, please visit < http://www.atis.org >.

## Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. ATIS SHALL NOT BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY ATIS FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL ATIS BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. ATIS EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

NOTE - The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to whether use of an invention covered by patent rights will be required, and if any such use is required no position is taken regarding the validity of this claim or any patent rights in connection therewith.

ATIS-1000032, *SIP User-Network Interface Testing Framework*

Is an ATIS Standard developed by the **Signalling, Architecture, and Control (SAC) Subcommittee** under the **ATIS Packet Technologies and Systems Committee (PTSC)**.

*Published by*

**Alliance for Telecommunications Industry Solutions**
**1200 G Street, NW, Suite 500**
**Washington, DC 20005**

Printed in the United States of America.

**ATIS-1000032**

Technical Report on

# SIP User-Network Interface Testing Framework

**Alliance for Telecommunications Industry Solutions**

Approved August 2008

**Abstract**
This Technical Report (TR) describes a framework for testing User-Network IP interconnection for VoIP.

## Foreword

The Alliance for Telecommunication Industry Solutions (ATIS) serves the public through improved understanding between providers, customers, and manufacturers. The Packet Technologies and Systems Committee (PTSC) develops and recommends standards and technical reports related to services, architectures, and signaling, in addition to related subjects under consideration in other North American and international standards bodies. PTSC coordinates and develops standards and technical reports relevant to telecommunications networks in the U.S., reviews and prepares contributions on such matters for submission to U.S. ITU-T and U.S. ITU-R Study Groups or other standards organizations, and reviews for acceptability or per contra the positions of other countries in related standards development and takes or recommends appropriate actions.

The mandatory requirements are designated by the word *shall* and recommendations by the word *should*. Where both a mandatory requirement and a recommendation are specified for the same criterion, the recommendation represents a goal currently identifiable as having distinct compatibility or performance advantages. The word *may* denotes a optional capability that could augment the standard. The standard is fully functional without the incorporation of this optional capability.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, PTSC, 1200 G Street NW, Suite 500, Washington, DC 20005.

At the time of consensus on this document, PTSC, which was responsible for its development, had the following roster:

J. Zebarth, PTSC Chair (Nortel)
G. Munson, Technical Editor (AT&T)
C. Underkoffler, ATIS Chief Editor

The **SAC** Subcommittee was responsible for the development of this document.

**Table of Contents**

**1 Introduction .......... 5**
1.1 Types of 'Users' .......... 6
**2 Definitions .......... 6**
**3 Abbreviations .......... 6**
**4 Physical and Data Link Layers .......... 7**
**5 Network Layer .......... 7**
5.1 IP Addresses .......... 7
5.2 Priority Differentiation .......... 7
5.3 IP Payload Protocol Indication .......... 7
**6 Signaling Transport .......... 7**
6.1 UDP .......... 7
6.2 TCP .......... 8
6.3 TLS over TCP .......... 8
**7 Configuration .......... 8**
7.1 Automatic IP Address Acquisition .......... 8
7.2 Locating a SIP Server .......... 9
7.2.1 Transport Protocol Discovery .......... 9
7.2.2 SIP Server IP Address / Port Number Discovery .......... 9
7.2.3 Relationship of "Outbound Proxy" to Locating a SIP Server .......... 10
7.3 Time Acquisition .......... 10
**8 SIP Address of Record Registration .......... 10**
8.1 Initial Registration .......... 10
8.2 Other Registration Scenarios .......... 11
8.2.1 Registration Update .......... 11
8.2.2 Request for Current Contact List .......... 11
8.2.3 Registration Cancellation .......... 11
**9 Call Control/Session Establishment .......... 11**
9.1 SIP - Call Stimulus Scenarios .......... 11
9.2 Media - Call Stimulus Scenarios .......... 21
**10 SIP OPTIONS .......... 24**
**11 Presence .......... 25**
11.1 A Watcher Wants Information about a Presentity .......... 25
11.2 A Watcher Wants Information about the Set of Watchers of a Presentity .......... 26
11.3 Additional Observations .......... 26
11.4 Presentity Application in an Enterprise Network on the User Side of the UNI .......... 27
**12 Instant Messaging .......... 27**
12.1 Page Mode .......... 27
12.2 Session Mode .......... 28
12.3 isComposing Status Messages .......... 29
**13 Conference Control .......... 30**
13.1 Conference Control .......... 30
13.2 Subscription to Conference State .......... 32
**14 Network Address Translator (NAT) .......... 33**
14.1 Simple Traversal of UDP over NATs (STUN) .......... 33
14.2 Interactive Connectivity Establishment (ICE) .......... 34
**15 SIP Message Boundary Testing .......... 35**
**16 Testing Organization .......... 35**
**17 Security .......... 35**
17.1 Encryption .......... 35
17.2 Authentication of SIP Requests .......... 36
**18 References .......... 37**

## Table of Figures

Figure 1 - Basic Presence System ........ 25
Figure 2 - Presence Application in Enterprise Network........ 27

## Table of Tables

Table 1 - Call Stimulus Scenarios for SIP ........ 12
Table 2 - Call Stimulus Scenarios for Media ........ 22
Table 3- Call Stimulus Scenarios for SIP - Typically for Conferencing ........ 30

Technical Report on –

# SIP User-Network Interface Testing Framework

## 1 Introduction

This Technical Report (TR) describes a framework for testing User-Network IP interconnection for VoIP.

The testing framework described herein is for the SIP user-network interface (UNI) as specified in the ATIS American National Standard Interoperable SIP UNI Specification. [1] That document is referred to in this TR as the SIP UNI ANS.

A common testing framework has the benefits of

1) re-use, avoiding the inefficiency of pair-wise re-creation by network operators and customer equipment (CE) vendors
2) industry-wide input
3) being a common reference for understanding test results (e.g., when network A tests with customer equipment B and then with customer equipment C)

This testing framework

- addresses pair-wise network-CE interoperability according to the SIP UNI ANS, rather than equipment or vendor testing or certification, or protocol standards conformance
- is agnostic regarding the particulars of test environments and configurations – e.g., it could be with a live network and commercial customer equipment or with a network-emulating lab and beta customer equipment
- does not assume any details about internal network architectures or systems
- describes the kinds of things to be tested, but does not provide explicit, enumerated test cases; and also does not specify particular testing methods

This testing framework does not address

- identification of the responsible organization performing the testing
- testing for services; however, services may be used as stimuli to invoke aspects of protocols that are to be tested
- load testing
- simultaneous testing of multiple UNIs between a network and customer equipment. This TR addresses testing of a single UNI at a time.
- failure scenarios
- test cases that intentionally result in a protocol timeout

Successful interoperability testing should be verifiable strictly by observing what happens across the UNI. Test generation and some verification methods may pragmatically involve the placement of actual telephone calls and observing results from that user point of view.

Before defining specific test plans, the network operator and CE operator involved shall agree on what subset of the SIP UNI ANS they are implementing. Testing could follow various models. For example:

a. Conduct a full suite of tests exercising the entire interface as defined in the SIP UNI ANS, and verify which pass. An advantage of this approach is that it can discover capabilities supported across the interface that may not otherwise have been known to or agreed to by the network operator and CE operator.
b. Conduct a test suite that exercises a *subset* of the entire interface in the SIP UNI ANS, where that subset is limited to aspects of the SIP UNI ANS that are agreed to by the network operator and CE operator. An advantage of this approach is that it may involve

less testing. The subset may depend to some degree, for example, on the nature of the IP network or CE.

The exercise of the testing guidelines and call stimulus scenarios contained in this document are intended to have no impact on the integrity or reliability of existing services being supported in a live network.

### 1.1 Types of 'Users'

The SIP UNI ANS describes two SIP profiles, one for each of the basic 'User' types of the UNI:

- *Device*, where the devices can be, e.g., SIP phones, software applications that run on PCs and provide a SIP UA interface, or analog terminal adaptors that provide a SIP UA interface,
- *Enterprise Network*

The difference between the two profiles in the SIP UNI ANS is in which protocol aspects are mandatory, as opposed to optional. Therefore, all of the material in this TR potentially pertains to both types.

## 2 Definitions

For protocol-specific terminology, the reader should consult the corresponding protocol specifications.

## 3 Abbreviations

| | |
|---|---|
| AIB | Authenticated Identity Body |
| ANS | American National Standard |
| APRI | Address Presentation Restricted Indicator |
| AS | Application Server |
| ATIS | Alliance for Telecommunications Industry Solutions |
| CE | Customer Equipment |
| CPIM | Common Presence and Instant Messaging |
| DHCP | Dynamic Host Configuration Protocol |
| DiffServ | Differentiated Service |
| DNS | Domain Name System |
| DTMF | Dual Tone Multi-Frequency |
| ETS | Emergency Telecommunications Service |
| FQDN | Fully Qualified Domain Name |
| IAM | Initial Address Message |
| ICE | Interactive Connectivity Establishment |
| IETF | Internet Engineering Task Force |
| IFP | Internet Fax Protocol |
| IP | Internet Protocol |
| ISDN | Integrated Services Digital Network |
| ISUP | ISDN User Part |
| ITU | International Telecommunication Union |
| LEC | Local Exchange Carrier |
| MIME | Multi-part Internet Mail Extensions |
| MS | Media Server |
| NAPTR | Naming Authority Pointer |
| NAT | Network Address Translator |
| NI | Network Interconnect |
| NP | Number Portability |
| OSI | Open Systems Interconnect |
| POTS | Plain Old Telephone Service |
| PSTN | Public Switched Telephone Network |
| RFC | Request For Comments |
| RR | Resource Record |
| RTCP | Real-Time Control Protocol |
| RTP | Real-time Transport Protocol |
| SACK | Selective ACKknowledge |
| SCTP | Stream Control Transmission Protocol |
| SDP | Session Description Protocol |
| SIP | Session Initiation Protocol |
| SIPS | SIP-Secure |
| SLA | Service Level Agreement |
| S/MIME | Secure MIME |
| SNTP | Simple Network Time Protocol |
| SRV | Service Location |
| STUN | Simple Traversal of UDP over NATs |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| TR | Technical Report |
| TURN | Traversal Using Relay NAT |
| UDP | User Datagram Protocol |
| UNI | User-Network Interface |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| VoIP | Voice over IP |
| WPS | Wireless Priority Service |
| XR | Extended Reports |

# 4 Physical and Data Link Layers

At the physical and data link layers there is no distinctive treatment of SIP or media versus data, or among SIP or media traffic types, and thus nothing specific to test in regard to SIP or media. Tests should be conducted to verify the satisfactory operation of those layers, but since there is no distinction from treatment of data traffic, testing at those layers is not addressed in this document.

# 5 Network Layer

Aspects of the network layer that may involve treatment of SIP or media different from data are addressed in this TR.

## 5.1 IP Addresses

The use of the IP addresses and port numbers as required and agreed to by the network and/or CE operator should be verified. Note that IP addresses for different types of IP packet payload may be different; for example, IP addresses for UDP payload versus TCP payload, or IP addresses for SIP signaling payload over UDP versus RTP media payload over UDP.

## 5.2 Priority Differentiation

Traffic priority across the UNI may be differentiated at the IP packet level through the use of DiffServ code points. When employed, the use of such differentiation in DiffServ should be verified.

The SIP UNI may more generally be an IP UNI that carries other kinds of traffic besides voice calls; for example, video, web-browsing, email, file transfer or presence information. Those other kinds of traffic may also receive differentiated treatment.

## 5.3 IP Payload Protocol Indication

Proper indication within IP of the payload protocol should be verified (i.e., use of the Protocol field in IPv4 or the Next Header field in IPv6).

# 6 Signaling Transport

The SIP UNI ANS identifies the use of TCP, UDP, or TLS over TCP for SIP signaling transport.

In addition, TCP, TLS over TCP and especially UDP may be used for other rudimentary signaling-related activity across the UNI, as indicated in later sections in this TR; for examp
le, for initial IP address acquisition, DNS server location, time acquisition, or NAT traversal.

## 6.1 UDP

The use of UDP for SIP transport is the same as for other types of UDP payload. Therefore, there are no specific capabilities of UDP that need to be exercised for SIP payload. However, it should be verified that the intended UDP port numbers for SIP transport are actually used, if they differ from those for other types of UDP payload.

Likewise, there is nothing special to test for other uses of UDP transport in the SIP UNI ANS.

### 6.2 TCP

As with UDP, there is nothing special about the use of TCP for SIP signaling transport versus other kinds of TCP payload. However, unlike UDP, TCP involves the establishment of a connection. Therefore it should be verified that the specific TCP connection(s) for conveying SIP signaling are established, including the use of the intended TCP port numbers and the desired TCP connection characteristics of window size and maximum segment size.

Likewise, there is nothing special to test for other uses of TCP transport in the SIP UNI ANS.

### 6.3 TLS over TCP

TLS is an encryption method for security. See section 17.

## 7 Configuration

The primary references for this section are

- RFC 4504  SIP Telephony Device Requirements and Configuration
- RFC 3263 Session Initiation Protocol (SIP): Locating SIP Servers
- RFC 4330 Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI

### 7.1 Automatic IP Address Acquisition

A device needs to acquire an IP address for itself before it can do anything else across the UNI. DHCP is used for that purpose. DHCP uses the User Datagram Protocol (UDP), i.e., DHCP messages ride as payload in UDP datagrams.

The DHCP server may supply other information in addition to a device IP address, such as domain name, DNS server address, SNTP (time) server address, or TCP/IP subnet mask and default gateway.

1) The device sends a DHCPDISCOVER message that includes among other things

- UDP source address of 0.0.0.0
- UDP destination address typically 255.255.255.255
- the device MAC address
- a Magic Cookie value *0x63825363*

2) One or more DHCP servers may respond to the device with DHCPOFFER messages that includes

- UDP source address as the IP address of the DHCP server
- UDP destination address as the offered IP address that may be assigned to the device
- the device MAC address
- the same Magic Cookie value
- duration of the IP address lease
- other TCP/IP configuration information such as subnet mask and default gateway
- possibly other information such as domain name, DNS server addresses, SNTP server addresses

The device chooses an offer from one of the DHCP servers.

3) The device broadcasts a DHCPREQUEST message (that basically informs all the DHCP servers which offer the device is accepting) that includes

- UDP source address is 0.0.0.0
- UDP destination address is 255.255.255.255
- the device MAC address

- the same Magic Cookie value
- the IP address of the DHCP server whose offer the device is accepting
- the IP address that that DHCP server is offering

4) The DHCP server whose offer the device is accepting sends a DHCPACK message to the (IP address of) the device; the message content reconfirms what was in the original offer from that DHCP server.

### 7.2 Locating a SIP Server

It is assumed that the device has a SIP request that it wants to send to a resource identified by a SIP or SIPS URI (e.g., a called party). The device needs to locate a SIP server to which it can send the request. The device needs to discover an IP address and port number of such a SIP server, along with the transport protocol(s) the server supports. Once a device discovers a SIP server, it may remember that server's address, port number and transport protocol so that it does not have to re-discover that information for each subsequent SIP request pertaining to the same domain.[1] One or more SIP servers may be located.

The description here assumes the device has no knowledge of a SIP server protocol or address/port number. Discovery happens in two steps (a) transport protocol and then (b) address/port number. For both steps the device will utilize a DNS server (known, e.g., through the device's IP address acquisition by a DHCP query as described in Section 7.1 above.)

DNS queries may use either UDP or TCP for transport.

#### 7.2.1 Transport Protocol Discovery

1) The device sends a NAPTR-type DNS query[2] to the DNS server, where the key information contained is the "target" (in RFC 3263 jargon, which is the value of the URI maddr parameter, else the value of the domain part of the URI). The target information is passed in a "Question" field within the DNS query.

2) The DNS server sends a response with NAPTR resource records (RRs) in "Answer" fields. Those NAPTR RRs provide a mapping from the "target" (i.e, the called domain) to the set of transport protocols that target/domain prefers to use, and the addresses of corresponding SIP servers supporting those transport protocols. A given server may appear multiple times with multiple transport protocols.

(With the received information, the device can determine the intersection of its own transport protocol capabilities with what transport protocol it preferentially would use with some SIP server.)

#### 7.2.2 SIP Server IP Address / Port Number Discovery

1) The device sends a SRV-type DNS query[3] to the DNS server, where the key information contained is the URI type plus the target value plus the transport protocol type (e.g., _sip._tcp.example.com). That information is passed in a "Question" field within the DNS query.

[1] It may also be that, e.g., a SIP URI includes identification of a transport protocol (say SCTP) that the currently known SIP server doesn't support. In that case the device would have to attempt to discover another SIP server that can support that transport protocol.

[2] Binary coding of decimal 1 or 28 in the Type field of the DNS query header indicates a NAPTR-type DNS query for servers with IPv4 addresses or with IPv6 addresses, respectively. Such queries are also referred to as A and AAAA, respectively, which are the codepoint labels for 1 and 28, respectively.

[3] Binary coding of decimal 33 in the Type field of the DNS query header indicates a SRV-type DNS query.

2) The DNS server returns a DNS response that contains SRV RRs in “Answer” fields, where each RR has a server IP address, port number and other information. There may be multiple SRV RRs (i.e., multiple SIP servers) for reliability and load-sharing purposes. (Note that the IP address may typically be in FQDN format such as “server22.example.com”, as opposed to numeric format.)

### 7.2.3 Relationship of “Outbound Proxy” to Locating a SIP Server

The UNI Specification includes the concept of an “Outbound Proxy”. It is a SIP server in the network to which the user-side device or enterprise network would send all SIP requests. That Outbound Proxy either

- may be provisioned into the device/enterprise network as such, in which case the method above for locating a SIP server is not performed

or

- may be located by the method described above, because the network DNS data is such that there is only one logical SIP server.

### 7.3 Time Acquisition

SNTP uses the User Datagram Protocol (UDP), i.e., DHCP messages ride as payload in UDP datagrams.

The description here assumes the device is using the SNTP unicast mode, where the device sends a request to a specific server.[4] The device may periodically send SNTP request messages.

1) The Device sends a SNTP request in a UDP datagram to the IP address of the SNTP Time Server (whose address is known to the client, e.g., from the previous acquisition of that IP address and other info from a DHCP server).

2) The SNTP server receives the request and issues a SNTP reply in a UDP datagram with time-related information.

## 8 SIP Address of Record Registration

### 8.1 Initial Registration

A SIP device shall register with a SIP registrar any SIP address of record (AOR) associated with the device. Registrations have finite lifetimes, so must be occasionally renewed. An enterprise network may register multiple devices. The message flow is as follows:

<u>REGISTER</u> device → network with

- address of the SIP Registrar in the Req-URI (if known – see Note below)
- both the To and From headers containing the AOR(s) to be registered
- the Contact header containing the address(es) that the registered address is to be bound to (may be, e.g., in the form of the IP address of the device or another SIP address); optionally, with an expires parameter for each address with a requested registration duration value

<u>200 OK</u> device ← network with

- the contact address (from the Register message) in the Req-URI
- both the To and From headers containing the registered AOR

[4] The other modes are (a) broadcast where the device simply waits to hear from one or more broadcasting SNTP servers and (b) manycast where the device sends a request to a broadcast address that may result in replies to the device from multiple SNTP servers.

- the Contact header containing the registered-at address(es) (i.e. the address(es) in the Contact header in the REGISTER message) *and* an expires parameter with a registration duration for each address

Note: If the device does not have the address of a SIP registrar, it may alternatively use as the address in the Req-URI either
a) the hostpart of the AOR to be registered (and not the user part)
or
b) the “all SIP servers” multicast address – “sip.mcast.net”

The Contact header in the REGISTER request may contain multiple addresses, all of which the AOR is to be bound to. Even if not, Contact header in the 200 Okay response may still contain multiple addresses, because it contains a complete list of contact address bindings, some of which may have occurred through other registration requests for other contact addresses for the same AOR.

### 8.2 Other Registration Scenarios

#### 8.2.1 Registration Update

The device may want to change the registration binding to another contact address. The message flow and content is the same as that for initial registration above, *except* the new value for the contact address would appear in the Contact header.

#### 8.2.2 Request for Current Contact List

The device may wish to request the current contact address list for a registered AOR. The message flow and content is the same as that for initial registration above *except* that the Contact header is absent from the REGISTER message.

#### 8.2.3 Registration Cancellation

The device may cancel an existing Registration. The flow and content is the same as that for the initial registration except that

- The REGISTER message contains an Expires header with value “0”; and contains a Contact header with vacuous content
- The 200 Okay contains a Contact header with value “*”

## 9 Call Control/Session Establishment

### 9.1 SIP - Call Stimulus Scenarios

This subsection contains a set of call scenarios that could be used as stimuli to test aspects of the SIP protocol that it would be especially desirable to verify. The set below does *not*

- exhaustively exercise all possible facets of SIP use allowed according to the SIP UNI ANS
- in particular, error scenarios eliciting 4XX, 5XX and 6XX SIP responses are absent

Some of the scenarios below could be combined into a single stimulus call. The reason to separate out into different scenarios here is to focus on the different aspects of the protocol to be tested, which are in the third column of Table 1.

Note that if the user equipment is sending a SIP request via an outbound SIP proxy server in the network, the SIP request will carry the proxy server address in the Req-URI and the target address in the Route header, as per RFC 3261. That would be the case for any SIP request using an outbound proxy server, and not just those for session establishment.

In any of the scenarios below involving a SIP INVITE to establish a session, use of the PRACK method for reliable provisional responses may be employed, but is generally not explicitly included in the description.

**Table 1 - Call Stimulus Scenarios for SIP**

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| | | | |
| 1. | Simple voice call; destination address types | Address formats in INVITE Request-URI and To header<br>▪ sip:x…x@hostname;user=phone<br>▪ tel:+x…x<br>▪ sip:username@hostname | *RFC 3261,*<br>*RFC 3966,*<br>*RFC 3824*<br>The "x…x" string represents telephone number information.<br>The destination address could be domestic or international, POTS, Toll-Free or 911, etc. |
| 2. | Simple call; answered and cleared | Normal SIP messaging sequence with final response of 200 OK to the INVITE, including<br>▪ 183 Session Progress with SDP content<br>▪ 180 Ringing<br>▪ the use of PRACK messages | *RFC 3261,*<br>*RFC 3262* |
| 3. | Simple call; called party busy | Final response of 486 Busy Here to the INVITE | *RFC 3261* |
| 4. | Simple voice call between SIP end-devices; answered | Use of a MIME-encoded message body with SDP offer content in the SIP INVITE and SDP answer content in one or more of the 183 Session Progress, 180 Alerting and 200 OK responses; with the Content-Type header containing "application/sdp"; and with the Content-Disposition header, if present, containing "session". | *RFC 3261,*<br>*RFC 3264*<br>The provisional responses 183 Session Progress and 180 Alerting are optional.<br>If the SIP PRACK procedure is used (making provisional responses reliable), the SDP answer content is not required to be carried in a 200 OK response. |
| 5. | Audio + video multimedia call between SIP end-devices. | Use of a MIME-encoded SDP body in the SIP INVITE and one or more of the 180 Alerting, 183 Session Progress or 200 OK response messages, characterizing the single multimedia offer. | *RFC 3264* |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 6. | Simple call attempt; ring no answer for X seconds | The call and its SIP dialog attempt should remain in progress until cleared after X seconds from the network side with a 480 Temporarily Unavailable final response to the INVITE. | *RFC 3261*<br>The value of X depends on the network policy for clearing call attempts rather than letting them continue indefinitely (i.e., reflecting the PSTN practice of going to 'lockout') |
| 7. | Simple call, but with multiple SDP offers in the INVITE | 183 Session Progress or 200 OK response, with a single SDP entry that is one of those offered in the INVITE | *RFC 3261,*<br>*RFC 3262,*<br>*RFC 3264* |
| 8. | Incoming call (i.e., to the U side of the UNI) originated to a toll-free number; a network has performed the toll-free translation to a routing number | Routing number passed in the INVITE Req-URI, original dialed toll-free number in the To header | *RFC 3261* |
| 9. | Simple call, PSTN-originated, IAM Calling party number parameter APRI = *Presentation restricted*, no Generic Address parameter | INVITE P-Asserted Identity header includes the calling party number, and priv-value=; "id"; From header carries "anonymous" display name and 'Anonymous URI' | *T1.679-2004,*<br>*RFC 3325* |
| 10. | Simple call, PSTN-originated, IAM Calling party number parameter with APRI = *Presentation allowed*, no Generic Address parameter | INVITE P-Asserted Identity header includes the calling party number; From header also includes the calling party number | *T1.679-2004,*<br>*RFC 3325* |
| 11. | Incoming call attempt, caller hangs up before called party answers | Network sends a CANCEL message after the INVITE | *RFC 3261* |
| 12. | Simple fax call, fax passed as voiceband (aka "pass-through") | Address format in INVITE To header<br>▪ fax:+12234567890<br>and SDP content indicates audio G.711 with silence suppression *off* | *RFC 2806*<br>There is a variety of additional information that can be conveyed as parameters with this kind of URL that is not identified here |
| 13. | Simple modem call, data passed as voiceband | Address format in INVITE To header<br>▪ modem:+12234567890;type=*type info*<br>and SDP content indicates audio G.711 with silence suppression *off* | *RFC 2806*<br>There is a variety of additional information that can be conveyed as parameters with this kind of URL that is not identified here |
| 14. | Call from a wireline phone to an Emergency Telecommunications Service (ETS) destination number | SIP INVITE contains a Resource Priority header with namespace.value ets.X with an appropriate integer value for X from 0 to 4 | *RFC 4412* |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 15. | Call from a wireless phone to a Wireless Priority Service (WPS) access code + destination number | SIP INVITE contains a Resource Priority header with a pair of namespace.values ets.X, wps.Y | *RFC 4412* |
| 16. | Incoming call (i.e., to the U side of the UNI) involving an Application Server, such as a prepaid card service, during which a called party may be put on hold<br>Or, *alternatively*, a SIP phone-originated call answered by a called party, and the calling SIP phone at some point puts the called party on hold in such a way that the phone itself initiates a re-INVITE without an SDP offer. | Sending a second INVITE (re-INVITE) without an SDP offer, and receiving a 200 OK response confirming the SDP offer | *RFC 3261*<br>Caller may be able to input DTMF or a card balance timer may expire, causing the AS to return the caller to a MS and putting the called party temporarily on hold. |
| 17. | Same as above, but now the called party is reconnected after being on hold | Sending a second INVITE (re-INVITE) with an SDP offer, and receiving a 200 OK response confirming the SDP offer | *RFC 3261*<br>AS is reconnecting the called party to the calling party. |
| 18. | A VoIP connection exists between SIP end-devices A and B. Device A invokes a call transfer from itself to Device C. The connections A-B and B-C are each across a UNI. | Between A and B: Use of the REFER method, including passing of REFER, 202 Accepted and NOTIFY messages. The REFER message includes the Refer-To and Referred-By headers as well as the escaped Replaces header.<br>The “triggered INVITE” from B to C contains the Referred-By and Replaces headers.<br>The REFER message may also include use of the message/sipfrag technique to carry the status of referenced requests. | *RFC 3515*<br>*RFC 3891*<br>*RFC 3892*<br>*RFC 3420*<br>*draft-ietf-sip-referredby* |
| 19. | A SIP end-device that can include caller preferences originates a call to another SIP end-device | Passing of the Request-Disposition, Accept-Contact and Reject-Contact headers in the INVITE; the particular response to the INVITE is not specified here | *RFC 3841*<br>The response of the network receiving the INVITE will depend on what it does or doesn’t do with the information in any of those ‘caller preferences’ headers |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 20. | Fax call between IP fax devices. | The SIP INVITE is coded for fax transmission using IFP. – The SDP details will vary depending on whether transport is UDPTL/UDP, TPKT/TCP, or RTP over UDP. | *ITU-T T.38,*<br>draft-ietf-sipping-realtimefax,<br>draft-jones-avt-audio-t38 |
| 21. | Call received by a clearmode-capable SIP device from an ISDN endpoint that originates the call as 64kps clear channel; application could be, e.g., low grade video or Group 4 fax | SDP in the SIP INVITE indicates CLEARMODE/8000. | *RFC 4040*<br>*T1.679* |
| 22. | Voice call from a SIP phone to a called party; the called party deliberately does not answer, and the caller hangs up upon hearing ringing. | After the SIP INVITE from the caller side network and 1XX a provisional response from the called side network, a SIP CANCEL request should appear from the caller-side network, with a 200 OK response to the CANCEL *and* with a 487 Request Terminated final response to the INVITE. | *RFC 3261* |
| 23. | Call to an unassigned number in the terminating PTSN network. | A final response of 404 Not Found is received in response to the SIP INVITE. | *RFC 3261,*<br>*T1.679*<br><br>Note: the terminating PSTN network is responding with an ISUP Release message with Cause 1 (unallocated(unassigned number)) |
| 24. | Call to a number in the terminating PSTN that is recently changed and unassigned. | A final response of 410 Gone is received in response to the SIP INVITE. | *RFC 3261,*<br>*T1.679*<br><br>Note: the terminating PSTN network is responding with an ISUP Release message with Cause 22 (number changed) |
| 25. | Call to a PSTN-connected phone that is busy (and that does not have call waiting, call forwarding, etc). | A final response of 486 Busy is received in response to the SIP INVITE. | *RFC 3261,*<br>*T1.679*<br><br>Note: the terminating PSTN network is responding with an ISUP Release message with Cause 17 (user busy) |
| 26. | Call to a SIP phone that can be set to temporarily refuse calls (e.g., a SIP phone ‘do not disturb’ feature) | A final response of 480 Temporarily Unavailable is received in response to the SIP INVITE. | *RFC 3261* |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 27. | IP end-devices capable of treating early media separately from regular media as per RFC 3959; calling IP device offers regular media in the INVITE, the called IP device accepts it in a 183 Session Progress but also offers early media, which the calling IP device accepts in the 200 OK to the 183 Session Progress.<br>The caller device may be a SIP phone. The called device may be SIP equipment performing a prompt & collect activity with the caller before deciding to actually accept the caller's media offer.<br>Alternatively, a calling SIP device may interact with a network AS and MS that supports early media (pre-answer interaction with the caller). | INVITE carries<br>- an "early-session" option tag in the Supported header field,<br>- Content-Type "application/sdp",<br>- Content-Disposition "session",<br>- SDP description for a media session (e.g., for a voice call)<br>183 Session Progress with<br>- Content-Type "application/sdp"<br>- Content-Disposition "session"<br>- SDP content<br>--- that accepts the regular media offer in the INVITE<br>and<br>- Content-Type "application/sdp"<br>- Content-Disposition "early-session"<br>- SDP description for another media session (e.g., for a voice call)<br>---- that offers the caller early media<br>200 OK to the 183 Session Progress with<br>- Content-Type "application/sdp"<br>- Content-Disposition "early-session"<br>- SDP content<br>---- that accepts the called party's early media offer<br>200 OK to the INVITE | *RFC 3959,*<br>*RFC 3960*<br>Typically the early media and regular media session IP addresses/port #s would not be the same; e.g., at least the port numbers would be different so the two IP end-devices can distinguish between the two media streams<br>The early media stream can start as soon as the early media offer/answer exchange has occurred (the 183 and the 200 OK to it).<br>The early media stream ends and the regular media stream begins once the 200 OK to the INVITE has occurred (that transitions the SIP exchange status from early dialog to regular dialog). |
| 28. | A call between SIP end-devices that support SIPS URIs and therefore also TLS and TCP or SCTP. | Use of a SIPS URI in the INVITE Request-URI; in which case the TLS method of encrypted transport must shall be employed to carry all the SIP messages for that SIP dialog.<br>The transport parameter of the SIPS Request URI must also indicate a reliable transport method (i.e., *tls* in the case of TCP).<br>The Contact header in the INVITE must contain a SIPS URI. Likewise, the Contact header in the 200 OK response must contain a SIPS URI. | *RFC 3261,*<br>*RFC 2246,*<br>*draft-ietf-sip-sctp*<br>TLS must be used from the caller to the domain of the called party, which in this scenario would include going across the UNI.<br>Use of a SIPS URI and therefore TLS also means that a reliable transport must be in use (i.e., TCP; not UDP). |
| 29. | Simple voice call between SIP end-devices that support the regular form of SIP header names. | Use of the regular form of SIP header names in SIP messages. | *RFC 3261*<br>If the networks happen to convert to compact form, then this scenario is not applicable. |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 30. | Simple voice call between SIP end-devices that support the compact form of SIP header names. | Use of the compact form of SIP header names in SIP messages. | *RFC 3261*<br>Not all headers have a compact form for their names.<br>If the networks happen to convert to regular form, then this scenario is not applicable. |
| 31. | Voice call between user equipment and a network–based media server, where the user-media server interaction occurs as 'early media'.<br>The user device alters the initially requested media characteristics (e.g., request a different codec type). | Initially, an *early* dialog is established, with an initial INVITE and a reliable provisional response, where the INVITE contains an SDP offer and the initial INVITE and subsequent responses contain "UPDATE' in the Allow header<br>Then an UPDATE with a new SDP offer is sent, which receives a prompt 200 OK response<br>Then a 200 OK response to the original INVITE and subsequent ACK, neither of which would contain SDP; the 200 OK Contact header contains the same value as in the 200 OK response to the UPDATE | *RFC 3311*<br>Note that RFC 3311 allows an UPDATE to also be used in a *confirmed* dialog to change media characteristics, but it recommends using a re-INVITE instead. |
| *32.* | *Incoming* call to the user where the originating call attempt has involved a network application such as call forwarding that includes the use of the History-Info header. | The SIP INVITE includes a History-Info header.<br>The SIP INVITE may also include a 'histinfo' option tag in the Supported header, in which case the user's responses should include the History-Info header. | *RFC 4244*<br>The content of the History-Info header would include a set of other addresses attempted and, typically, reasons for the call not being accepted at those addresses.<br>RFC 4244 requires the use of TLS at the signaling transport layer for security purposes in order to pass History-Info in a request. |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 33. | SIP end devices that both support the qos precondition; outgoing voice call; calling device requires resource reservations end-to-end in both directions | INVITE with<br>- Requires header that includes the "preconditions" tag<br>- Supported header that includes the "100rel" tag<br>- Allow header that includes the "UPDATE" tag<br>- SDP content that includes precondition elements<br>a=curr:qos e2e none<br>a=des:qos e2e mandatory sendrecv<br><br>then response 183 Session Progress with SDP content that includes precondition elements<br>a=curr:qos e2e none<br>a=des:qos e2e mandatory sendrecv<br>a=conf:qos e2e recv<br><br>[at this point, both devices engage in reservation for their respective send paths using, e.g., RSVP]<br><br>then UPDATE (from calling device once is reservation is complete) with SDP content that includes precondition elements<br>a=curr:qos e2e send<br>a=des:qos e2e mandatory sendrecv<br><br>200 OK (to the UPDATE) (from the called device once its reservation is complete)<br>a=curr:qos e2e sendrecv<br>a=des:qos e2e mandatory sendrecv<br><br>180 Ringing<br><br>200 OK (to the INVITE) | *RFC 3312*<br><br>The use of preconditions requires the use of the PRACK (*RFC 3262*) and UPDATE (*RFC 3311*) methods.<br><br>The qos precondition assumes that the end devices employ a protocol such as RSVP (*RFC 2205*) for resource reservation.<br><br>With preconditions, the called party is not alerted until the preconditions are met. |
| 34. | SIP device that employs the SIP Privacy header for some purpose on some type of call; outgoing voice call | INVITE includes the Privacy header with one or more values<br><br>Other request and response messages may also include the Privacy header | *RFC 3323*<br><br>Use of the Privacy header shields the identity of the user agent from intermediate SIP entities in the path or from the far end user agent. It assumes the existence of 'privacy services' in the network to support the types of privacy indicated in the header. |
| 35. | SIP device behind a Network Address Translator (NAT); outgoing voice call | INVITE Via header includes an rport parameter | *RFC 3581* |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 36. | A simple call between SIP end-devices that support S/MIME message content.<br>The particular S/MIME content that is passed is not specified as part of this scenario. It could be, for example, an attachment file or some SIP message headers that could have significance to the end-devices but that the networks do not act upon. | Passing in the INVITE request, and possibly a 180 Alerting, 183 Session Progress or 200 OK response message, of S/MIME-encoded message body content, as indicated in the Content-Type (e.g. “application/x-pkcs7-mime”) and Content-Disposition (e.g., “attachment; filename=smime.p7m”) headers. | *RFC 3261,*<br>*RFC 2311*<br>With the use of S/MIME, the end-devices would also exchange key/certificate information to allow them to decrypt/authenticate. |
| 37. | A call between SIP end-devices, where the called device forwards the call to another endpoint, and employs the 181 Call Is Being Forwarded response to inform the originator of status | The INVITE receives a temporary response of 181 Call Is Being Forwarded | *RFC 3261* |
| 38. | Simple voice call between SIP end-devices, where the called device temporarily does not accept the call (e.g., it may be busy on another call or calls), and employs the 182 Queued response to inform the originator of status | The INVITE receives a temporary response of 182 Queued<br>The 182 Queued response may contain a reason phrase providing information about the queuing status. | *RFC 3261* |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 39. | Call between SIP end-devices that support the SIP Authenticated Identity Body (AIB) capability for verifying user identity | INVITE request has a “multipart/mixed” MIME body that includes a body for the SDP offer *and* additional body content for the AIB information. The exact format can vary. An illustrative example is:<br>Content-Type: multipart/signed; protocol="application/pkcs7-signature";<br>micalg=sha1; boundary=boundary*XX*<br>Content length: *NNN*<br>--boundary*XX*<br>Content-Type: message/sipfrag<br>Content-Disposition: aib; handling=optional<br>*{SIP headers constituting the actual AIB body}*<br>--boundary*XX*<br>Content-Type: application/pkcs7-signature; name=smime.p7s<br>Content-Transfer-Encoding: base64<br>Content-Disposition: attachment; filename=smime.p7s;<br>handling=required<br>*{cryptographic signature}*<br>--boundary*XX*-- | *RFC 3893*<br>The AIB body may optionally be encrypted. |
| 40. | Call between SIP end-devices that support SIP content indirection, with content (e.g., a jpeg file) located elsewhere that will be identified by a URI from the calling device and retrieved by the called device | INVITE request contains a multi-part body with<br>▪ SDP offer content,<br>▪ content indirection information that will minimally consist of<br>Content-Type: message/external-body; access-type="URL";URL=" *address*"<br>The content indirection part may contain additional information such as content description, expiration date of the URL, or content size.<br>The INVITE may also carry “message/external-body” in the Accept header. | draft-ietf-sip-content-indirect-mech |

| # | Scenario name/description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 41. | A call between SIP end-devices A and B already exists. A third SIP end-device C, across the VoIP UNI from B, invites itself to join the A-B SIP call at device B, creating a 3-way call.<br>Both SIP end-devices B and C must support the SIP Join capability, and additionally the device B must have a 3-way bridging capability. | The SIP INVITE from device C addressed to device B includes a Join header with call ID, to-tag and from-tag information from the device A – device B SIP dialog.<br>That SIP INVITE should also include the option tag "join" in a Supported header, and may also include that option tag in a Required header.<br>The Contact header in the 200 OK response contains a conference URI corresponding to device B, and an "isfocus" parameter (that identifies which party is controlling the conference) | *RFC 3911*<br>How the device A – device B SIP dialog information is provided to device C is not specified here. Presumably the information can be identified at device B and somehow conveyed to and input into device C.<br>The dialog information must be as it is known to device B. |
| 42. | SIP end-devices A and B are engaged in a SIP dialog. Another SIP device C, across the VoIP IP UNI from device B, subscribes with device B for the dialog state of the device A – device B SIP dialog.<br>After the subscription has begun, device B ends its call with device A.<br>Devices B and C support the SIP Subscribe/Notify capability and the 'dialog' event package.<br>This scenario assumes that the subscription interval does not expire before the device A – device B dialog is ended. | SIP SUBSCRIBE request from the network serving device C, addressed to device B, with an Event header containing the event package name "dialog" with parameters: to-tag, from-tag, call-ID of the device A – device B dialog, and optionally an include-session-description parameter, and optionally an Accept header indicating what kind of data format is acceptable to be returned in a NOTIFY<br>A response of either 200 OK or 202 Accepted is returned<br>A first NOTIFY request indicates a dialog state of "confirmed", and also provides the (media) session description information if it was requested<br>That first NOTIFY request receives a 200 OK response<br>A second NOTIFY request indicates a dialog state of "terminated"<br>That second NOTIFY request receives a 200 OK response.<br>(That second exchange of NOTIFY and 200 OK will implicitly end the SUBSCRIBE dialog, because the A-B dialog state was "terminated".) | *RFC 3265,*<br>*RFC 4235*<br>How the device A – device B SIP dialog information is provided to device C is not specified here. Presumably the information can be identified at device B and somehow conveyed to and input into device C.<br>The dialog information must be as it is known to device B.<br>Inclusion of the include-session-description parameter means device C wants to know the nature of the media session between A and B.<br>The default NOTIFY data format for the dialog event package is "application/dialog-info+xml" |

### 9.2 Media - Call Stimulus Scenarios

This subsection contains a set of call scenarios that could be used as stimuli to test aspects of media support that it would be especially desirable to verify. Analogous to the SIP-related section above, the set below does *not*

- exhaustively exercise all possible facets of media use allowed according to the IP NI ASN

- indicate other levels of testing that could be performed at the same time, such as verification related to signaling transport or SIP

Some of the scenarios below could be combined into a single stimulus call. The reason to separate out into different scenarios here is to focus on the different aspects of the protocol to be tested, which are the third column of Table 2.

When RTP and RTCP are used for media transport, there are certain situations involving Network Address Translation (NAT) where their use must be *symmetric,* as per RFC 4961.

**Table 2 - Call Stimulus Scenarios for Media**

| # | Scenario Name/Description | Media aspect to be exercised & verified | Comments |
|---|---|---|---|
| | | | |
| 1. | Simple SIP originated calls that support various audio coding formats | Various audio coding formats such as G.711, G.729A, etc. | *RFC 3264* |
| 2. | Outgoing video or audio+video call | Various video coding formats such as H.263 or H.264 | *RFC 3984,*<br>*RFC 4629* |
| 3. | Call where digits are entered from the keypad of the calling or called device after the call is answered<br>The far end device may be an ordinary telephone off the PSTN | Passing of 'DTMF' digits using the RTP Payload Format for Named Events | *RFC 4733* |
| 4. | Call between SIP end-devices that starts with one audio coding format (INVITE with, e.g., G.711) and later switches to another audio coding format (a re-INVITE with, e.g., G.726) | RTP packets passed according to the first coding and then the other coding format | *RFC 3264* |
| 5. | Call between IP end-devices performing a fax (facsimile) transmission using the IFP protocol over UDPTL/UDP. | Support for the IFP protocol, and for IFP running over UDPTL/UDP. | *ITU-T T.38*<br>In this case IFP is transported using UDP with the UDPTL header. |
| 6. | Call between IP end-devices performing a fax  transmission using the IFP protocol over TPKT/TCP | Support for the IFP protocol, and for IFP running over TPKT/TCP. | *ITU-T T.38*<br>In this case IFP is transported using TCP with the TPKT header. |
| 7. | Call between IP end-devices performing a fax  transmission using the IFP protocol over RTP | Support for the IFP protocol, and for IFP running over RTP. | *ITU-T T.38, draft-jones-avt-audio-t38-05 in the IETF Network working group*<br>In this case IFP is transported using RTP (over UDP). |

| # | Scenario Name/Description | Media aspect to be exercised & verified | Comments |
|---|---|---|---|
| 8. | Call between devices performing a Group 3 fax transmission.<br>The far end device may be a SIP device or a device off the PSTN.<br>The particular image transmission method/rate is not specified as part of this scenario. It will be negotiated by the fax end-devices. | Passing of an inband fax transmission.<br>Passing of fax control/communication tone-based information using the RTP Payload Format for Named Events<br>- ANS<br>- CNG<br>- V.21 channel 1, "0" bit<br>- V.21 channel 1, "1" bit<br>- V.21 channel 2, "0" bit<br>- V.21 channel 2, "1" bit | *ITU-T T.30,*<br>*RFC 4733*<br>Note: In this case, where an audio coding for fax is used (so the fax is passed as audio using G.711 and *not* using IFP), the distinction between voice and fax in RTP is that conveyance of fax tones will involve 'named events' different from DTMF. |
| 9. | Call from one VoIP device to another using G.711. The calling VoIP device also supports Voice Activity Detection and Comfort Noise Generation.<br>The caller does not speak and there is no background noise (i.e., audio input into the VoIP device is silence). | Passing of comfort noise payload in RTP Silence Insertion Descriptor frames | *RFC 3389*<br>For this kind of originating device, during a period when the caller isn't speaking the G.711 encoder will not generate RTP audio packets encoding the non-voice. Instead, the device will send RTP comfort noise packets. |
| 10. | Call between devices performing a Super Group 3 fax transmission.<br>The far end device may be a SIP device or a device off the PSTN.<br>The particular image transmission method/rate is not specified as part of this scenario. It will be negotiated by the fax end-devices. | Same as the Group 3 fax scenario above, except that ANSam is seen from the called fax terminal instead of ANS. | *ITU-T V.34,*<br>*ITU-T T.30,*<br>*RFC 4733*<br>The fax or data is passed as audio using G.711. |
| 11. | Call between fax or modem devices using V.8bis.<br>The far end device may be a SIP device or a device off the PSTN. | Passing of tone-based capability or mode information using the RTP Payload Format for Named Events<br>CRdi<br>CRdr<br>CRe<br>ESi<br>ESr<br>MRdi<br>MRdr<br>MRe | *ITU-T V.8bis,*<br>*RFC 4733*<br>Note: V.8bis would precede V.8 (ANS, ANSam) in the end-end communication.<br>The fax or data is passed as audio using G.711. |

| # | Scenario Name/Description | Media aspect to be exercised & verified | Comments |
|---|---|---|---|
| 12. | Call received by a clearmode-capable SIP device from an ISDN endpoint that originates the call as 64kps clear channel; application could be, e.g., low grade video or Group 4 fax | Support for the equivalent of a 64 kbps clear channel connection; | *RFC 4040*<br>*T1.679*<br>SDP in the SIP INVITE would indicate CLEARMODE/8000. |
| 13. | Voice call between two endpoints. | Passing of RTCP packets associated with the voice RTP stream<br>In this case, compound RTCP packets consisting of Sender Report (SR) and Source Description (SDES) RTCP packets. | *RFC 3550*<br>RTCP itself does not pass media, but provides for the exchange of performance and control information related to the RTP media stream. |

## 10 SIP OPTIONS

The SIP OPTIONS method allows a SIP User Agent to query another SIP User Agent or proxy about the SIP capabilities that it supports. A typical would purpose would be, e.g., for a device to find out information about the supported SIP methods, content types, extensions, codecs, etc. of a potential called device/party without "ringing" the other party.

An OPTIONS request may be sent outside of any other established dialog. It may also be sent within an established dialog (e.g., an already-established SIP session for a voice call).

A specific stimulus scenario may be a manual triggering of a SIP Phone device to perform an OPTIONS query of another SIP Phone device as identified by its pubic address. Regardless of scenario, the message exchange is essentially the same and consists of:

OPTIONS device → network including

- the Req-URI containing the address of the target entity for this request
- an Accept header with content indicating the type of message body that the sending client wishes to receive in the response

200 OK device ← network including

- an Allow header with the set of SIP methods supported; (note: this header is *not* present in the response from a SIP *proxy*)
- an Accept header with information on media types supported
- an Accept-Encoding header with information on content-codings supported
- an Accept-Language header with a list of languages acceptable for reason phrases, session descriptions or status responses carried in message bodies
- a Supported header with a list of option tags for all the SIP extensions supported
- optionally a message body; for example if the OPTIONS request in its Accept header included ‘application/sdp’, then the target may include corresponding application/sdp content in this message body

The target could respond with some other response instead of a 200 OK, even in a ‘normal’ situation. For example, the target could respond with a 486 Busy Here if it is not prepared to accept a call.

## 11 Presence

While Presence in its entirety is a fairly complicated subject, its use of a SIP interface is fairly simple. RFC 2778 describes the Presence framework. RFC 3265 describes the underlying SUBSCRIBE/NOTIFY method. Other RFCs such as 3856 through 3859 and 3861 through 3863 provide additional information.

See Figure 1. To meaningfully exercise presence-related SIP capabilities across the UNI, the network has a Presence application, there is a presentity that can be watched, and there is at least one watcher. The watcher's user device shall  have a presence/watcher application.

Watchers can get information about a presentity or they can get information about the set of watchers of a presentity. A presentity can also be a watcher of itself.

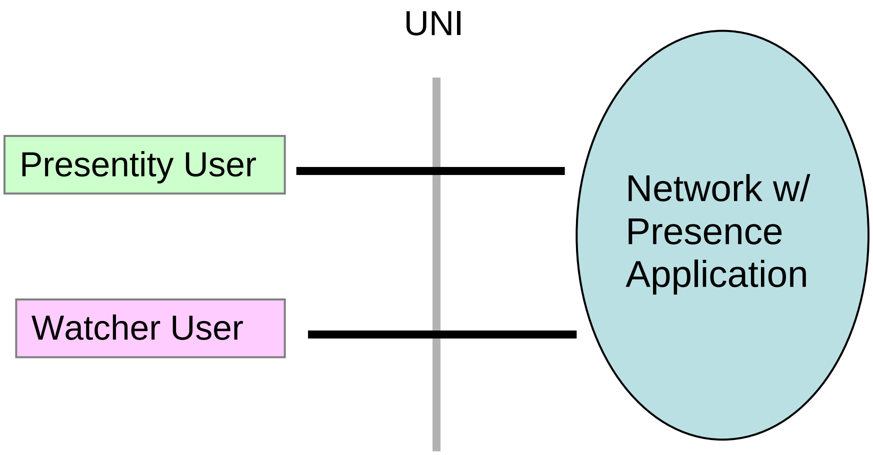

**Figure 1 - Basic Presence System**

This section focuses on the watcher user of the UNI. How the presentity user subscribes to a network-based presence application or makes its presence information known to that application[5] falls outside the scope of the UNI.

As with other sections of this TR, the message content listed below is not exhaustive; it only highlights the particulars for supporting presence.

### 11.1 A Watcher Wants Information about a Presentity

This scenario assumes that the watcher's request is accepted.

SUBSCRIBE, user → network with

- Req URI: the address of the presentity in SIP, SIPS or pres(ence) form
  - pres URI *should* be used in the req URI if known (else SIP or SIPS URI)
- From header: if the watcher has a pres URI as well, it *should* be used here; otherwise a SIP or SIPS URI of the watcher
- Event header: indicates "presence"
- Expires header: *should* be present, providing a proposed duration for the subscription

[5] The only exception is the use of a REGISTER message, which would be sent for a purpose other than presence, but which a presence application may make use of, for example, to obtain a current Contact address for the presentity.

- Accept header, optional; if present, it must include “application/pidf+xml”, which is the default data format for a presence document; but may additionally indicate other formats
- Message body, optional; it may contain, e.g.
- , a filter document indicating the types of presence information desired by this watcher

200 OK device ← network with

- Expires header: with the duration of the subscription

NOTIFY device ← network with

- Event header: indicates “presence”
- Subscription-State header: indicates “active”
- Message body containing a presence document, where the default is the “application/pidf+xml” format, but could be any other format listed in the Accept header of the SUBSCRIBE

200 OK device→ network

Additional NOTIFY/200 Okay exchanges occur whenever presentity status changes and the watcher is to be informed. One way to cause a presentity status change would be for it to disconnect from the network and then REGISTER through another device, causing the presentity Contact address information to change.

### 11.2 A Watcher Wants Information about the Set of Watchers of a Presentity

This scenario assumes that the watcher’s request is accepted.

SUBSCRIBE, user → network with

- Req URI: the address of the presentity
  - pres URI *should* be used in the req URI if known (else SIP or SIPS URI)
- Event header: indicates “presence.winfo”
- Expires header: *should* be present, providing a proposed duration for the subscription
- Accept header, optional; if present, it must include “application/watcherinfo+xml”, which is the default data format for a watcher document; but may additionally indicate other formats
- Message body, optional; it may contain, e.g., a filter document indicating the types of watcher information desired by this watcher

200 OK device ← network with

- Expires header: with the duration of the subscription

NOTIFY device ← network with

- Message body containing a presence document, where the default is the “application/watcherinfo+xml” format, but could be any other format listed in the Accept header of the SUBSCRIBE

200 OK device→ network

Additional NOTIFY/200 Okay exchanges occur whenever the set of watchers status changes and the watcher is to be informed.

### 11.3 Additional Observations

All presence subscriptions must be authenticated, which includes the exchange of additional messages not shown above. See section 17.2.

It may be that a user is barred by the presentity from viewing its presence or watcher information. In that case the response to the SUBSCRIBE would be a 403 Forbidden or 603 Decline.

A watcher can request a one-time 'fetch' of presentity or watcher information. That is accomplished by setting the Expires value to 0 seconds in the SUBSCRIBE message.

Since subscriptions are of finite duration, they must be periodically refreshed in order to persist indefinitely, which is accomplished as per RFC 3265 with another SUBSCRIBE with same dialog info as the existing dialog. A watcher may terminate a subscription by sending a new SUBSCRIBE with the same dialog information with an Expires value of 0 seconds.

### 11.4 Presentity Application in an Enterprise Network on the User Side of the UNI

As illustrated in Figure 2 below, when the User side of the UNI is an enterprise network, it may contain a presence application $P_2$ that keeps track of a presentity locally within that enterprise network. In order for a presence application $P_1$ in the network on the network side of the UNI to have current and global information on that presentity, $P_1$ would have to subscribe to presence information from $P_2$. In that role where $P_1$ is a watcher, the message flows described above would still apply, but would go in the opposite direction (SUBSCRIBE network → user, etc.).

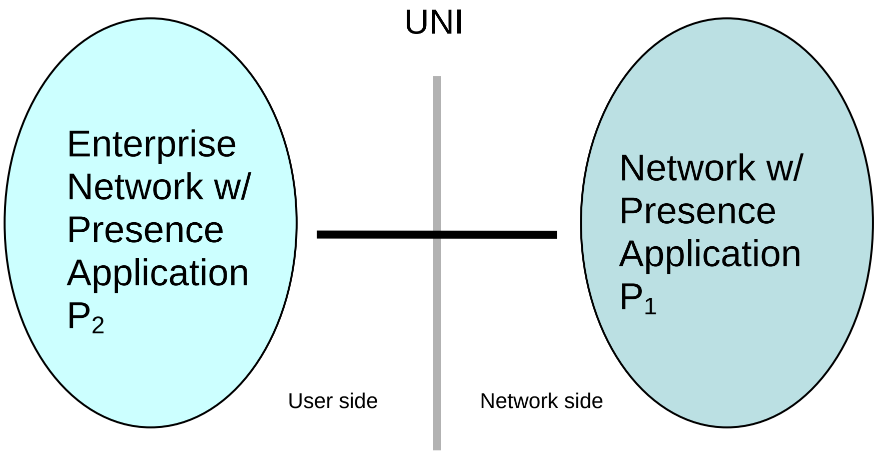
UNI
Enterprise Network w/ Presence Application $P_2$
Network w/ Presence Application $P_1$
User side
Network side

**Figure 2 - Presence Application in Enterprise Network**

## 12 Instant Messaging

### 12.1 Page Mode

The page mode of instant messaging operates as described in RFC 3428, where each message is stand-alone, as opposed to the session mode described next. Employing two SIP devices that support the SIP MESSAGE method, the essence of a successful exchange observed across the UNI (where the user side is originating the instant message) is

MESSAGE, user → network with

- the instant message carried as MIME payload

200 OK user ← network

Notes:

a. The MIME payload can be of any type.
b. The 200 OK must not contain a body or a Contact header field.
c. The response to the MESSAGE request may be a 202 Accepted instead of 200 OK.
d. The use of the URI format im:user@domain is not supported across the UNI.

## 12.2 Session Mode

The session mode of instant messaging uses the Message Session Relay Protocol (MSRP) and operates as described in RFC 4975. A media session for instant messaging is established using SIP. The SDP offer/answer within SIP identifies the use of MSRP as the message exchange protocol, the addresses of the two messaging endpoints, and the type of connection (e.g.,TCP) to be used. The actual instant message exchange uses MSRP, not SIP.

Note that the SIP session including instant messaging could be multi-media. The SDP offer/answer could be, e.g., for voice and instant messaging.

Employing end-user devices that support MSRP, the salient content of the SIP session establishment across the UNI when the user originates the session are

INVITE, user → network with

- the SDP offer containing, by way of illustration, (only SDP lines pertinent to instant messaging content are show here)

```
c=IN IP4 atlanta.example.com
m=message 7654 TCP/MSRP *
a=accept-types: message/cpim text/plain
a=path:msrp://atlanta.example.com:7654/jshA7weztas;tcp
```

where
- in the c-line, atlanta.example,com illustrates the “authority” portion of the MSRP URI
- the m-line indicates ‘message’ as the media type, a port number (7654) for receiving MSRP messages, and TCP/MSRP indicates the protocol, and ‘*’ is a null value for the irrelevant format list field.
- the first a-line indicates the types of message formats the user is willing to accept
- the second a-line indicates the MSRP URI for receiving messages; note that the random string jshA7weztas illustrates the MSRP session identifier for this particular MSRP session.

200 OK , user ← network with

- SDP answer with similar information for the far-end user (its MSRP URI, port number)

ACK, user → network

The reader is advised to be aware, as RFC 4975 says, that “MSRP URIs that identify sessions are ephemeral; an MSRP device will use a different MSRP URI for each distinct session. An MSRP URI that identifies a session has no meaning outside the scope of that session.”

Immediately after the SIP session is established, the initiating user device (the one that sent the SIP INVITE) sends a SEND ‘request’ whether or not there is actual user instant messaging content to be sent at that point, and that SEND receives an MSRP 200 OK response.

Note: MRSP messages cannot be sent until the underlying TCP connection is established between the two users. The TCP connection may be a pre-existing one or a new one.

When an instant message is to be sent, it is conveyed as payload in a SEND request, which normally receives an MRSP 200 OK response:

SEND user → network with

- a transaction identifier in the start line
- the msrp URI of the intended recipient in the To-Path header
- the msrp URI of the sender in the From-Path header
- a Success-Report header (optional)
- a Failure-Report header (optional)
- a Byte-Range header indicating what bytes of the instant message contained as payload (it may be a 'chunk' or the entire message)
- Content-Type indicating the payload format
- payload with all or a portion of the instant message
- an end line of seven hyphens plus the same transaction identifier as in the start line, plus a final character of '$'

200 OK response user ← network with

- the same transaction identifier in the start line as in the SEND request
- To-Path with the msrp URI of the sender of the SEND
- From-Path header with the msrp URI of the recipient of the SEND
- Byte-Range header indicating the byte segment of the instant message that was received
- An end line of seven hyphens plus the transaction identifier plus a final character of '$'

Note: A single instant message can be 'chunked' across multiple SEND requests. In that case, the final character of the end line of each SEND and each corresponding 200 OK is a '+' instead of '$' *except* for the SEND that contains the final chunk of the message and its corresponding 200 OK. Each SEND indicates the byte range of the chunk of the message contained therein, e.g. 10-50 out of 160 bytes.

Note: If a SEND message contains a Success-Report header set to "Yes", it will elicit a corresponding REPORT request from the recipient of the SEND (or SENDs) conveying the instant message. The REPORT message for a successfully received instant message will look like:

REPORT, user ←network with

- Same transaction identifier in the start line as came in the SEND
- To-Path header with the msrp URI of the sender of the SEND
- From-Path header with the msrp URI of the recipient of the SEND
- Message-ID header with the same value as came in the SEND
- Byte-Range header indicating the range of bytes covered
- Status header with namespace value of "000", status code of "200", and possibly additional status information in a comment phrase

Note: The issuer of the REPORT request may send a single REPORT covering the entire instant message once received, or may send periodic reports for the portion of the instant message received thus far.

The SIP session would eventually be closed out with the usual SIP BYE and 200OK.

### 12.3 isComposing Status Messages

For both page mode and session mode instant messaging exchange, status messages can be conveyed that indicate that the user is composing a new message, as described in RFC 3994.

Such indication is exchanged as payload
- in the SIP MESSAGE request in page mode
- in the MSRP SEND request in session mode

where the payload content type is “application/im-iscomposing+xml”. In session mode, to be used, that content type would be listed in the SDP offer as one of the types of media payload format this is acceptable and would be acknowledged in the SDP answer (i.e., ‘a’ line accept-types).

## 13 Conference Control

SIP requests may be subject to authentication. See Section 17.2.

### 13.1 Conference Control

The call stimulus scenarios contained in this section reflect aspects of SIP that may typically involve, and therefore may be conveniently exercised by, a conference situation where the user equipment and a conference application are operating as per RFC 4579. Note, however, that while some of the message content may be conferencing-specific, the SIP methods have broader applicability than just conferencing.

RFC 4579 assumes a conference-aware SIP User Agent known as the conference focus. When a user wants to join the conference, for example, the user sends an INVITE to the conference URI associated with the focus. The discussion here assumes the focus is on the network side of the UNI, although there is nothing to preclude it from being in customer equipment and therefore on the user side of a UNI.

The scenarios below all assume user equipment and a network-based conference application that support capabilities as described in RFC 4579. Several of the scenarios assume the involvement of another user that may, for example, be added to or dropped from the conference. In such cases where another user is involved, the signaling with that user is not shown.

**Table 3- Call Stimulus Scenarios for SIP - Typically for Conferencing**

| # | Scenario Name/Description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| | | | |
| 1. | User calls into conference using the conference URI | INVITE with the conference URI in the Request-URI and To header<br>200 OK response contains the isfocus feature parameter for the URI in the Contact header | *RFC 4579*<br>The conference URI does not necessarily look different from other URIs.<br>In general, the focus should include the isfocus feature parameter whenever it is sending a SIP message with a Contact header. |

| # | Scenario Name/Description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 2. | User requests the focus to add a new conference participant | ▪ REFER from the User with the conference URI as the Req URI and the intended participant's URI in the Refer-To header<br>▪ 202 Accepted from the focus<br>▪ NOTIFY from the focus with payload of a SIP Trying message fragment<br>▪ 200 OK from the User<br>When the participant is successfully added:<br>▪ NOTIFY from the focus with payload of a 200 OK message fragment<br>▪ 200 OK from the User | *RFC 4579*<br>*RFC 3515*<br>The User making the request does not need to be a conference participant. The REFER may be sent within an existing SIP dialog, or establish a new one. |
| 3. | User requests the focus to request another user to join the conference | ▪ REFER from the User with the conference URI as the Req URI; and in the Refer-To header:<br>- the intended participant's URI<br>- method=REFER<br>- an escaped Refer-To header with the conference URI<br>▪ 202 Accepted from the focus<br>▪ NOTIFY from the focus with payload of a SIP Trying message fragment<br>▪ 200 OK from the User<br>Next, when the focus sends a REFER to the intended participant:<br>▪ NOTIFY from the focus with payload of a 202 Accepted message fragment<br>▪ 200 OK from the User<br>When the participant is successfully added:<br>▪ NOTIFY from the focus<br>▪ 200 OK from the User | *RFC 4579*<br>*RFC 3515*<br>This is essentially a 'REFER in a REFER' |
| 4. | User requests to join the conference, as identified by a SIP dialog of one of the existing legs with a participant | ▪ INVITE with the Conference URI as the Req URI and in the To header, and with a Join header containing the SIP dialog information of the existing call leg with a participant | *RFC 4579*<br>*RFC 3911*<br>This scenario assumes that there is a call conference call leg already established with one participant, and its SIP dialog information can be known and used by our user's device. |

| # | Scenario Name/Description | SIP aspect to be exercised & verified | Comments |
|---|---|---|---|
| 5. | User is a participant/leg on the conference (SIP dialog D1) and wants to move his/her participation to another device. Initiates the request from that second device. | ▪ New INVITE (*not* a 're-INVITE' for existing SIP dialog D1) with the Conference URI as the Req URI and in the To header, and a Replaces header with the SIP Dialog D1 information | *RFC 4579*<br>*RFC 3891*<br>This scenario assumes an existing SIP session from a user's device, and that its SIP dialog information can be known and used by our user's second device.<br>The focus will also initiate clearing of the SIP session with the first device. |
| 6. | User requests a participant be dropped from the conference | ▪ REFER from the User with the conference URI as the Req URI; and in the Refer-To header:<br>- the intended droppee's URI<br>- method=BYE<br>▪ 202 Accepted from the focus<br>▪ NOTIFY from the focus with payload of a SIP Trying message fragment<br>▪ 200 OK from the User<br>When the participant is successfully dropped:<br>▪ NOTIFY from the focus with payload of a 200 OK message fragment<br>▪ 200 OK from the User | *RFC 4579*<br>*RFC 3515*<br>This scenario assumes the User has the authority to drop the participant. |
| 7. | User discovers/confirms that a URI is a conference focus and discovers SIP-related capabilities of the focus | ▪ OPTIONS from the User with the conference URI as the Req URI<br>▪ 200 OK from the focus with the conference URI and isfocus feature parameter in the Contact header; and with additional header fields (Allow, Accept, Accept-Encoding, Accept-Language, Supported) listing additional capabilities supported by the focus | *RFC 4579*<br>*RFC 3261* |

### 13.2 Subscription to Conference State

RFC 4575 defines a subscription package for conference state. The event package name is *conference*. Subscription is handled by the conference focus. Because subscription to conference state employs the SIP SUBSCRIBE/NOTIFY method, it resembles subscription for any other purpose, such as presence. The event name and document format are different, and what triggers the end of the subscription is typically the end of the conference.

A subscriber to the conference package *need not* be a conference participant.

The scenario below assumes that the subscriber's request is accepted.

<u>SUBSCRIBE</u>, user → network with

- Req URI: the conference URI in SIP, SIPS form
- From header: a SIP or SIPS URI of the subscriber

- Event header: indicates “conference”
- Expires header: *should* be present, providing a proposed duration for the subscription
- Accept header, optional; if present, it must include “application/conference-info+xml”, which is the default data format for a conference document; but may additionally indicate other formats
- Message body, optional; it may contain, e.g., a filter document indicating the types of conference information desired by this subscriber

200 OK device ← network with

- Expires header: with the duration of the subscription

NOTIFY device ← network with

- Event header: indicates “conference”
- Subscription-State header indicating “active”
- Message body containing a conference document, where the default is the “application/conference-info+xml” format, but could be any other format listed in the Accept header of the SUBSCRIBE

200 OK device→ network

Additional NOTIFY/200 Okay exchanges occur whenever conference status changes and the user/subscriber is to be informed.

When the conference ends, the subscription is terminated by the focus sending a final NOTIFY with “noresource” in the Subscription-State header.

## 14 Network Address Translator (NAT)

A Network Address Translator (NAT) may translate the IP address and/or port number of the user device. The SIP UNI ANS recognizes two methods that may be employed for media traversal of NATs: STUN (RFC 3489) and ICE (draft-ietf- mmusic-ice). Both STUN and ICE are relevant when UDP is used for media transport.[6] (Note that SIP message traversal of a NAT is accomplished by methods other than those for media transport, such as use of a SIP outbound proxy and the rport parameter.)

Note that RTP and RTCP involve the use of multiple and possibly different IP addresses and port numbers. Likewise, multimedia sessions may involve the use of multiple IP addresses and port numbers.

STUN and ICE are complicated topics, especially the latter. The sections below explain the basics of what would be observed across the UNI. The documents cited above and related material should be studied for a full understanding.

### 14.1 Simple Traversal of UDP over NATs (STUN)

The use of STUN requires a STUN server in the network. The STUN server address may be provisioned in the device, or may be determined as part of DNS discovery, or may be directly discovered using procedures as specified in RFC 2782. It does not work with all types of NATs; in particular it does not work with symmetric NATs.

The device (containing a STUN client) sends one or more STUN Binding requests to the STUN server. Depending on whether a Binding response or no response is received for each request, the device can learn whether a NAT is present, and if so, what public IP address is being used by

[6] ICE could be extended for other types of transport as well, such as TCP.

the NAT for the device, what port was opened by the NAT to allow incoming traffic back to the device, and the type of NAT encountered.

Before a Binding request can be sent, the device (STUN client) sends a STUN Shared Secret request, using TLS/TCP for transport. The STUN server returns a Shared Secret response containing a temporary ‘username’ and ‘password’. That information is employed in formulating a subsequent Binding request to the STUN server.

### 14.2 Interactive Connectivity Establishment (ICE)

Interactive Connectivity Establishment (ICE) (draft-ietf- mmusic-ice) makes use of STUN (specifically draft-ietf-behave-rfc3489bis) and its extension Traversal Using Relay NAT (TURN) (draft-ietf-behave-turn). ICE is more versatile than STUN. ICE may use STUN or TURN servers in the network.

Suppose a device A wants to establish a SIP-based media session with a device B across a network using UDP transport for the media, and in which NAT traversal may be involved. The basic premise of ICE is

1) Device A determines a set of “candidate” receive media addresses for itself (IP addresses and port numbers) to which Device B might be able to send media; determination of some candidate addresses may involve the use of STUN or TURN servers in the network.

2) Device A sends a SIP INVITE to device B, where the included SDP offer indicates use of ICE by way of the a=ice attribute and includes the candidate addresses by way of the a=candidate attribute.

3) Device B, upon receiving the INVITE, determines its own set of candidate receive media addresses.

4) Device B sends a SIP response, typically a 183 Session Progress, where the SDP content indicates B’s own candidate receive media addresses. Device B may periodically repeat sending that response until it receives a first STUN Binding request from Device A (see next step 3).

5) Devices A and B each perform testing of the candidate addresses by sending STUN Binding requests to each other (each acting as a STUN server for the other) and discover which candidate addresses actually work. In this process a form of authentication is also performed, where username/password information is included in the Binding requests that was earlier shared in the SIP request and response messages (contained in the SDP a=ice-pwd and a=ice-ufrag attributes).

6) Given that devices A and B now know which pairs of candidate addresses work, they select one pair and use that. That selection is automatic and mutually understood, governed by additional information that was included in the SIP request and response SDP content. Media flow could begin at this time.

7) Device B sends a 200 OK for the original INVITE, whose SDP content c/m lines contain the receive IP address/port number for itself as selected in (6), followed by an ACK from device A.

8) If the IP address/port number information in the c/m lines of the SDP offer in the INVITE happens to match those selected address arrived at in (6) for device A, then no

further steps are taken; *otherwise,* device A may send a re-INVITE whose SDP offer c/m lines contain the selected media address information.[7]

## 15 SIP Message Boundary Testing

SIP-based messages are text-based, can be relatively long, and may contain a multiplicity of fields, tags, etc. in the various headers. Within the SIP syntax, there can also be variations in the way the same information is contained in a message. It would be prudent to demonstrate the ability of each network to robustly and efficiently process long, unusual, and/or complicated, but legitimate, SIP messages received from another network, and to handle allowed variations in information representation; and to detect the limits, if any, of a network to be able to do so.

The recommended way to exercise this SIP 'message boundary testing' is to use VoIP Test Set equipment on either the network side or the customer equipment side of the interface, where the Test Sets can run test scripts, and where the relevant tests would involve SIP signaling and/or call establishment across the UNI. It should be recognized that there is the possibility that, when the Test Set equipment is not directly connected to the UNI, intermediate equipment may normalize/homogenize some of the possible variations in SIP content produced by the Test Set equipment.

One source of examples for such test scripts is the set of so-called SIP "torture tests", described in an IETF information draft[8], and born out of SIPIT[9] (SIP Interoperability Testing) events. Another example of a set of SIP tests is the PROTOS suite[10]. Note: Each of those contains examples of messages that are legitimate and others that are not. The latter is out of the scope of this TR.

## 16 Testing Organization

This TR does not recommend any particular levels of aggregation or sequencing for conducting tests. That is because the best testing strategy may be different depending on various factors, especially the level of confidence that testing will be successful. For example, that level of confidence may depend on whether the network provider or CE operator is testing the interface for the first time or the Nth time, or whether the whole interface is being tested or only increments to prior testing. Furthermore, the degree to which there are different perceived levels of risk/concern may vary by specific area (e.g., a simple voice SIP call versus one involving a SIP REFER interaction, or a voice call versus a fax call, or a fax call using T.30 versus T.38, or the use of UDP versus SCTP for SIP signaling transport, or the workings of RTCP, etc.).

## 17 Security

### 17.1 Encryption

When employed, encryption must be working correctly for other tests to succeed, so there are no recommended tests for it provided here. (If some other test does *not* succeed, trouble-shooting should take into account that faulty/absent encryption on one side could be a cause.)

Encryption between the User and the Network, as described in the SIP UNI ANS, may be facilitated by:

[7] Useful for ensuring that 'middleboxes' in the SIP signaling path have the right media IP addresses, for purposes related to QoS, diagnostics, monitoring, firewalling, etc.

[8] http://www.ietf.org/rfc/rfc4475.txt

[9] http://www.sipit.net

[10] http://www.ee.oulu.fi/research/ouspg/protos/testing/c07/sip/

- Transport Layer Security (TLS) over TCP for transport-layer encryption and authentication, as per RFC 4346
  - also Mutual TLS as in RFC 2246, where both sides of the interface exchange certificates; used with an Enterprise on the User side of the UNI, and *not* with a Device on the User side of the UNI

- IPsec for encryption and authentication at the network layer, as per RFCs 4301 through 4309
  - encrypts SIP messages sent across the UNI using either TCP or UDP for transport

- MD5 Message Digest for authentication of SIP request challenges as per RFCs 3261, 1321;
  - used with a Device on the User side of the UNI and *not* with an Enterprise on the User side of the UNI

- Secure RTP for encrypting media, as per RFC 3711

### 17.2 Authentication of SIP Requests

The network side may demand authentication for SIP requests[11] from the user side of the UNI. Authentication steps are not shown in other sections of this document. The incremental authentication steps are described here.

The Digest method of authentication described in RFC 3261 would be used. The particular credentials used may depend on the type of the request, e.g., a REGISTER request for device configuration or an INVITE for initiating a call. Generically, the incremental message steps are

1) a response challenging the initial request message
2) resending the original request message, but with the necessary credentials

In response to an initial request from the user side of the UNI, the network side may respond with a challenge of 401 Unauthorized or 407 Proxy-Authentication Required.

The message flow, depending on the type of challenge response, would be

*either*

| <u>User</u> | <u>Network</u> |
|---|---|
| Initial request → | |
| | ← 401 Unauthorized with a WWW-Authenticate header |
| Same request →<br>but now including credentials<br>in the Authorization header | |
| | ← response (resumption of normal message flow) |

*or*

| <u>User</u> | <u>Network</u> |
|---|---|
| Initial request → | |
| | ← 407 Proxy Authentication Required with a<br>Proxy-Authenticate header |
| Same request → | |

[11] Except for ACK and CANCEL, as per RFC 3261.

but now including credentials
in the Proxy-Authorization header

← response (resumption of normal message flow)

## 18 References

For the specific protocols, extensions, versions, etc. specified as part of the SIP UNI ANS,[1] the reader is referred to that document.

[1] IP Device (SIP UA) to Network Interface Standard ATIS-1000028.2008[12]

[12] This document is available from the Alliance for Telecommunications Industry Solutions (ATIS), 1200 G Street N.W., Suite 500, Washington, DC 20005. < https://www.atis.org/docstore/default.aspx >