IWD – SIP Telephony

Content

1 INTRODUCTION 4

1.1 PURPOSE 4
1.1.1 IMS ISC 5
1.2 VERSIONS 5
1.3 RELATED PROTOCOLS 5
1.4 SUPPORTED STANDARDS 5
1.4.1 RFC 2046 "Multipurpose Internet Mail Extension (MIME) Part Two: Media Types" 5
1.4.2 RFC 2976 "The SIP INFO Method" 5
1.4.3 RFC 3261 "SIP: Session Initiation Protocol" 6
1.4.4 RFC 3262 "Reliability of Provisional Responses in the Session Initiation Protocol (SIP)" 7
1.4.5 RFC 3264 "An Offer/Answer Model with the Session Description Protocol (SDP)" 7
1.4.6 RFC 3323 "A Privacy Mechanism for the Session Initiation Protocol" 7
1.4.7 RFC 3325 "Private Extension to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks" 8
1.4.8 RFC 3326 "The Reason Header Field for the Session Initiation Protocol (SIP)" 8
1.4.9 RFC 3455 "Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP) 8
1.4.10 RFC 3966 "The tel URI for Telephone Numbers" 8
1.4.11 RFC 4244 "An Extension to the Session Initiation Protocol (SIP) for Request History Information" 8
1.4.12 RFC 4458 "Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR)" 9
1.4.13 RFC 4566 "SDP: Session Description Protocol" 9
1.4.14 draft-ietf-sip-privacy-04 "SIP Extensions for Caller Identity and Privacy" 9
1.4.15 draft-levin-mmusic-xml-schema-media-control-03 "XML Schema for Media Control" 10
1.4.16 draft-levy-sip-diversion-08 "Diversion Indication in SIP" 10
1.4.17 ETSI TS 183 004 V1.1.1 (2006-04) "Communication Diversion (CDIV)" 10
1.4.18 ETSI TS 183 007 V1.1.1 (2006-03) "Originating Identification Presentation (OIP) and Originating Identification Restriction (OIR)" 10
1.4.19 GTD 11
1.4.20 RFC 4585 and draft-ietf-avt-avpf-ccm-10 11

2 DEFINITIONS 11

2.1 EXTERNAL CLIENT 11
2.2 SYSTEM 11
2.3 SESSION 11

3 SESSION 12

3.1 STATES 12
3.2 INBOUND SESSION 12

3.2.1 Inbound INVITE with SDP offer 12
3.2.2 Inbound INVITE without SDP offer 13
3.2.3 Inbound session for which the System plays early media 14
3.3 OUTBOUND SESSION 16
3.3.1 Outbound session when no early media is received 16
3.3.2 Outbound session for which the System receives early media 18
3.4 SESSION MEDIA 19

4 PROCEDURES 20

4.1 RENEGOTIATE A SESSION 20
4.1.1 Initiated by System 20
4.1.2 Received by System 20
4.2 TERMINATE A SESSION 20
4.2.1 Initiated by System 20
4.2.2 Received by System 21
4.3 CANCEL A SESSION CREATION REQUEST 21
4.3.1 Initiated by System 21
4.3.2 Received by System 22
4.4 QUERY FOR CAPABILITIES 22
4.4.1 Initiated by System 22
4.4.2 Received by System 22
4.5 REQUEST VIDEO FAST UPDATE 22
4.5.1 Initiated by System 23
4.5.2 Received by System 23

5 MESSAGES 23

5.1 SIP REQUESTS 23
5.1.1 General request handling 23
5.1.2 INVITE 25
5.1.3 ACK 26
5.1.4 CANCEL 26
5.1.5 BYE 26
5.1.6 OPTIONS 27
5.1.7 PRACK 27
5.1.8 REGISTER 27
5.1.9 INFO 27
5.2 SIP RESPONSES 27
5.2.1 Receiving responses 27
5.2.2 Sending responses 27

6 HEADER FIELDS 31

6.1 ACCEPT 34
6.2 ACCEPT-ENCODING 34
6.3 ACCEPT-LANGUAGE 35
6.4 ALLOW 35
6.5 CALL-ID 35
6.6 CALL-INFO 35
6.7 CONTACT 35
6.8 CONTENT-DISPOSITION 36
6.9 CONTENT-ENCODING 36
6.10 CONTENT-LANGUAGE 36
6.11 CONTENT-LENGTH 36
6.12 CONTENT-TYPE 37
6.13 CSEQ 37

6.14 DIVERSION 37
6.15 EXPERIENCED-OPERATIONAL-STATUS 38
6.16 EXPIRES 38
6.17 FROM 38
6.18 HISTORY-INFO 39
6.19 MAX-FORWARDS 40
6.20 P-ASSERTED-IDENTITY 40
6.21 P-CHARGING-VECTOR 41
6.22 PRIVACY 42
6.23 RACK 42
6.24 REASON 42
6.25 RECORD-ROUTE 43
6.26 REMOTE-PARTY-ID 43
6.27 REQUEST-URI 44
6.28 REQUIRE 44
6.29 ROUTE 44
6.30 RSEQ 45
6.31 SUPPORTED 45
6.32 TIMESTAMP 45
6.33 TO 45
6.34 UNSUPPORTED 46
6.35 USER-AGENT 46
6.36 VIA 46
6.37 WARNING 47

7 BODY CONTENT 48

7.1 SDP 49
7.1.1 Fields 49
7.1.2 Attributes 52
7.1.3 SDP examples 55
7.2 MEDIA CONTROL 57
7.3 GTD 58
7.3.1 Fields 58

8 PROPERTIES 59

8.1 SUPPORTED URI SCHEMES 59
8.2 ERROR HANDLING 59
8.3 TIMING PROPERTIES 59
8.3.1 SIP timing properties 59
8.3.2 System specific timing properties 60
8.4 ABANDONED SESSION DETECTION 60
8.5 RETRIEVAL OF CALL PARTY IDENTIFICATION FROM URI 61
8.5.1 User and Host 61
8.5.2 Telephone Number 61
8.6 RETRIEVAL OF CALL PARAMETERS FOR TESTING PURPOSE 61
8.6.1 Called party 62
8.6.2 Calling party 62
8.6.3 Redirecting party 62

9 REFERENCES 63

10 TERMINOLOGY 65

History

Version	Date	Adjustments
A	2007-06-11	First version. Replaces the MAS specific parts of the system document IWD – SIP: 7/IWD-1/HDB 101 02.
B	2008-08-05	Updated chapter 7.1 regarding ptime, maxptime and fmtp. Updated bandwidth parameter. Updated for FE27, Orange, video fast update. Updated with GTD RNI from Meteor merge. (ematsha/estberg)

1 Introduction

1.1 Purpose

This document specifies the SIP (Session Initiation Protocol) interface used between the access network and the SIP telephony functions in the CompEdge system (from here on referred to as the System, see section 2.2).

The following access networks are supported:

- ISUP through an ISUP/SIP interworking gateway
- IMS over the ISC interface
- VoIP through a soft switch

One and the same interface is used for all types of access networks. Therefore, this document contains for example not only the supported IMS ISC interface but also details supported in order to work well with VoIP networks and ISUP/SIP interworking gateways.

This interface is based on the SIP protocol version 2.0 specified in [4]. Other versions of the SIP protocol are not supported.

This interface assumes that the System is placed within a trusted network since no emphasis has been placed on supporting security solutions in SIP.

This document only describes the parts of the SIP protocol that are supported and how, not the protocol itself. For a full understanding of the SIP protocol, see [4].

If not otherwise stated in section 1.4, all MUST and SHALL defined in the SIP specification are implemented. This document also tries to describe the SHOULD and MAY defined in the SIP specification that are implemented since the document describes the SIP behavior on a detailed level.

1.1.1 IMS ISC

The System acts as an IMS AS. This interface is thus based on the ISC interface specified in [18], [19], and [20].

This SIP interface may be used by clients over other IMS interfaces as well (such as for example Mr) if it complies with the clients expectations for that IMS interface.

Currently, not all mandatory parts of the above specifications are supported, e.g. the precondition SIP extension is not supported. See section 1.4 for the supported standards.

1.2 Versions

This paragraph is intentionally left blank.

1.3 Related Protocols

In SIP, SDP (Session Description Protocol) is used in SIP messages to carry information about the media details (such as the type of media, codec or sampling rate) of a session. SDP is specified in [14]. Within a SIP session the SDP is exchanged as described in the offer/answer model specified in [6].

1.4 Supported Standards

1.4.1 RFC 2046 "Multipurpose Internet Mail Extension (MIME) Part Two: Media Types"

This interface supports SIP message bodies containing MIME multipart as specified in [1].

The following parts of the specification are **not** supported:

- The *Content-Transfer-Encoding* header field. If received in a SIP message body it is ignored and the message content is interpreted as UTF-8 regardless of transfer encoding.

The MIME media types used by the System are listed in section 6.12.

1.4.2 RFC 2976 "The SIP INFO Method"

This interface supports the SIP INFO method specified in [3].

The following parts of the specification are **not** supported:

- Reception of a SIP INFO request. If received it is responded with a SIP 405 "Method Not Allowed" response. However if a session is bridged to another session, a received SIP INFO request is accepted and forwarded to the other session. In a similar manner, the response to the SIP INFO request is tunneled back to the original requester.

How the SIP INFO method is used by the System is described in section 4.5.

1.4.3 RFC 3261 "SIP: Session Initiation Protocol"

This interface supports the SIP protocol version 2.0 specified in [4].

The following parts of the specification are **not** supported:

1.4.3.1 Multicast conferences and forking

If an outbound SIP INVITE is forked, responses after the first one received are ignored.

The fact that forking is not supported is not a problem when the System is connected to an ISUP/SIP interworking gateway since forking is never done then.

For other access networks (IMS or VoIP) the system currently does NOT have any support for avoiding forking. If forking is performed by any of these access networks, the System will not behave according to standard.

1.4.3.2 Authentication

If an outbound SIP INVITE is responded to with a SIP 407 "Proxy Authentication Required" no new SIP INVITE with credentials are sent.

Credentials are never added to SIP messages sent by the System. Received credentials are ignored.

1.4.3.3 Authorization

If an outbound SIP INVITE is responded to with a SIP 401 "Unauthorized", no new SIP INVITE with credentials are sent.

Credentials are never added to SIP messages sent by the System. Received credentials are ignored.

1.4.3.4 Content encoding

No content encodings are supported. If a SIP request is received with an encoded mandatory content, the SIP request is rejected since the mandatory content cannot be parsed.

1.4.3.5 TCP

The TCP transport is not supported. No SIP messages can be sent or received over TCP.

1.4.3.6 TLS/SIPS

The TLS transport is not supported. No SIP messages can be received over TLS.

1.4.3.7 S/MIME

SIP message content for which S/MIME has been used (e.g. content of type "multipart/signed") cannot be parsed. If a SIP request is received with a mandatory S/MIME content, the SIP request is rejected since the mandatory content cannot be parsed.

1.4.3.8 Header fields contained in a URI

Header fields contained in a URI are ignored if received.

1.4.3.9 Re-negotiation of media (using re-INVITE)

Only some limited types of media re-negotiation are allowed. See section 4.1 for a description on the supported variants of media re-negotiation.

1.4.3.10 Alert-Info header field carrying ring tones

The *Alert-Info* header field is not supported. If received in a SIP message, it is ignored.

1.4.3.11 Updating the Timestamp header with request processing time

If the Timestamp header field is received in a SIP INVITE it is copied into the SIP 100 "Trying" response sent back. But, the processing time is not added to the *Timestamp* header as recommended in the standard.

1.4.3.12 Redirection of a session initiated by the System

If a SIP INVITE sent by the System is redirected, no new session will be initiated by the System.

1.4.4 RFC 3262 "Reliability of Provisional Responses in the Session Initiation Protocol (SIP)"

This interface supports the extension *100rel* specified in [5] which enables sending provisional responses reliably.

The available session creation procedures making use of *100rel* are illustrated in section 3.

1.4.5 RFC 3264 "An Offer/Answer Model with the Session Description Protocol (SDP)"

This interface supports the offer/answer model for SDP specified in [6].

The following parts of the specification are **not** supported:

- Setting the *time description* field in an SDP answer to the same value as in the SDP offer. The System always sets the time description to: *t=0 0*.
- An SDP received in a SIP OPTIONS request. If received, it is ignored and the SIP response will not contain an SDP.

1.4.6 RFC 3323 "A Privacy Mechanism for the Session Initiation Protocol"

This interface supports the privacy mechanisms specified in [7].

The following parts of the specifications are supported:

- To request network-provided privacy.
- The privacy preferences *header* and *id*.

The SIP *Privacy* header field is used as described in section 6.22.

1.4.7 RFC 3325 "Private Extension to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks"

This interface supports the asserted identity mechanisms specified in [8].

The following parts of the specifications are supported:

- The *P-Asserted-Identity* header field

The SIP *P-Asserted-Identity* header field is used as described in section 6.20.

1.4.8 RFC 3326 "The Reason Header Field for the Session Initiation Protocol (SIP)"

This interface supports the *Reason* header field specified in [9]. Currently only the "Q.850" protocol is supported and the *Reason* header field is only read, never set.

The *Reason* header field is used as described in section 6.24.

1.4.9 RFC 3455 "Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)

The P-header extension to SIP is specified in RFC 3455 [10]. How it shall be used for the IMS ISC interface is described in [20].

The following parts of the specifications are supported:

- The *P-Charging-Vector* header field

The SIP *P-Charging-Vector* header field is used as described in section 6.21.

1.4.10 RFC 3966 "The tel URI for Telephone Numbers"

This interface supports URIs for telephone calls as specified in [11].

1.4.11 RFC 4244 "An Extension to the Session Initiation Protocol (SIP) for Request History Information"

This interface supports the history information as specified in [12].

The following parts differ from what is suggested in the specification:

- Although the *Supported* header field in a received INVITE contains *histinfo*, no *History-Info* header field is included in any response to the INVITE.
- The redirecting reason is selected from the *History-Info* header field as specified for the TISPAN CDIV function (see [23]) which is in conflict with RFC 4244.

Since the System assumes to be located in a trusted network and thus TLS and authentications is not supported (see section 1.4.3), all security requirements defined in [12] is not supported.

The SIP *History-Info* header field is used as described in section 6.18.

1.4.12 RFC 4458 "Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR)"

This interface supports mapping of a SIP response code to a redirection reason as specified in [13]. Only the SIP response code mapping part of RFC 4458 (specified in section 2.2 of [13]) is supported and is used when parsing the SIP *History-Info* header field.

The *History-Info* header field is used as described in section 6.18.

1.4.13 RFC 4566 "SDP: Session Description Protocol"

This interface supports the SDP protocol version 0 specified in [14]. The following parts of that specification are **not** supported:

- Interpretation of the *time description* field. All values are accepted and ignored.
- Interpretation of the attributes *sdplang* and *lang*. Regardless of value of these attributes, the SDP is interpreted as written in US English.
- Interpretation of the attribute *framerate*. All values are accepted and ignored.
- Interpretation of the attribute *fmtp*. All values are accepted and ignored.
- Interpretation of the attribute *maxptime*. All values are accepted and ignored.
- Interpretation of the attribute *bandwidth* set to *CT*. All values are accepted and ignored.
- The attribute *bandwidth* set to *AS* on session level is only handled as a media level default value. No interpretation of session level bandwidth is supported.
- Non-contiguous ports for RTP/RTCP.

1.4.14 draft-ietf-sip-privacy-04 "SIP Extensions for Caller Identity and Privacy"

This interface supports the caller identity mechanism specified in [15].

The following parts of the specification are **not** supported:

- IP address privacy using the *Anonymity* header field.
- Adding a *Proxy-Require* header when requesting privacy using the *Remote-Party-ID* header.
- Sending the *Remote-Party-ID* header in SIP responses.
- Interpretation of parameters *rpi-screen* and *rpi-id-type*.
- The *other-user* value of "private".
- Removal of user identity in headers such as *To*, *From*, and *Contact* when requesting "uri" or "full" privacy.

The *Remote-Party-ID* header field is used as described in section 6.26.

1.4.15 draft-levin-mmusic-xml-schema-media-control-03 "XML Schema for Media Control"

This interface supports media control specified in [16].

The following parts of the specification are **not** supported:

- Responding to a media control request. A media control request received in a SIP INFO request will be handled as described in section 1.4.2. A media control request received in a SIP NOTIFY request will be responded with a SIP 501 "Not Implemented" response and no action will be taken.
- Sending media control commands other than Video Picture Fast Update. Only Video Picture Fast Update commands are sent by the System.

1.4.16 draft-levy-sip-diversion-08 "Diversion Indication in SIP"

This interface supports the diversion indication mechanism specified in [17].

How the specified *Diversion* header field is used is described in section 6.14.

1.4.17 ETSI TS 183 004 V1.1.1 (2006-04) "Communication Diversion (CDIV)"

This interface supports communication diversion as specified in [23].

The following parts differ from what is suggested in the specification:

- Although a received INVITE contains a *History-Info* header field, no *History-Info* header field is included in any response to the INVITE.

While the CDIV specification mandates that the draft draft-jennings-sip-voicemail-uri shall be used for the interpretation of the cause parameter, the System uses RFC 4458 [13] instead. RFC 4458 is the RFC that was written after the draft was approved.

1.4.18 ETSI TS 183 007 V1.1.1 (2006-03) "Originating Identification Presentation (OIP) and Originating Identification Restriction (OIR)"

This interface supports originating identification presentation and restriction as specified in [24] with one exception.

According to the OIP/OIR specification, the *From* header field should be set as follows when presentation is restricted:

```
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=_tag_
```

where _tag_ is the current dialog from tag.

However, the System sets the *From* header field as follows when presentation is restricted:

```
From: "Anonymous" <sip:_user_@_host_;user=phone>;tag=_tag_
```

where _user_ is the telephone number of the user, _host_ is the System UA host and _tag_ is the current dialog from tag.

Thus, the *From* header field will include end-user information. The System uses the *Privacy* header field instead to request the access network to remove the end-user information from common SIP header fields.

1.4.19 GTD

See section 7.3 for details.

1.4.20 RFC 4585 and draft-ietf-avt-avpf-ccm-10

This interface supports only one type of value for the RTCP-based feedback attribute ("rtcp-fb") as specified in [25] and [26]: the "Full Intra Request" (FIR).

This should be specified in a "RTP/AVPF" profile, similar to a previously specified "RTP/AVP" video profile. See example in section 7.1.3.

2 Definitions

This section defines some of the terms used in this document. Also, the terminology specified in [4] is used in this document.

2.1 External Client

External Client is the name used in this document to denote the external SIP gateway, SIP soft switch or IMS S-CSCF not part of the System.

2.2 System

System is the name used in this document to denote the CompEdge system.

When referring to a specific SIP function in the System, **System UA** will be used further on in this document.

The System UA is realized with the System component MAS.

2.3 Session

Session as defined here is the same as the session defined in [4] with the following clarifications.

A session is a peer-to-peer SIP relationship between two or more user agents that persists for some time. A session is a collection of participants, and streams of media between them, for the purpose of communication. A session is initiated using the SIP INVITE message, resulting in one or more SIP dialogs (see [4]). A dialog is communication between two parties. Since multicast conferencing is not supported (see section 1.4.3.1) currently one session contains one single dialog only.

SIP also contains out-of-dialog handling of SIP messages that are not related to a particular session.

This document describes how SIP messages are handled by the System both within and outside a session.

3 Session

This section describes how a session is created over SIP.

Apart from session creation, the following procedures are handled over SIP:

- Renegotiate a session
- Terminate a session
- Cancel a session creation request
- Query for capabilities (might be out-of-dialog procedure)
- Request video fast update

3.1 States

For a detailed description of the SIP protocol, see [4].

3.2 Inbound session

An inbound session is initiated by an External Client sending a SIP INVITE to the System.

The procedure of setting up an inbound session differs depending on if the INVITE contains an SDP offer or not, if the System wants to transmit early media or not and if provisional responses are sent reliably or not. The following alternative ways of creating an inbound session are illustrated below with sequence charts:

- Inbound INVITE with SDP offer
- Inbound INVITE without SDP offer
- Inbound session for which the System plays early media

In all the above alternatives, provisional responses may be sent reliably or unreliably.

Provisional responses are sent reliably by the System in the following situations:

- When a received SIP INVITE indicates that the *100rel* extension is required (using the *Require* header field).
- When a received SIP INVITE indicates that the *100rel* extension is supported but not required (using the *Supported* header field) and configuration of the System indicates that provisional responses should be sent reliably. The System configuration allows for provisional responses to be sent reliably or unreliably or to only send provisional responses with SDP information reliably.

3.2.1 Inbound INVITE with SDP offer

If the inbound INVITE contains an acceptable SDP offer and no early media is available at the System, the session is created using the procedure illustrated below.

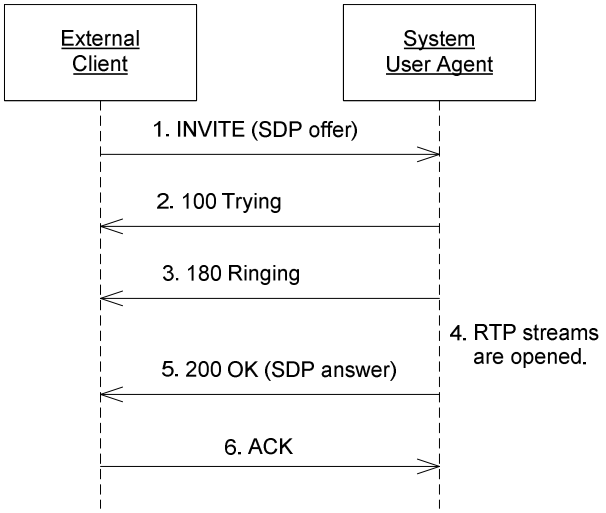

Figure 1 Inbound INVITE with SDP offer

If the SDP offer could not be parsed or if it is unacceptable, a SIP 488 "Not Acceptable Here" response is sent as described in section 5.2.2.16. Section 3.4 describes how to determine if the remote SDP offer is acceptable and how the SDP answer is created.

If the SIP 180 "Ringing" response shall be sent reliably, the session is created using the procedure illustrated below. Section 5.2.2 describes when provisional responses are sent reliably.

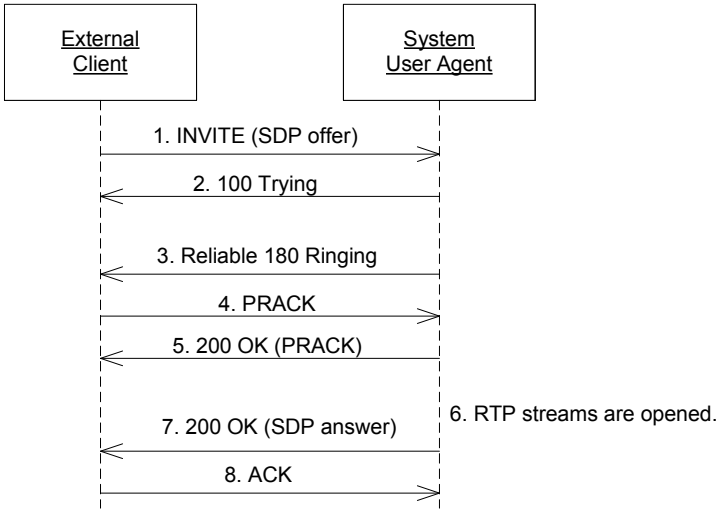

Figure 2 Inbound INVITE with SDP offer and 180 is sent reliably

3.2.2 Inbound INVITE without SDP offer

If the inbound INVITE does not contain an SDP offer the session is created using the procedure illustrated below.

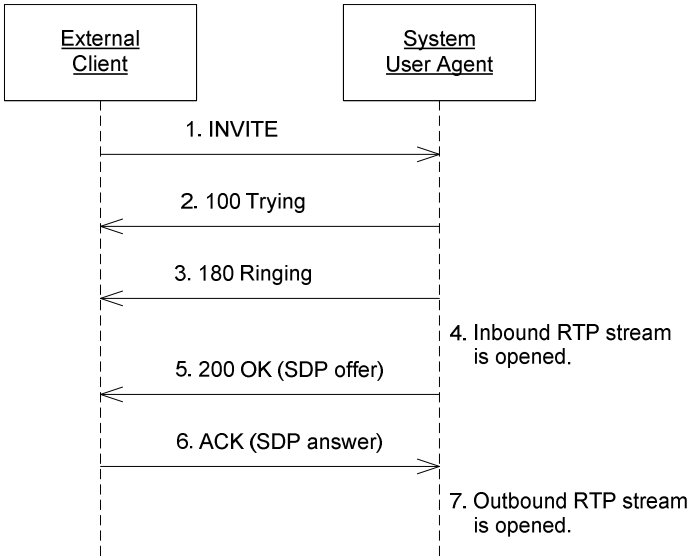

Figure 3 Inbound INVITE without SDP offer

If the SDP answer could not be parsed or if it is unacceptable, the session will be terminated immediately as described in section 4.2. Section 3.4 describes how to determine if the remote SDP answer is acceptable and how the SDP offer is created.

If the SIP 180 "Ringing" response shall be sent reliably, the session is created using the procedure illustrated below. Section 5.2.2 describes when provisional responses are sent reliably.

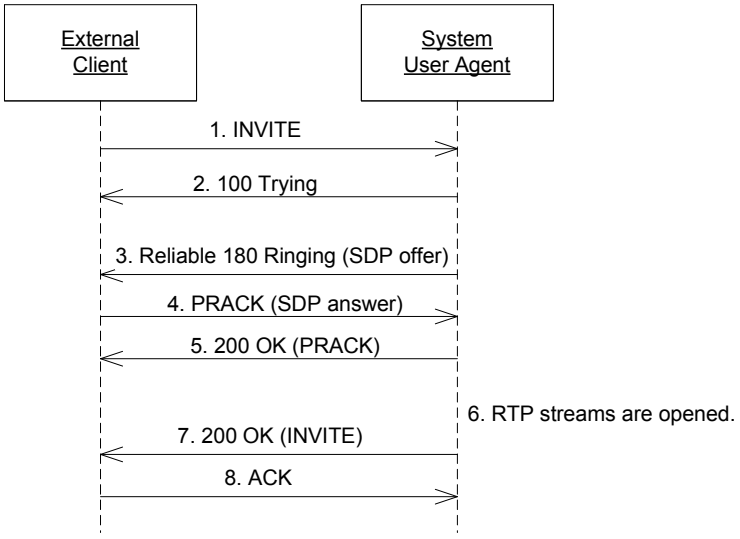

Figure 4 Inbound INVITE without SDP offer and 180 is sent reliably

3.2.3 Inbound session for which the System plays early media

Early media can only be played by the System during session creation if the inbound INVITE contains an SDP offer. The procedure is illustrated below.

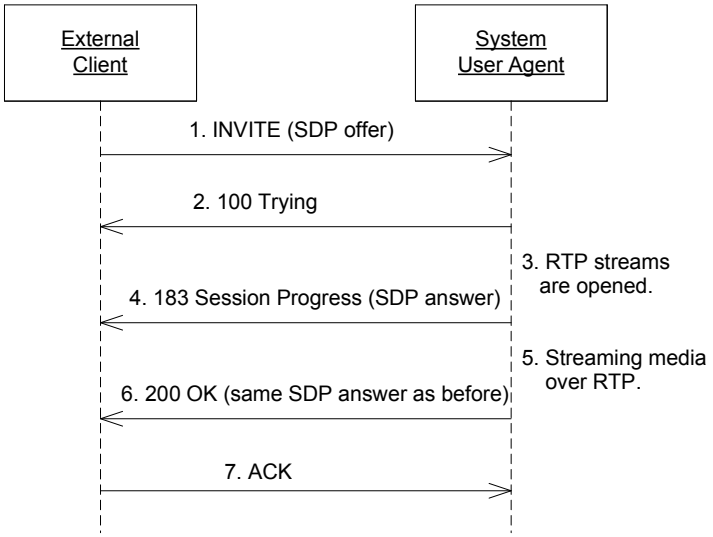

Figure 5 Inbound session with early media

If the SDP offer could not be parsed or if it is unacceptable, a SIP 488 "Not Acceptable Here" response is sent as described in section 5.2.2.16. Section 3.4 describes how to determine if the remote SDP offer is acceptable and how the SDP answer is created.

If the SIP 183 "Session Progress" response shall be sent reliably, the session is created using the procedure illustrated below. Section 5.2.2 describes when provisional responses are sent reliably.

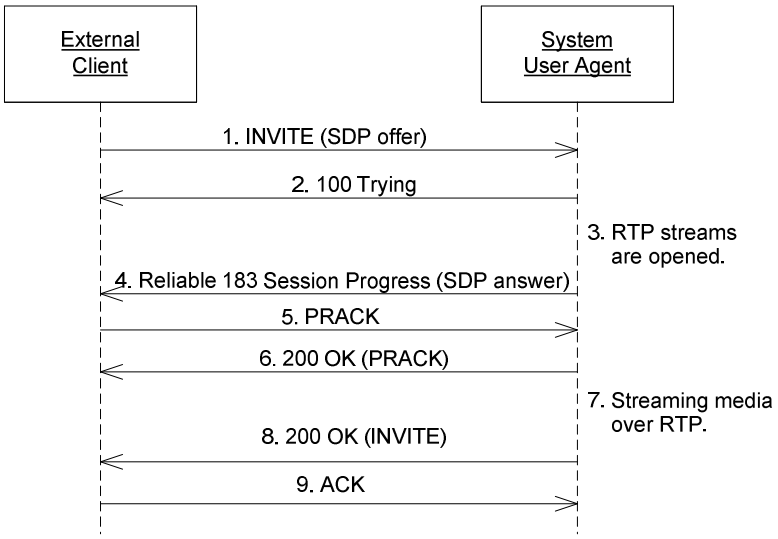

Figure 6 Inbound session with early media and 183 is sent reliably

In the scenario illustrated above, the external client may send a new SDP offer in the PRACK request in which case the System UA will include an SDP answer in the SIP 200 "OK" response to the SIP PRACK request. This is a re-negotiation of the previously negotiated media.

If the media negotiation is not allowed, the generated SDP answer will indicate that all included media streams shall be unused (see section 3.4) and the session

is terminated (as described in section 4.2) by the System UA after the ACK has been received. The supported types of media re-negotiations are described in section 4.1.

3.3 Outbound session

An outbound session is initiated by the System UA sending a SIP INVITE to the External Client.

The procedure of setting up an outbound session differs depending on if the External Client wants to transmit early media or not. The following alternative ways of creating an outbound session are illustrated below with sequence charts:

- Outbound session when no early media is received
- Outbound session for which the System receives early media

Generally, a SIP response to an outbound INVITE is handled as described below:

- **100, 180, 181, 182**:
  does not result in any action in the System UA.
- **Other 1xx response that contains no SDP**:
  is handled as a 180 response.
- **Other 1xx response that contains an SDP**:
  indicates that early media will be sent to the System UA.
- **2xx**:
  an ACK is sent by the System UA and the session creation is considered complete.
- **3xx - 6xx**:
  the session creation fails. No new session creation is initiated.

3.3.1 Outbound session when no early media is received

The outbound INVITE will contain an SDP offer. If the first SDP answer is received in the SIP 200 "OK" response, no early media will be streamed to the System UA. The procedure is illustrated below.

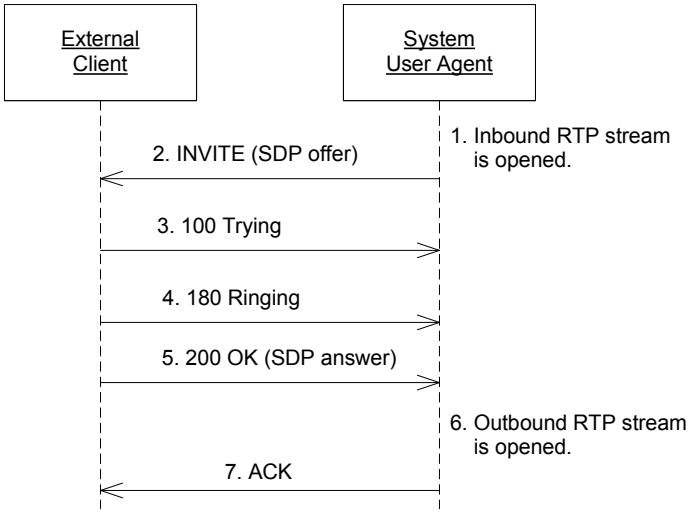

Figure 7 Outbound session without early media

If the SDP answer could not be parsed or if it is unacceptable, the session is terminated (as described in section 4.2) by the System UA after the ACK has been sent. Section 3.4 describes how to determine if the remote SDP answer is acceptable and how the SDP offer is created.

If the SIP 180 "Ringing" response was sent reliably, the session is created using the procedure illustrated below.

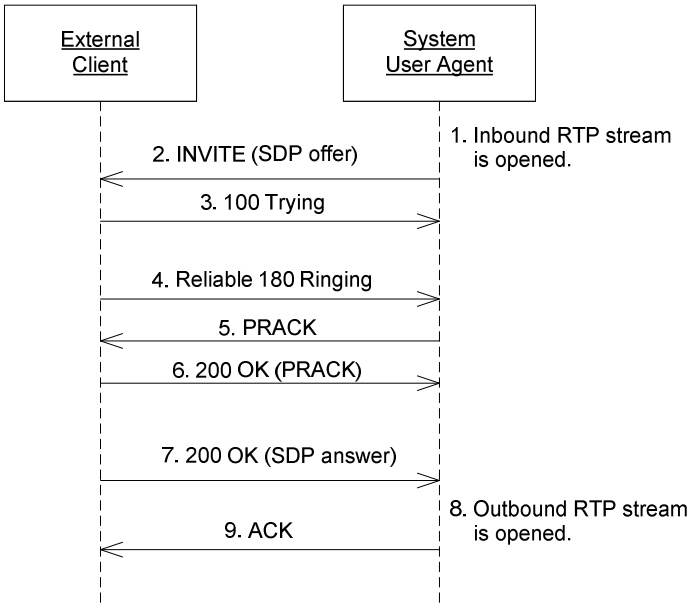

Figure 8 Outbound session without early media when 180 is sent reliably

3.3.2 Outbound session for which the System receives early media

The outbound INVITE will contain an SDP offer. If the first SDP answer is received in a SIP 183 "Session Progress" response (or equivalent) the System UA is prepared to receive early media. The procedure is illustrated below.

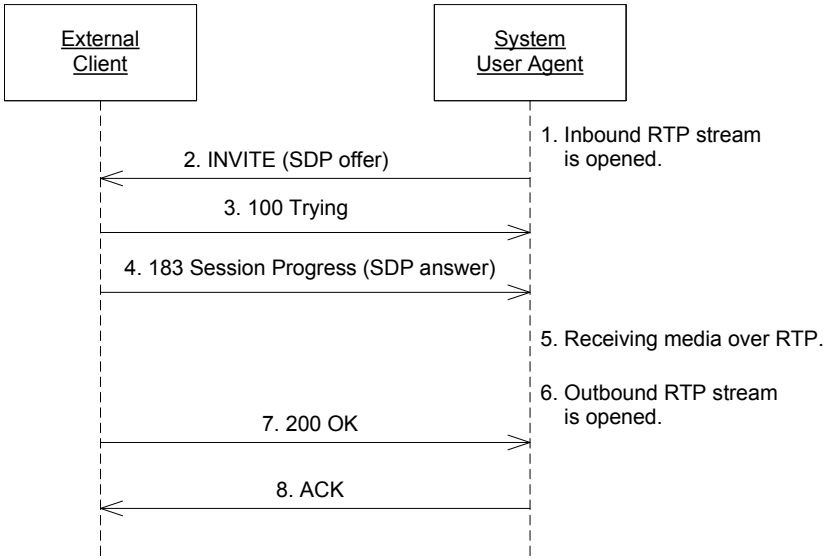

Figure 9 Outbound session with early media

If the SDP answer could not be parsed or if it is unacceptable, the System UA cancels the session (as described in section 4.3). Section 3.4 describes how to determine if the remote SDP answer is acceptable and how the SDP offer is created.

If the SIP 183 "Session Progress" response was sent reliably, the session is created using the procedure illustrated below.

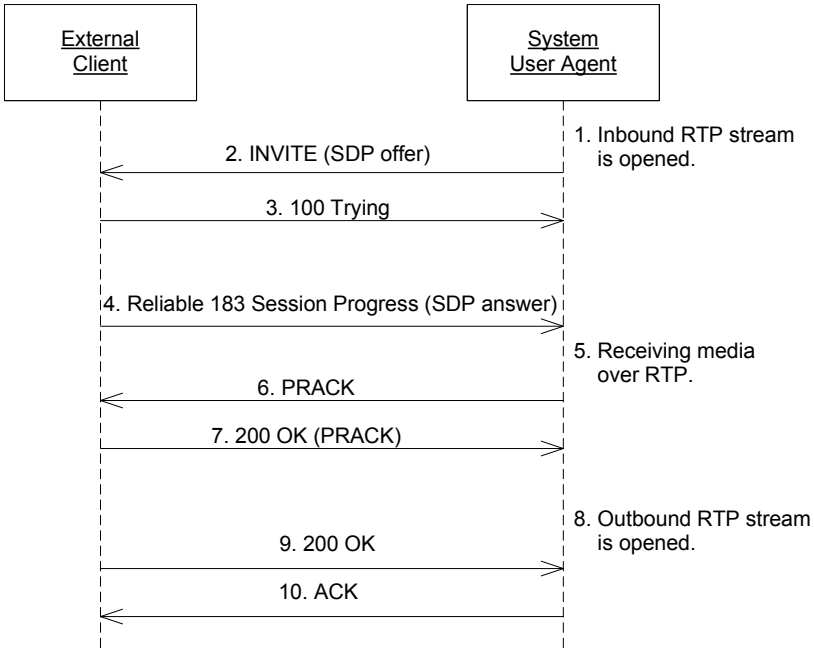

Figure 10 Outbound session with early media and 183 is sent reliably

3.4 Session media

The media format to use for a SIP session is determined based upon the SDP offer and answer.

An SDP offer/answer created by the System UA always consists of one audio codec and zero or one video codec depending on the type of session (i.e. voice or video). The audio and video codec are selected based upon the format in which the System wishes to store received media. They are configurable in the System.

For an outbound session, whether to create a voice or video session is determined based upon end-user activities. For an inbound session, the type of session to create is given by the SDP offer in the SIP INVITE. If the SDP offer contains video media a video session is created, otherwise a voice session is created. If no SDP offer was included in the SIP INVITE the type is determined based upon the Call-Info header field (as described in section 6.6) if such exists, or otherwise selected from configuration of the System.

An SDP offer/answer from an External Client is accepted by the System UA if the following applies:

- The external SDP offer/answer contains all media formats the System UA might use when streaming media.
- The session type indicated by an external SDP answer matches the session type of the Systems SDP offer. (The session type can be video or voice and depends upon if video is included or not as media format in the SDP.)
- The SDP offer does not try to re-negotiate a feature for which re-negotiation is not supported. The supported types of media re-negotiations are described in section 4.1.

When the System UA receives an SDP offer that is unacceptable but an SDP answer must be generated and sent before the call can be disconnected, the System UA copies the SDP offer and sets all media ports to zero to indicate that they shall not be used. This is all according to the SDP offer/answer model specified in [6].

4 Procedures

This section describes all non-session creating procedures. This section only describes procedure specific behavior and not general SIP message handling which is described in section 5.

4.1 Renegotiate a session

A session can be re-negotiated with a SIP re-INVITE request as described in section 14 of [4]. Other means for re-negotiation (such as for example SIP UPDATE) are not supported.

The following re-negotiations are allowed by the System:

- An SDP offer identical to the previously received remote SDP.

4.1.1 Initiated by System

The System UA never initiates a re-negotiation.

4.1.2 Received by System

A SIP re-INVITE received while no final response has been sent or received (depending on session direction) for the initial INVITE is rejected with the SIP 491 "Request Pending" response as described in section 5.2.2.17.

A SIP re-INVITE received for an established session containing a media re-negotiation that is allowed by the System is accepted with a SIP 200 "Ok" response.

A SIP re-INVITE received for an established session containing a media re-negotiation that is unsupported by the System is rejected with the SIP 488 "Not Acceptable Here" response as described in section 5.2.2.16.

4.2 Terminate a session

A session is terminated with a SIP BYE request as described in section 15 of [4]. When a SIP BYE request is sent or received by the System UA, all open streams in the session are closed.

4.2.1 Initiated by System

Depending on the state of a session, the session can either be terminated using BYE or canceled using CANCEL (if it is an outbound session). The reasons for termination or cancellation are mainly the same. A session for which a CANCEL has been sent might (due to timing issues) become established before the CANCEL has been received at the other end and in this situation a BYE request

will be sent to terminate the session. For information about System initiated cancellation, see section 4.3.1.

The System UA terminates the session in the following situations:

- Due to end-user activities (e.g. certain input of DTMF).
- Max duration for an outbound session has been reached, see section 8.3.2.1. Can only occur if a max duration has been configured in the System.
- Surveillance of the inbound stream shows no activity for a configurable amount of time. This surveillance is further described in section 8.4.
- If an SDP answer from the External Client cannot be parsed, is unacceptable or is missing.
- If outbound RTP media streams cannot be created.
- If a SIP 481 "Call/Transaction Does Not Exist" response or a SIP 408 "Request Timeout" response is received for a SIP request sent by the System UA.
- The maximum time to wait for a session to be created has been reached, see section 8.3.2.2. This timer is configurable in the System.
- Due to manual operation and maintenance.

4.2.2 Received by System

If a SIP BYE request is received within a session a SIP 200 "OK" response is sent by the System UA, otherwise a SIP 481 "Call/Transaction Does Not Exist" response is sent.

If a SIP BYE request is received for an inbound session while the System UA has not yet sent a final response for the INVITE, a SIP 487 "Request Terminated" is sent as response to the INVITE.

4.3 Cancel a session creation request

A session creation request can be canceled with a SIP CANCEL request as described in section 9 of [4]. When a SIP CANCEL request is sent or received by the System UA, all open streams in the session are closed.

4.3.1 Initiated by System

The System UA cancels an outbound session in the following situations:

- Due to end-user activities (e.g. certain input of DTMF).
- Surveillance of the inbound stream shows no activity for a configurable amount of time. This surveillance is further described in section 8.4.
- If an SDP answer from the External Client cannot be parsed or is unacceptable.
- If outbound RTP media streams cannot be created.

- If a SIP 481 "Call/Transaction Does Not Exist" response or a SIP 408 "Request Timeout" response is received for a SIP request sent by the System UA.
- The maximum time to wait for a session to be created has been reached, see section 8.3.2.2. This timer is configurable in the System.
- Due to manual operation and maintenance

4.3.2 Received by System

If a SIP CANCEL request is received for an inbound session for which the System has sent a final response no cancellation is possible since the session creation procedure has completed. A SIP "OK" response is sent for the SIP CANCEL request.

If a SIP CANCEL request is received for an inbound session while the System UA has not yet sent a final response for the INVITE the session is canceled and a SIP 487 "Request Terminated" response is sent for the INVITE.

4.4 Query for capabilities

Capabilities of a SIP UA are queried using a SIP OPTIONS request as described in section 11 of [4].

4.4.1 Initiated by System

A SIP OPTIONS request is never sent by the System UA.

4.4.2 Received by System

If a SIP OPTIONS request is received by the System UA within a session a SIP 200 "OK" response is sent by the System UA.

If a SIP OPTIONS request is received by the System UA outside of a session, the response sent is the same that would have been chosen had the request been an INVITE. Thus if an INVITE would have been accepted at the time an OPTIONS is received a SIP 200 "OK" response is sent, otherwise a SIP 503 "Service Unavailable" response is sent. However, if the system is experiencing a load close to the maximum allowed, a received SIP OPTIONS request will be rejected with a SIP 503 "Service Unavailable" response a bit earlier than had it been a SIP INVITE request. The reason for doing this is to send out an early warning that the System UA is about to become overloaded.

The response to an OPTIONS request contains the System UA capabilities with regards to SIP only. No SDP is included in the response (see section 1.4.5 for not supported features).

4.5 Request video fast update

A SIP user agent can manage a video stream using a SIP INFO request (see [3]) as described in [16]. Only the Video Picture Fast Update command is supported and can be sent by the System UA in a SIP INFO request. The System UA is not

able to generate a picture update and therefore is not able to respond to a Video Picture Fast Update request.

4.5.1 Initiated by System

The System UA will send a Video Picture Fast Update command using SIP INFO when the System is about to start recording the incoming media of a video stream.

4.5.2 Received by System

Normally, reception of a SIP INFO request is not supported. If received it is responded to with a SIP 405 "Method Not Allowed" response. However if a session is bridged to another session, a SIP INFO request received by a System UA is forwarded to the other session. The response to the SIP INFO request is tunneled in a similar manner and sent to the original requester.

5 Messages

The supported SIP messages are described in [3], [4] and [5].

This section describes how a specific SIP message is handled or generated by the System. It does not define or describe the content of the SIP request.

For details on the content of header fields in the messages, see section 6.

5.1 SIP Requests

This section describes how SIP requests are received by the System.

For information on the content of SIP requests sent by the System, see section 6.

5.1.1 General request handling

All SIP requests are validated according to the rules defined in [4]. If a request is considered invalid an error response is sent (unless the request is an ACK for which no response is sent) and the request is discarded. All validations are described in the sections below and listed in the order they occur.

5.1.1.1 Session not found

If the request indicates being sent within a session but the session or transaction cannot be found, a SIP 481 "Call/Transaction Does Not Exist" response is sent by the System.

5.1.1.2 Request type validation

If the request method is known but reception is not supported a SIP 405 "Method Not Allowed" response is sent. This occurs if:

- a SIP REGISTER request is sent to a System UA
- a SIP INFO request is sent to a System UA for a session that is not bridged to another session

If the request method is unknown a SIP 501 "Not Implemented" response is sent. This occurs for all methods other than INVITE, ACK, BYE, CANCEL, OPTIONS, REGISTER, INFO, and PRACK.

5.1.1.3 Address validation

If the *To* header is invalid a SIP 403 "Forbidden" response is sent. Currently all *To* headers are supported.

If the URI scheme used in the *Request-URI* is unsupported a SIP 416 "Unsupported URI Scheme" response is sent. Currently all URI schemes are allowed in the *Request-URI*.

If the *Request-URI* is invalid a SIP 404 "Not Found" response is sent. Currently all *Request-URI* values are supported.

5.1.1.4 Loopback detection

If a loopback is detected (as described in [4]) by the System UA a SIP 482 "Loop Detected" response is sent.

5.1.1.5 Required extensions validation

The *Require* header is inspected in all requests except ACK and CANCEL (in which the header should be ignored). If the *Require* header contains an extension not supported by the System a SIP 420 "Bad Extension" response is sent. The supported extensions are described in section 6.31.

5.1.1.6 Content validation

For all body parts in the SIP request a validation is made of the content if the content disposition indicates that the body part is required (see section 6.8). If the content of a required body part is invalid, a SIP 415 "Unsupported Media Type" response is sent.

The content is invalid due to one of the following reasons:

- The content language is unsupported. The supported content languages are described in section 6.10.
- The content encoding is unsupported. The supported content encodings are described in section 6.9.
- The content type is unsupported. The supported content types are described in section 6.12.
- The *charset* attribute in the content type is unsupported. The supported character sets are described in section 6.12.

5.1.1.7 Version validation

The version in the request line is inspected in all requests. If the version is not supported by the System a SIP 505 "Version Not Supported" response is sent. Currently only version "sip/2.0" is supported.

5.1.2 INVITE

When a SIP INVITE has been received and validated ok by the System UA the request is handled as described in section 3.2. The following information is retrieved from the request:

- Call type
- SDP offer
- Called party
- Calling party
- Redirecting party

If the parameter *test=on* is found in the *Request-URI* or (if not there) in the *To* header field, the called, calling and redirecting parties may be retrieved as described in section 8.6 if the System UA is configured to allow it. This is not the common retrieval of the call party information; it is used solely for testing purposes. Default behavior of the System UA is to not allow the above but to retrieve the call party information as described below.

Currently, the only supported format for call parties is an E.164 telephone number that can be included either in a SIP URI or in a TEL URI.

5.1.2.1 Call type

The call type is determined from the *Call-Info* header field if present as described in section 6.6. The call type is used to determine which type of session that shall be created (voice or video) if the received INVITE lacks an SDP offer (see section 3.4).

5.1.2.2 SDP offer

The last found body part with content type *application/sdp* is retrieved as SDP offer. For information on what parts of the SDP that are used by the System UA, see section 7.1.

5.1.2.3 Called party

The called party is retrieved from the *Request-URI* as described in section 6.27.

The called party may either be an end-user identity or a public service identity (PSI) identifying a specific service in the system. While the end-user identity must be a telephone number, a public service identity may be either a name or a telephone number. The services available in the System depend on the specific customer installation.

5.1.2.4 Calling party

The calling party is retrieved in the order listed below.

1. If the *P-Asserted-Identity* header field is present, the calling party is retrieved from the *P-Asserted-Identity* and *Privacy* header fields as described in sections 6.20 and 6.22.

2. Otherwise, if the *Remote-Party-ID* header field is present, the calling party is retrieved from the *Remote-Party-ID* header field as described in section 6.26.
3. Otherwise, the calling party is retrieved from the *From* and *Privacy* header fields as described in sections 6.17 and 6.22.

If a GTD is present in the INVITE request, the number completion information is extracted. See section 7.3.

When an INVITE request is initiated by the System UA, all the headers listed above will be added containing the calling party information. However, the System UA may be instructed (using configuration) to exclude a specific header field such as the *Remote-Party-ID* header field. This feature has been added to ensure that no private information is sent to an access network that are unable to understand a specific header and therefore cannot make it anonymous or remove it. Currently only the header fields *Remote-Party-ID* and *P-Asserted-Identity* can be excluded in configuration.

5.1.2.5 Redirecting party

The redirecting party is retrieved in the order listed below.

1. If the *Diversion* header field is present, the redirecting party is retrieved from the *Diversion* header field as described in section 6.14.
2. Otherwise, if the *History-Info* header field is present, the redirecting party is retrieved from the *History-Info* header field as described in section 6.18.

5.1.3 ACK

When a SIP ACK request has been received and validated ok by the System UA the request is handled as described in section 3.2. The following information is retrieved from the request:

- SDP answer (for the session creating procedure illustrated in section 3.2.2)

5.1.3.1 SDP answer

The last found body part with content type *application/sdp* is retrieved as SDP answer. For information on what parts of the SDP that are used by the System UA, see section 7.1.

5.1.4 CANCEL

When a SIP CANCEL request is received, the Q.850 value of the *Reason* header field (if present) is used as release reason. For details on the *Reason* header field, see section 6.24.

The request is otherwise handled as described in section 4.3.2.

5.1.5 BYE

When a SIP BYE request is received, the Q.850 value of the *Reason* header field (if present) is used as release reason. For details on the *Reason* header field, see section 6.24.

The request is otherwise handled as described in section 4.2.2.

5.1.6 OPTIONS

After validation of the SIP OPTIONS request, no further header fields are investigated. The request is handled as described in section 4.4.

5.1.7 PRACK

After validation of the SIP PRACK request, the request is checked to see if it matches the previously sent reliable provisional response as described in [5]. No further header fields are investigated in the PRACK request.

5.1.8 REGISTER

A SIP REGISTER request is rejected during request validation with a SIP 405 "Method Not Allowed" response as described in section 5.1.1.2.

5.1.9 INFO

After validation of the SIP INFO request, no further header fields are investigated. The request is handled as described in section 4.5.

5.2 SIP responses

5.2.1 Receiving responses

This purpose of this section is to describe how SIP responses are received by the System.

When an outbound session is being rejected with a SIP error response, the release reason is retrieved in the following order:

1. If present, from the Q.850 value of the *Reason* header field.
2. Otherwise from the SIP response code.

5.2.2 Sending responses

This section describes the SIP responses sent by the System and the reason for it being sent. For information on the content of SIP responses sent by the System, see section 6.

Provisional responses are sent reliably by the System for the following reasons:

- The System UA receives a SIP INVITE requiring the use of the *100rel* extension.
- The System UA receives a SIP INVITE indicating support for the *100rel* extension and the System is configured to send provisional responses reliably. The System can be configured to use reliability for all responses, no responses or only for those responses that contain SDP information.

5.2.2.1 100 Trying

A SIP 100 "Trying" response is sent by the System UA when a received SIP INVITE has been verified ok and a session creation procedure is started as illustrated in section 3.2.

5.2.2.2 180 Ringing

A SIP 180 "Ringing" response is sent by the System UA during an inbound session creation when no early media shall be played, see sections 3.2.1 and 3.2.2.

5.2.2.3 183 Session Progress

A SIP 183 "Session Progress" response is sent by the System UA during an inbound session creation when early media shall be played, see section 3.2.3. The response will contain an SDP answer.

5.2.2.4 200 Ok

A SIP 200 "OK" response is sent by the System in the following situations:

- In a capability query procedure initiated by the External Client (see section 4.4.2) an ok response is sent for the OPTIONS request if received within a session or received out-of-dialog and the current state indicates that an INVITE would be accepted if received.
- In a session termination procedure initiated by the External Client (see section 4.2.2) an ok response is sent for the BYE request.
- In a session cancellation procedure initiated by the External Client (see section 4.3.2) an ok response is sent for the CANCEL request.
- In an inbound session creation procedure (see section 3.2).

5.2.2.5 400 Bad Request

A SIP 400 "Bad request" response is sent by the System UA if a received SIP request could not be parsed (unless the request is an ACK for which no response is sent).

5.2.2.6 403 Forbidden

A SIP 403 "Forbidden" response is sent by the System UA if the end-user service rejects an inbound INVITE request, e.g. if the caller is not allowed to access the system.

The response would also be sent if the *To* header field received in a SIP request is invalid as described in section 5.1.1.3.

The response is also sent for a PRACK request that is received by the System UA when not expected.

5.2.2.7 404 Not Found

A SIP 404 "Not Found" response would be sent by the System UA if the *Request-URI* received in a SIP request is invalid as described in section 5.1.1.3.

5.2.2.8 405 Method Not Allowed

A SIP 405 "Method Not Allowed" response is sent by the System UA if the received request method is known but reception is not supported. Situations when this occurs are described in section 5.1.1.2.

5.2.2.9 408 Request Timeout

A SIP 408 "Request Timeout" response is sent by the System UA if an inbound SIP INVITE is not accepted by the end-user service within a configurable amount of time, see section 8.3.2.3.

5.2.2.10 415 Unsupported Media Type

A SIP 415 "Unsupported Media Type" response is sent by the System UA for the following reasons:

- The content language of a required body part in a SIP request is unsupported. The supported content languages are described in section 6.10.
- The content encoding of a required body part in a SIP request is unsupported. The supported content encodings are described in section 6.9.
- The content type of a required body part in a SIP request is unsupported. The supported content types are described in section 6.12.
- The *charset* attribute in the content type of a required body part in a SIP request is unsupported. The supported character sets are described in section 6.12.

5.2.2.11 416 Unsupported URI Scheme

A SIP 416 "Unsupported URI Scheme" response would be sent by the System UA if the URI scheme used in the *Request-URI* of a received SIP request is unsupported as described in section 5.1.1.3.

5.2.2.12 420 Bad Extension

A SIP 420 "Bad Extension" response is sent by the System if a *Require* header included in a received request (other than ACK or CANCEL) indicates an unsupported extension. For supported extensions, see section 6.28.

5.2.2.13 481 Call/Transaction Does Not Exist

A SIP 481 "Call/Transaction Does Not Exist" response is sent by the System UA if it receives a SIP request for an unrecognized session.

5.2.2.14 482 Loop Detected

A SIP 482 "Loop Detected" response is sent by the System UA if a loop is detected as described in section 8.2.2.2 of [4].

5.2.2.15 487 Request Terminated

A SIP 487 "Request Terminated" response is sent by the System UA for an inbound INVITE request if the session creation procedure is terminated prior to connect due to one of the following reasons:

- Due to end-user activities (e.g. certain input of DTMF).
- A session termination procedure is initiated by the External Client (see section 4.2.2).

- A session cancellation procedure is initiated by the External Client (see section 4.3.2).
- The expiration time for the INVITE request (received in the *Expires* header field, see section 6.16) has elapsed.

5.2.2.16 488 Not Acceptable Here

A SIP 488 "Not Acceptable Here" response is sent by the System UA when an INVITE cannot be accepted due to one of the following reasons:

- The SDP offer received in the INVITE request cannot be parsed.
- The SDP offer received in the INVITE cannot be accepted, see sections 3.4 and 7.1.
- The INVITE is a re-INVITE and the suggested re-negotiation is not supported by the System.

5.2.2.17 491 Request Pending

A SIP 491 "Request Pending" response is sent by the System UA when a re-INVITE request is received while the initial session creating procedure has not completed.

5.2.2.18 500 Server Internal Error

A SIP 500 "Server Internal Error" response is sent by the System UA for an inbound INVITE request due to one of the following reasons:

- An unexpected internal error occurs to which there is no recovery.
- One of the sessions RTP stream cannot be created.
- An SDP offer/answer cannot be created.
- The end-user service cannot be loaded.

A SIP 500 "Server Internal Error" response is also sent by the System UA for a SIP request received out of sequence.

5.2.2.19 501 Not Implemented

A SIP 501 "Not Implemented" response is sent by the System UA when an unknown request is received, i.e. a request other than INVITE, ACK, BYE, CANCEL, OPTIONS, REGISTER, INFO, or PRACK.

5.2.2.20 503 Service Unavailable

A SIP 503 "Service Unavailable" response is sent by the System UA in the following situations:

- If an inbound INVITE cannot be accepted due to the operational state of the System UA, i.e. the System is under operation and maintenance and cannot take calls.
- If an inbound INVITE cannot be accepted since the System UA has reached its maximum capacity.

- In a capability query procedure initiated by an External Client (see section 4.4.2) a SIP 503 "Service Unavailable" response is sent for a SIP OPTIONS request if received out-of-dialog and the current state indicates that an INVITE would not be accepted if received (for the reasons listed above).

When this response is sent by the System UA, the reason phrase is exchanged from "Service Unavailable" to a string describing why the service is unavailable.

5.2.2.21 504 Server Time-out

A SIP 504 "Server Time-out" response is sent by the System UA if a PRACK request is expected to be received but is not received before SIP timer H expires.

5.2.2.22 505 Version Not Supported

A SIP 505 "Version Not Supported" response is sent by the System if the version in the request line of a received request is unsupported as described in section 5.1.1.7.

6 Header Fields

This section describes the SIP headers supported by the System. Non-supported headers will be ignored if received. The headers described are all defined in [4] unless otherwise stated.

The table below shows the SIP headers supported by the System. The table illustrates in which SIP messages headers are inserted by the System and from which messages headers are read by the System.

The following notes are used in the table for more detailed information:

- (R) indicates that the header is required when processing a message
- (C) indicates that the header is copied from the original request
- (U) indicates that the header is unused, i.e. not set or read, due to the fact that that's the way to indicate support for a certain functionality
- (B) indicates that a header is read if the message has a content
- (L) indicates that the header is included only if loopback prevention is required
- (REL) indicates that reliability is used for a provisional response

Table 1 Header Fields

SIP Header	Added by System to following requests	Added by System to following responses	Processed by System if received in following requests	Processed by System if received in following responses
Accept	INVITE CANCEL (C)	183 (INVITE) 200 (INVITE) 200 (OPTIONS) 415 (Any)		

SIP Header	Added by System to following requests	Added by System to following responses	Processed by System if received in following requests	Processed by System if received in following responses
Accept-Encoding	INVITE CANCEL (C)	183 (INVITE) 200 (INVITE) 200 (OPTIONS) 415 (Any)		
Accept-Language	(U)	(U)		
Allow	INVITE CANCEL (C)	183 (INVITE) 200 (INVITE) 200 (OPTIONS) 405 (Any)		
Call-ID	All	All (C)	All (R)	All (R)
Call-Info			INVITE	
Contact	INVITE CANCEL (C)	100 (INVITE) 180 (INVITE) 183 (INVITE) 200 (INVITE) 200 (OPTIONS)	INVITE	200 (INVITE) (R)
Content-Disposition			All (B)	183 (INVITE) (B) 200 (INVITE) (B) Any (INFO) (B)
Content-Encoding			All (B)	183 (INVITE) (B) 200 (INVITE) (B) Any (INFO) (B)
Content-Language			(U)	(U)
Content-Length	All	All	All	183 (INVITE) (B) 200 (INVITE) (B) Any (INFO) (B)
Content-Type	INVITE INFO	183 (INVITE) 200 (INVITE)	All (B)	183 (INVITE) (B) 200 (INVITE) (B) Any (INFO) (B)
CSeq	All	All (C)	All (R)	All (R)
Diversion (defined in [17])	INVITE (L) CANCEL (C)	302 (INVITE) (C)	INVITE	
Experienced-Operational-Status (Defined by System for internal use.)		All (OPTIONS)		

SIP Header	Added by System to following requests	Added by System to following responses	Processed by System if received in following requests	Processed by System if received in following responses
Expires	INVITE CANCEL (C)		INVITE	
From	All	All (C)	All (R)	All (R)
History-Info (defined in [12])			INVITE	
Max-Forwards	All	All (C)	INVITE	
P-Asserted-Identity (defined in [8])	INVITE CANCEL (C)		INVITE	
P-Charging-Vector (defined in [10])	All	All (within a session)	INVITE	
Privacy (defined in [7])	INVITE CANCEL (C)		INVITE	
RAck (defined in [5])	PRACK		PRACK (R)	
Reason (defined in [9])			BYE, CANCEL	3xx-6xx (INVITE)
Record-Route		All (C)		
Remote-Party-ID (defined in [15])	INVITE CANCEL (C)		INVITE	
Request-URI	All		All	
Require		1xx (INVITE) (REL)	All except ACK and CANCEL	
Route				
RSeq (defined in [5])		1xx (INVITE) (REL)		1xx (INVITE) (REL)
Supported	INVITE CANCEL (C)	183 (INVITE) 200 (INVITE) 200 (OPTIONS)		
Timestamp		All (C)	All (C)	
To	All	All (C)	All (R)	All (R)
Unsupported		420 (Any)		
User-Agent	INVITE CANCEL (C)		INVITE	
Via	All	All (C)	All (R)	All (R)
Warning		488 (INVITE)		

The following SIP headers (defined in [4]) are not used:

- Alert-Info
- Authentication-Info
- Authorization
- Date
- Error-Info
- In-Reply-To
- MIME-Version
- Min-Expires
- Organization
- Priority
- Proxy-Authenticate
- Proxy-Authorization
- Proxy-Require
- Reply-To
- Retry-After
- Server
- Subject
- WWW-Authenticate

6.1 Accept

The *Accept* header field lists the supported content types.

When inserted in SIP messages by the System, the format of the header is:

```
Accept: application/sdp,application/gtd,application/media_control+xml
```

6.2 Accept-Encoding

The *Accept-Encoding* header field lists the supported content encodings.

If the header field is empty or is set to *identity*, no content encodings are supported.

The System does not support any content encodings. Therefore when inserted in SIP messages by the System, the format of the header is:

```
Accept-Encoding: identity
```

6.3 Accept-Language

The *Accept-Language* header field lists the supported content languages. If no header field exists, all languages are accepted.

The System supports all content languages and therefore no *Accept-Language* header is inserted by the System in SIP messages.

6.4 Allow

The *Allow* header field lists the supported SIP methods.

When inserted in SIP messages by the System, the format of the header is:

```
Allow: OPTIONS,CANCEL,BYE,INFO,INVITE,ACK,PRACK
```

6.5 Call-ID

The *Call-ID* header field uniquely identifies a particular invitation. The header field is generated using MD5.

When inserted in SIP messages by the System, the header will look something like this:

```
Call-ID: b692904897340edee4b3632193b8eafd@host.company.com
```

6.6 Call-Info

The *Call-Info* header field provides additional information about the session.

This header is never inserted by the System.

When an INVITE lacking SDP offer is received by a System UA, this header field is checked for the session type (voice or video). If the header field contains the word "voice" a voice session shall be created. If the header field contains the word "video" a video session shall be created. Otherwise the session type is unknown, and configuration is used to determine which type of session to create (see section 3.4). This procedure is not according to any specification but is added to support certain external clients.

When received by the System, the header might look something like this:

```
Call-Info: <Media:Voice>;purpose=info
```

Note that the System is case insensitive when looking for the words voice or video.

6.7 Contact

The *Contact* header field is used as defined in [4].

When inserted by the System UA in SIP messages, the header will contain the configured name for the System UA together with the installed host name (or IP address) and port number:

```
Contact: <sip:mas@host.company.com:5060>
```

When received by the System, the *Contact* header field will be used to determine where to send responses and requests within the session.

6.8 Content-Disposition

The *Content-Disposition* header field describes how a message body part is to be interpreted.

The message body part is handled in the same way regardless of the content disposition type (see [4]).

The *handling* parameter describes whether or not the message body part is "optional" or "required". If the *handling* parameter is missing or if the *Content-Disposition* header is missing, the value "required" is assumed by the System UA.

Since all content sent by the System requires handling by the External Client, the System UA never inserts this header.

When received in a SIP request with content, this header is used during request validation as described in section 5.1.1.6.

6.9 Content-Encoding

The *Content-Encoding* header field indicates which encoding mechanisms that have been used for a message body part.

No content encodings are supported by the System.

This header is never inserted into SIP messages by the System.

When received in a SIP request with content, this header is used during request validation as described in section 5.1.1.6.

6.10 Content-Language

The *Content-Language* header field indicates the language used for a message body part.

All content languages are supported by the System.

This header is never inserted into SIP messages by the System.

When received in a SIP request with content, this header is used during request validation as described in section 5.1.1.6.

6.11 Content-Length

The *Content-Length* header field indicates the length of the message body.

This header is inserted by the System for all SIP message sent, regardless of if the message contains a body or not.

When received in a SIP message, this header is used when retrieving the message body.

6.12 Content-Type

The *Content-Type* header field gives the type of the message body.

The following content types are supported:

- application/sdp (defined in [14])
- application/media-control+xml (defined in [16])
- application/gtd (see section 7.3)
- multipart/* (defined in [1], subtype is ignored)

If this header is missing, the content type is assumed unsupported.

The *Content-Type* header may contain a *charset* attribute. The following character sets are supported: "UTF-8".

If the *charset* attribute is missing, "UTF-8" is assumed.

This header is inserted by the System when a SIP message with content is sent. For a message with an SDP:

```
Content-Type: application/sdp
```

For a message with a Video Picture Fast Update command:

```
Content-Type: media-control+xml
```

The System never sends a SIP message with multipart content.

The system never sends a body with content-type application/gtd.

When received in a SIP request with content, this header is used during request validation as described in section 5.1.1.6.

6.13 CSeq

The *CSeq* header field contains a single decimal sequence number and the original request method for which the SIP message is sent. This header is used by the System as described in [4].

The System always starts the sequence numbering for a session at 1.

6.14 Diversion

The *Diversion* header field is defined in [17]. This header indicates from whom and why a call was diverted.

When received by the System in an INVITE, the redirecting party is retrieved from this header:

- The redirecting party identification is retrieved from the *Diversion* header field URI as described in 8.5.
- If the *User-Agent* header field indicates that the External Client is a Cisco gateway (see section 6.35) the redirecting party presentation indicator is retrieved from the display name of the *Diversion* header field. If the display name is set to *anonymous* the presentation is restricted, if it is set to

unknown the presentation is unknown, otherwise it is allowed. This is not as described in [17] but has been implemented to support the Cisco implementation.

- For other user agents, the redirecting party presentation indicator is retrieved from the *privacy* parameter of the *Diversion* header field. If the parameter is set to "off" the presentation is allowed. If the parameter is set to something else, the presentation is restricted. If the parameter is missing the presentation is considered unknown.
- The redirecting cause is retrieved from the *reason* parameter of the *Diversion* header field. Note that the reason can be overridden by the GTD RNI attribute, see 7.3.1.2.

The *Diversion* header field is set by the System UA when sending an INVITE for which loopback should be prevented. Whether loopback prevention should be requested for an INVITE or not is determined by the end-user service. Loopback prevention is done by setting the *counter* parameter in the *Diversion* header to the value 5. This of course requires that the External Client supports loopback prevention using the *counter* parameter. The *Diversion* header set by the System UA would look something like this:

```
Diversion: <sip:host.company.com >;counter=5
```

6.15 Experienced-Operational-Status

The *Experienced-Operational-Status* header field is defined by the System. It is used internally within the System to determine the operational status of a System UA. If received in a SIP message by an external client, it can be ignored.

6.16 Expires

The *Expires* header field gives the relative time after which a sent message expires. When included in an INVITE, this header indicates the time within which the session must be setup. If this time elapses before the session has been setup, the INVITE procedure shall be terminated.

This header is inserted by the System UA when a SIP INVITE is sent. The time set is configurable in the System. The *Expires* header in the SIP CANCEL has no meaning. The *Expires* header set by the System would look something like this:

```
Expires: 51
```

When received by the System UA in a SIP INVITE request, a timer is started. If this timer expires before the session has been setup completely, the System UA rejects the call as described in section 8.3.1.1.

6.17 From

The *From* header field indicates the initiator of the request.

When received by the System in an INVITE, the calling party may be retrieved from this header (see section 5.1.2.4):

- The calling party identification is retrieved from the *From* header field URI as described in 8.5.

When the calling party is retrieved from this header, the calling party presentation indicator is retrieved from the *Privacy* header as described in section 6.22.

The *From* header field set by the System UA when sending an INVITE contains the calling party responsible for initiating the session.

If the calling party is a phone number, the *From* header field set by the System UA would look something like this:

```
From: <sip:4321@host.company.com;user=phone>;tag=504333693
```

The URI parameter *user* is set to "phone" to indicate that the user part of the URI is a phone number.

If the calling party is not a phone number, the *From* header field set by the System UA would look something like this:

```
From: <sip:theUser@company.com>;tag=219518124
```

The System does not set the *From* header field completely according to section 8.1.1.3 in [4] but has been implemented like this to accommodate SIP gateway vendors. The following parts differ:

- The host part of the URI may contain an IP address or a host name depending on how the System is installed. According to the specification an IP address should not be used.
- When presentation of the calling party shall be restricted the System does not reflect this completely in the *From* header field. The URI still contains a correct host part. According to the specification the identity of the user should be hidden and a meaningless URI should be used instead of user identification. The *From* header field set by the System UA would look something like this when the calling party number is restricted:

```
From: "Anonymous" <sip:invalid@host.company.com>;tag=21951812
```

6.18 History-Info

The *History-Info* header field is defined in [12]. This header indicates from whom and why a call was diverted.

When received by the System in an INVITE, the redirecting party may be retrieved from this header. If multiple redirections have occurred, the System retrieves the last redirection as described below:

- The redirecting party identification is retrieved from the second last *Hi-targeted-to-uri* in the *History-Info* header field as described in 8.5.
- The redirecting party presentation indicator is retrieved from the *Privacy* header field in the second last *Hi-targeted-to-uri* in the *History-Info* header field or from the *Privacy* header field of the entire SIP request.
- The redirecting reason is retrieved from the *Reason* header field of the last *Hi-targeted-to-uri* in the *History-Info* header field entry added.

The above is all according to TISPAN CDIV specification [23] but differs slightly from RFC 4244 [12] since the TISPAN and RFC specifications are in conflict when it comes to how to retrieve the redirecting reason.

In order to retrieve presentation indicator and reason from the *History-Info* header field, the System is able to retrieve SIP header fields from a SIP URI. This is only possible if the URI is contained in the *History-Info* header field and if the URI is a SIP URI (since the TEL URI does not support header fields in a URI).

This header field is never inserted in any SIP message sent by the System.

6.18.1.1 Redirecting party privacy

The redirecting party privacy is considered unknown in the following situations:

- no *Privacy* header field exists in the *Hi-targeted-to-uri* or in the entire SIP request.

The redirecting party is considered to have restricted presentation in the following situations:

- the *Privacy* header field of the *Hi-targeted-to-uri* contains the privacy value *history*
- the *Privacy* header field of the entire SIP request contains the privacy value *history*, *header*, or *session*.

Otherwise, the redirecting party privacy is considered allowed.

6.18.1.2 Redirecting party reason

The redirecting party reason is selected from the *Reason* header field of the *Hi-targeted-to-uri* and is mapped according to what is specified in RFC 4458 [13].

6.19 Max-Forwards

The *Max-Forwards* header field is used to limit the amount of times a SIP request can be forwarded.

The System UA sets this header in all SIP requests sent, the value 70 is used as recommended in [4]:

```
Max-Forwards: 70
```

6.20 P-Asserted-Identity

The *P-Asserted-Identity* header field is defined in [8]. This header is used by the System to identify the calling party.

When received by the System in an INVITE, the calling party identification is retrieved from this header (see section 5.1.2.4):

- The calling party identification is retrieved from the header field URI as described in section 8.5.

If multiple *P-Asserted-Identity* header fields exist, the first one containing a telephone number is chosen. If a telephone number cannot be found in any of the header fields, the first header field instance is chosen.

When the calling party is retrieved from this header, the calling party presentation indicator is retrieved from the *Privacy* header as described in section 6.22.

The *P-Asserted-Identity* header field set by the System UA when sending an INVITE contains the calling party responsible for initiating the session.

If the calling party is a phone number, the *P-Asserted-Identity* header field set by the System UA would look something like this:

```
P-Asserted-Identity: <sip:4321@host.company.com;user=phone>
```

The URI parameter *user* is set to "phone" to indicate that the user part of the URI is a phone number.

If the calling party is not a phone number, the *P-Asserted-Identity* header field set by the System UA would look something like this:

```
P-Asserted-Identity: <sip:theUser@company.com>
```

6.21 P-Charging-Vector

The *P-Charging-Vector* header field is defined in [10] and its usage over the IMS ISC interface is described in [20]. This header is used by the System to indicate charging information related to the session.

When no *P-Charging-Vector* header field is included in a SIP INVITE request received by the System, the System will generate an ICID value itself and add it in a *P-Charging-Vector* header field in the response. Otherwise, this header field is handled as required in [20].

This header field is also added to the following SIP requests sent by the System:

- SIP INVITE sent out of dialog
- Any request sent within dialog

The *P-Charging-Vector* header set by the System differs on situation. A header field generated for an out of dialog SIP INVITE or NOTIFY would look something like this:

```
P-Charging-Vector: icid-value=1234bc9876e-mas1.company.com;icid-generated-at=
mas1.company.com;orig-ioi=mas1.company.com
```

A header field included in a SIP response or a SIP request sent within dialog would look something like this:

```
P-Charging-Vector: icid-value=1234bc9876e;icid-generated-
at=host.company.com;orig-ioi=host.company.com;term-ioi=mas1.company.com
```

where the orig-ioi and icid-generated parameters only are included if received in the initial SIP INVITE request.

The ICID value generation follows the requirements specified in [21].

6.22 Privacy

The *Privacy* header field is defined in [7] with extensions in [8]. This header is used by the System to request network-provided privacy.

When received by the System in an INVITE, the calling party presentation indicator may be retrieved from this header (see section 5.1.2.4):

- Presentation is allowed if the *Privacy* header field has the value "none" or "history".
- Presentation is restricted if the *Privacy* header field has any other value.
- Presentation is unknown when no *Privacy* header field is present.

When received by the System within a URI contained in the *History-Info* header field in an INVITE, the redirecting party presentation indicator may be retrieved from this header as described in section 6.18.

The *Privacy* header field is always set by the System UA when sending an INVITE.

For allowed presentation:

```
Privacy: none
```

For restricted presentation:

```
Privacy: id
```

6.23 RAck

The *RAck* header field contains two decimal sequence numbers and the original request method for which the SIP message is sent. This header is used by the System as described in [5].

6.24 Reason

The *Reason* header field is defined in [9]. This header is used by the System to identify the release cause when a SIP BYE, a SIP CANCEL or a SIP error response is received. This header is also used by the System to identify the redirection reason when a SIP INVITE is received containing a *History-Info* header field (see section 6.18).

When received by the System as a header field in an INVITE, only the protocol "Q.850" is supported. Apart from retrieving the release cause from this header, the release location is also retrieved if present. The release location is not specified in [9] and is thus retrieved in a proprietary manner. The release location is retrieved from the parameter "eri-location".

When only cause is present:

```
Reason: Q.850;cause=16
```

When location is present as well:

```
Reason: Q.850;cause=17;eri-location=0
```

When received by the System within a URI contained in the *History-Info* header field in an INVITE, the redirecting reason may be retrieved from this header. In that case, only the protocol "SIP" is supported. How the SIP reason is mapped to a redirecting reason is described in section 6.18.

This header field is never inserted in any SIP message sent by the System.

6.25 Record-Route

The *Record-Route* header field is used as described in [4].

This header field is never inserted in any request sent by the System. If received in a SIP request, the header field will be copied to any response sent by the System.

6.26 Remote-Party-ID

The *Remote-Party-ID* header field is defined in [15]. This header is used by the System to identify the calling party.

When received by the System in an INVITE, the calling party presentation indicator is retrieved from this header (see section 5.1.2.4):

- If the header field parameter *rpi-pty-type* is set to "calling" (i.e. *party=calling*), the calling party identification is retrieved from the header field URI as described in section 8.5. Also, according to specification if the parameter *rpi-pty-type* is missing it should be interpreted as "calling". Note that the first remote-party-id header instance with party set to "calling" (or with the party parameter missing) is chosen when retrieving the calling party since multiple remote-party-id headers are allowed.
- The calling party presentation indicator is retrieved from the header field parameter *rpi-privacy*:
    - Presentation is allowed if the parameter has the value "none" (i.e. *privacy=none*).
    - Presentation is restricted if the parameter has any other value.
    - Presentation is unknown when no *privacy* parameter is present.

The *Remote-Party-ID* header field is set by the System UA when sending an INVITE. Apart from the URI specifying the calling party identity, the header field contains the parameters *rpi-pty-type* and *rpi-privacy*.

For allowed presentation:

```
Remote-Party-ID: <sip:4321@host.company.com;user=phone>;party=calling;
privacy=off
```

For restricted presentation:

```
Remote-Party-ID: <sip:4321@host.company.com;user=phone>;party=calling;
privacy=full
```

When the System UA requests privacy "full", removal of user identity in headers such as *To*, *From*, and *Contact* is **not** done. The reason for this is to be able to support multiple External Client vendors. Not all have support for the *Remote-*

Party-ID header field and need to retrieve calling party information from for example the *From* header field.

6.27 Request-URI

The *Request-URI* indicates the recipient of the request.

When received by the System in an INVITE, the called party is retrieved from this information (see section 5.1.2.3):

- The called party identification is retrieved from the *Request-URI* as described in 8.5.

The retrieved called party information could either identify an end-user or a public service identity as described in section 5.1.2.3.

The *Request-URI* header field set by the System UA when sending an INVITE contains the end-user party called in the session.

If the called party is a phone number, the *Request-Line* (containing the *Request-URI*) set by the System UA would look something like this:

```
INVITE sip:1234@host.company.com:5060;user=phone SIP/2.0
```

The URI parameter *user* is set to "phone" to indicate that the user part of the URI is a phone number.

If the called party is not a phone number, the *Request-Line* set by the System UA would look something like this:

```
INVITE sip:theUser@company.com:5060 SIP/2.0
```

6.28 Require

The *Require* header field is used to require support of specific SIP extensions.

The supported extension are listed in section 6.31

When received in a SIP request, the *Require* header field is used during request validation as described in section 5.1.1.5. Also, when received in a SIP INVITE request, the *Require* header field is used to determine whether provisional responses should be sent reliably or not.

A *Require* header field is inserted by the System UA when sending a provisional response reliably:

```
Require: 100rel
```

6.29 Route

The *Route* header field is used to force routing for a request through the listed set of proxies.

This header field is never set or read by the System UA.

6.30 RSeq

The *RSeq* header field is used in provisional responses in order to transmit them reliably. It contains a single numeric sequence number. This header is used by the System as described in [5].

The System starts the sequence numbering for a session at a random number between 1 and (2^32-1).

6.31 Supported

The *Supported* header field is used to indicate support of specific SIP extensions.

The following extensions are supported:

- *100rel* – to send provisional responses reliably

When received in a SIP INVITE request, the *Supported* header field is used to determine whether provisional responses should be sent reliably or not.

The following *Supported* header field is inserted into SIP messages sent by the System:

```
Supported: 100rel
```

6.32 Timestamp

The *Timestamp* header field describes when a request was sent by a UA.

If received by the System in SIP requests, this header field is copied into the generated corresponding SIP response. No update of the timestamp value is done.

The System does not insert this header in SIP requests.

6.33 To

The *To* header field indicates the recipient of the request.

When received in a SIP message, the *To* header field is used by the System UA to identify the session to which the message belongs as described in [4].

The *To* header field set by the System UA when sending an INVITE contains the party called in the session.

If the called party is a phone number, the *To* header set by the System UA would look something like this:

```
To: <sip:1234@host.company.com;user=phone>
```

The URI parameter *user* is set to "phone" to indicate that the user part of the URI is a phone number.

If the called party is not a phone number, the *To* header set by the System UA would look something like this:

```
To: <sip:theUser@company.com>
```

6.34 Unsupported

The *Unsupported* header field lists SIP extensions not supported.

The supported SIP extensions are listed in section 6.31. When a SIP request containing a *Require* header field with an unsupported extension is received by the System, a 420 "Bad Extension" response is sent (see section 5.1.1.5). In this response the System adds the *Unsupported* header field listing all unsupported extensions from the previously received header field *Require*. For example:

```
Unsupported: timer
```

This header is never checked when received in SIP messages.

6.35 User-Agent

The *User-Agent* header field contains information about the UA sending a SIP request.

This header is inserted by the System when a SIP INVITE is sent. The value is configurable and the header field could look something like this:

```
User-Agent: mas
```

This header is parsed when received in a SIP INVITE and used for the situations described in the table below. The table below lists the External Clients receiving special treatment due to the content of the *User-Agent* header and the string searched for to determine the identity of the External Client.

Table 2 Special treatment for some External Clients

External Client	Searched for in header	Used in situations
Cisco	"cisco"	When retrieving the redirecting party from the *Diversion* header field as described in section 6.14. When retrieving a call party telephone number from a URI as described in section 8.5.2.
Radvision	"radvision"	When retrieving a call party telephone number from a URI as described in section 8.5.2.

It is possible to configure additional user-agents that receive special handling when retrieving a call party telephone number from a URI as described in section 8.5.2.

Note that the System is case insensitive when looking for the specific user agents.

6.36 Via

The *Via* header field indicates the path taken by a SIP request.

The System copies the header(s) from received SIP requests into the generated corresponding SIP response messages.

The System adds a *Via* header in every SIP request sent as described in [4]. An example of generated *Via* header is:

```
Via: SIP/2.0/UDP
host.company.com:5060;branch=z9hG4bKdddfd52968151ba60a252d0c1d2f2f7b
```

The *Via* branch id is generated using MD5.

If more than one *Via* header field value is present in a SIP response, the System picks the first value and uses that even though according to [4] it is recommended to discard the message.

The branch ID in the *Via* header must always begin with the magic cookie "z9hG4bK" as specified in [4]. The System can handle SIP messages lacking the magic cookie, but the performance will degrade significantly.

6.37 Warning

The *Warning* header field is used to carry additional information about the status of a SIP response.

The *Warning* header field is included by the System when sending a 488 "Not Acceptable Here" response. How the warning code and warning text is set is described in the table below.

Table 3 Warning codes and texts used by the System

Warning code	Warning text	Scenarios causing a 488 response to be sent
300	"Only network protocol IN is supported. "	The network type in an SDP *connection* field is unsupported, see section 7.1.1.4.
301	"Only network address format IP4 is supported. "	The address type in an SDP *connection* field is unsupported, see section 7.1.1.4.
302	"Only transport protocol RTP/AVP or RTP/AVPF are supported. "	The media transport in an SDP *media* field is unsupported, see section 7.1.1.8.
304	"Only media types "audio" and "video" are supported. "	The media type in an SDP *media* field is unsupported, see section 7.1.1.8.
305	"Required media formats not supported by session description. "	The following scenarios both cause this warning code: • The media formats in an SDP is not acceptable by the System as described in section 3.4 • A media format in an SDP *media* field is a non-integer, see section 7.1.1.8.

Warning code	Warning text	Scenarios causing a 488 response to be sent
306	"Media attribute is not understood. "	Error occurred when parsing an attribute in an SDP *media* field.
307	"Parameter is not understood. "	The following scenarios both cause this warning code: • Error occurred when parsing the following parts of an SDP: *media* field, *session description*, *media description*, *connection* field, *origin* field. • No media level or session level *connection* field is included in the SDP.
330	"Multicast is not supported. "	The address in an SDP *connection* field is a multicast address, see section 7.1.1.4.
399	"Modifying an ongoing session is not supported. "	A re-INVITE suggesting a non-supported re-negotiation is received for an established session, see section 4.1.2.
399	"Encrypted SDP is not supported. "	An SDP contains unsupported encryption keys, see section 7.1.1.6.
399	"Charset specified in SDP is not supported. Only UTF-8 is supported. "	The *charset* attribute in an SDP is unsupported, see section 7.1.2.7.
399	"Port count in media field is not supported. Only port count one is supported. "	The media port count in an SDP *media* field is unsupported, see section 7.1.1.8.
399	"Only SDP version 0 is supported"	An unsupported protocol version is used in an SDP, see section 7.1.1.1.

An example of a *Warning* header field generated by the System:

```
Warning: 304 host.company.com "Only media types %22audio%22 and %22video%22
are supported."
```

7 Body Content

This section describes the different body contents used by the System. Apart from MIME multipart (which is only a structuring of body parts, see [1]) the supported content types are:

- application/sdp
- application/media-control+xml
- application/gtd

Below is a description of how these content types are used within the System.

7.1 SDP

An SDP describes the media to use for a session as described in [14] and [6].

This section describes the SDP fields and attributes supported by the System. Non-supported fields and attributes will be ignored if received.

7.1.1 Fields

The table below shows the SDP fields supported by the System. The table illustrates which SDP fields are inserted by the System and which are read by the System.

The fields described are all defined in [14] unless otherwise stated.

Table 4 SDP field usage

SDP field	Added by System to SDP Session Description	Added by System to SDP Media Description	Processed by System if received in SDP Session Description	Processed by System if received in SDP Media Description
Protocol version (v)	X		X	
Origin (o)	X		X	
Session name (s)	X			
Connection (c)	X	X	X	X
Time (t)	X			
Encryption keys (k)			X	
Attribute (a)		X (See Table 5)	X (See Table 5)	X (See Table 5)
Media (m)		X		X
Bandwidth (b)		X	X	X

The following SDP fields (defined in [14]) are not used:

- Session information (i)
- URI (u)
- Email address (e)
- Phone number (p)
- Repeat times (r)
- Time zone adjustments (z)

7.1.1.1 Protocol version

The *protocol version* field gives the SDP version. Currently only version 0 exists.

When inserted by the System in SIP messages, the format is:

```
v=0
```

7.1.1.2 Origin

The *origin* field gives the originator of the session.

When inserted by the System in SIP messages, it contains the configured name of the System UA:

```
o=mas 0 0 IN IP4 10.16.128.23
```

When received by the System, all values are accepted in this field.

7.1.1.3 Session name

The *session name* field conveys the subject of the session.

When inserted by the System in SIP messages, it is set as suggested in [6]:

```
s=-
```

7.1.1.4 Connection

The *connection* field contains connection data.

When inserted by the System in SIP messages, only the connection address will vary:

```
c=IN IP4 10.16.128.23
```

An SDP received by the System is unacceptable if the *connection* field cannot be parsed or if the network type, address type or address is unsupported.

Supported network types are: "IN".

Supported address types are: "IP4"

Only unicast addresses are supported which means that an address must not contain a TTL section or an address count larger than one.

7.1.1.5 Time

The *time* field specifies the start and stop time for a session.

When inserted by the System in SIP messages, it is set as suggested in [6]:

```
t=0 0
```

When received by the System, all values are accepted in this field.

7.1.1.6 Encryption keys

The *encryption keys* field conveys information on how the SDP is encrypted.

No SDP encryptions are supported by the System. Therefore, this field is never inserted into SIP messages by the System.

An SDP received by the System containing *encryption keys* is not accepted. It is only processed by the system in order to be able to reject a call containing encryption keys.

7.1.1.7 Attribute

Attributes are used to extend the SDP with additional information. The supported attributes are described in section 7.1.2.

7.1.1.8 Media

The *media* field describes a media stream in the session.

Supported media types are: "audio", "video".

Supported media transports are: "RTP/AVP", "RTP/AVPF".

A port count larger than one is not supported.

When inserted by the System into SIP messages, the format is:

```
m=audio 22220 RTP/AVP 0 101
```

The media type can be either "audio" or "video". For the "audio" media type, the format list will contain one audio codec and (if configured to do so) the codec for DTMF over RTP. For the "video" media type, the format list will contain one video codec.

When received by the System, the SDP is not acceptable if the media type, media transport or port count is unsupported or if the media format list contains a non-integer value.

When an SDP offer is received by the System, the System selects the media streams appropriate to use and creates an SDP answer based on them. The other media streams are left unused as described in [6]. For a voice session one audio media stream is selected and for a video session one audio and one video media stream are selected. If multiple media streams of the same type are present in an offer, the System selects only one of them for use.

According to [6], media formats in an SDP are listed in order of preference. Since only one audio and one video codec is supported for an installed system, this fact is ignored by the System which chooses to use the only supported media format regardless of position in the *media* field.

According to [6], media formats in an SDP answer should be listed in the same order as in the SDP offer. Since only one audio and one video codec is supported for an installed system, this fact is ignored by the System which lists the media formats in the SDP answer regardless of order in the SDP offer.

7.1.1.9 Bandwidth

The *bandwidth* field denotes the proposed bandwidth to be used by the session or media. The bandwidth types supported are:

- AS, application specific bandwidth
- RS, RTCP bandwidth allocated to active data senders
- RR, RTCP bandwidth allocated to other participants

The bandwidth for AS is interpreted as kilobits per second but RS and RR are interpreted as bits per second.

When inserted by the System in SIP messages, the bandwidth will vary depending on the media format used. An example of bandwidth field inserted by the System UA is:

```
b=AS:64
b=RR:800
b=RS:2400
```

An SDP received by the System is unacceptable if the bandwidth is too small for the selected media format or if the RTCP bandwidths are unacceptable for the selected media format.

7.1.2 Attributes

The table below shows the SDP attributes supported by the System. The table illustrates which SDP attributes are inserted by the System and which are read by the System.

The attributes described are all defined in [14] unless otherwise stated.

Table 5 SDP attribute usage

SDP attribute	Added by System to SDP Session Description	Added by System to SDP Media Description	Processed by System if received in SDP Session Description	Processed by System if received in SDP Media Description
ptime		X	X	X
maxptime		X		X
recvonly			X	X
sendrecv			X	X
sendonly			X	X
inactive			X	X
charset			X	
fmtp		X	X	X
rtpmap		X	X	X
rtcp-fb				X

The following SDP attributes (defined in [14]) are not used:

- cat
- framerate
- keywds
- tool
- orient
- type
- sdplang

- lang
- quality

7.1.2.1 Ptime

The *ptime* attribute gives the length of time in milliseconds represented by the media in a packet.

It is inserted by the System for the audio media description and will contain a System configured value:

```
a=ptime:40
```

When received by the System, it is retrieved and used during media streaming.

The ptime value is unacceptable if it contains a non-integer value.

7.1.2.2 Maxptime

The maxptime attribute gives the maximum amount of media that can be encapsulated in each packet expressed as time in milliseconds.

It is inserted by the System for the audio media description and will contain a System configured value:

```
a=maxptime:40
```

When received by the System, it is retrieved and used during media streaming. The maxptime value is unacceptable if it contains a non-integer value.

7.1.2.3 Recvonly

The *recvonly* attribute specifies that the media stream should be started in receive-only mode.

The System media streams must be bi-directional. Therefore, this attribute is never set by the System in an SDP. When received by the System, a media description with this attribute set will not be used by the System.

7.1.2.4 Sendrecv

The *sendrecv* attribute specifies that the media stream should be started in send and receive mode. This is the default behavior if not specified.

The System media streams must be bi-directional. Since that is the default behavior, this attribute is never set by the System in an SDP. When received by the System, media descriptions with this attribute set will be used by the System if the listed media formats are acceptable.

7.1.2.5 Sendonly

The *sendonly* attribute specifies that the media stream should be started in send-only mode.

The System media streams must be bi-directional. Therefore, this attribute is never set by the System in an SDP. When received by the System, a media description with this attribute set will not be used by the System.

7.1.2.6 Inactive

The *inactive* attribute specifies that the media stream should be inactive.

This attribute is never set by the System in an SDP. When received by the System, a media description with this attribute set will not be used by the System.

7.1.2.7 Charset

The *charset* attribute specifies the character set that is used within the SDP. If not specified, UTF-8 encoding is used.

Only UTF-8 encoding is supported by the System. Therefore, this attribute is never set by the System in an SDP.

When this attribute is received by the System with an unsupported character set, the SDP is not accepted.

7.1.2.8 Fmtp

The *fmtp* attribute is used to carry parameters specific to a particular media format.

This attribute is currently set by the System for the following media formats:

- When DTMF over RTP (see [2]) is included in the SDP audio media description:

  ```
  a=fmtp:101 0-16
  ```

- When the AMR media format is included in the SDP audio media description a System configured value for fmtp will be used:

  ```
  a=fmtp:96 mode-set=7; octet-align=1
  ```

When this attribute is received by the System in an SDP, it is parsed and handled depending on the particular media format it belongs to. Currently it is only handled when received for the AMR media format:

When fmtp is received for the AMR media format the System configured fmtp is compared with the received fmtp and the following conditions must hold true:

- The received parameter *mode-set*, if present, must match the System configured mode-set exactly. Only one single mode is supported at a time.
- The received parameter *octet-align* must be present and match the System configured octet-align. Only the value "1" is currently supported.
- The received parameter *channels*, if present, must be equal to "1".
- The received parameters *crc*, *robust-sorting* and *interleaving*, if present, must be equal to "0".
- Other received parameters are ignored.

If any of the above fails to match the corresponding SDP media description will not be considered for use.

7.1.2.9 Rtpmap

The *rtpmap* attribute is used to describe an RTP payload type. Its main purpose is to describe dynamic payload types but can also be used to describe static payload types.

This attribute is added to the SDP by the System for each media format listed in the SDP. Typical examples are:

```
a=rtpmap:101 telephone-event/8000/1

a=rtpmap:0 PCMU/8000/1
```

When received by the System, it is retrieved and used during media streaming.

7.1.2.10 rtcp-fb

The "rtcp-fb" attribute is for RTCP feedback messages, specified in [25]. Currently, the support is limited, and only the "ccm fir" type is answered, as described in [26].

7.1.3 SDP examples

Below are some SDP offer/answer examples. The examples are only intended as guidelines not as references.

7.1.3.1 Sample SDP offer with AMR audio (from SIP phone)

```
v=0
o=sip:16191700 1193645988 1193648090 IN IP4 20.10.10.1
s=Kapanga [1193645988]
c=IN IP4 20.10.10.1
t=0 0
m=audio 23018 RTP/AVP 96 101
a=rtpmap:96 amr/8000
a=fmtp:96 mode-set=7; octet-align=1; robust-sorting=0
a=sendrecv
a=rtcp:23019
a=maxptime:20
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15,36
```

7.1.3.2 Sample SDP answer with AMR audio (from MAS)

```
v=0
o=mas 0 0 IN IP4 20.10.10.2
s=-
c=IN IP4 20.10.10.2
t=0 0
m=audio 23008 RTP/AVP 96 101
c=IN IP4 20.10.10.2
b=AS:13
a=ptime:20
a=maxptime:40
a=rtpmap:96 amr/8000/1
a=rtpmap:101 telephone-event/8000/0
a=fmtp:96 mode-set=7; octet-align=1; robust-sorting=0
a=fmtp:101 0-16
```

7.1.3.3 Sample SDP offer with AMR and H.263 (from MAS)

```
v=0
o=mas 0 0 IN IP4 20.10.10.2
s=-
c=IN IP4 20.10.10.2
t=0 0
m=audio 23012 RTP/AVP 96 101
c=IN IP4 20.10.10.2
b=AS:13
a=ptime:20
a=maxptime:40
a=rtpmap:96 AMR/8000/0
a=rtpmap:101 telephone-event/8000/0
a=fmtp:96 mode-set=7; octet-align=1
a=fmtp:101 0-16
m=video 23014 RTP/AVP 34
c=IN IP4 20.10.10.2
b=AS:52
a=rtpmap:34 H263/90000/0
```

7.1.3.4 Sample SDP answer, AMR only (from SIP phone)

```
v=0
o=sip:16191700 1193645989 1193650444 IN IP4 20.10.10.1
s=Kapanga [1193645989]
c=IN IP4 20.10.10.1
t=0 0
m=audio 23024 RTP/AVP 96 101
a=rtpmap:96 AMR/8000/0
a=fmtp:96 mode-set=7; octet-align=1
b=AS:13
a=sendrecv
a=maxptime:20
a=ptime:20
a=rtpmap:101 telephone-event/8000/0
a=fmtp:101 0-16
```

7.1.3.5 Sample SDP offer with G.711 audio (from gateway)

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 3907 2703 IN IP4 10.10.10.2
s=SIP Call
c=IN IP4 10.10.10.2
t=0 0
m=audio 22916 RTP/AVP 0 101
c=IN IP4 10.10.10.2
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:40
```

7.1.3.6 Sample SDP answer with G.711 audio (from MAS)

```
v=0
o=mas 0 0 IN IP4 10.10.10.3
s=-
c=IN IP4 10.10.10.3
t=0 0
m=audio 25324 RTP/AVP 0 101
c=IN IP4 10.10.10.3
b=AS:64
a=ptime:40
a=maxptime:40
a=rtpmap:0 PCMU/8000/1
```

```
a=rtpmap:101 telephone-event/8000/0
a=fmtp:101 0-16
```

7.1.3.7 Sample SDP offer with PCMU audio and H.263 video with FIR RTCP feedback (from gateway)

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 5928 8051 IN IP4 10.11.0.68
s=SIP Call
c=IN IP4 10.11.0.68
t=0 0
m=audio 7600 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
m=video 7602 RTP/AVP 34
c=IN IP4 10.11.0.68
a=sendrecv
a=rtpmap:34 H263/90000/0
a=fmtp:34 QCIF=2/MaxBR=520
m=video 7602 RTP/AVPF 34
c=IN IP4 10.11.0.68
a=sendrecv
a=rtpmap:34 H263/90000/0
a=fmtp:34 QCIF=2/MaxBR=520
a=rtcp-fb:34 ccm fir
```

7.1.3.8 Sample SDP answer with PCMU audio and H.263 video with FIR RTCP feedback (from MAS)

```
v=0
o=mas 0 0 IN IP4 10.11.0.147
s=-
c=IN IP4 10.11.0.147
t=0 0
m=audio 23000 RTP/AVP 0 101
c=IN IP4 10.11.0.147
b=AS:0
a=ptime:40
a=rtpmap:101 telephone-event/8000/0
a=rtpmap:0 PCMU/8000/1
a=fmtp:101 0-16
m=video 23002 RTP/AVP 34
c=IN IP4 10.11.0.147
b=AS:0
a=rtpmap:34 H263/90000/0
m=video 23002 RTP/AVPF 34
c=IN IP4 10.11.0.147
b=AS:0
a=rtpmap:34 H263/90000/0
a=rtcp-fb:34 ccm fir
```

7.2 Media Control

The media control content type is used to carry video control commands as described in [16].

The only video control command supported by the System is the Video Picture Fast Update. When a Video Picture Fast Update command is sent by the System, the body content will look like this:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<media_control xmlns="urn:ietf:params:xml:ns:media_control">
  <vc_primitive>
    <to_encoder>
      <picture_fast_update/>
    </to_encoder>
  </vc_primitive>
</media_control>
```

When a response for a Video Picture Fast Update is received by the System, the *general_error* tag of the response is parsed and logged.

7.3 GTD

The GTD is a proprietary structure, which is based on Q.1980.1 defined in [22].

7.3.1 Fields

The table below shows the GTD fields supported by the System. The table illustrates which GTD fields are inserted by the System and which are read by the System.

The fields described are all defined in [22] unless otherwise stated.

Table 6 GTD field usage

GTDfield	Added by System to GTD	Processed by System if received in GTD
CGN		X
RNI		X

7.3.1.1 CGN

From the CGN (calling party number), the following fields are processed:

- Number completion.

This is an example of a CGN line, with the number completion field in bold style:

CGN,04,**y**,1,y,4,1133

7.3.1.2 RNI

From the RNI (redirection information), the following field is processed:

- Redirecting reason. Note: Only if redirecting reason is 6 it will be used. For all other values the redirecting reason will be extracted from the Diversion header, see 6.14.

This is an example of a RNI line, with the Redirecting reason field in bold style:

RNI,03,N,1,**6**

8 Properties

This section lists various properties that do not fit into any of the above sections.

8.1 Supported URI schemes

The following URI schemes are supported by the System:

- SIP URI (specified in [4])
- Tel URI (specified in [11])

All URIs created by the System are SIP URIs.

8.2 Error handling

If an error occurs when the System sends a SIP message within an established session, the session is considered broken and the RTP streams are immediately closed.

If an error occurs when the System sends a SIP message not related to a session, the error is logged but no other action is performed.

If an error occurs during outbound session creation a SIP CANCEL or BYE request is sent by the System depending on the current status of the session.

If an error occurs when the System sends a SIP CANCEL or BYE request, the session is considered terminated anyway.

A SIP message lacking one of the headers fields *To*, *From*, *CSeq*, *Call-ID*, or *Via* is immediately discarded when received by the System.

8.3 Timing Properties

8.3.1 SIP timing properties

The following SIP timers (specified in [4]) are configurable in the System:

- T2
- T4
- Timer B
- Timer C
- Timer D
- Timer F
- Timer H
- Timer J

The timers are configurable as units of T1. T1 is 500 ms as recommended in [4].

The System handles SIP timeouts as specified in [4].

8.3.1.1 Expires timer

When a SIP INVITE request with the *Expires* header field (see section 6.16) set is received by the System, a timer is started.

If this timer expires before the session has been setup completely, the System UA rejects the call with a SIP 487 "Request Terminated" response (see section 5.2.2.15).

8.3.2 System specific timing properties

The System has specific timers for the following situations:

- Max Call Duration
- Call Not Connected
- Call Not Accepted

8.3.2.1 Max call duration timer

The System can be configured with max call duration for an outbound session. If this feature is activated, there is a maximum time for an established outbound session. The timer is started when the outbound session creation has completed. When the timer expires the System terminates the session with a SIP BYE request as described in section 4.2.1.

8.3.2.2 Call not connected timer

The System has a timer to prevent too long outbound session creation procedures. The System is configured with a maximum time it can take for an outbound session creation to complete. The timer is started when the System initiates a new outbound session. When the timer expires, the System cancels the session with a SIP CANCEL request as described in section 4.3.1.

8.3.2.3 Call not accepted timer

The System has a timer to prevent too long inbound session creation procedures. The System is configured with a maximum time it can take for an inbound session creation to be accepted by the System. The timer is started when the System receives a SIP INVITE request. When the timer expires, the INVITE is rejected with a SIP 408 "Request Timeout" response (see section 5.2.2.9).

8.4 Abandoned Session Detection

The System has a mechanism to prevent abandoned sessions. This is done by inspecting the status of the inbound media stream for a session. If no packets have been received on the inbound media stream for a configurable amount of time, the session is considered abandoned. The System terminates the session with a SIP BYE or CANCEL request depending on the status of the session.

The media stream detection is started as soon as the inbound media stream is created, i.e. when a session is created or when there is early media to receive.

8.5 Retrieval of call party identification from URI

This section describes how a call party is retrieved from a URI. Three parts are extracted from the URI and represents a call party:

- The URI as is
- The user and host (if the URI scheme is SIP)
- The telephone number (if the URI scheme is SIP or TEL)

8.5.1 User and Host

The user and host are retrieved as is from the URI if the scheme is SIP. For other URI schemes, the user and host are not retrieved.

8.5.2 Telephone Number

If the scheme is TEL, the telephone number is retrieved as is (as described in [11]).

If the scheme is SIP it is retrieved as follows:

- If the URI parameter *user* is set to "phone" (i.e. user=phone), the telephone number is retrieved from the user part of the URI. If the user part is not included, the telephone number is retrieved from the host part of the URI.
- If the URI parameter *user* is not included or not set to "phone" but the *User-Agent* header field indicates that the External Client requires special handling (see section 6.35), the telephone number is retrieved from the user part of the URI.

For other URI schemes, the telephone number is not retrieved.

8.6 Retrieval of call parameters for testing purpose

This section describes an alternative way call parameters are retrieved by the System. This is not the common retrieval of the call party information; it is used solely to simplify testing. Common SIP phones normally do not allow specifying arbitrary SIP header and in order to set call parameter information the System allows for alternative specification in the *Request-URI* or *To* header field.

Note: this way of specifying call parameters should only be used for testing purposes.

If the *Request-URI* or the *To* header field URI contains the parameter *test* given the value "on" (i.e. test=on) and the System UA is configured to allow test input in this way, the called, calling and redirecting parties are retrieved from special URI parameters. The following information is retrieved from the URI: called party number, calling party number, calling party presentation indicator, redirecting

party number, redirecting party presentation indicator, and redirection party cause. An example of URI that contains all parameters:

```
<sip:user@host.com;test=on;called=1234;calling=4321;calling_privacy=off;redir
=5678;redir_privacy=on;redir_cause=time-of-day>
```

8.6.1 Called party

The called party number is retrieved as is from the URI parameter *called*:

```
<sip:user@host.com;test=on;called=1234>
```

8.6.2 Calling party

The calling party number is retrieved as is from the URI parameter *calling*:

```
<sip:user@host.com;test=on;calling=4321>
```

The calling party presentation indicator is retrieved from the URI parameter *calling_privacy*:

```
<sip:user@host.com;test=on;calling_privacy=off>
```

If the parameter is set to *on* the presentation is restricted, if the parameter is set to *off* the presentation is allowed, otherwise the presentation indicator is unknown.

8.6.3 Redirecting party

The redirecting party number is retrieved as is from the URI parameter *redir*:

```
<sip:user@host.com;test=on;redir=1234>
```

The redirecting party presentation indicator is retrieved from the URI parameter *redir_privacy*:

```
<sip:user@host.com;test=on;redir_privacy=on>
```

If the parameter is set to *on* the presentation is restricted, if the parameter is set to *off* the presentation is allowed, otherwise the presentation indicator is unknown.

The redirecting cause is retrieved as is from the URI parameter *redir_cause*:

```
<sip:user@host.com;test=on;redir_cause=time-of-day>
```

Possible values for the redirecting cause are:

- "away"
- "deflection"
- "do-not-disturb" (or the escaped version "do%2dnot%2ddisturb")
- "follow-me" (or the escaped version "follow%2dme")
- "no-answer" (or the escaped version "no%2danswer")
- "out-of-service" (or the escaped version "out%2dof%2dservice")

- "time-of-day" (or the escaped version "time%2dof%2dday")
- "unavailable"
- "unconditional"
- "unknown"
- "user-busy" (or the escaped version "user%2dbusy")

9 References

[1] Multipurpose Internet Mail Extension (MIME) Part Two: Media Types
RFC 2046
http://ietf.org/rfc/rfc2046.txt?number=2046

[2] RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals
RFC 2833
http://ietf.org/rfc/rfc2833.txt?number=2833

[3] The SIP INFO Method
RFC 2976
http://ietf.org/rfc/rfc2976.txt?number=2976

[4] SIP: Session Initiation Protocol
RFC 3261
http://ietf.org/rfc/rfc3261.txt?number=3261

[5] Reliability of Provisional Responses in the Session Initiation Protocol (SIP)
RFC 3262
http://ietf.org/rfc/rfc3262.txt?number=3262

[6] An Offer/Answer Model with the Session Description Protocol (SDP)
RFC 3264
http://ietf.org/rfc/rfc3264.txt?number=3264

[7] A Privacy Mechanism for the Session Initiation Protocol (SIP)
RFC 3323
http://ietf.org/rfc/rfc3323.txt?number=3323

[8] Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks
RFC 3325
http://ietf.org/rfc/rfc3325.txt?number=3325

[9] The Reason Header Field for the Session Initiation Protocol (SIP)
RFC 3326
http://ietf.org/rfc/rfc3326.txt?number=3326

[10] Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)

RFC 3455
http://ietf.org/rfc/rfc3455.txt?number=3455

[11] The tel URI for Telephone Numbers
RFC 3966
http://ietf.org/rfc/rfc3966.txt?number=3966

[12] An Extension to the Session Initiation Protocol (SIP) for Request History Information
RFC 4244
http://ietf.org/rfc/rfc4244.txt?number=4244

[13] Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR)
RFC 4458
http://ietf.org/rfc/rfc4458.txt?number=4458

[14] SDP: Session Description Protocol
RFC 4566
http://ietf.org/rfc/rfc4566.txt?number=4566

[15] SIP Extensions for Caller Identity and Privacy
draft-ietf-sip-privacy-04
http://ietfreport.isoc.org/all-ids/draft-ietf-sip-privacy-04.txt

[16] XML Schema for Media Control
draft-levin-mmusic-xml-schema-media-control-03
http://www1.cs.columbia.edu/sip/drafts/mmusic/draft-levin-mmusic-xml-media-control-03.txt

[17] Diversion Indication in SIP
draft-levy-sip-diversion-08
http://ietfreport.isoc.org/all-ids/draft-levy-sip-diversion-08.txt

[18] IP Multimedia (IM) session handling; IM call model
3GPP TS 23.218 V7.2.0 (2006-06)

[19] IP Multimedia Subsystem (IMS)
3GPP TS 23.228 V7.4.0 (2006-06)

[20] IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)
3GPP TS 24.229 V7.4.0 (2006-06)

[21] IP Multimedia Subsystem (IMS) Charging
3GPP TS 32.260 V6.6.0 (2006-06)

[22] The Narrowband Signalling Syntax (NSS) – Syntax definition
ITU-T Recommendation Q1980.1

[23] TISPAN; PSTN/ISDN simulation services; Communication Diversion (CDIV)
ETSI TS 183 004 V1.1.1 (2006-04)

[24] TISPAN; PSTN/ISDN simulation services; Originating Identification Presentation (OIP) and Originating Identification Restriction (OIR)
ETSI TS 183 007 V1.1.1 (2006-03)

[25] Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)
RFC 4585
http://ietf.org/rfc/rfc4585.txt?number=4585

[26] Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)
draft-ietf-avt-avpf-ccm-10
http://ietfreport.isoc.org/all-ids/draft-ietf-avt-avpf-ccm-10.txt

10 Terminology

3GPP	3rd-Generation Partnership Project
AS	Application Server
DTMF	Dual-Tone Multi Frequency
GTD	Generic Transparency Descriptor, a proprietary structure used to convey ISDN data to for example SIP. It contains information about the involved parties.
ICID	IMS Charging Identity
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISC	IMS Service Control interface between an AS and an S-CSCF
MAS	Media Access Server
MIME	Multipurpose Internet Mail Extensions
PSI	Public Service Identity
RTP	Real-time Transport Protocol
RTP/AVP	RTP Audio Video Profile
SDP	Session Description Protocol
S-CSCF	Serving Call Session Control Function
SIP	Session Initiation Protocol
SIPS	SIP Secure
S/MIME	Secure MIME
SSP	SIP Serving Proxy
TLS	Transport Layer Security
TTL	Time to Live
UA	User Agent
URI	Uniform Resource Identifier