Implementation Agreement for CORE SIP Profile, for Voice over IP Version 2

MSF-IA-SIP.002v2-FINAL

MultiService Forum
Implementation Agreement

Contribution Number: msf2005.040.02

Document Filename: MSF-IA-SIP.002v2-FINAL

Working Group: Protocol and Control

Title: Implementation Agreement for CORE SIP Profile, for Voice over IP Version 2

Editors: Shinsaku Ogasawara
NTT Resonant Inc.
s.ogasawara@nttr.co.jp

Satoshi Miyayama
NTT Resonant Inc.
s.miyayama@nttr.co.jp

Paul Drew
MetaSwitch
pd@metaswitch.com

Mitch Laman
Tekelec
mitchell.laman@tekelec.com

Wayne Cutler
Marconi Communications
wayne.cutler@marconi.com

Working Group Chairperson: Chris Gallon, Fujitsu

Date: 14 December 2005

Abstract: This contribution is the draft updated Core MSF Implementation Agreement for SIP Profile, for Voice over IP Version 2. This is a SIP Profile. It specifies a common profile for SIP to be used as part of the other MSF SIP IAs. This is an update to the published Implementation Agreement for Core SIP Profile for Voice over IP MSF-IA-SIP.002-FINAL based on contributions msf2004.124 and msf2005.019, adding support for IPv6 and correcting issues send at GMI 2004.

The goal of the MSF is to promote multi-vendor interoperability as part of a drive to accelerate the deployment of next generation networks. To this end the MSF looks to adopt pragmatic solutions in order to maximize the chances for early deployment in real world networks.

To date the MSF has defined a number of detailed Implementation Agreements and detailed Test Plans for the signaling protocols between network components and is developing additional Implementation Agreements and Test Plans addressing some of the other technical issues such as QoS and Security to assist vendors and operators in deploying interoperable solutions.

The MSF welcomes feedback and comment and would encourage interested parties to get involved in this work program. Information about the MSF and membership options can be found on the MSF website http://www.msforum.org/

DISCLAIMER

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and the MultiService Forum is not responsible for any errors or omissions. The MultiService Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither the MultiService Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind whether based on theories of tort, contract, strict liability or otherwise, shall be assumed or incurred by the MultiService Forum, its member companies, or the publisher as a result of reliance or use by any party upon any information contained in this publication. All liability for any implied or express warranty of merchantability or fitness for a particular purpose is hereby disclaimed.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

Any express or implied license or right to or under any MultiService Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor

Any warranty or representation that any MultiService Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

Any commitment by a MultiService Forum company to purchase or otherwise procure any product(s) and/or service(s) that embody any or all of the ideas, technologies, or concepts contained herein; nor

Any form of relationship between any MultiService Forum member companies and the recipient or user of this document.

Implementation or use of specific MultiService Forum Implementation Agreements, Architectural Frameworks or recommendations and MultiService Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in the MultiService Forum.

For addition information contact:

MultiService Forum
39355 California Street, Suite 307
Fremont, CA 94538
USA
Phone: +1 510 608-5922
Fax: +1 510 608-5917
info@msforum.org
http://www.msforum.org

1. MultiService Switching Forum

The goal of the MSF is to promote multi-vendor interoperability as part of a drive to accelerate the deployment of next generation networks. To this end the MSF looks to adopt pragmatic solutions in order to maximize the chances for early interoperability in real world networks.

To date the MSF has defined a number of detailed Implementation Agreements and detailed Test Plans for the signaling protocols between network components and is developing additional Implementation Agreements and Test Plans addressing some of the other technical issues such as QoS and Security to assist vendors and operators in deploying interoperable solutions.

In 2004, the MSF held a “Global MSF Interoperability 2004” (GMI 2004) event that tested interoperability between next generation network elements situated in Asia, Europe and North America. GMI 2004 validated the MSF release 2 architectural framework and Implementation Agreements by subjecting them to interoperability testing based on realistic network scenarios.

Global MSF Interoperability 2004 (GMI 2004) demonstrated a deployable and operationally ready IP telephony network with Network Management, enhanced Quality-of-Service (QoS) and security features. GMI2004 also demonstrated a service layer with application server, media server, and service broker functionality.

GMI2004 provided an industry showcase that:

- Assist carriers achieve their goal: to deploy flexible, best of breed products.
- Assist vendors achieve their goal: to market products more cost effectively.
- Display the global interoperability of the MSF architecture as referenced in the Release 2 architecture document.
- Demonstrate a network scenario that can be managed to specific quality standards.

The MSF welcomes feedback and comment and would encourage interested parties to get involved in this work program. Information about the MSF and membership options can be found on the MSF website http://www.msforum.org/

2. Scope Of This Document

This Implementation Agreement describes a core profile for using SIP to create and tear down voice over IP sessions between endpoints via SIP. It specifies a common profile for SIP to be used as part of the specific MSF SIP IAs. It is an update to MSF-IA-SIP.002-FINAL applying feedback from GMI 2004 and adding support for IPv6.

Figure 1 shows the MSF architecture diagram for GMI 2004.

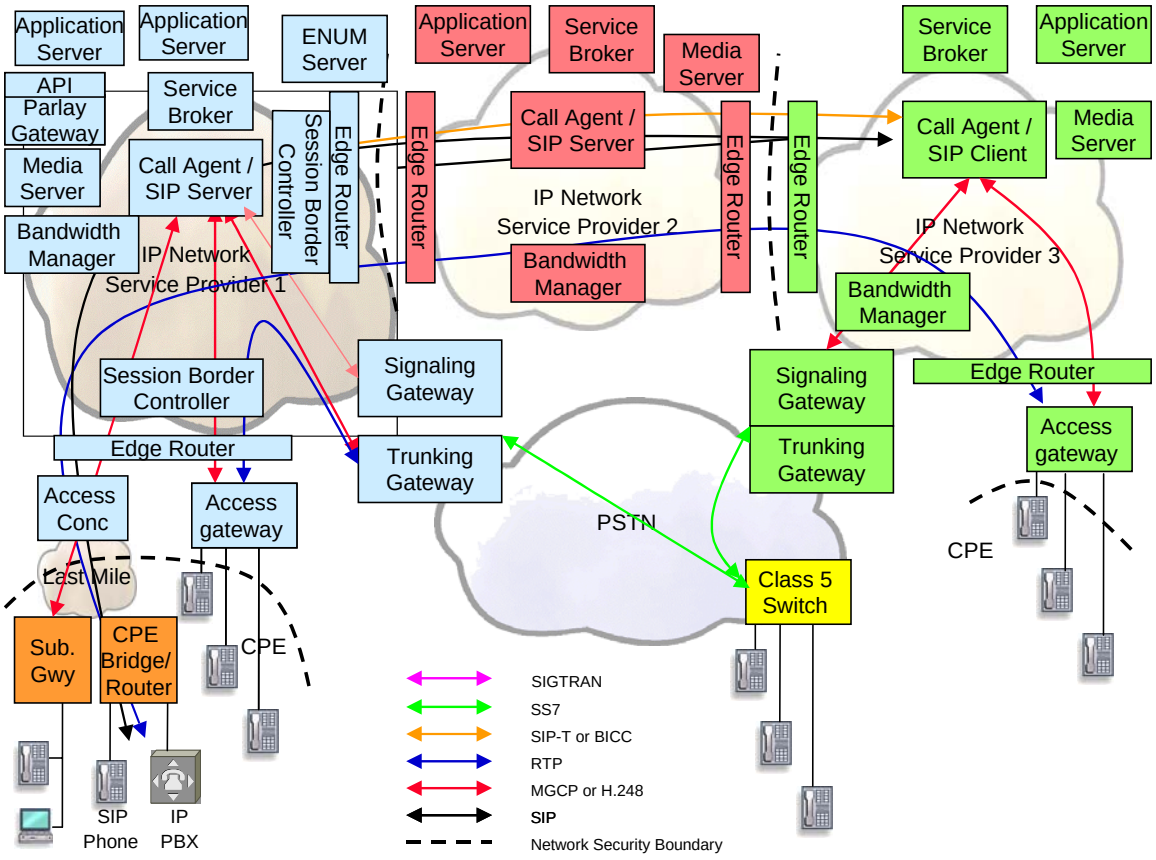


Figure 1: MSF GMI 2004 Architecture

SIP is applicable across many of the interfaces shown in the diagram above. Each SIP interface is explicitly covered in one of the following IAs, but all of them are based on this core SIP IA profile.

- SIP Call Agent to Bandwidth Manager MSF-IA-SIP.010-FINAL
- SIP Call Agent to Call Agent MSF-IA-SIP.004-FINAL
- SIP Call Agent to Service Broker Interface MSF-IA-SIP.005-FINAL
- SIP Call Agent to User Agent interface MSF-IA-SIP.003-FINAL
- SIP Media Server Interface MSF-IA-SIP.009-FINAL
- SIP Service Broker to Application Server MSF-IA-SIP.006-FINAL
- SIP Service Broker to Service Broker MSF-IA-SIP.007-FINAL
- SIP Signaling Security for GMI 2004 MSF-IA-SIP.008-FINAL
- SIP-T Profile for Media Gateway Controller MSF-IA-SIP-T.001.02-FINAL

2.1 Network Components

This IA applies to the following network components.

- SIP User Agent
  This is any kind of device that directly provides phone service using SIP signaling including the following
  - SIP Phones
  - SIP-based Access gateways – these are devices resident at either Central Office or Remote Cabinet locations.
  - ATAs (analog telephony adapters) using SIP
  - IADs (integrated access devices) using SIP

- Call Agent
  This is located in the service provider's network and provides call logic, call control and routing functions, typically maintaining call state for every call in the network.
  Some Call Agents provide support for SIP User Agents (equivalent to a Class 5 switch in the PSTN), other Call Agents just provide call routing function (equivalent to a Class 4 switch in the PSTN).
  The Call Agent will participate in SIP signaling and typically acts as a B2BUA. It is synonymous with a "SIP Server" and "Media Gateway Controller (MGC)".
  Call Agents providing support for SIP User Agents act as a SIP Registrar and typically include service logic for supplementary services, e.g. Caller ID, Call Waiting, as well as supporting interactions with application servers to supply services that are not directly hosted on the Call Agent.

- Application Server
  The Application Server is located in the service provider's network and provides the service logic and execution for one or more applications or services that are not directly hosted on the Call Agent. For example, it may provide voice mail or conference calling facilities. Typically the Call Agent will route calls to the appropriate application server when a service is invoked that the Call Agent cannot itself support.

- Service Broker
  The Service Broker is an optional component located in the service provider's network and enables the deployment and integration of Application Servers. When Service Brokers are deployed they define the rules by which Application Servers exist and interact.

- Media Server
  This Media Server is located in the service provider's network. It provides services such as playing announcements, mixing and interactive voice response (IVR) processing.

- Bandwidth Manager
  The Bandwidth Manager is located in the service provider's network and is responsible for providing the required QoS from the network. It is responsible for the setting up and tearing down of bandwidth within the network and for controlling the access of individual calls to this bandwidth. It is responsible for installing the appropriate policy in edge routers to police the media flows on a per call basis.

2.2 Terminology

This IA uses the following terms as per RFC 3261 [1]

- SIP Network Entity – this refers to any network component that supports SIP signaling, which includes all of the network components described in section 2.1

- UAC (User Agent Client) – this refers to the logical entity within each network component that creates new requests. Most but not all of the network components described in section 2.1 include a UAC. For example Media Servers may not include a UAC.

- UAS (User Agent Server) – this refers to the logical entity within each network component that generates a response to a SIP request. All the network components described in section 2.1 include a UAS.

3. Definitions

SIP is the session initiation protocol defined by IETF in RFC 3261 [1]. It defines the control messages to create and terminate communications sessions between multiple endpoints on the IP networks.

3.1 SIP Transport

RFC 3261 [1] allow SIP requests to be sent using reliable or unreliable protocols: UDP, TCP, or SCTP.

Where the SIP interface crosses a trust boundary, the MSF architecture mandates the use of TLS as described in the SIP Signaling Security IA. This requires support for TCP and optionally SCTP as a transport protocol and applies to the following interfaces

- Call Agent to User Agent interface
- Call Agent to Call Agent interface
- Service Broker to Service Broker interface.

For the following other SIP interfaces support for UDP is required and support for TCP or SCTP is optional

- Call Agent to Bandwidth Manager interface
- Call Agent to Service Broker interface
- Media Server Interface
- Service Broker to Application Server

When UDP is used, only unicast is required.

This IA includes support for both IPv4 and IPv6 although it is implementation dependent whether to support IPv4 or IPV6 or both.

3.1.1 SIP Messages

SIP Messages are the basic language elements in SIP. Each SIP message contains SIP headers and maybe a message body. There are two types of SIP Messages: Request and Response.

3.1.2 SIP Request Methods

SIP Requests are messages, typically, sent by the SIP client to initiate a transaction. For example, an INVITE method starts a new call, an INVITE request sent within an existing dialog is known as a re-INVITE. Note that a single re-INVITE can modify the dialog and the parameters of the session at the same time. Either the caller or callee can modify an existing session. A CANCEL method cancels the request.

3.1.3 SIP Responses

SIP Responses are messages received by the SIP client during a transaction that initiated by a request. One or more responses can take place in answer to a single request.

3.1.4 SIP Headers

SIP Headers are a set of parameters that could be assigned specific values inside a SIP message. They convey information about the SIP request or response.

3.1.5 SIP Profiles

SIP Profiles specify the usage of the SIP protocol for specific applications. They cover which headers and methods are supported, header values supported and which RFC and Internet Drafts are supported. Profiles may be defined by any organization. This implementation agreement defines one such SIP profile for use in MSF compliant deployments.

4. SIP Profile

Support for SIP version 2 as specified in RFC 3261 [1] is mandatory.

Note support for RFC 2543, is not included as part of the MSF GMI 2004.

4.1 SIP Requests

The following sections specify the default requirement for supporting SIP requests (methods) for all MSF compliant devices that support SIP, unless otherwise stated in the detailed SIP IAs.

4.1.1 INVITE

Support for INVITE is mandatory for both UACs and UASs.

The INVITE request is used in many ways: to start a new call, refresh session-timer, reroute (redirect) calls. An INVITE request should have the appropriate headers and may include a SDP body. In case the corresponding INVITE has an SDP body, the ACK should not have an SDP body. The SDP body is necessary for providing early audio. Headers convey specific information about the call such as calling, called party, and the call's unique identification. Subsequent re-INVITE requests are used to refresh session time, or to change RTP receiving address port and codec.

The INVITE request initiates or renegotiates a session may include a SDP body. If originating SIP server does not know its desired characteristic of session until it receives the 200 OK response, the initial INVITE should have not SDP body. In this case the ACK request for this INVITE will contain a SDP body. The INVITE requesting the session-timer refresh requires SDP body. If the INVITE doesn't contain any SDP, the following ACK shall contain a SDP. It is mandatory for a UAS to support receiving INVITE requests with no SDP body.

IPv4 Example:

INVITE SIP:9724401004@172.80.2.100:5060 SIP/2.0
Via: SIP/2.0/UDP 172.17.2.29:5060
From: IP30 Phone 1 <SIP:2143302105@172.17.2.29>;tag=949D.80A7
To: <SIP:9724401004@172.20.2.31>
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21479 INVITE
Content-Length: 148
Content-Type: application/sdp
Contact: SIP:2143302105@172.17.2.29
Session-Expires: 120
Supported: timer, 100rel

v=0
o=- 7323 3661 IN IP4 172.17.2.29
s=- SIP Call
c=IN IP4 12.0.0.102


```
t=0 0
m=audio 9000 RTP/AVP 0 18
a=ptime:20
a=sendrecv
```

IPv6 Example:

```
INVITE SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]:5060 SIP/2.0
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21479 INVITE
Content-Length: 148
Content-Type: application/sdp
Contact: SIP:2143302105@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
Session-Expires: 120
Supported: timer, 100rel

v=0
o=- 7323 3661 IN IP6 3ffe:302:feed:beef:208:74ff:fee1:d3ca
s=- SIP Call
c=IN IP6 3ffe:302:feed:beef:208:74ff:fee1:d3ca
t=0 0
m=audio 9000 RTP/AVP 0 18
a=ptime:20
a=sendrecv
```

4.1.2 ACK

Support for ACK is mandatory for both UACs and UASs.

The ACK request confirms that the client has received a final response to an INVITE request. (ACK is used only with INVITE requests.). The ACK request does not generate responses for any transport protocol. The ACK request contains a message body with the final session description if it is an acknowledgement to an initial INVITE that has no body (see Section 4.1.1). If the ACK message body is empty, the callee uses the session description in the INVITE request.

IPv4 Example:

```
ACK SIP:9724401004@172.80.2.100:5060 SIP/2.0
Via: SIP/2.0/UDP 172.17.2.29:5060
From: IP30 Phone 1 <SIP:2143302105@172.17.2.29>;tag=949D.80A7
To: <SIP:9724401004@172.20.2.31>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21479 ACK
Content-Length: 0
Contact: SIP:2143302105@172.17.2.29
```

IPv6 Example:

```
ACK SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]:5060 SIP/2.0
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>;tag=36
```


Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21479 ACK
Content-Length: 0
Contact: SIP:2143302105@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]

4.1.3 PRACK (Provisional Response ACK)

Support for PRACK is mandatory for both UACs and UASs that support early media, this includes both setting up and receiving early media streams before a 200 OK response has been sent. It is optional but recommended for other UACs and UASs.

The PRACK request confirms that the UAC has received a reliable provisional response (with the exception of '100 Trying') in 1xx reliability. The PRACK request requires a (200 OK) final response to indicate it has been received. The PRACK request MAY contain a message body to establishing early media session.

IPv4 Example:

PRACK SIP:9724401004@172.80.2.100:5060 SIP/2.0
Via: SIP/2.0/UDP 172.17.2.29:5060
From: IP30 Phone 1 <SIP:2143302105@172.17.2.29>;tag=949D.80A7
To: <SIP:9724401004@172.20.2.31>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21480 PRACK
Content-Length: 0
Contact: SIP:2143302105@172.17.2.29
RAck: 14763 21479 INVITE

IPv6 Example:

PRACK SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]:5060 SIP/2.0
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21480 PRACK
Content-Length: 0
Contact: SIP:2143302105@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
RAck: 14763 21479 INVITE

4.1.4 CANCEL

Support for CANCEL is mandatory for both UACs and UASs.

The CANCEL request cancels a pending request with the same Call-ID, To, From, and CSeq (sequence number only) header field values, but does not affect a completed request or existing calls. The UAC can use a BYE request to terminate a call if the CANCEL arrived too late. The CANCEL request can only be sent after at least one 1xx provisional response has been received.

IPv4 Example:

CANCEL SIP:9724401004@172.100.2.100:5060 SIP/2.0
Via: SIP/2.0/UDP 172.17.2.29:5060
From: IP30 Phone 2 <SIP:2143302105@172.17.2.29>; tag=949D.80A7

To: <SIP:9724401004@172.20.2.31>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21479 CANCEL
Content-Length: 0
Contact: SIP:2143302105@172.17.2.29

IPv6 Example:

CANCEL SIP:9724401004@172.100.2.100:5060 SIP/2.0
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>; tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21479 CANCEL
Content-Length: 0
Contact: SIP:2143302105@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]

4.1.5 BYE

Support for BYE is mandatory for both UACs and UASs.

A UA uses a BYE request to indicate to the other UA that it wishes to release the call. A BYE request from either caller or callee terminates any pending INVITE at a UA, but the INVITE request transaction MUST be completed with a final response and an ACK. If the INVITE request contained a Contact header, the callee should send the BYE request to that address rather than the From address. If the UAC included a ROUTE header field in the original INVITE, it should include the same route set in any BYE it sends.

IPv4 Example:

BYE SIP:2143302105@172.17.2.29 SIP/2.0
Via: SIP/2.0/UDP 172.80.2.100:5060
From: <SIP:9724401004@172.20.2.31>;tag=36
To: IP30 Phone 1 <SIP:2143302105@172.17.2.29>;tag=949D.80A7
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 7582 BYE
Supported: 100rel
Supported: timer
Content-Length: 0

IPv6 Example:

BYE SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce] SIP/2.0
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 7582 BYE
Content-Length: 0

4.1.6 INFO

Support for INFO (RFC 2976 [2]) is required for the following scenarios

- Supporting SIP-T (SIP-I).
- Certain applications to pass information between a UA and an Application Server.

Therefore support INFO is mandatory for the following network components

- Call Agents supporting SIP-T (SIP-I)
- SIP User Agents to support services requiring INFO
- Call Agents – pass through to Service Brokers
- Service Brokers – pass through to Application Servers
- Application Servers supporting services requiring INFO.

4.1.7 UPDATE

Support for UPDATE (RFC 3311 [11]) is optional but recommended for both UACs and UASs.

The caller begins with an INVITE transaction, which proceeds normally. Once a dialog is established, either early or confirmed, the caller can generate an UPDATE method that contains an SDP offer for the purposes of updating the session. The response to the UPDATE method contains the answer. Similarly, once a dialog is established, the callee can send an UPDATE with an offer, and the caller places its answer in the 2xx to the UPDATE. UPDATE is the recommended method for the SIP Session Timer.

UPDATE is the recommended method for providing resource management (QoS), see section 4.5.

4.1.7.1 Determining Support for UPDATEs

When a UAS compliant to this specification receives an INVITE request for a new dialog, and generates a reliable provisional response containing SDP, that response SHOULD contain an Allow header field that lists the UPDATE method. This informs the caller that the callee is capable of receiving an UPDATE request at any time. An unreliable provisional response MAY contain an Allow header field listing the UPDATE method, and a 2xx response SHOULD contain an Allow header field listing the UPDATE method.

4.1.7.2 Handling an UPDATE request

UPDATE is a target refresh request. As specified in RFC 3261 [1] and RFC 3311 [11], this means that it can update the remote target of a dialog. If a UA uses an UPDATE request or response to modify the remote target while an INVITE transaction is in progress, and it is a UAS for that INVITE transaction, it MUST place the same value into the Contact header field of the 2xx to the INVITE that it placed into the UPDATE request or response.

If a UA receives an UPDATE for an existing dialog, it MUST check any version identifiers in the session description or, if there are no version identifiers, the content of the session description to see if it has changed. If the session description has changed, the UAS MUST adjust the session parameters accordingly and generate an answer in the 2xx response.

4.1.8 OPTIONS

Support for OPTIONS is optional for both UACs and UASs.

The OPTIONS request is used to query the capabilities of a UA.

4.1.9 REGISTER

Support for REGISTER is mandatory for SIP User Agents and Call Agents supporting SIP User Agents. It is optional for all other network components, including Call Agents just providing SIP or SIP-T trunking.

SIP User Agents must register with a Call Agent (SIP Server/SIP Registrar) using the REGISTER method.

4.1.10 SUBSCRIBE

Support for SUBSCRIBE (RFC 3265 [17]) is mandatory for the following network components

- SIP User Agents
- Call Agents supporting SIP User Agents
- Service Brokers
- Application Servers supporting either voicemail, call transfer, presence or other services requiring SUBSCRIBE

It is optional for all other network components.

SUBSCRIBE is a dialog-creating method, as described in RFC 3261 [1].

When a subscriber wishes to subscribe to a particular state for resource, it forms a SUBSCRIBE message. If the initial SUBSCRIBE represents a request outside of a dialog (as it typically will), its construction follows the procedures outlined in RFC 3261 [1] for UAC request generation outside of a dialog.

This SUBSCRIBE request will be confirmed with a final response. 200-class responses indicate that the subscription has been accepted, and that a NOTIFY will be sent immediately. A 200 response indicates

that the subscription has been accepted and that the user is authorized to subscribe to the requested resource. A 202 response merely indicates that the subscription has been understood, and that

authorization may or may not have been granted.

The "Expires" header in a 200-class response to SUBSCRIBE indicates the actual duration for which the subscription will remain active (unless refreshed).

Non-200 class final responses indicate that no subscription or dialog has been created, and no subsequent NOTIFY message will be sent. All non-200 class responses (with the exception of "489", as described in RFC 3265 [17]) have the same meanings and handling as described in RFC 3261 [1].

SUBSCRIBE is used to provide services such as message waiting indicator for voicemail and call transfer using the REFER method.

4.1.11 NOTIFY

Support for NOTIFY (RFC 3265 [17]) is mandatory for the following network components

- SIP User Agents
- Call Agents supporting SIP User Agents
- Service Brokers

- Application Servers supporting either voicemail, call transfer, presence or other services requiring NOTIFY.

It is optional for all other network components.

NOTIFY messages are sent to inform subscribers of changes in state to which the subscriber has a subscription. Subscriptions are typically put in place using the SUBSCRIBE method; however, it is possible that other means have been used.

A NOTIFY does not terminate its corresponding subscription; in other words, a single SUBSCRIBE request may trigger several NOTIFY requests.

NOTIFY is used along with SUBSCRIBE to provide services such as message waiting indicator for voicemail and call transfer using the REFER method.

4.1.12 REFER

Support for REFER (RFC 3515 [13]) is mandatory for the following network components

- SIP User Agents
- Call Agents supporting SIP User Agents
- Service Brokers
- Application Servers supporting call transfer or other services requiring REFER.

It is optional for all other network components.

REFER is used along with SUBSCRIBE and NOTIFY to provide call transfer services. It also requires support for the message/sipfrag MIME media type as specified in RFC 3420 [16].

4.2 Supported Responses

The following sections specify the default requirement for supporting SIP responses for all MSF compliant devices that support SIP, unless otherwise stated in the detailed SIP IAs.

4.2.1 1xx Provisional Responses

Support for the following SIP responses is mandatory for all UAs.

- 100 Trying
- 180 Ringing
- 183 Session Progress

Support for sending other provisional responses below is optional and no specific action is required when one is received.

- 181 Call Is Being Forwarded
- 182 Queued

4.2.1.1 100 Trying

The 100 response is sent whenever there is not immediate response available. It does not have a body.

IPv4 Example:

SIP/2.0 100 Trying
Via: SIP/2.0/UDP 172.17.2.29:5060
From: IP30 Phone 1 <SIP:2143302105@172.17.2.29>
To: <SIP:9724401004@172.20.2.31>
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21479 INVITE
Content-Length: 0

IPv6 Example:

SIP/2.0 100 Trying
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21479 INVITE
Content-Length: 0

4.2.1.2 180 Ringing

The 180 response is sent when the UAS is alerting the endpoint. This response may contain a SDP body, which also includes information of the codec and audio port for early audio session. If this response has no SDP body the UAC will provide the ring-back tone.

IPv4 Example:

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 172.17.2.29:5060
From: IP30 Phone 1 <SIP:2143302105@172.17.2.29>;tag=949D.80A7
To: <SIP:9724401004@172.20.2.31>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21479 INVITE
Contact: <SIP:9724401004@172.80.2.100:5060>
Require: 100rel
RSeq: 14763
Content-Length: 0

IPv6 Example:

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21479 INVITE
Contact: <SIP:9724401004@[3ffe:302:feed:beef:208:74ff:fee0:ca70]:5060>
Require: 100rel
RSeq: 14763
Content-Length: 0

4.2.1.3 183 Session Progress

The 183 response is sent when audio has to be sent before the phones are connected. This response may contain a SDP body, which also includes information of the codec and audio port for early audio session. This response must contain a SDP body, if the UAS wants to provide early media using reliable delivery of provisional responses, for example an announcement or its own version of ring-back tone.

IPv4 Example:

```
SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 172.17.2.19:5060
From: "9721112222" <SIP:9721112222@172.17.2.29>
To: <SIP:2143302105@172.17.2.29>;tag=11F7.00A0
Call-ID: afb90200-58fc0-a1fe-2e323731@172.17.2.19
CSeq: 101 INVITE
Content-Length: 204
Content-Type: application/sdp
Content-Disposition: session
Contact: SIP:2143302105@172.17.2.29

v=0
o=- 7300 3650 IN IP4 172.17.2.29
s=- SIP Call
c=IN IP4 172.17.2.33
t=0 0
m=audio 32793 RTP/AVP 18 101
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

IPv6 Example:

```
SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 101 INVITE
Content-Length: 204
Content-Type: application/sdp
Content-Disposition: session
Contact: <SIP:9724401004@[3ffe:302:feed:beef:208:74ff:fee0:ca70]:5060>

v=0
o=- 7300 3650 IN IP6 3ffe:302:feed:beef:208:74ff:fee0:ca70
s=- SIP Call
c=IN IP6 3ffe:302:feed:beef:208:74ff:fee0:ca70
t=0 0
m=audio 32793 RTP/AVP 18 101
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```


4.2.1.4 Provisional Response Supported Headers

All SIP 1xx responses support To, From, Via, Call-ID, CSeq, and Content-Length headers. Besides these, they also support other headers including, but not limited to those, in Tables 2 and 3 of RFC 3261 [1].

4.2.2 2xx Successful Responses

Support for 200 OK SIP responses is mandatory for all UAs.

Support for 202 Accepted SIP responses is mandatory for UACs supporting SUBSCRIBE and NOTIFY methods and optional for UASs supporting SUBSCRIBE and NOTIFY methods. Support is not required for other UAs not supporting SUBSCRIBE or NOTIFY.

4.2.2.1 200 OK

This successful response is sent whenever a request is correctly processed. A 200 OK response will be sent to INVITE, PRACK, INFO, BYE and CANCEL if the request is properly processed. It may contain a body to exchange the callee's capabilities. The 200 OK response to a BYE or CANCEL has no body.

The 200 OK response supports To, From, Via, Call-ID, CSeq, Content-Length, RSeq, and Contact headers, but are not limited to these headers.

IPv4 Example:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.17.2.29:5060
From: IP30 Phone 1 <SIP:2143302105@172.17.2.29>;tag=949D.80A7
To: <SIP:9724401004@172.20.2.31>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21479 INVITE
Contact: <SIP:9724401004@172.80.2.100:5060>
Allow: OPTIONS, CANCEL, INVITE, ACK, PRACK, INFO, BYE
Session-Expires: 120
Require: timer
Content-Length: 152
Content-Type: application/sdp

v=0
o=- 7582 14763 IN IP4 172.80.2.100
s=SIP Media Capabilities
c=IN IP4 172.70.2.8
t=0 0
m=audio 5082 RTP/AVP 18
a=sendrecv
a=ptime:20
```

IPv6 Example:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>;tag=36
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21479 INVITE
```


Contact: <SIP:9724401004@[3ffe:302:feed:beef:208:74ff:fee0:ca70]:5060>
Allow: OPTIONS, CANCEL, INVITE, ACK, PRACK, INFO, BYE
Session-Expires: 120
Require: timer
Content-Length: 152
Content-Type: application/sdp

v=0
o=- 7582 14763 IN IP6 3ffe:302:feed:beef:208:74ff:fee0:ca70
s=SIP Media Capabilities
c=IN IP6 3ffe:302:feed:beef:208:74ff:fee0:ca70
t=0 0
m=audio 5082 RTP/AVP 18
a=sendrecv
a=ptime:20

4.2.2.2 202 Accepted

The 202 Accepted response is sent when a SUBSCRIBE request has been accepted for processing, but the processing has not been completed. The request might or might not eventually be acted upon, as it might be disallowed when processing actually takes place. There is no facility for re-sending a status code from an asynchronous operation such as this.

The 202 response is intentionally non-committal. Its purpose is to allow a server to accept a request for some other process (perhaps a batch-oriented process that is only run once per day) without requiring that the user agent's connection to the server persist until the process is completed. The entity returned with this

response SHOULD include an indication of the request's current status and either a pointer to a status monitor or some estimate of when the user can expect the request to be fulfilled.

4.2.3 3xx Redirection Responses

Support for 3xx Redirection Responses is optional for all UAs.

3xx Redirection Responses support To, From, Via, Call-ID, CSeq, Content-Length, and Contact headers, but are not limited to these headers. UACs that support 3xx Redirection Responses only interpret the Contact headers in the 3xx Response, which can be used to redirect the call. The 3xx responses in SIP are

- 300 Multiple Choices
- 301 Move Permanently
- 302 Moved Temporarily
- 305 Use Proxy
- 380 Alternative Service

UACs that do not support 3xx Redirection Responses treat a 3xx response as a 4xx Request Failure response.

IPv4 Example of a 300 Multiple Choice response:

SIP/2.0 300 Multiple Choices

Via: SIP/2.0/UDP 172.17.2.29:5060
From: IP30 Phone 1 <SIP:2143302105@172.17.2.29>;tag=949D.80A7
To: <SIP:9724401004@172.20.2.31>
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21478 INVITE
Content-Length: 0
Contact: SIP:9724401004@172.80.2.100:5060

IPv6 Example of a 300 Multiple Choice response:

SIP/2.0 300 Multiple Choices
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21478 INVITE
Content-Length: 0
Contact: SIP:9724401005@[3ffe:302:feed:ffff::d49d:cdce]:5060

4.2.4 4xx Request Failure Responses

Support for 4xx Request Failure responses is mandatory for all UAs.

When a UAC receives a 4xx response it terminates the request. When the 4xx response is received for an INVITE an ACK is sent to acknowledge it. Receiving a 4xx response to a re-INVINITE does not terminate the dialog except as specified in section 14.1 of RFC 3261 [1].

4.2.4.1 400 Bad Request

The request could not be understood due to malformed syntax. The Reason-Phrase SHOULD identify the syntax problem in more detail, for example, "Missing Call-ID header field".

4.2.4.2 401 Unauthorized

A 401 response is sent when user authentication is required as described in the MSF Implementation Agreement for SIP Signalling Security for GMI 2004 [15].

4.2.4.3 403 Forbidden

A 403 response is sent when the request is refused.

4.2.4.4 404 Not Found

A 404 response is sent when the called number cannot be found in the database.

4.2.4.5 405 Method Not Allowed

A 405 response is sent when the method is not supported

4.2.4.6 406 Not Acceptable

A 406 response is sent when the UAS can only generate responses that have characteristics not acceptable according to the Accept header sent in the request.

4.2.4.7 407 Proxy Authentication Required

A 407 response is sent when user authentication is required by a proxy as described in the MSF Implementation Agreement for SIP Signalling Security for GMI 2004 [15].

4.2.4.8 408 Request Timeout

A 408 response is sent when the UAS could not resolve the request within a suitable amount of time.

4.2.4.9 410 Gone

A 410 response is requested resource is no longer available at the UAS.

4.2.4.10 413 Request Entities Too Large

A 413 response is sent when the UAS could not resolve the request because the request's entity-body is larger than the UAS expects. When this response is received the UAC may retry the request with a shorter or no body, but the mandatory information for the session description should not be removed.

4.2.4.11 414 Request URI Too Long

A 414 response is sent when the request's URI is longer than the UAS expects.

4.2.4.12 415 Unsupported Media Type

A 415 response is sent when the request's body is in a format that is not supported by the UAS. When this response is received the UAC should retry the request with the Accept, Accept-Encoding, and/or Accept-Languages headers.

4.2.4.13 416 Unsupported URI Scheme

The server cannot process the request because the scheme of the URI in the Request-URI is unknown to the UAS.

4.2.4.14 420 Bad Extension

A 420 response is sent by a UAS if it could not understand the extension in Proxy-Require or Require header. The receiving UAC may retry the request. If retrying, the UAC must omit any extension listed in this response's Unsupported header.

4.2.4.15 421 Extension Required

The UAS needs a particular extension to process the request, but this extension is not listed in a Supported header field in the request. Responses with this status code MUST contain a Require header field listing the required extensions.

A UAS SHOULD NOT use this response unless it truly cannot provide any useful service to the client. Instead, if a desirable extension is not listed in the Supported header field, servers SHOULD process the request using baseline SIP capabilities and any extensions supported by the client.

4.2.4.16 422 Session Interval Too Small

It is generated by a UAS or proxy when a request contains a Session-Expires header field with a duration that is below the minimum timer for the server. The 422 response MUST contain a Min-SE header field with the minimum timer for that server.

4.2.4.17 423 Interval Too Brief

The UAS is rejecting the request because the expiration time of the resource refreshed by the request is too short. This response can be used by a registrar to reject a registration whose Contact header field expiration time was too small.

4.2.4.18 480 Temporarily not available

A 480 response is sent when the phone is temporary unavailable to receive call.

4.2.4.19 481 Call Leg/Transaction Does Not Exist

A 481 response is sent when a call cannot be found and the To header of the request has a tag.

4.2.4.20 482 Loop Detected

A 482 response is sent when the UAS has detected a loop.

4.2.4.21 483 Too Many Hops

A 483 response is sent when the UAS receives a request that contains a Max-Forwards header field with the value zero.

4.2.4.22 484 Address Incomplete

A 484 response is sent when the UAS receives a request with a Request-URI that was incomplete. Additional information SHOULD be provided in the reason phrase.

This status code allows overlapped dialing. With overlapped dialing, the client does not know the length of the dialing string. It sends strings of increasing lengths, prompting the user for more input, until it no longer receives a 484 (Address Incomplete) status response.

4.2.4.23 485 Ambiguous

A 485 response is sent when the UAS receives a request with a Request-URI that is ambiguous.

4.2.4.24 486 Busy Here

A 486 response is sent when the called party is busy.

4.2.4.25 487 Request Cancelled

A 487 response is sent when a request has not sent a final response upon receiving a CANCEL request.

4.2.4.26 488 Not Acceptable Here

A 488 response is sent when a request body contains session description that is not acceptable by the receiving UAS.

4.2.4.27 491 Request Pending

The request was received by a UAS that had a pending request within the same dialog.

4.2.4.28 493 Undecipherable

The request was received by a UAS that contained an encrypted MIME body for which the recipient does not possess or will not provide an appropriate decryption key. This response MAY have a single body containing an appropriate public key that should be used to encrypt MIME bodies sent to this UA. Details of the usage of this response code can be found in Section 23.2 of RFC 3261 [1].

4.2.5 5xx Server Error Responses

Support for 5xx Request Failure responses is mandatory for all UAs.

When a UAC receives a 5xx response it terminates the request. When the 5xx response is received for an INVITE an ACK is sent to acknowledge it. Receiving a 5xx response to a re-INVINITE does not terminate the dialog except as specified in section 14.1 of RFC 3261 [1].

4.2.5.1 500 Internal Server Error

A 500 response is sent when there is some kind of error at the UAS side.

4.2.5.2 501 Not Implemented

A 501 response is sent when the UAS does not support the functionality required to fulfill the request.

4.2.5.3 502 Bad Gateway

A 502 response is sent when the UAS, while acting as a proxy or gateway, received an invlided response from the downstream server.

4.2.5.4 503 Service Unavailable

A 503 response is sent when the UAS is temporary unable to process the request.

4.2.5.5 504 Server Time-out

A 504 response is sent when the UAS did not receive a timely response from an external server it accessed in attempting to process the request.

4.2.5.6 505 Version Not Supported

A 504 response is sent when the UAS does not support the version of SIP specified in the request.

4.2.6 6xx Global Failure Responses

Support for 6xx Global Failure responses is mandatory for all UAs.

When a UAC receives a 6xx response it terminates the request. When the 6xx response is received for an INVITE an ACK is sent to acknowledge it. Receiving a 6xx response to a re-INVINITE does not terminate the dialog except as specified in section 14.1 of RFC 3261 [1].

4.2.6.1 600 Busy Everywhere

A 600 response is sent when the called party is busy and does not wish to take the call at this time.

4.2.6.2 603 Decline

A 603 response is sent when the called party does not wish to take the call at this time.

4.2.6.3 604 Does Not Exist Anywhere

A 604 response is sent when the UAS has authoritative information that user indicated in the Requets-URI does not exist anywhere.

4.2.6.4 606 Not Acceptable

A 606 response is sent when some aspects of the session description such as codec were not acceptable.

4.3 Supported Headers

The following sections specify the default requirement for supporting SIP headers for all MSF compliant devices that support SIP, unless otherwise stated in the detailed SIP IAs.

The following is an example INVITE showing a variety of headers. There are other headers not shown in the example INVITE request.

IPv4 example

INVITE SIP:9724401004@172.80.2.100:5060 SIP/2.0
Via: SIP/2.0/UDP 172.17.2.29:5060
Max-Forwards: 70
From: IP30 Phone 1 <SIP:2143302105@172.17.2.29>;tag=949D.80A7
To: <SIP:9724401004@172.20.2.31>
Call-ID: 0800.20CE.A152.3C44.949D.80A7@172.17.2.29
CSeq: 21479 INVITE
Content-Length: 148
Content-Type: application/sdp
Contact: SIP:2143302105@172.17.2.29
Session-Expires: 120
Supported: timer, 100rel
Resource-Priority: dsn.flash

IPv6 example

INVITE SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]:5060 SIP/2.0
Via: SIP/2.0/UDP [3ffe:302:feed:beef:208:74ff:fee1:d3ca]:5060
Max-Forwards: 70
From: IP30 Phone 1 <SIP:2143302105@[3ffe:302:feed:ffff::d49d:cdce]>;tag=949D.80A7
To: <SIP:9724401004@[3ffe:302:feed:ffff::d49d:cdce]>
Call-ID: 0800.20CE.A152.3C44.949D.80A7@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
CSeq: 21479 INVITE
Content-Length: 148
Content-Type: application/sdp
Contact: SIP:2143302105@[3ffe:302:feed:beef:208:74ff:fee1:d3ca]
Session-Expires: 120
Supported: timer, 100rel
Resource-Priority: dsn.flash

4.3.1 Request Line

Support for the request line is mandatory for both UACs and UASs.

The request line has the method name (e.g. INVITE) and the request URI (e.g. 9724401004@172.80.2.100:5060 or 9724401004@[3ffe:302:feed:ffff::d49d:cdce]:5060). Unless specified in a specific SIP IA, the request URI consists of a phone number and an IP address or domain name.

4.3.2 Via

Support for the Via header is mandatory for both UACs and UASs.

Via header is inserted to the request message as it passes through a user agent. It is used for loop detection as well as response path.

4.3.3 From

Support for the From header is mandatory for both UACs and UASs.

The From header conveys the calling party information, except in privacy call. In a non-privacy call, it may have the name (e.g. IP30 Phone 1) and URL of the calling party. The phone number is parsed from the URL. This means that a phone number must be present. The From field MUST contain a unique tag parameter on the first and subsequent requests.

For calls that withhold caller id (privacy calls) the name and user part are replaced with the string 'anonymous' as per the example below.

From: Anonymous<sip:anonymous@domain.com>;tag=server+1+24710+b8441f

4.3.4 To

Support for the To header is mandatory for both UACs and UASs.

The To header conveys the called party information. This information remains unchanged even when the call is redirected. The To header on the first INVITE message must not contain a tag parameter. If a tag parameter exists, it indicates that this is a re-INVITE.

4.3.5 Call-ID

Support for the Call-ID header is mandatory for both UACs and UASs.

The Call-ID header contains a globally unique value for identifying the call the request is associated.

4.3.6 CSeq

Support for the CSeq header is mandatory for both UACs and UASs.

The CSeq header indicates the current sequence number for the request. Subsequent requests must have a monotonically increasing number. Besides the number, the method name must be present.

4.3.7 Content-Length

Support for the Content-Length header is mandatory for both UACs and UASs.

The Content-Length header indicates the length of the body that follows the headers. When the length is zero, the body is absent and Content-Type header is not present.

4.3.8 Content-Type

Support for the Content-Type header is mandatory for both UACs and UASs.

4.3.9 Contact

Support for the Contact header is mandatory for both UACs and UASs.

The Contact header contains a SIP or SIPS URI that represents a direct route to contact, usually composed of a fully qualified domain name (FQDN) or IP address.

4.3.10 Session-Expires

Support for the Session-Expires header is optional for both UACs and UASs.

The Session-Expires header defines the time interval for session heartbeat timer (See section 4.7 - SIP Session Timer).

4.3.11 Supported

Support for the Supported header is mandatory for both UACs and UASs.

The Supported header indicates which optional extensions are supported. Session timer support is indicated by the value of “timer” and reliable 1xx responses is indicated by the value “100rel”. Support for session timers is optional but support for reliable 1xx responses is conditionally mandatory.

4.3.12 Record-Route

Support for the Record-Route header is mandatory for both UACs and UASs.

The Record-Route header is inserted to INVITE request by a Proxy to ensure that it will receive all subsequence requests and responses belong to the same session. Reference is made to section 16.12 of RFC 3261 for example flows on the use of this header.

4.3.13 Route

Support for the Route header is mandatory for both UACs and UASs.

The Route headers are a set of addresses of the proxies that a request must traverse as it routed between the UAC and UAS. Reference is made to section 16.12 of RFC 3261 for example flows on the use of this header.

4.3.14 Resource-Priority

Support for the resource priority headers, as defined in [18], is mandatory for the following network components

- Call Agents
- Service Brokers
- Application Servers

- Media Server
- Bandwidth Manager

"Resource-Priority" and "Accept-Resource-Priority" are two SIP header fields for communicating resource priority. The Resource-Priority header field can influence the behavior of SIP UAs, such as PSTN gateways, and SIP proxies. The header fields MAY be used by SIP user agents, including PSTN gateways and terminals, and SIP proxy servers to influence their treatment of SIP requests, including the priority afforded to PSTN calls. For PSTN gateways, the behavior translates into analogous schemes in the PSTN, for example the ITU Recommendation Q.735.3 prioritization mechanism, in both the PSTN-to-IP and IP-to-PSTN directions. If the network element does not act upon the header fields it is simply required to ignore and forward them on.

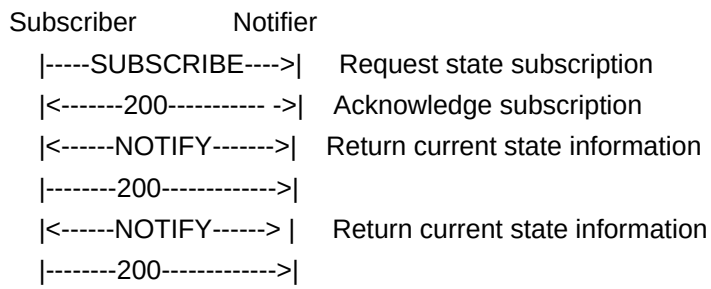
4.4 Event Notification

Support for event notification is mandatory for the following network components

- SIP User Agents
- Call Agents
- Service Brokers
- Application Servers supporting either voicemail or call transfer

Event notification is used by entities in the network to subscribe to resource or call state for various resources or calls in the network, and those entities (or entities acting on their behalf) can send notifications when those states change as specified in RFC 3265 [17]

A typical flow of messages would be:



Subscriptions are expired and must be refreshed by subsequent SUBSCRIBE messages.

If necessary, clients may probe for the support of SUBSCRIBE and NOTIFY using the OPTIONS request defined in RFC 3261 [1].

The presence of the "Allow-Events" header in a message is sufficient to indicate support for SUBSCRIBE and NOTIFY.

4.5 Resource Management (QOS)

The MSF QoS architecture MSF-AF-QOS.001-FINAL [MSF-3] recommends the use of preconditions as specified by RFC 3312 [12].

This minimizes the chances of a session establishment failure due to the inability to reserve network resources once the callee has been alerted, "ghost rings". RFC 3312 [12] introduces the concept of a precondition. A precondition is a set of constraints about the session which are introduced in the offer. The recipient of the offer generates an answer, but does not alert the user or otherwise proceed with session establishment. That only occurs when the preconditions are met. This can be known through a local event (such as a confirmation of a resource reservation), or through a new offer sent by the caller. This is described in more detail in msf2004.001 [MSF-12].

The following alternatives are also allowed

- No support for RFC 3312 [12]
- No SDP on initial INVITE (deferred model)

4.6 SIP 100rel Extension

RFC 3262 [7] defines a mechanism for providing reliable responses using SIP. This is required for interworking with the PSTN and support is conditionally mandatory for all UAs.

When applicable the mechanism requires the use of the following

- Rseq header
- RAck header
- 100rel option tag
- PRACK method

4.6.1 RSeq Header

RSeq header is an optional response header. The RSeq header is used in provisional responses in order to transmit them reliably. It contains a single numeric value from 1 to 2**32.

4.6.2 RAck Header

The RAck header is sent in a PRACK request to support reliability of provisional responses. It contains two numbers and a method tag. The first number is the value from the RSeq header in the provisional response that is being acknowledged. The next number, and the method, are copied from the CSeq in the response that is being acknowledged. The method name in the RAck header is case sensitive.

4.7 SIP Session Timer Extension

Session Timer is a session heartbeat monitoring logic, where UAS or UAC sends periodic re-INVITEs or UPDATE to keep the session alive. If a re-INVITE or UPDATE is not received before the interval passes, UAS/UAC will send a BYE to terminate the session. Similarly, if there is no response to the session timer re-INVITE or UPDATE, UAS/UAC will also send a BYE to terminate the session.

The interval for re-INVITEs and UPDATEs is negotiable. The rules for this negotiation are defined in RFC 4028 [5].

A UA may not support session timer, and yet will interoperate with one that does not support the session timer. This UA will see them as repetitive re-INVITEs or UPDATEs.

A UA that does not support session timer MAY use a re-INVITE as a heartbeat mechanism. The consequence of this is that a UA that does support session timer shall accept and respond to re-INVITES from the UA that does not support session timer.

4.7.1 Supported Header

A UAC starts by sending an INVITE. It includes a Supported header that may contain the option tag "timer", indicating support for this extension.

4.7.2 Require Header

The UAC MAY include a Require header field in the request with the value "timer" to indicate that the UAS must support the session timer to participate in the session. This does not mean that the UAC is requiring the UAS to perform the refreshes; just that it is requiring the UAS to support the extension. In addition, the UAC MAY include a Proxy-Require header field in the request with the value "timer" to indicate that proxies must support session timer in order to correctly process the request. However, usage of either Require or Proxy-Require by the UAC is NOT RECOMMENDED. They are not needed, since the extension works even when only the UAC supports the extension. The Supported header field containing "timer" MUST still be included even if the Require or Proxy-Require header fields are present containing "timer".

4.7.3 Session-Expires Header

Session-Expires header can be placed only in INVITE or UPDATE requests, and in any 2xx final response to an INVITE or UPDATE. Its value, <delta time in seconds>, indicates the desired expiration time of the session.

4.7.4 Min-SE Header

Min-SE, minimum Session-Expires, header value, <delta time in seconds>, indicates the minimum value of the session interval the UAC is willing to accept. When it is not present in the INVITE, the default value is ninety (90) seconds. It must be present in the 422 response.

4.7.5 Behavior as a UAC

Specified in RFC 4028 [5] section 7.

4.7.6 Behavior as a UAS

Specified in RFC 4038 [5] section 9.

4.8 SIP Privacy Extension

Privacy is an extension to SIP that allows parties in a SIP session to withhold their identity and remain anonymous. This is equivalent to withholding caller id in the PSTN. It is specified by RFC 3323 [6] and support is mandatory for all UAs. A UAC indicates that it wishes to make an anonymous call it uses the value “anonymous” From header field, e.g.

From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748

The Call Agent or a suitable Application Server selected by the Service Broker is responsible for acting as the “anonymizer” function as described in RFC 3323. In addition Call Agents must support the P-Asserted-Identity header as specified in RFC 3325 [10] to provide network asserted identity to other trusted network elements such as Service Brokers, suitable Application Servers and other Call Agents.

4.8.1 Privacy Header

The UAC can use the Privacy Header to indicate that it wishes to use the services of the network "anonymizer" function for this call.

4.8.2 P-Asserted-Identity Header

The P-Asserted Identity Header is added by a Call Agent to provide a network asserted identity for a user. It must only be sent within a trust domain and hence not be sent by or to a SIP User Agent.

```
INVITE sip:14085551212@pstn.net SIP/2.0
Via: SIP/2.0/TCP useragent.com;branch=z9hG4bK-124
To: <sip:14085551212@pstn.net>
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748
Call-ID: 245780247857024504
CSeq: 2 INVITE
Contact: <sip:anonymous@pstn.net>
Max-Forwards: 69
P-Asserted-Identity: "Joe Bloggs" <sip:15073451340@pstn.net>
Privacy: id
```

4.9 SIP Diversion Header

Before and during GMI 2004 the SIP Diversion header was used but this is now replaced with History-Info header as specified in section 4.10. This section is now historic but left as a description of the use of the SIP Diversion header.

The SIP Diversion Header [8] is used when a call is forwarded to indicate the forwarding party and reason.

Support is optional for all SIP UAs that provide call forwarding support, in particular Call Agents, Service Brokers and Application Servers. Support for the Diversion Header is not required when using SIP-T (SIP-I) as the diversion information is already present in the tunneled ISUP.

While the Diversion Header is widely supported today it is a proprietary header and it not on the IETF standards track. The IETF mandated alternative is to use the History header as specified in draft-ietf-sip-history-info-0 [19]. For the purposes of GMI 2004 the Diversion header will be used but it will be replaced by the History header after this event.

For example

```
INVITE sip:orig-party@appserver.com SIP/2.0
Via: SIP/2.0/UDP callagent.com;branch=z9hG4bK-metaswitch.com-1-166d14-4ff89cbb
Max-Forwards: 70
To: 12012002345@callagent.com
From: 12013001234@callagent.com;tag= callagent.com+1+24710+b8441f77
Call-ID: F5329DFF@callagent.com
CSeq: 427144613 INVITE
Contact: <sip:12013001234@callagent.com>
Expires: 180
```


Route: <sip: F5329DFF@callagent.com;lr>

P-Charging-Vector: icid-value=F5329DFF@callagent.com

Diversion: 12014003456@callagent.com; reason=unconditional

Referred-By: 12014003456@callagent.com

Content-Type: application/sdp

Content-Length: 128

<SDP not shown>

4.9.1 Diversion Header

Diversion headers should be added every time a SIP UA changes the ultimate endpoint, which will receive the call. There may be more than one Diversion header in a request. UA adds a Diversion header when features such as call forwarding change the Request-URI of a message. Diversion headers should not be added for normal call routing changes to the Request-URI.

A typical use of the header would be by a voicemail application server to identity the mailbox and greeting to use for a call forwarded to voicemail.

4.9.2 UAS/C Behavior

When a request is received, the SIP-URL in the Diversion header is compared to the Request-URI for call forwarding loop detection. If a loop is detected, the session invitation is terminated.

When a request is sent, the SIP-URLs in the Diversion headers for internal call forwards are compared to those in Diversion headers of external diversions. If there is a match, a call forwarding loop has been detected.

4.10 SIP History-Info header

The SIP History-Info header [19] is used when a call is forwarded to indicate the forwarding party and reason. Support is optional for all SIP UAs that provide call forwarding support, in particular Call Agents, Service Brokers and Application Servers. Support for the History header is not required when using SIP-T (SIP-I) as the diversion information is already present in the tunneled ISUP.

For example

INVITE sip:orig-party@appserver.com SIP/2.0

Via: SIP/2.0/UDP callagent.com;branch=z9hG4bK-metaswitch.com-1-166d14-4ff89cbb

Max-Forwards: 70

To: 12012002345@callagent.com

From: 12013001234@callagent.com;tag= callagent.com+1+24710+b8441f77

Call-ID: F5329DFF@callagent.com

CSeq: 427144613 INVITE

Contact: <sip:12013001234@callagent.com>

Expires: 180

Route: <sip: F5329DFF@callagent.com;lr>
P-Charging-Vector: icid-value=F5329DFF@callagent.com
History-Info: <sip: 12014003456@callagent.com >; index=1
Referred-By: 12014003456@callagent.com
Content-Type: application/sdp
Content-Length: 128

<SDP not shown>

4.10.1 History-Info Header

History-Info headers should be added every time a SIP UA changes the ultimate endpoint, which will receive the call. There may be more than one index for the History-Info header in a request. UA adds a History-Info header or index when features such as call forwarding change the Request-URI of a message. History-Info headers should not be added for normal call routing changes to the Request-URI.

A typical use of the header would be by a voicemail application server to identity the mailbox and greeting to use for a call forwarded to voicemail.

4.11 Registration

SIP User Agents must register with their configured Call Agent/SIP Server/SIP Registrar. Registration is unnecessary for other network components.

4.12 Authentication

Authentication is required for network components that support interfaces cross a trust boundary, in particular between SIP User Agents and their configured Call Agent/SIP Server. Detailed requirements are specified in the MSF SIP Signaling Security for GMI 2004 IA [MSF-2].

4.13 SIP Signaling Security

SIP signaling security, TLS and/or S/MIME is required for network components that support interfaces cross a trust boundary, in particular between Call Agents/SIP Servers in different network domains. Detailed requirements are specified by the MSF SIP Signaling Security for GMI 2004 IA [MSF-2].

4.14 Failure detection through no-response

According to RFC3261 if an unreliable protocol (such as UDP) is used as the transport for SIP then an INVITE must be retried a number of times before the destination is considered unreachable; the mechanism specified is to retry at intervals starting at T1 and doubling after that up to a maximum of 64*T1, where T1 is an estimate of RTT. The default value for T1, based on the behavior of the Internet is 500ms and results in six retry attempts at 0.5s, 1.5s, 3.5s, 7.5s, 15.5s, 31.5s with the destination being declared unreachable after 32 seconds. During this period the user will hear silence, which is significantly longer than equivalent timers in the PSTN today and is unacceptable for a PSTN equivalent service.

Therefore it is recommended that the number of retries be modified to limit retries to a maximum of 8*T1 and for T1 to be configurable, with a default of 250ms.

4.15 Symmetric Response Routing

SIP extensions for symmetric response routing for supporting SIP devices (phones or clients) behind a NAT devices are specified in RFC 3581 [9]. However this extension cannot resolve the NAT matter for the media path as well as through the signaling path. So the implementation of this extension is described in this IA is optional.

The SIP extensions for symmetric response routing add a new rport parameter to the Via header.

4.15.1 Via Header

The Via header includes two parameters, rport and received. The rport parameters is inserted in to the request's Via header by the SIP client (phone) to signify to the SIP server that the response to the request must be sent to the IP and port that the request is sent from and not the IP/port values in Contact or Via header. In a request, the client will not assign a value to the rport parameter. The receiving SIP server (or proxy) must complete the Via header by inserting received:<IP Address> and rport:<IP port> before forwards or responds to this request

Example:

1) A SIP Phone with private IP address of 12.11.10.9, locates behind a NAT "firewall", sends an INVITE request to a SIP server via port 5060. The INVITE sent by SIP phone looks in part like

INVITE SIP:4083302105@domain SIP/2.0

Via: SIP/2.0/UDP 12.11.10.9:5060;rport

...

2) However, after pass through the NAT, the INVITE actually is sent via public address and port (170.30.20.10:8899). The SIP server serving the SIP phone sends responses back to the public IP address and port, after complete the Via header

SIP/2.0 100 Trying

Via: SIP/2.0/UDP 12.11.10.9:5060;received=170.30.20.10;rport=8899

...

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP 12.11.10.9:5060;received=170.30.20.10;rport=8899

...

SIP/2.0 200 OK

Via: SIP/2.0/UDP 12.11.10.9:5060;received=170.30.20.10;rport=8899

...

5. Management Information Model

The Management Information Model will be specified in the individual SIP IAs.

6. References

Table 2: This SIP Profile uses the following IETF references

	IETF RFC Identification	Document Title	Mandatory/ Optional
1	RFC 3261	SIP: Session Initiation Protocol	Mandatory
2	RFC 2976	SIP Info Method	Optional
3	RFC 2327	SDP: Session Description Protocol	Mandatory
4	RFC 3264	An Offer/Answer Model with the Session Description Protocol (SDP)	Mandatory
5	RFC 4028	Session Timers in the Session Initiation Protocol (SIP)	Optional
6	RFC 3323	A Privacy Mechanism for the Session Initiation Protocol (SIP)	Mandatory
7	RFC 3262	Reliability of Provisional Response in SIP	Conditionally Mandatory
8	Draft-levy-sip-diversion-07	Diversion Indication in SIP	Deprecated (used in GMI 2004)
9	RFC 3581	An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing	Optional
10	RFC 3325	Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks	Conditionally Mandatory
11	RFC 3311	SIP UPDATE Method	Optional
12	RFC 3312	Resource Management QOS	Optional
13	RFC 3515	The Session Initiation Protocol (SIP) Refer Method	Conditionally Mandatory
14	RFC 3892	The SIP Referred-By Mechanism	Conditionally Mandatory
15	RFC 3326	The Reason Header Field for the Session Initiation Protocol (SIP)	Optional
16	RFC 3420	Internet Media Type message/sipfrag	Conditionally Mandatory
17	RFC 3265	Session Initiation Protocol (SIP)-Specific Event Notification	Optional
18	draft-ietf-sip-resource-priority-10	Resource priority	Conditionally Mandatory
19	draft-ietf-sip-history-info-06	An Extension to the Session Initiation Protocol for Request History Information	Optional
20	RFC2460	Internet Protocol, Version 6 (IPv6) Specification	Optional
21	RFC3513	Internet Protocol Version 6 (IPv6) Addressing Architecture	Optional

Table 3: This SIP Profile uses the following MSF references

	MSF Title	Document Title	Mandatory/ Optional
MSF-1	MSF-IA-SDP.001-FINAL (MSF2003-059)	SDP Usage & Codec Negotiation for GMI2004	Mandatory
MSF-2	MSF-IA-SIP.008-FINAL (MSF2003-116)	SIP Signaling Security for GMI 2004	Mandatory
MSF-3	MSF-AF-QOS.001-FINAL (MSF2003-105)	MSF QoS Solution Framework	Mandatory
MSF-4	MSF-IA-SIP.010-FINAL (msf2004.027)	SIP Call Agent to Bandwidth Manager	Conditionally Mandatory
MSF-5	MSF-IA-SIP.004-FINAL (msf2004.041)	SIP Call Agent to Call Agent	Conditionally Mandatory
MSF-6	MSF-IA-SIP.005-FINAL (MSF2003.064)	SIP Call Agent to Service Broker Interface	Conditionally Mandatory
MSF-7	MSF-IA-SIP.003-FINAL (msf2004.042)	SIP Call Agent to User Agent interface	Conditionally Mandatory
MSF-8	MSF-IA-SIP.009-FINAL (MSF2004.006)	SIP Media Server Interface	Conditionally Mandatory
MSF-9	MSF-IA-SIP.006-FINAL (MSF2003.063)	SIP Service Broker to Application Server	Conditionally Mandatory
MSF-10	MSF-IA-SIP.007-FINAL (MSF2003.065)	SIP Service Broker to Service Broker	Conditionally Mandatory
MSF-11	MSF-IA-SIP-T.001.02-FINAL (msf2004.030)	SIP-T Profile for Media Gateway Controller	Conditionally Mandatory
MSF-12	msf2004.001	Use case for SIP terminals with NAT supported by separate SBCs	Recommended

7. Call Flows

Call flows will be contained in the individual SIP IAs.

8. Redundant Call Agent/SIP Server

There are at least two commonly implemented mechanisms for providing a redundant Call Agent/SIP Server

- Seamless 1:1 redundancy
- Active-Standby redundancy mechanism

8.1 Seamless 1:1 redundancy

This mechanism for providing redundancy uses two Call Agents/SIP Server in a fault tolerant pair, one acting as a primary and the other as a backup. They share a single IP address as far as the remainder of the network is concerned. Typically they support replication of call state and configuration between the primary and backup and if the primary fails, the backup takes over seamlessly. The backup may require an auditing step to verify that all calls are still active, but this will simply be a reINVITE or UPDATE and is no different from the mechanisms described in section 4.7.

8.2 Active/Standby Mechanism

This alternate mechanism describes an implementation for the SIP server redundancy scheme, where there are two servers provide an Active-Standby system. In this schema, a SIP client must be provisioned with both SIP servers in a redundant system. The SIP client should register to both SIP servers in a redundant system. The Active server responds to all requests sent to it. The Standby server responds only to REGISTER request.

8.2.1 SIP Client Behavior

- If a SIP client should send its request to the last SIP server (Active server) that responds to its last request.
- If there is no response from this server, the client must resend the request to other server (Standby server) it has registered.
- If there are no responses from either server, the client may continue to resend the request to both servers, one-at-a-time, until it receives a response or it could abort the request. It is a matter of implementation.
- If a SIP client receives a request from the Standby server, it must respond to the request and must send its subsequence request the new server.

8.2.2 Active SIP Server Behavior

The behavior of the Active SIP server is very simple:

- It must respond to all requests and ACK to responses in a timely manner to avoid case where the client resends request to the Standby.
- It must handle client registrations.
- It must exchange the client registrations and call status with the Standby server.

8.2.3 Standby SIP Server Behavior

The behavior of the Standby SIP server is a bit more demanding

- It must poll the Active server to determine if the Active server is ‘alive’
- It must keep all active call status
- It must handle client registrations
- If it has determined the Active server is out of service (a matter of implementation, outside the scope of this contribution). It must re-INVITE all active calls to alerts the endpoints that the servers have been swapped. It then changes its state to Active.

