INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.178

(05/2003)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS
IPCablecom

IPCablecom CMS to CMS Signalling

CAUTION !

PREPUBLISHED RECOMMENDATION

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU [had/had not] received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

ITU-T Recommendation J.178
IPCablecom CMS to CMS Signalling
(Geneva, 2003)

Summary

This Recommendation describes the IPCablecom Call Management Server (CMS) to CMS Signalling protocol intended for use by a CMS to communicate with another CMS in order to support packet-based voice and other real-time multimedia applications. The protocol exchanges between a CMS and a Media Gateway Controller (MGC) are identical to those between CMSes, and so for purposes of this specification the MGC is considered identical to a CMS.

1 INTRODUCTION

1.1 Scope

This document describes the IPCablecom Call Management Server (CMS) to CMS Signalling protocol intended for use by a CMS to communicate with another CMS in order to support packet-based voice and other real-time multimedia applications. The protocol exchanges between a CMS and a Media Gateway Controller (MGC) are identical to those between CMSes, and so for purposes of this specification the MGC is considered identical to a CMS. CMSes currently support multimedia endpoints (within the IPCablecom infrastructure) that use the Network-based Call Signalling [5] (NCS) protocol and the PSTN Gateway Call Signalling Protocol [6] (TGCP) for communicating signalling information between the endpoint and the CMS. In the future, other protocols may be supported as well, and the CMS to CMS protocol is intended to be sufficiently general to accommodate such protocols without change.

The CMS to CMS protocol uses the Session Initiation Protocol 2.0 (SIP) specification with extensions and usage rules that support commonly available local and CLASS^{SM} services. This protocol is referred to as the Call Management Server Signalling (CMSS) protocol.

The CMSS protocol takes into account the need to manage access to network resources and account for resource usage. The usage rules defined in this specification specifically address the coordination between CMS Signalling and IPCablecom Dynamic Quality of Service (QoS) mechanisms for managing resources over the cable access network. In addition, this specification defines the protocols and messages needed between Call Management Servers for supporting these services.

This document specifies the protocols and procedures to use between CMSes belonging to a single service provider as well as between CMSes that belong to different service providers. In the case that the CMSes are owned by multiple service providers, it is assumed that the service providers have a mutual trust relationship.

Other IPCablecom documents describe interfaces between other system elements. These documents cover areas such as: Event Message recording for billing and other back office functions [3]; Dynamic Quality of Service [4]; Operations and Provisioning [26]; Electronic Surveillance [24]; and Security [2]. These other specifications indirectly place requirements on the signalling protocol to ensure that it transports the correct data needed to implement a complete system. This document includes syntax and protocols for implementing these requirements. Currently, the document does not address interworking with non-IPCablecom-compliant devices.

1.2 Specification Language

It is understood that implementing this recommendation is optional. If this Recommendation is implemented, the key words "MUST" and "SHALL" as well as "REQUIRED" are to be interpreted as indicating a mandatory aspect of this specification. The key words indicating a certain level of significance of particular requirements that are used throughout this Recommendation are summarized in the table below:

"MUST"	This word or the adjective "REQUIRED" means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.

“SHOULD”	This word or the adjective “RECOMMENDED” means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
“SHOULD NOT”	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
“MAY”	This word or the adjective “OPTIONAL” means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

The following ITU-T Recommendations and other references contain provisions, which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[1] ITU-T Recommendation J.160 - Architectural Framework for the delivery of time critical services over cable television networks using cable modems. (02/02)

[2] ITU-T Recommendation J.170 - IPCablecom Security Specification. (02/02)

[3] ITU-T Recommendation J.164 - Event Message requirements for the support of real-time services over cable television networks using cable modems. (03/01)

[4] ITU-T Recommendation J.163 - Dynamic quality of service for the provision of real-time services over cable television networks using cable modems. (03/01)

[5] ITU-T Recommendation J.162 - Network call signalling protocol for the delivery of time critical services over cable television networks using cable modems (03/01)

[6] ITU-T Recommendation J.171 - IPCablecom Trunking Gateway Control Protocol (TGCP). (02/02)

[7] ITU-T Recommendation J.174 - IPCablecom Interdomain Quality of Service. (02/02)

[8] IETF RFC 3261, Rosenberg, J., Schulzrinne, H. Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M, E. Schooler, SIP: Session Initiation Protocol, June 2002.

[9] ITU-T Recommendation E.123 - Notation for national and international telephone numbers, e-mail addresses and Web addresses (02/01)

[10] ITU-T Recommendation E.164 - The international public telecommunication numbering plan. (05/97)

[11] IETF RFC 2327 - Handley, M, V. Jacobson, SDP: Session Description Protocol, April 1998.

[12] IETF RFC 1890 - Schulzrinne, H, RTP Profile for Audio and Video Conferences with Minimal Control, January, 1996.

[13] IETF RFC 768 - Postal, J, User Datagram Protocol, August 1980.

[14] IETF RFC 3312 - Integration of Resource Management and SIP for IP Telephony, October, 2002.

[15] IETF RFC 3262 - Reliability of Provisional Responses in SIP, June, 2002.

[16] IETF RFC 3323 - J. Peterson, A Privacy Mechanism for the Session Initiation Protocol (SIP), November 2002.

[17] IETF RFC 3325 - C. Jennings, J. Peterson, M. Watson; Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks; November 2002.

[18] IETF RFC 3265 - Adam Roach, SIP-Specific Event Notification, June 2002.

[19] IETF RFC 3311 - J. Rosenberg, "The SIP UPDATE Method", October, 2002

[20] IETF RFC 3263 - J. Rosenberg, H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", June 2002.

[21] IETF RFC 2234 - Crocker, D. and P. Overell, “Augmented BNF for Syntax Specifications: ABNF”, November 1997.

[22] IETF RFC 2397 - L. Masinter, “The “data” URL Scheme”, August 1998.

[23] IETF RFC 2396 - T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", August 1998.

[24] ANSI/SCTE 24-13 2001 IPCablecom Part 13, Electronic Surveillance Standard

[25] IETF RFC 3420 - Robert Sparks, Internet Media Type message/sipfrag, November 2002.

2.2 Reference Acquisition

ITU-T Recommendations available at: http://www.itu.int/ITU-T

IETF RFCs available at: http://www.ietf.org/rfc/

SCTE Standards available at: www.scte.org

3 TERMS AND DEFINITIONS

This Recommendation defines the following terms:

Active	A service flow is said to be “active” when it is permitted to forward data packets. A service flow must first be admitted before it is active.
Admitted	A service flow is said to be “admitted” when the CMTS has reserved resources (e.g. bandwidth) for it on the DOCSIS network.
Announcement Server	An announcement server plays informational announcements in IPCablecom network. Announcements are needed for communications that do not complete and to provide enhanced information services to the user.
Authentication	The process of verifying the claimed identity of an entity to another entity.
Authorization	The act of giving access to a service or device if one has the permission to have the access.
CNAM	Calling Name
Encryption	A method used to translate information in plaintext into ciphertext.
Encryption Key	The key used in a cryptographic algorithm to translate the plaintext to ciphertext.
Endpoint	A Terminal, Gateway or MCU
Event Message	Message capturing a single portion of a connection
Gateway	Devices bridging between the IPCablecom IP Voice Communication world and the PSTN. Examples are the Media Gateway which provides the bearer circuit interfaces to the PSTN and transcodes the media stream, and the Signalling Gateway which sends and receives circuit switched network signalling tot he edge of the IPCablecom network.
Header	Protocol control information located at the beginning of a protocol data unit.
Key	A mathematical value input into the selected cryptographic algorithm.
LNP	Local Number Portability
Network Layer	Layer 3 in the Open System Interconnection (OSI) architecture that provides network information that is independent from the lower layers.
Off-Net Call	A communication connecting a IPCablecom subscriber out to a user on the PSTN
On-Net Call	A communication placed by one customer to another customer entirely on the IPCablecom Network
Privacy	A way to ensure that information is not disclosed to any one other then the intended parties. Information is usually encrypted to provide confidentiality. Also known as confidentiality.
Proxy	A facility that indirectly provides some service or acts as a representative in delivering information there by eliminating a host from having to support the services themselves.

Public Key	The key used in public key cryptography that belongs to an individual entity and is distributed publicly. Other entities use this key to encrypt data to be sent to the owner of the key.
Public Key Cryptography	A procedure that uses a pair of keys, a public key and a private key for encryption and decryption, also known as asymmetric algorithm. A user’s public key is publicly available for others to use to send a message to the owner of the key. A users private key is kept secret and is the only key which can decrypt messages sent encrypted by the users public key.
Trunk	An analog or digital connection from a circuit switch which carries user media content and may carry voice signalling (MF, R2, etc.).

4 ABBREVIATIONS AND ACRONYMS

This Recommendation defines the following abbreviations and acronyms.

BCID	Billing Correlation ID
BLV	Busy Line Verification
CA	Call Agent
CFNA	Call Forwarding No Answer
CLASS	Custom Local Area Signalling Services
CMTS	Cable Modem Termination System
CNAM	Calling Name
DCS	Distributed Call Signalling
EI	Emergency Interrupt
E.164	Telephone number standard of ITU
FQDN	Fully Qualified Domain Name
GC	Gate Controller
IP	Internet Protocol
LNP	Local Number Portability
LRN	Local Routing Number
MF	Multi-Frequency
MG	Media Gateway
MGC	Media Gateway Controller
MTA	Multimedia Terminal Adapter
NCS	Network Call Signalling
OSPS	Operator Services Positioning System
RFC	Request for Comments (IETF standard)
SDP	Session Description Protocol

5 BACKGROUND AND MOTIVATION

The material contained in this section and its subsections is informative. The design of the Call Management Server Signalling (CMSS) architecture recognizes the trend towards use of packet networks as the underlying framework for communications. These networks will provide a broad range of services, including traditional best-effort data service, as well as enhanced value-added services such as telephony and gaming. The Network based Call Signalling (NCS) and PSTN Gateway Call Signalling (TGCP) protocols are used to communicate between limited-function multimedia end-points, such as standard telephone sets and trunking gateways, and Call Management Servers (CMS). However, the NCS and TGCP protocols do not address the need for communication between multiple CMSes residing in one or more service providers' networks. This document covers the signalling performed between CMSes. The initial real-time multimedia service that is supported by the NCS and TGCP function is that of interactive telephony. The NCS and TGCP protocols represent the same architecture and are largely similar. In the following, where a distinction is not important, they will sometimes be simply referred to as NCS.

It is recognized that packet based networks may also offer additional real-time multimedia services to endpoints that are IP capable. Also, improvements in silicon will reinforce the trend towards increased functionality and “intelligence” in endpoints. These intelligent endpoints will be addressed in a future document to take advantage of the widespread availability of packet networks to enable a rich set of applications and services for users.

A key element of the CMSS architecture is a recognition of the need for coordination between call signalling, which controls access to telephony specific services, and resource management, which controls access to network-layer resources. This coordination is designed to meet the user expectations and human factors associated with telephony. For example, in a telephony environment, the called party should not be alerted until the resources necessary to complete the call are available. If resources were not available when the called party picked up, the user would experience a call defect. In addition, PSTN users expect to be charged for service only after the called party answers the phone. As a result, it must be possible to track usage accounting in a way that allows customer billing to start once the called party picks up. Coordination between call signalling and resource management is also needed to prevent fraud. The coordination between call signalling and Dynamic QoS [4] protocols ensures that users are authenticated and authorized before receiving access to the enhanced QoS associated with the telephony service.

In the NCS and TGCP protocols, the functionality required of the multimedia endpoint is simple, and more of the functionality resides in the network in call management servers, where the state of a session is maintained. The CMS is responsible for establishing and managing session legs, and for requesting and obtaining network layer QoS for the session. The NCS and TGCP protocols specify the information and message exchanges between the multimedia endpoint and the CMS. When the session has to be routed through multiple CMSes, additional functionality is required in the protocol to communicate the information related to the session. This includes information provided by the endpoint to the network as well as information that may reside in the CMS or other entities within the network that relates to the session. Examples of such additional information that may reside in the network include billing and data that may otherwise be kept private from untrusted multimedia endpoints.

5.1 Requirements and Design Principles

This section briefly describes the application requirements that led to the set of CMSS design principles.

The need to support primary line telephony service requires enhanced bearer channel and signalling performance, including:

- *Low delay* – end-to-end packet delay must be sufficiently small that it does not interfere with normal multimedia sessions. The ITU recommends no greater than 300 ms roundtrip delay for a telephony service.
- *Low packet loss* – packet loss must be sufficiently small that it does not perceptibly impede session quality or, in the case of telephony, performance of fax and voice band modems.
- *Short post-dial delay* – the delay between dialing the last digit and receiving positive confirmation from the network must be sufficiently short to ensure that users do not perceive a difference from post-dial delays typically experienced in the circuit switched network; in particular, the delay must not be so long that the user is led to believe that the network has failed.
- *Short post-pickup delay* – the delay between a user picking up a ringing phone or acknowledging a multimedia session and the voice or media path being cut through must be sufficiently short to ensure that the initial talk-spurt, *e.g.*, "hello", is not clipped.

A number of key design principles that arise from the requirements and philosophy above are identified:

- It is essential to provide network-layer Quality of Service while allowing the service provider to derive revenues from the use of such service.
- The CMSS architecture must allow for communication between CMSes in the network. At a high level, one may regard a CMS as performing complex signalling tasks on behalf of an endpoint. When the network includes multiple CMSes, CMSS should provide the call signalling function between the CMSes on an individual call basis. Within such a context, the CMSS architecture must allow the network to support limited-function multimedia endpoints, while allowing additional functions to be performed by the CMSes (including the maintaining call state in the CMSes).
- The architecture must ensure that the network is protected from fraud and theft of service. The service provider must be able to authenticate users requesting service and to ensure that only those authorized to receive a particular service are able to obtain it. Furthermore, the service provider must be able to track the usage of such services in order to support billing.
- The architecture must enable the service provider to add value by supporting the functions of a trusted intermediary. In the case of telephony, this includes protecting the privacy of calling and called party information, and ensuring the accuracy of the information that is provided in messages from the network.
- The architecture must be implementable, cost-effectively, at very large scale.

5.2 IPCablecom Architecture

The CMS to CMS Signalling (CMSS) Architecture follows the principles outlined above to support a robust multimedia service. Figure 1 introduces the key components in the architecture. In addition, the definitions of all IPCablecom elements (Domain, Realm, etc.) are provided in the IPCablecom Interdomain Architecture Technical Report [1].

Multimedia Terminal Adapters (MTAs) may either be embedded into the Cable Modem (CM) or they may be stand-alone. The cable access network interfaces to an IP backbone through a CMTS that is the first trusted element within the provider's network. The CMTS performs network resource management, acts as a policy enforcement point, and as a source of event messages that can be used for billing.

The CMS establishes and receives sessions on behalf of an endpoint by using the NCS protocol to communicate with the MTA. The CMS uses the protocol specified here to communicate with other CMSes. In addition, it may also perform the function of a Gate Controller (GC), which is responsible for authorizing the enhanced Quality of Service for the media stream. The CMS acts as a source of event messages that can be used for billing.

Media Service Nodes represent network-based components that operate on media flows to support the service. Media service nodes perform audio bridging, play announcements, provide interactive voice response services, etc. The protocol exchanges between a CMS and a Media Service Nodes are identical to those between CMSes, and so for purposes of this document a Media Service Node is considered identical to a CMS. Media Service Nodes may be decomposed into a controller and a player, in which case CMSS signalling is performed with the controller.

PSTN gateways interface to the Public Switched Telephone Network. The PSTN gateway may be decomposed into a Media Gateway Controller (MGC), a signalling gateway (SG), and a Media Gateway (MG). The TGCP protocol is used between the MGC and the MG. The protocol exchanges between a CMS and an MGC are identical to those between CMSes, and so for purposes of this document, the MGC is considered identical to a CMS.

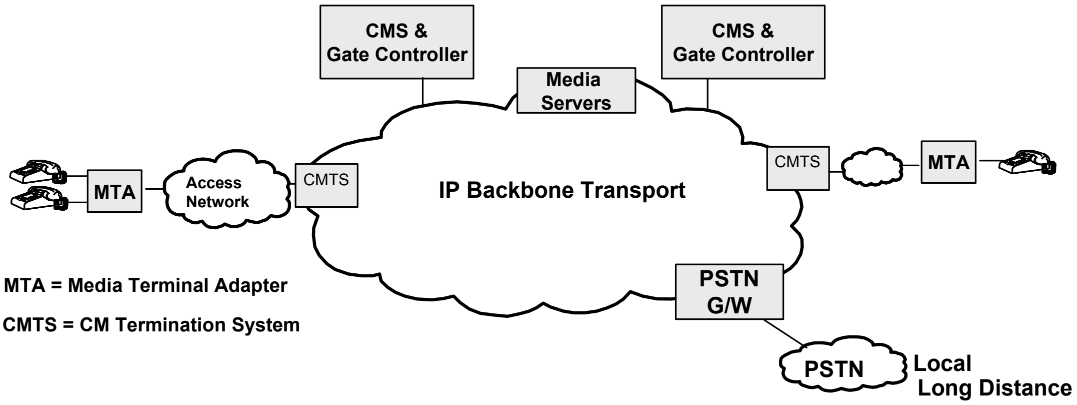

Figure 1: System Architecture – Realm

Access to network resources must be controlled by the service provider. The CMTS receives resource management requests from endpoints, and is responsible for ensuring that packets are provided the QoS they are authorized to receive (either through packet marking, or through routing and queuing the packets as a specific QoS-assured flow). The CMTS requires authorization from a Gate Controller (on a session by session basis for the multimedia service) before providing access to enhanced QoS for an end-to-end IP flow. Thus, the CMTS is able to ensure that enhanced QoS is provided only for end-to-end flows that have been authorized and for which usage accounting is being performed. Since the CMTS knows about the resource usage associated with individual IP flows, it generates the usage events that allow a user to be charged for service [3].

DQoS [4] introduces the concept of a "gate" in the CMTS. Conceptually, gates manage access to enhanced quality of service. The gate is a packet classifier and policer that ensures that only those IP flows that have been authorized by the CMS are granted access to enhanced QoS in the access and backbone networks. Gates are "admitted" selectively for a flow. For a multimedia service, gates are opened and controlled for individual sessions. Admitting a gate involves an admission control check that is performed when a resource reservation or commit request is received from the endpoint, and it may involve resource reservation in the backbone network if necessary. The packet filter in the gate allows a flow of packets to receive enhanced QoS for a session from a specific IP source address and port number to a specific IP destination address and port number.

CMSes implement a set of service-specific control functions required to support the telephony service:

- Authentication and authorization: Since services are only provided to authorized subscribers, CMSes authenticate signalling messages and authorize requests for service on a session-by-session basis.
- Name/number translation and call routing: CMSes translate dialed numbers or names to a terminating IP address based on call routing logic.
- Service-specific admission control: CMSes can implement a broad range of admission control policies for the telephony service. For example, CMSes may provide precedence to particular calls, *e.g.*, emergency calls initiated by dialing a special number such as 911. Admission control may also be used to implement overload control mechanisms, *e.g.*, to restrict the number of calls to a particular location or to restrict the frequency of call setup to avoid signalling overload.
- Signalling and service feature support: CMSes maintain and track all signalling activities to insure compliance and to manage subscriber features. For example, in the case of telephony, 3-way calling, caller ID, etc.

A CMS is responsible for a set of endpoints and the associated CMTSes. While endpoints are not trusted, there is a trust relationship between the CMTS and its associated CMSes, since the Gate Controllers in the CMSes play a role as a policy server that controls when the CMTS can provide enhanced QoS service. There is also a trust relationship among CMSes. Details of the security model and mechanisms are specified in [2].

CMSS supports inter-working with the circuit switched telephone network through PSTN gateways. A PSTN gateway may be realized as a combination of a Media Gateway Controller (MGC), Media Gateway (MG), and a Signalling Gateway (SG). A media gateway acts as the IP peer of an endpoint for media packets, converting between the data format used over the IP network and the format required for transmission over the PSTN, *e.g.*, PCMU. The signalling gateway acts as the IP peer of a PSTN endpoint for call signalling, providing signalling inter-working between the IPCablecom network and conventional telephony signalling protocols such as ISUP/SS7. The MGC uses the PSTN Gateway Call Signalling Protocol (TGCP) to control the operation of the media gateway.

There are additional system elements that may be involved in providing the multimedia service [1]. For example, in the case of telephony service, the CMS may interface with other servers that implement the authorization or translation functions. Similarly, announcements, voicemail, and three-way calling may be supported using media service nodes in the network. Management of security interfaces between system elements is explained in [2].

This document provides generic capabilities that can be used to implement additional features. Features that have an intra/inter-domain (CMS-CMS) impact are considered and specifically addressed below.

5.3 CMSS Trust Model

CMSS defines a trust boundary around the various systems and servers that are within a single domain. These trusted systems include the Internal and External Border Proxies[1], CMSes, CMTSes of the cable access network, and various servers such as bridge servers, voicemail servers, announcement servers, etc. Outside of the trust boundary are the customer premises equipment, *i.e.*, the MTAs, the Public Switched Telephone Network (PSTN)[2], and various media service nodes operated by third-party service providers. At the boundary of the trusted domain are CMTSes/Edge Routers at the transport level and EBP/CMSes at the signalling level. The EBP interfaces to other IPCablecom domains. Although these other IPCablecom domains are outside the trust boundary, CMSS still trusts call signalling sent to and received from these other IPCablecom domains.

[1] More generally referred to as tandem proxies or simply proxies.

[2] but not the PSTN GW.

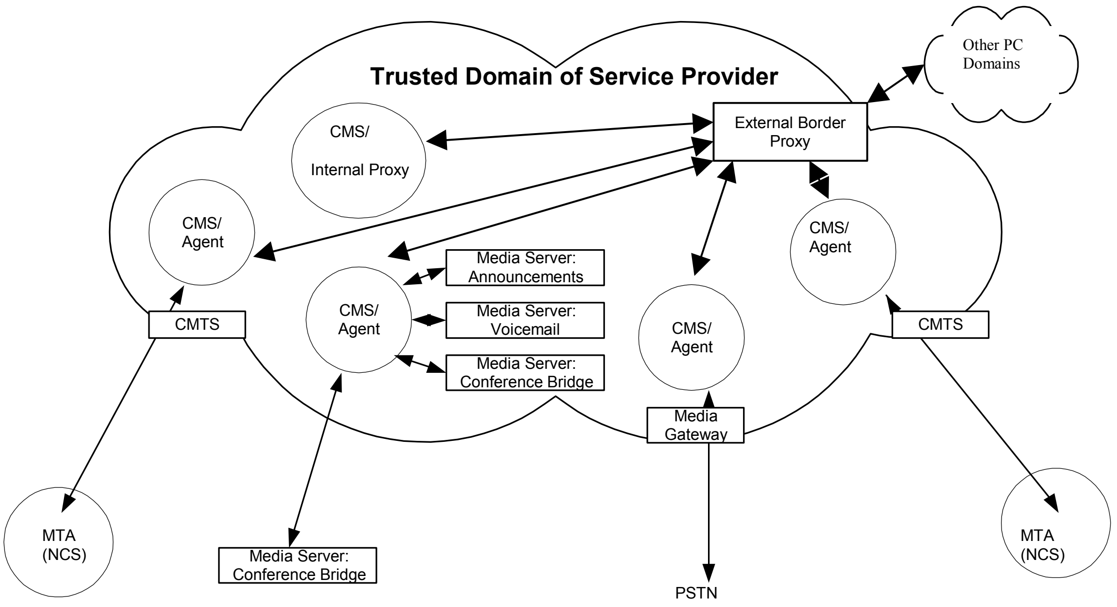

Figure 2: Trusted Domain of IPCablecom Service Provider

5.4 CMS to CMS Architectural Model

The Call Management Server (CMS) is an architectural entity that performs those services necessary to enable endpoints to establish IP multimedia sessions. The CMS is a complex of server functions that support session signalling, number translation, and feature support. In addition to processing signalling messages, the CMS provides functions for service and feature authorization, call routing, and service-specific admission control. As a trusted decision point, the CMS may also coordinate with Gate Controllers (which act as Policy Decision Points from a resource management point of view) to control when resource reservations are authorized for particular users and media flows.

This document describes the messages required to support IP Telephony between entities that support one or more of the roles indicated in the following table:

Role	Distinguishing function
Call Agent (CA) or CMS as defined above.	Support of endpoints implementing Network-Based Call Signalling (NCS) [5].
Media Gateway Controller (MGC)	Interworking with the PSTN. Use of PSTN Gateway Call Signalling (TGCP) [6] to control trunks.
Announcement Servers, Bridge Servers, or VoiceMail Servers	Provide various media services.
Tandem server within a domain, or a gateway server between service provider domains.	Routing functions only.

The list of roles may be expanded in the future. Although trust level varies between providers, the document assumes CMSes within a REALM and across multiple domains trust each other. MTAs, however, are untrusted NCS endpoints. Note: where multiple roles are combined within a single node, the interface between them is hidden and untestable.

All of the various types of endpoint management systems currently fall into one of two different categories of CMSes. A CMS[3] is a trusted entity that establishes calls on behalf of an untrusted endpoint, *e.g.*, an MTA, in the customer premise. The role of the CMS is to verify the signalling messages from the untrusted source, and provide various network services, such as translation, authentication and accounting. The second category is the Proxy. Proxies are classified into two types: proxies used within a domain, and proxies used between domains. The Interior Border Proxy (IBP) [36] is a proxy that can be used for inter-realm (intra-domain) signalling and the Exterior Border Proxy (EBP) [36] is required for inter-domain signalling. See the IPCablecom Inter-Domain Architecture Document [36] for further details. Where a distinction between the different types of proxy is either unimportant or is evident from the text, they will be simply referred to as proxies or tandem proxies.

[3] CMS/MGC and other types of centralized control CMSes fall within this category as well.

A CMS is a SIP User Agent (UA). If the CMS controls NCS endpoints, the CMS furthermore contains a Gate Controller (GC) with a hidden and untestable interface between the UA and the GC as shown in Figure 3. The Tandem Proxy, EBP and IBP entities are SIP proxies. Specification of a GC-to-UA interface is left as for further study.

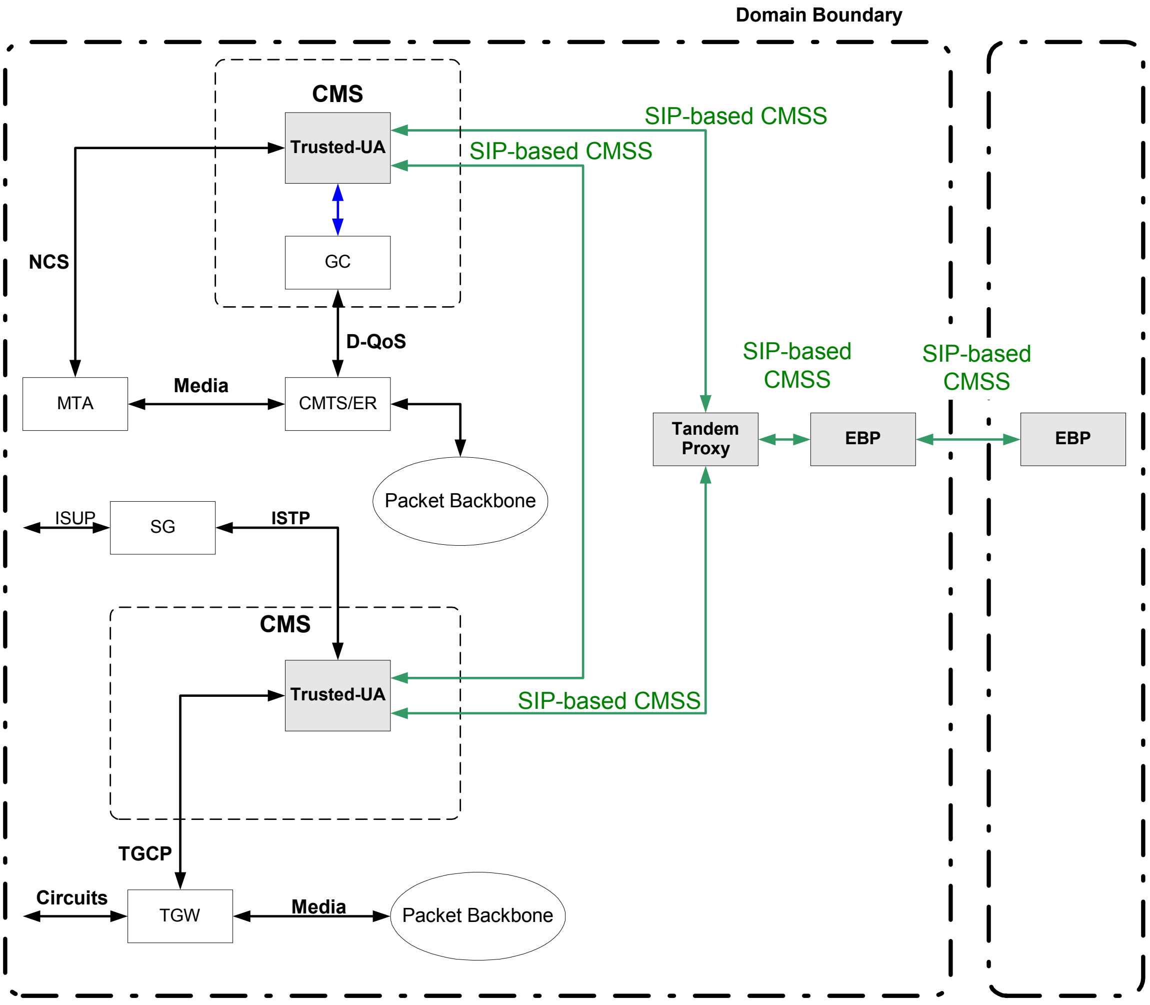

Figure 3: CMS Signalling Model

A CMS establishes connections either on its own behalf or on behalf of a non-SIP endpoint. Examples of the former are voicemail and conference bridge servers; in these cases the CMS is a trusted network entity that establishes connections on its own behalf. Examples of the latter are the Call Agent (CA) described in NCS [5], the Media-Gateway-Controller (MGC) described in [6], and the Announcement Controller (ANC) described in [28] in all of these cases there is centralized call control, protocol messages, *e.g.*, NCS, are exchanged between the CMS and the endpoint device, and the endpoint device does not participate in the SIP signalling exchanges directly.

The term CMS in this document refers to either of the above categories. Where only one category is being described, the term proxy[4] or CMS will be used as a representative for the category being described.

Unless otherwise stated in this document, a CMS MUST follow the requirements given for SIP user agents in [8] and a proxy MUST follow the requirements given for SIP proxies and redirect servers in [8].

Tandem proxies act as call routers and security association aggregation points. They may also provide additional functions such as signalling transformation gateways, signalling firewalls, etc. Depending on its role, a tandem proxy may remain in the call-signalling path for the duration of the call. More detailed tandem proxy information can be found in section 8.1.2.

[4] This may be further refined, e.g. into IBP and EBP. The term tandem proxy may also be used instead of just proxy.

5.5 Overview of CMS Behavior

IPCablecom defines the Call Management Server (CMS) as a complex of server functions which support call signalling, number translation, call routing, feature support and admission control. Within the CMS complex, the IPCablecom architecture allocates many of these responsibilities to the Proxy/CA/MGC and the Gate Controller (GC) function. In addition to processing session-signalling messages, a CMS provides functions for service and feature authorization, name/number translation, session routing, and service-specific admission control. As a trusted decision point, the CMS may also coordinate with Gate Controllers (which act as Policy Decision Points from a resource management point of view) to control when resource reservations are authorized for particular users. While the CMS is responsible for session control functions associated with proxying signalling requests, the Gate Controller is responsible for the policy decision regarding whether a requested QoS level should be admitted. Upon receipt of signalling information, a CMS instructs the Gate Controller to authorize a QoS level in advance of any resource reservation signalling (see [4] for more details).

The CMS associated with the endpoint originating a call is referred to as the originating CMS and is denoted by CMS_O. The CMS associated with the terminating endpoint is referred to as the terminating CMS and is denoted by CMS_T. The Gate Controllers (GC_O, GC_T) are the trusted policy decision points for controlling when and which resources are allowed to be reserved by endpoints; they coordinate with the CMTSes ($CMTS_O$, $CMTS_T$) through DQoS signalling. The CMTSes are the policy enforcement points, and ensure that the media path is provided the QoS it is authorized to receive.

The IPCablecom CMS-CMS architecture extends the use of the basic INVITE/200-OK/ACK SIP transaction. A provisional response, the 183-Session-Progress, and its acknowledgement, the PRACK/200-OK, are used with the initial INVITE to exchange capabilities and establish session state in the network prior to alerting the user. Following this exchange, the endpoints engage in resource reservations to obtain the resources they will need for the media streams. If the resource reservations are successful, the originating CMS performs an UPDATE/200-OK exchange. At this point the initial INVITE continues with a 180-Ringing or 183-Session-Progress, than a PRACK/200-OK followed by the final 200-OK response and ACK to the initial INVITE. In all cases, all provisional and final responses to an INVITE message traverse the path taken by the original INVITE through one or more CMSes and proxies; other messages, however, pass end-to-end directly between the originating CMS and the terminating CMS.

In support of billing functions, the originating CMS (CMS_O) includes information containing the account number of the caller and the Billing-Correlation-ID in the INVITE message that it sends.

Operator services such as Busy Line Verification (INVITE(BLV)) and Emergency Interrupt (INVITE(EI)) are initiated from a Media-Gateway-Controller type of CMS and sent to the number being verified/interrupted.

In call sequences associated with three-way calling, inter-domain call transfer, and inter-domain call forwarding, the CMS performs redirection. One possibility is that the CMS simply passes revised SDP to the other CMS, causing a redirection of media flow without changing the call topology. Alternatively, the CMS makes use of the REFER method to redirect the entire call. The REFER causes the receiver to initiate a new call using the information provided.

5.6 Basic Telephony Call Flow

Figure 4 presents a high-level overview of an example basic call that uses the CMSS protocol between the CMSes through a proxy while the end-points (MTAs) are using the NCS protocol.

In this example[5], when the MTA goes off-hook and the user dials a telephone number, the originating MTA (MTA_O) collects the dialed digits and exchanges initial NCS messages with the originating CMS (CMS_O) to notify it of the dialed digits and create a (media) connection. As a result of creating the connection, MTA_O returns a session description using the Session Description Protocol (SDP), which will subsequently be passed to CMS_T. When CMS_O wishes to ensure that adequate resources are available in the network before users who wish to communicate are alerted, it includes additional information in the SDP. This additional information is a “Pre-Condition” (possibly a set of pre-conditions including both security and resource requirements) that needs to be satisfied before the terminating user is alerted. CMS_O verifies that MTA_O is a valid subscriber of the telephony service and determines that this subscriber is authorized to place this call. CMS_O then translates the dialed number into the address of a terminating CMS (CMS_T) and sends the (1) INVITE message to it containing the SDP with the added pre-conditions.

It is assumed that the originating and terminating CMSes trust each other. CMS_O includes additional information, such as billing data containing the telephone number of the caller, in the INVITE message that it sends to CMS_T via the proxy. CMS_T then translates the dialed number into the address of the terminating MTA (MTA_T) and exchanges NCS

[5] Call flows throughout this document are examples only; IPCablecom does not mandate particular call flows.

signalling with MTA_T to create a (media) connection for the terminating endpoint. As part of the task of creating the connection, MTA_T selects the encoding and bandwidth requirements for the media streams and returns to CMS_T a session description containing a subset of the capabilities that were present in the NCS Create Connection request that are acceptable to MTA_T. CMS_T sends a GATE-SET message to the terminating CMTS ($CMTS_T$); this GATE-SET message conveys policy instructions allowing $CMTS_T$ to create a gate for the IP flow associated with this phone call subsequent to the admission control that is performed following a resource reservation request. CMS_T may send information in the GATE-SET message to notify $CMTS_T$ of billing-related information such as the IP address of the terminating RKS, the Billing Correlation ID (BCID) (see [3] for details) of the terminating event message stream, etc. (see [4] for further details). CMS_T then sends the (2) 183- Session-Progress response back to CMS_O via the proxy. Included in the 183-Session-Progress response is the SDP from MTA_T, with an indication added by CMS_T that the terminating side agrees to meet the preconditions specified in the INVITE before alerting the user. CMS_O then sends a GATE-SET message to the originating CMTS ($CMTS_O$) to indicate that it can admit a gate for the IP flow associated with the phone call. CMS_O then sends an NCS ModifyConnection request to MTA_O, enabling MTA_O to start reserving resources.

The initial INVITE request and the 183-Session-Progress response contain a SIP Contact header to indicate the IP address of the remote CMS to be used for subsequent end-to-end SIP signalling exchanges as well as the BCID and the Financial Entity ID (FEID) of the CMS sending the message. CMS_O acknowledges the 183-Session-Progress directly to CMS_T using the (3) Provisional Reliable Ack (PRACK) message. The terminating CMS_T acknowledges the PRACK message with the (4) 200-OK message. At this point, resource reservation has not yet completed, and thus the preconditions have not yet been met. CMS_T now issues a modify connection command to MTA_T instructing it to reserve network resources.

Once MTA_O has successfully completed its resource reservation thereby meeting its precondition, it sends an NCS signalling message to CMS_O which in turn sends the (5) UPDATE message directly to CMS_T. CMS_T acknowledges the UPDATE with the (6) 200-OK. When MTA_T has reserved its resources it exchanges NCS signalling with CMS_T. At this point in time, all preconditions have been met, and CMS_T can exchange NCS signalling with MTA_T instructing it to alert the user (ring the destination telephone). CMS_T then sends the (7) 180-Ringing message to CMS_O via the proxy indicating that the terminating phone is ringing, and that the calling party should be given a ringback call progress tone. CMS_O exchanges NCS signalling with MTA_O instructing it to provide ringback, and CMS_O sends another (8) Provisional ACK (PRACK) directly to CMS_T to acknowledge receipt of the (7) 180-Ringing message.

The terminating CMS_T acknowledges the PRACK with a (9) 200-OK. When the called party answers, by going off-hook, MTA_T exchanges NCS signalling with CMS_T to notify the off-hook and enable a full duplex connection. CMS_T also sends a (10) 200-OK final response to the (1) INVITE to CMS_O via the proxy. CMS_O acknowledges the 200-OK directly with the (11) ACK and exchanges NCS signalling with MTA_O instructing it to stop local ringback and enable a full duplex connection. At this point the resources that were previously reserved are committed, and the call is “cut through”

Either party can terminate the call. When CMS_O receives an on-hook notification from MTA_O, CMS_O sends a (12) BYE message directly to CMS_T, which is acknowledged with (13) 200-OK. Each CMS exchanges NCS signalling with its MTA to delete the connection and release the resources reserved.

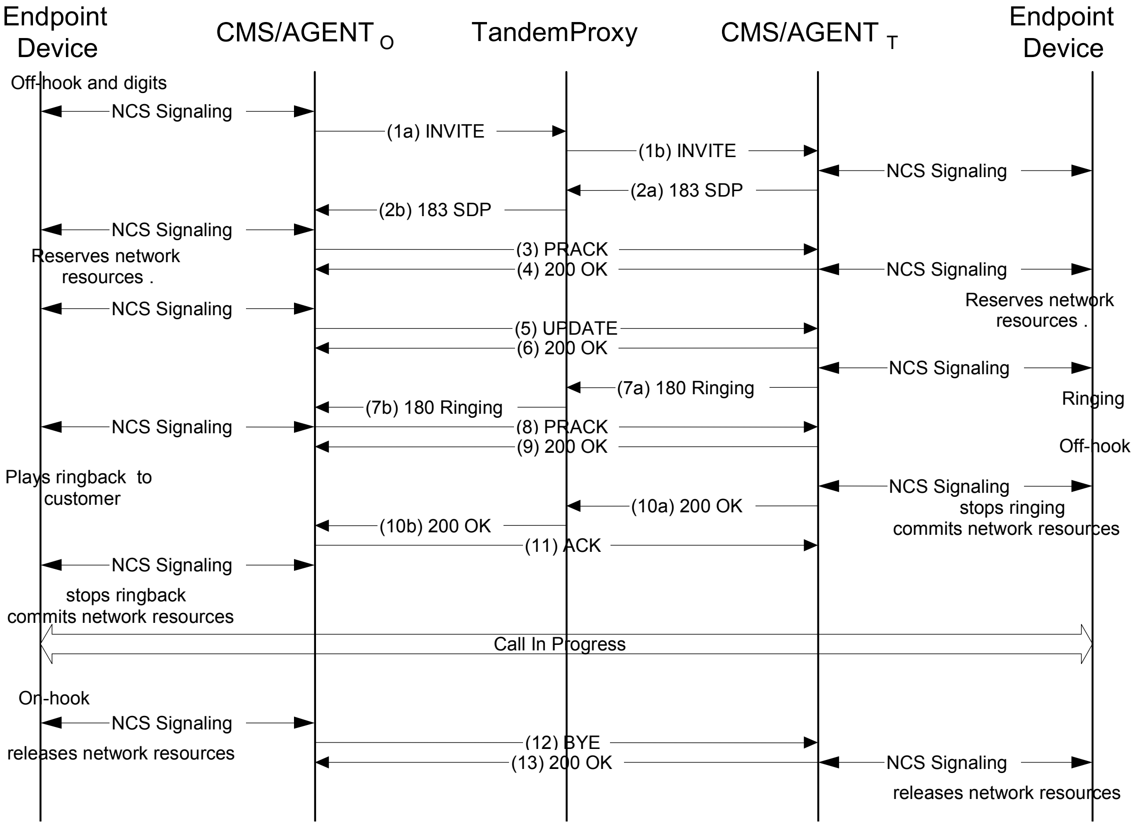

Figure 4: CMS – CMS Signalling Basic Call Flow

5.7 CMS-MGC Basic Telephony Call Flow

This section presents a high level overview of a basic call that uses the CMSS protocol between a CMS and an MGC to make an on-net to off-net call. The procedure shown follows the SIP to ISUP mapping recommendations provided in [35]. Note that CMSS in general is intended to be compatible with the interworking recommendations provided in [35].

The following procedure may be used to make a basic on-net to off-net call using CMSS as the protocol between a CMS and a MGC.

1. The subscriber goes off hook.
2. The MTA exchanges NCS signalling message with the CMS to notify the CMS that the subscriber went off hook.
3. The CMS instructs the MTA to start sending dial tone to the user by exchanging NCS signalling messages with the MTA.
4. The subscriber dials a valid telephone number.
5. The MTA collects the dialed digits and exchanges NCS messages with the CMS to notify the CMS of the dialed digits.
6. The CMS exchanges NCS messages with the MTA to create a (media) connection. As a result of creating the connection, the MTA returns a session description using the Session Description Protocol (SDP).
7. The CMS sends the MGC a (1) INVITE message containing the SDP.
8. The MGC exchanges TGCP signalling messages with the MG to create a (media) connection. During this exchange, session description information is also exchanged.
9. The MGC then sends a (2) 183-Session-Progress response back to the CMS with the SDP from the MG.
10. The CMS exchanges DQoS messages with the CMTS and NCS messages with the MTA so that the MTA will start reserving resources.
11. The CMS acknowledges the 183-Session-Progress using the (3) Provisional Reliable Ack (PRACK) message.
12. The MGC acknowledges the PRACK message with a (4) 200-OK message.

13. Once the MTA has successfully completed its resource reservation, the MTA exchanges NCS signalling messages with the CMS.
14. The CMS sends the (5) UPDATE message to the MGC.
15. The MGC sends an IAM message to the SG.
16. The MGC acknowledges the UPDATE message with a (6) 200-OK.
17. The MGC receives an ACM message from the SG.
18. If the ACM indicates, that the called party is being alerted, the MGC sends a (7) 180-Ringing message to the CMS. If the ACM instead indicated progress or in-band information available, the MGC would have sent a 183-Session-Progress instead (see [35] for details). In this latter case, the MGC would also exchange TGCP signalling with the MG instructing the MG to send packets to the MTA so in-band media provided by the PSTN can be heard by the subscriber.
19. The CMS exchanges NCS signalling with the MTA instructing it to play ringback to the subscriber.
20. The CMS sends another (8) Provisional ACK (PRACK) to the MGC to acknowledge the receipt of the 18x message.
21. The MGC acknowledges the PRACK with a (9) 200-OK.
22. The MGC receives an ANM message from the SG.
23. The MGC exchanges TGCP signalling with the MG instructing it to enable a full duplex connection.
24. The MGC sends a (10) 200-OK final response to the initial INVITE from the CMS.
25. The CMS exchanges NCS signalling with the MTA instructing it to enable a full duplex connection.
26. The CMS acknowledges the 200-OK with an (11) ACK message, the call is “cut through”.
27. The subscriber goes on hook.
28. The MTA exchanges NCS signalling with the CMS to notify the CMS of the on hook condition.
29. The CMS exchanges NCS signalling with the MTA instructing it to delete the connection and release resources.
30. The CMS sends the MGC a (12) BYE message.
31. The MGC sends the CMS a (13) 200-OK to acknowledge the BYE message.
32. The MGC sends a REL message to the SG.
33. The SG sends a RLC message to the MGC.
34. The MGC exchanges TGCP messages with the MG instructing the MG to delete the connection.

Figure 5 shows a basic on-net to off-net call flow, using CMSS as the protocol between the CMS and the MGC.

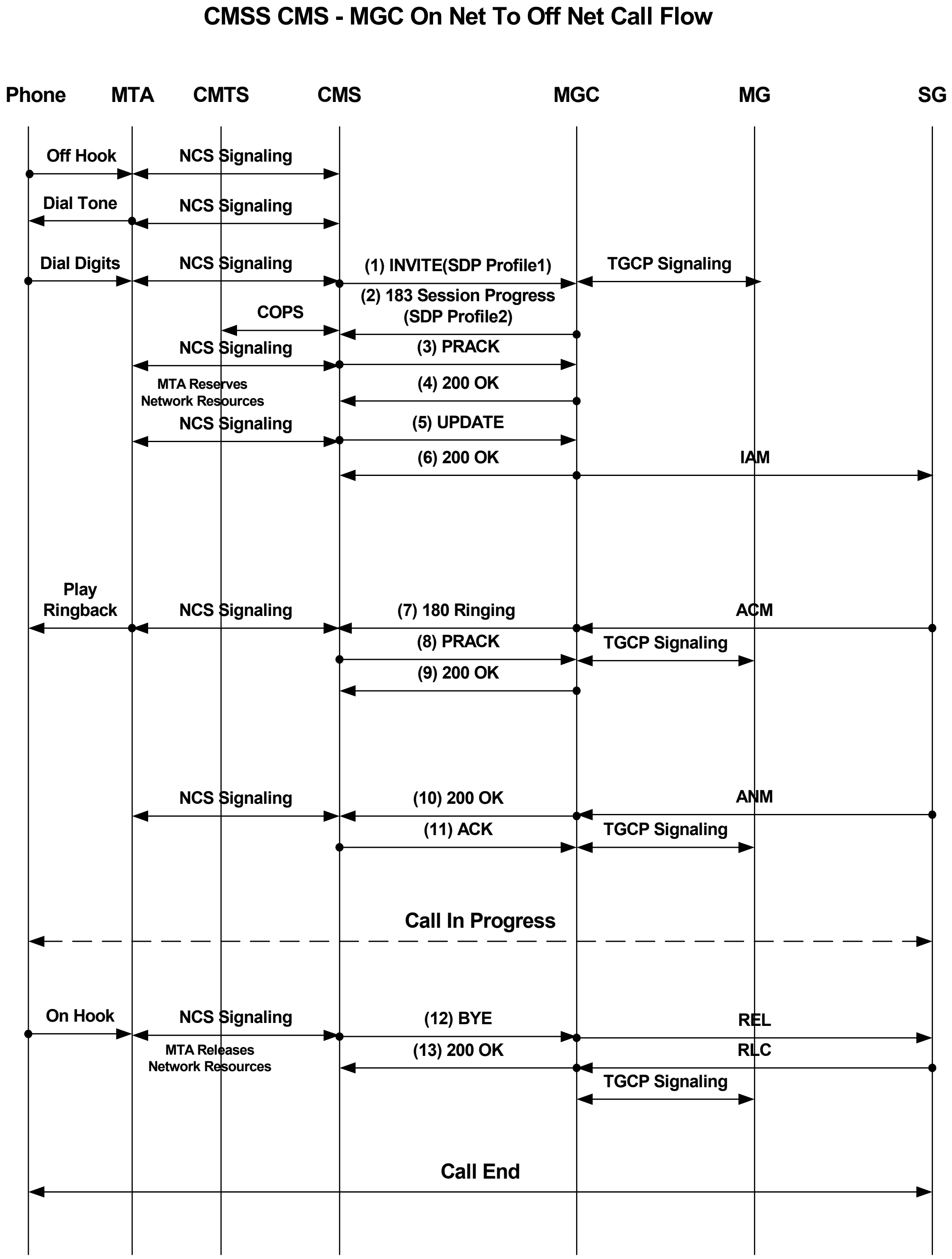

Figure 5: CMS to MGC Signalling

6 SIP PROFILE

This section defines a SIP [8] profile for usage in CMSS compliant systems. This section is structured to mirror the SIP document and its section numbering. The subsections of this section are numbered such that the second digit tracks the SIP section numbers of the SIP specification [8], and section titles at all header levels track the section titles of the SIP specification [8].

This section and section 7 define the nearly complete set of enhancements and restrictions to a standard SIP implementation based on [8]. However, not all details of the required behavior can be captured in these sections. Later sections provide details needed for certification and interoperability testing, which are generally not present in [8]. Sections 6 through 9 are considered normative. Informative call flow examples for the use of CMSS are provided in [26].

Unless otherwise stated in this document, a CMS MUST follow the requirements given for SIP user agents in [8] and a proxy MUST follow the requirements given for SIP proxies and redirect servers in [8].

6.1 Introduction

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 1.

Since there are no requirements in SIP/2.0 [8] section 1, CMSS implementations are automatically compliant with it.

6.2 Overview of SIP Functionality

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 2.

Since there are no requirements in SIP/2.0 [8] section 2, CMSS implementations are automatically compliant with it.

6.3 Terminology

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 3.

Since there are no requirements in SIP/2.0 [8] section 3, CMSS implementations are automatically compliant with it.

6.4 Overview of Operation

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 4.

Since there are no requirements in SIP/2.0 [8] section 4, CMSS implementations are automatically compliant with it.

6.5 Structure of the Protocol

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 5.

Since there are no requirements in SIP/2.0 [8] section 5, CMSS implementations are automatically compliant with it.

6.6 Definitions

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 6.

The reader should note that the term "client" in this section covers both UAs and proxies.

Since there are no requirements in SIP/2.0 [8] section 6, CMSS implementations are automatically compliant with it.

6.7 SIP Messages

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 7, except as noted below.

6.7.1 Requests

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 7.1, except as defined in this section.

The INVITE, ACK, CANCEL, BYE, and OPTIONS methods MUST be supported. The REGISTER method MAY be supported.

The Request-URI MUST be a SIP URI, as defined in [8], or a tel URI as defined in Section 7.1. The SIPS URI format MAY be supported.

The Request-URI in an initial INVITE for a basic telephone call[6] MUST identify the called party using a tel URI or by using the telephone-subscriber syntax (*i.e.*, the dialed phone number) in a SIP URI. Refer to Section 8.3 for details on forming the associated Request-URI. When with the Request-URI is a SIP URI, the host part MUST identify the CMS or entity to which the message is addressed.

The Request-URI for other requests associated with a basic telephone call MUST identify the targeted host using IPv4address or FQDN syntax, as given by the Contact header.

The host part of the Request-URI typically agrees with one of the host names of the receiving server. However, if the Request-URI of a received INVITE does not so agree, the server SHOULD proxy the request to another entity based on saved translation information or pre-provisioned policy information.

6.7.2 Responses

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 7.2.

6.7.3 Header Fields

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 7.3 and its subsections.

Furthermore, CMSS compliant applications MUST be able to both generate and accept short and long form header field names as defined in [8] section 7.3.3.

6.7.4 Bodies

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 7.4 and its subsections except as defined in this section.

6.7.4.1 Message Body Types

CMSS compliant applications MUST support the message body type “application/sdp”.

The message body type "application/sdp" MUST be supported with the INVITE, UPDATE, and PRACK methods as well as any non-failure response to these methods.

Refer to Appendix II for a complete list of supported values.

6.7.4.2 Message Body Length

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 7.4.2.

6.7.5 Framing SIP Messages

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 7.5.

6.8 General User Agent Behavior

Behavior of CMSS User Agents (UA) MUST be in accordance with chapter 8 of this document and with [8] except as noted in this section.

Note that the behavior defined in this section applies only to requests and responses outside of a dialog. Behavior within a dialog is defined in 6.12.

6.8.1 UAC Behavior

Support for the REGISTER method and SIPS URIs is OPTIONAL, however if supported, it MUST be as specified in [8] section 8.1.

Support for multiple simultaneous media streams for a single call is OPTIONAL for an MTA. Consequently, UACs MUST NOT accept multiple dialogs for a given request, if such dialogs result in the establishment of multiple

[6] This includes INVITEs generated as a result of forwarding.

simultaneous media streams, unless the MTA for which the dialog is being established supports multiple simultaneous media streams.

6.8.1.1 Generating the Request

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 8.1.1, except as noted below.

Request-URI in the request contains the address of the callee. This will normally be a telephone-number, but it may also be a general SIP-URI[7]. The From and To fields in the request might contain random strings that protect the Privacy of the session originator.

Refer to section 6.20 for further details of various header field values to be used.

The IETF allows option tags to be defined for their purpose only in standard-track RFCs. In addition to the standards-track RFCs' option tags, option tags from non-IETF documents MAY also be used, as long as they are defined in this document.

6.8.1.2 Sending the Request

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 8.1.2 and its subsections.

6.8.1.3 Processing Responses

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 8.1.3 except as noted in this section.

When receiving a 401 (Unauthorized) or 407 (Proxy Authentication Required) response, it is OPTIONAL to follow the SIP authorization procedures.

When receiving a 420 (Bad Extension) response, it is OPTIONAL to follow the SIP retry procedures.

6.8.2 UAS Behavior

The CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 8.2 and its subsections.

6.8.3 Redirect Servers

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 8.3.

6.9 Canceling a Request

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 9 and its subsections.

6.10 Registrations

Proxies MUST, and UACs MAY, support the SIP REGISTER method in accordance with [8] section 10. Support for registrars is OPTIONAL; however if supported, it MUST be as specified in [8] Section 10.

6.11 Querying for Capabilities

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 11 and its subsections.

6.12 Dialogs

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 12 and its subsections, except as noted in the following subsections.

[7] This can, for example, be used when forwarding to a an Interactive Voice Response (IVR) system.

6.12.1 Creation of a Dialog

Support for SIPS URIs is OPTIONAL; however if supported, it MUST be as specified in [8] section 12.1 and its subsections. The IPCablecom security requirements [2] MUST still be honored[8].

6.12.2 Requests within a Dialog

Support for SIPS URIs is OPTIONAL, however if supported, it MUST be as specified in [8] section 12.2 and its subsections. The IPCablecom security requirements [2] MUST still be honored[9].

6.12.3 Termination of a Dialog

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 12.3.

6.13 Initiating a Session

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 13 and its subsections, except as noted in this section.

The UAC MUST include a message body of type "application/sdp" with the initial INVITE.

To support codec selection, an SDP session description MUST be included in:

- the initial INVITE request (offer); and
- the first reliable non-failure response to the INVITE (*e.g.*, 183-Session-Progress sent reliably).

6.14 Modifying an Existing Session

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 14 and its subsections, except as noted in this section.

If a re-INVITE is sent, it MUST include a message body of type "application/sdp" with a new offer. Furthermore, CMSS compliant applications MUST support the procedures for modifying an existing session described in Section 8.4.4.

6.15 Terminating a Session

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 15 and its subsections.

6.16 Proxy Behavior

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 16 and its subsections, except as noted in this section.

Support for multiple simultaneous media streams for a single call is OPTIONAL for an MTA. Since parallel forking may result in multiple simultaneous media streams for a single call, CMSS compliant implementations SHOULD NOT use parallel forking.

6.17 Transactions

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 17 and its subsections except as noted in this section.

Behavior of CMSS servers (proxies) MUST be in accordance with section 8 of this document, which takes precedence over [8] section 17 in case of any conflicts.

[8] Thus, if IPsec is required by the PacketCable Security architecture, use of SIPS URIs will lead to using TLS over IPsec.

[9] Thus, if IPsec is required by the PacketCable Security architecture, use of SIPS URIs will lead to using TLS over IPsec.

CMSS compliant User Agent Servers MAY return an error code 486 (Busy Here) to an INVITE request for a user if a dialog already exists for that user and the new INVITE is not part of that dialog.

6.18 Transport

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 18 and its subsections.

6.19 Common Message Components

6.19.1 SIP and SIPS URI Component

The definition of a SIP/SIPS-URI is as given in [8] section 19.1.1 and extended in Section 7.1.1 in this document.

6.20 Header Fields

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 20 and its subsections, except as defined in this section.

In addition to the extensions listed in section 7, the following SIP headers MUST be supported by CMSS compliant applications.

1) Accept
2) Accept-Encoding
3) Accept-Language
4) Allow
5) Call-ID
6) Contact
7) Content-Disposition
8) Content-Encoding
9) Content-Length
10) Content-Type
11) CSeq
12) From
13) Max-Forwards
14) MIME-Version
15) Proxy-Require
16) Record-Route
17) Require
18) Route
19) Supported
20) Timestamp
21) To
22) Unsupported
23) Via

Other SIP headers MAY be supported by CMSS compliant applications. CMSS compliant applications SHOULD ignore unsupported optional headers.

Listed below is each SIP header defined in [8], and the requirements for supporting each in CMSS are identified.

6.20.1 Accept

The Accept header MUST be supported as specified in [8] section 20.1.

6.20.2 Accept-Encoding

The Accept-Encoding header MUST be supported as specified in [8] section 20.2, except as noted below.

The Accept-Encoding header MAY be used by CMSS compliant implementations. The "identity" encoding value MUST be supported; other encodings MAY be supported.

6.20.3 Accept-Language

The Accept-Language header MUST be supported as specified in [8] section 20.3, except as noted below.

The “Accept-Language” header SHOULD be present in requests with a value of “en” for English, which MUST be supported. Other values MAY be supported.

6.20.4 Alert-Info

Support for the Alert-Info header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.4.

It is noted that there are security risks associated with acting on the Alert-Info header as described in [8] section 20.4.

6.20.5 Allow

The Allow header field MUST be supported as specified in [8] section 20.5. The “Allow” header MUST be present in the initial INVITE and the 200-OK response to the initial INVITE.

Refer to Appendix II for a list of supported methods.

6.20.6 Authentication-Info

Support for the Authentication-Info is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.6.

See also Section 6.22.

6.20.7 Authorization

Support for the Authorization header field is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.7.

See also Section 6.22.

6.20.8 Call-ID

The Call-ID header MUST be supported as specified in [8] section 20.8, except as noted below.

CMSS restricts contents of the Call-ID header in order to support user Privacy.

When Privacy is requested by the session originator, the Call-ID MUST NOT contain the “@” sign and hence consists of a single “word” as defined in [8] section 25.1. The “word” MUST be a random identifier, and MUST be unique across all possible UAs with probability of greater than 0.999999. A suggested implementation is a text encoding (which does not contain an “@”) of a cryptographic hash of phone number, time, a random number, and a quantity provisioned or manufactured to be unique across UAs of otherwise identical manufacture. The last quantity is suggested to help prevent UAs of an otherwise identical manufacture from producing identical “random” Call-IDs when presented with identical stimuli.

6.20.9 Call-Info

Support for the Call-Info header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.9.

It is noted that there are security risks associated with acting on the Call-Info header as described in [8] section 20.9.

6.20.10 Contact

The Contact header MUST be supported as specified in [8] section 20.10, except as noted below.

CMSS compliant applications MUST populate the Contact header field in an INVITE request, and in a 2xx response to an INVITE request, with a SIP-URI. Support for any other type of URI is OPTIONAL.

When the user is requesting Privacy, the Contact header field SHOULD NOT contain any domain names; the IP address form SHOULD be used instead. It should be noted that, in systems with multiple network interfaces, use of the (single) IP address form can reduce the overall system reliability. If multiple interfaces exists and reliability is a concern, it is considered it a reasonable trade-off to refrain from using the IP address form.

CMSS compliant applications MUST populate the Contact header field in a 3xx response to an INVITE request with a valid SIP-URI or tel-URI. If the new destination is a telephone number, it MUST contain a tel URI with the number of the new destination as described is section 7.1. Support for any other type of URI is OPTIONAL.

6.20.11 Content-Disposition

The Content-Disposition header MUST be supported as specified in [8] section 20.11, except as noted below.

The Content-Disposition header MAY be used by CMSS compliant implementations. The value "session" MUST be supported; other values MAY be supported.

Note that the default value for message bodies of type "application/sdp" is "session", whereas the default value for all other message body types (*e.g.*, "message/sipfrag") is "render". If the default value is not desired, then the Content-Disposition header MUST be included.

6.20.12 Content-Encoding

The Content-Encoding header MUST be supported as specified in [8] section 20.12, except as noted below.

The Content-Encoding header MAY be used by CMSS compliant implementations. The "identity" encoding value MUST be supported; other encodings MAY be supported.

6.20.13 Content-Language

Support for the Content-Language header is OPTIONAL, however if supported, it MUST be as specified in [8] section 20.13.

6.20.14 Content-Length

This header MUST be supported as specified in [8] section 20.14.

6.20.15 Content-Type

The “Content-Type” header MUST be supported as specified in [8] section 20.15.

Refer to Appendix II for a list of supported values.

6.20.16 CSeq

The CSeq header MUST be supported as specified in [8] section 20.16.

6.20.17 Date

Support for the Date header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.17.

6.20.18 Error-Info

Support for the Error-Info header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.18.

It is noted that there are security risks associated with acting on the Error-Info header as described in [8] section 20.18.

6.20.19 Expires

Support for the Expires header in the methods and responses defined in[8] is OPTIONAL[10]; however, if supported, it MUST be as specified in [8] section 20.19.

6.20.20 From

The From header MUST be supported as specified in [8] section 20.20, except as noted below.

In support of user Privacy, CMSS restricts the allowed contents of the SIP "From" header.

When the session originator requests Privacy, compliant applications MUST generate a From header according to the following rules:

- The display-name MUST be "Anonymous".
- The addr-spec MUST contain the identifier "anonymous" for userinfo.
- The addr-spec MUST contain the non-identifying hostname "anonymous.invalid".

6.20.21 In-Reply-To

Support for the In-Reply-To header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.21.

Note that use of this header is subject to security considerations as described in [8] section 20.21.

6.20.22 Max-Forwards

The Max-Forwards header MUST be supported as specified in [8] section 20.22, except as noted below.

When a CMSS compliant implementation of a back-to-back User Agent (B2BUA) forwards a request, it MUST use a Max-Forwards value equal to the incoming Max-Forwards value minus one.

6.20.23 Min-Expires

Support for the Min-Expires header is OPTIONAL (since support for the REGISTER method is optional); however, if supported, it MUST be as specified in [8] section 20.23.

6.20.24 MIME-Version

The MIME-Version header MUST be supported as specified in [8] section 20.24, except as noted below.

The MIME-Version header MAY be used by CMSS compliant implementations. The version "1.0" value MUST be supported; other values MAY be supported.

6.20.25 Organization

Support for the Organization header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.25.

6.20.26 Priority

Support for the Priority header is OPTIONAL; however if supported, it MUST be as specified in [8] section 20.26.

There are security ramifications for entities that act on this header.

6.20.27 Proxy-Authenticate

Support for the Proxy-Authenticate header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.27.

See also Section 6.22.

[10] Note that, per Section 7.5, support for the Expires header is required for the SUBSCRIBE method.

6.20.28 Proxy-Authorization

Support for the Proxy-Authorization header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.28.

See also Section 6.22.

6.20.29 Proxy-Require

The Proxy-Require header MUST be supported as specified in [8] section 20.29, except as noted below.

In addition to the standards-track RFCs' option tags, option tags from non-IETF documents MAY also be used, as long as they are defined in this document.

Refer to Appendix II for a list of supported values.

6.20.30 Record-Route

The Record-Route header MUST be supported as specified in [8] section 20.30.

6.20.31 Reply-To

Support for the Reply-To header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.31.

6.20.32 Require

The "Require" header MUST be supported as specified in [8] section 20.32, except as noted below.

In addition to the standards-track RFCs' option tags, option tags from non-IETF documents MAY also be used, as long as they are defined in this document.

Refer to Appendix II for a list of supported values.

6.20.33 Retry-After

Support for the Retry-After header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.33.

6.20.34 Route

The Route header MUST be supported as specified in [8] section 20.34.

6.20.35 Server

Support for the Server header is OPTIONAL; however, if supported, it MUST be as specified [8] section 20.35.

6.20.36 Subject

Support for the Subject header is OPTIONAL, however if supported, it MUST be as specified [8] section 20.36.

6.20.37 Supported

The "Supported" header MUST be supported as specified in [8] section 20.37, except as noted below.

In addition to the standards-track RFCs' option tags, option tags from non-IETF documents MAY also be used, as long as they are defined in this document.

Refer to Appendix II for a list of supported values.

6.20.38 Timestamp

The Timestamp header MUST be supported as specified in [8] section 20.38, except as noted below.

CMSS compliant implementations MAY send the Timestamp header in requests; if received, this header MUST be processed as described in [8] section 20.38.

6.20.39 To

The To header MUST be supported as specified in [8] section 20.39, except as noted below.

In support of user Privacy, CMSS restricts the allowable contents of the SIP “To” header. Typically, the “To” header indicates the dialed digits in a tel-URI (see section 7.1). This information is of end-to-end significance, and might reveal information about the caller’s location, *e.g.*, local, long-distance, PBX, or international.

When the call originator requests Privacy, CMSS compliant applications MUST generate a “To” header according to the following rules:

> The display-name MUST be absent.
>
> If a global telephone number is used, then the userinfo part of the addr-spec MUST contain a full E.164 number, including the country code.
>
> The host part of the addr-spec MUST contain the non-identifying hostname “anonymous.invalid”.

If anonymity is not requested by the call originator and the user dialed a telephone number, then the To: header SHOULD contain a tel-URI with the dialed digits.

6.20.40 Unsupported

The Unsupported header MUST be supported as specified in [8] section 20.40.

6.20.41 User-Agent

Support for the User-Agent header is OPTIONAL; however, if supported, it MUST be as specified [8] section 20.41.

6.20.42 Via

The Via header MUST be supported as specified in [8] section 20.42, except as noted below.

When the user is requesting Privacy, the Via header field SHOULD NOT contain any domain names; the IP address form SHOULD be used instead. Support for IP address Privacy is described in more detail in Section 8.4.1.1.3. It should be noted that, in systems with multiple network interfaces, use of the (single) IP address form can reduce the overall system reliability. If multiple interfaces exists and reliability is a concern, it is considered a reasonable trade-off to refrain from using the IP address form.

A border proxy (EBP) which is passing a request outside of the trusted domain of the service provider MAY encrypt all "Via" headers except the topmost header (*i.e.*, the "Via" header of the terminating proxy) to a non-recognizable string. The proxy MAY include the encrypted string in the Via header, or it may cache the encrypted "Via" headers and include a local token string in the Via header.

6.20.43 Warning

Support for the Warning header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.43.

6.20.44 WWW-Authenticate

Support for the WWW-Authenticate header is OPTIONAL; however, if supported, it MUST be as specified in [8] section 20.44.

See also Section 6.22.

6.21 Response Codes.

CMSS compliant applications MUST be in accordance with SIP/2.0 [8] section 21 and its subsections, except as specified below.

CMSS compliant applications MUST NOT issue a 401 or a 407 response; however, they MUST process any such received responses in accordance with section 6.8.1.2.

6.22 Usage of HTTP Authentication

Support of HTTP Authentication is OPTIONAL; however, if used, it MUST be as specified in [2] and [8] section 22. Where these documents conflict, [2] takes precedence.

See also Section 6.21.

6.23 S/MIME

Support of S/MIME is OPTIONAL; however, if used, it MUST be as specified in [2] and [8] section 23. Where these documents conflict, [2] takes precedence.

6.24 Examples

The examples provided in [8] Section 24 do not apply to CMSS compliant implementations. For equivalent examples, refer to Section 8 in this document.

6.25 Augmented BNF for the SIP Protocol

CMSS compliant applications MUST comply with SIP/2.0 [8] section 25.

6.26 Security Considerations: Threat Model and Security Usage Recommendations

CMSS complaint applications MUST comply with the IPCablecom Security specification [2].

Support for the SIP Security Considerations specified in [8] section 26 is considered OPTIONAL, unless they conflict with the IPCablecom Security specification [2], in which case they MUST NOT be used.

Note:
The IPCablecom Security specification may evolve to meet new architecture and security requirements and thus, may be subject for according changes. Therefore, it is not ruled out that consideration of further (SIP) security measures other than or in addition to the ones like IPSec currently specified in the IPCablecom Security specification might be considered in the future.

6.27 Table of Timer Values

CMSS compliant applications MUST comply with SIP/2.0 [8] Appendix A.

CMSS compliant applications MUST also support the timer values defined in Appendix I according to the procedures specified in 8.4.

7 SIP EXTENSIONS

SIP [8] has a flexible mechanism for adding extensions and new fields to the protocol for support of additional capabilities. This section defines a set of SIP extensions that enables IPCablecom CMSS-compliant systems to provide a robust multimedia service platform supporting basic telephony, CLASS, and custom calling features, while at the same time allowing the supported services to evolve to a multimedia environment. Many of the extensions have been documented in RFCs, to which this document provides cross-references. Several of these extensions have their base in the Distributed Call Signalling (DCS) framework, as described in [30].

This section describes procedures applicable to both NCS and SIP based endpoints; however, it should be noted that SIP based MTAs are out of scope of IPCablecom 1.2 and are described and listed in this section for reference purposes only. The term SIP User Agent (UA) in this section refers to an originator/terminator of SIP requests. The combination of a UA with its SIP Proxy is in many ways equivalent to a CMS; likewise a CMS may be decomposed into a UA and a SIP Proxy (with a hidden and untestable interface between them) as shown in Figure 3.

This section follows the naming convention of SIP [8], of User Agents, Clients, Servers, and Proxies. User Agent Clients initiate requests and in particular initiate sessions (*i.e.*, they are call originators), and User Agent Servers respond to requests and in particular accept session requests (*i.e.*, they are call terminators). A User Agent performs either role as required within the context of the call. The description of each extension in this section gives the specific procedures for CMSes and Proxies.

This specification extends SIP in several ways, which are summarized here. All these extensions MUST be supported:

- CMSS supports a resource reservation scheme in which network resources are reserved prior to alerting the user. This is done through specification of preconditions that must be met prior to continuing the session establishment. Confirmation that the preconditions are met is indicated by an additional end-to-end message exchange (UPDATE/200-OK), which is nested within the normal INVITE/200-OK/ACK message exchange. This extension allows network resources to be reserved prior to alerting the user and also allows network resources to be committed after the user has accepted the invitation, *i.e.*, answered the call. This extension is described further in [14].
- CMSS supports Privacy extensions to SIP. These extensions enable users to make connections without identifying themselves or revealing location information. When Privacy is not requested by the originator, calling number delivery and calling name delivery is provided to the destination (*i.e.*, Caller-ID service) in a reliable manner. Entity identity is also provided to support regulatory features such as Customer Originated Trace, enabling a destination party to report a harassing session even if the originator requested anonymity. This extension is further described in [16] and [17].
- CMSS supports the DCS proxy-to-proxy extensions to SIP that allow proxies to pass additional information between them to perform service-provider functions such as accounting, authorization, billing, coordination of resources, electronic surveillance, etc. This extension is further described in Appendix III.
- CMSS supports the ability to send a reliable provisional response to a SIP request, ensuring the delivery of the provisional response to the initiating UA, with retransmissions as needed. This extension is further described in [15].
- CMSS supports the ability to send a request to another user agent to instruct that other user agent to initiate a new INVITE. Three extensions are defined for this, as described in Appendix IV, [18] and Appendix V.
- CMSS supports the ability to send a request to another user agent to update that user agent with parameters of the session that do not impact the state of the session (*e.g.*, media parameters). This extension is further described in [19].

The remainder of this section defines further extensions to SIP required by a CMSS compliant application.

This section, and section 6, define the nearly complete set of enhancements and requirements to a standard SIP implementation based on [8]. However, not all details of the required behavior can be captured in these sections. Later sections provide details needed for certification and interoperability testing. Sections 6 through 9 of this document are normative.

7.1 URIs for Telephone Calls

CMSS compliant implementations MUST support URIs for telephone calls except as noted in this section.

The SIP URI can already contain a user part which reproduces the syntax of the tel URI [31] with characters other than those in the "unreserved" and "user-unreserved" sets escaped (see [8], Section 25.1). Such a user part is indicated by also including the "user=phone" parameter. CMSS defines extensions to the SIP URI to enable the use of telephone numbers as defined in [32] and [33]. These extensions relate to number portability and freephone service. At the time of writing, these extensions are works in progress, and hence the relevant parts to which CMSS compliant applications must adhere are included below.

7.1.1 Syntax

The "telephone-subscriber" production in [8] section 25 references the syntax of the tel URI [31]. The syntax is here modified and extended as suggested in [32] and [33]. Furthermore, a dial-around indicator is added to indicate when the user dialed a carrier access code. The resulting grammar is shown below.

The syntax definition follows RFC 2396 [23], indicating the actual characters contained in the URI. Note that the reserved characters "+", ";", "=", and "?" MUST NOT be escaped if shown in the grammar definitions below as they are delimiters for the "tel" URI scheme. These reserved characters MUST be escaped if they appear in parameter values.

Characters other than those in the "reserved" and "unsafe" sets (see [23]) are equivalent to their "% HEX HEX" encoding.

The "tel" URI has the following syntax:

```
telephone-uri       = "tel:" subscriber *param
subscriber          = global-number / local-number
global-number       = global-number-part [isdn-subaddress]
global-number-part  = "+" 1*phonedigit
local-number        = local-number-part [isdn-subaddress] [context]
local-number-part   = 1*phonedigit
isdn-subaddress     = ";isub=" 1*uric
context             = ";phone-context=" descriptor *("," descriptor)
descriptor          = domainname / global-number-part
domainname          = *( domainlabel "." ) toplabel [ "." ]
domainlabel         = alphanum
                          / alphanum *( alphanum / "-" ) alphanum
toplabel            = ALPHA / ALPHA *( alphanum / "-" ) alphanum
param               = ";" (pname ["=" pvalue ] / yu-tel-param)
pname               = 1*uric
pvalue              = *uric
phonedigit          = HEXDIG / visual-separator
visual-separator    = "-" / "." / "(" / ")"
alphanum            = ALPHA / DIGIT

yu-tel-param        = *1(routing-number)
                          *1(npdi-dip-indicator)
                          *1(carrier-id-code)
                          *1(dial-around-indicator)

routing-number      = "rn=" global-rn / local-rn
global-rn           = "+" 1*phonedigit
local-rn            = 1*phonedigit [context]

npdi-dip-indicator  = "npdi"
carrier-id-code     = "cic=" global-cic / local-cic
global-cic          = "+" 1*phonedigit
local-cic           = 1*phonedigit [context]
dial-around-indicator = "dai"
```

Production with missing non hash terminals can be found in [8] and RFC-2234 [21].

An example of a SIP URI that includes local-number-portability information is:

sip:+1-212-555-1212;rn=+1-212-234-8706;npdi@dcs-proxy;user=phone

7.1.1.1 Phone Numbers

The <subscriber> part of the URI indicates the number. The phone number can be represented in either global (E.164) or local notation. All phone numbers MUST use the global form unless they cannot be represented as such. Numbers from private numbering plans, emergency ("911", "112") and some directory assistance numbers (e.g., "411") and other "service codes" (numbers of the form N11 in the United States) cannot be represented in global (E.164) form, and need to be represented as a local number with a context. Local numbers MUST be tagged with a phone-context.

Implementations MUST NOT assume that telephone numbers have a maximum, minimum or fixed length, or that they always begin with a certain number.

7.1.1.1.1 Separators in Phone Numbers

Phone numbers MAY contain visual separators. Visual separators (<visual-separator>) merely aid readability and MUST be ignored by the CMSS compliant implementation.

Even though ITU-T E.123 [9] recommends the use of space characters as visual separators in printed telephone numbers, tel URIs MUST NOT use spaces.

7.1.1.1.2 Alphabetic Characters

In some countries, it is popular to write phone numbers using alphabetic characters which correspond to certain numbers on the telephone keypad. The URI format does not support this notation since the mapping from alphabetic characters to digits is not completely uniform internationally, although there are standards addressing this issue.

Since called and calling terminal numbers (TNs) are encoded in Binary Coded Decimal (BCD) in ISUP, this allows for six additional values per digit, sometimes represented as the hexadecimal characters A through F. However, in accordance with E.164, they may not be included in global numbers. Their use in local numbers is not defined, but is not prohibited.

7.1.1.1.3 Global and Local Numbers

Global (international) numbers are identified by the "+" character prefix. Global numbers MUST be composed with the country (CC) and national (NSN) numbers as specified in E.123 [9] and E.164 [10]. International numbers have the property of being context free everywhere in the world and are RECOMMENDED.

Local numbers only work within a certain geographical area or a certain part of the telephone network, *e.g.*, a private branch exchange (PBX) or a particular country. URIs with local phone numbers should appear only in environments where all local entities can successfully set up the call by passing the number to the dialing software. Digits needed for accessing an outside line, for example, are not included in local numbers.

The phone-context parameter can be used to identify unambiguously the local context where the local number is valid. The parameter value is defined by the "owner" of the local number. The parameter can contain a list of contexts that enumerate all the contexts where this number is valid. It does NOT in any way indicate a prefix that turns the local number into a global (E.164) number. The phone-context provides a unique identifier that lets a client know whether it can interpret the local number. It has no other meaning or function.

There are two ways to label the context: via a global number prefix (*e.g.*, "+33") and via a domain name, *e.g.*, "houston.example.com". The choice between the two is left to the "owner" of the local number and is governed by whether a global number prefix or a domain name is a valid identifier for a particular local number.

The two label types were chosen so that, in almost all cases, a local administrator can pick an identifier that is reasonably descriptive and does not require a new assigned number. It is left to the administrator to assign an appropriate identifier and to use it consistently. In many cases, an organization can choose many different identifiers.

The domain name does not have to resolve to any actual host, but it MUST be under the administrative control of the entity managing the local phone context.

For example, ";phone-context=+1,+49" indicates that the number is valid in country codes 1 (North America) and 49 (Germany). For a local number valid within a PBX, the organization can choose any number under its control to identify the context. For example, a context consisting of any of the organization's global numbers may be suitable, or a substring that is completely occupied by the organization. For example, +49-6151-16 would be a suitable prefix for the TU Darmstadt, as it uses all numbers starting with those digits.

The global number prefix does not imply that adding it to the number will generate a valid E.164 number. It might do this, but this behavior cannot be relied upon. (For example, "911" should be labeled with the context "+1", but "+1-911" is not a valid E.164 number.) As noted earlier, numbers that can be represented as valid E.164 numbers SHOULD NOT be represented as local numbers plus phone-context.

7.1.1.2 ISDN Subaddresses

A phone number MAY also contain an <isdn-subaddress> parameter which indicates an ISDN subaddress. Support for this parameter is mandatory, *i.e.*, a CMSS compliant implementation MUST reject the URI if it cannot interpret ISDN subaddresses. ISDN subaddresses typically contain IA5 characters, but may contain any octet value.

7.1.1.3 Other Parameters

Future extensions to this URI scheme may add other parameters (<param> in the ABNF). Such parameters can be either mandatory or optional. Mandatory parameters are required to start with "m-". An implementation MAY ignore optional parameters. An implementation MUST NOT use the URI if it contains unknown mandatory parameters. The "m-" prefix cannot be added to parameters that were already registered (except to create a new, logically distinct parameter). The "phone-context" parameter in this document is mandatory.

For example, <param> parameters can be used to store application-specific additional data about the phone number, its intended use, or any conversions that have been applied to the number.

All new parameters must be registered with IANA.

7.1.1.4 Routing Number, Number Portability, Carrier Identification Code, and Dial Around Indication

The yu-tel-param parameters are defined as follows:

- The presence of "npdi" indicates that a Number Portability Database (NPDB) dip has been performed. If "npdi" is not present, it indicates that either NPDB dip has not yet been performed or NP is not relevant.
- The first 1, 2 or 3 digits in the "global-cic" identify a country code. The rest of the digits identify a carrier ID code assigned in that country.

The "rn," "npdi," and "cic", if present, can each appear at most once. The "cic," "rn" or "npdi" may be removed when there is no need to carry it further in the call signalling messages. For example, when a freephone call reaches the freephone service provider/carrier serving that freephone number, the "cic" may no longer be needed when the call is to be routed to the called party or another network. Whether and when to remove the new parameters defined in this document are outside the scope of this document.
When the "rn" is present, the "npdi" may or may not be present. This is because the routing number may be present independent of NP. When the "npdi" parameter is not present, it indicates that either NPDB dip has not been performed or NP is not relevant. If a CMSS server is set to perform the NPDB queries, and if a received INVITE message does not contain the "npdi" parameter, the CMSS server will perform the NPDB query. The details of the NPDB query are outside the scope of this document. The routing number received in the response (converted to global-rn or local-rn format) will replace the routing number in the "rn" parameter if present, or will be used by the new "rn" parameter if the "rn" parameter is not present. The "npdi" parameter will be included in this case. The routing number can be a global routing number (*e.g.*, with "+" and the country code plus the national number) or a local (*e.g.*, network-specific) routing number.

Although rare, it is possible to have the "cic," "rn" and POTS number all in the same "tel" URI. When all three are present, the "cic" is used for call routing. When only the "rn" and the POTS number are present, the "rn" is used for making routing decisions (*e.g.*, to check against the routing tables). If the "cic" and "rn" parameters are not present, the telephone number immediately after "tel:" is used for call routing. Note that specific "cic" values can be reserved to indicate call routing information instead of indicating a valid CIC that is assigned to a carrier. For example, a "cic" value of "+1-0110" in a response from the freephone database in the U.S. indicates "local, translated number provided." In this particular case, the "cic" is ignored and the "rn" and the POTS number are used for call routing based on the rules described above.

The CICs in the U.S. are assigned to entities that purchase Feature Group B (FGB) or Feature Group D (FGD) access, FGB translation access, or are Local Exchange Carriers (LECs). CICs are also returned in the response to a freephone query, in which case they identify the freephone service provider that serves the queried freephone number. CICs in the U.S. are currently managed by the North American Numbering Plan Administration (NANPA).

The CIC can be expanded to include VoIP carriers and other types of carriers in the same country or under the same country code so that all carriers can be identified in the IP domain for routing purpose. International service providers and carriers can be identified by the E.164 country codes.

Per the definitions above, the yu-tel-param parameters are defined as optional in the tel-URI sense. However, CMSS compliant applications MUST support the yu-tel-param parameters and hence they MUST NOT be ignored.

CMSS compliant implementations will use the CIC feature to identify not only freephone carriers but also customers' pre-subscribed or dialed carrier access codes as follows:

- If the number dialed is a freephone number, then the CIC MUST identify the carrier serving that freephone number, irrespective of the customers presubscribed or dialed carrier.
- If the number dialed is a not a freephone number, then the CIC MUST identity the pre-subscribed carrier for the caller, unless the caller dialed a carrier access code, in which case that the CIC for that carrier MUST be used. In the latter case, the dial-around indicator MUST be included as well; the dial-around indicator MUST NOT be included in any other case.

7.1.2 Procedures at an Originating CMS

If the Request-URI of an initial INVITE contains a telephone number, a carrier-id-code parameter MUST be included in the telephone-subscriber part with a value corresponding to identity of the carrier preferred by the party paying for the call. Note that, for freephone numbers, this will be the carrier serving the freephone number. If CMS_O provides support for the caller to select a preferred carrier on a per-call basis, the carrier-id-code parameter MUST be included in the telephone-subscriber part and set to the carrier-id-code that the caller has dialed, unless the number dialed is a freephone number. The carrier-id-code MAY furthermore be included in the Refer-To URI of a REFER request, when the party performing the refer is the party paying for the call.

A SIP URI containing "npdi" in the telephone-subscriber part MUST NOT appear other than in an initial INVITE Request-URI or the Refer-To URI of a REFER request sent to a proxy or CMS_T.

An originating CMS that performs the local-number-portability lookup and passes the REFER or initial INVITE request to a proxy or a terminating CMS MUST generate a Request-URI containing a SIP URI with the "npdi" parameter in the telephone-subscriber part. Note that this applies irrespective of whether a received Request-URI was a tel URI or a SIP URI.

An originating CMS that performs the local-number-portability lookup and passes the REFER or initial INVITE request to a proxy or a terminating CMS MUST include the "rn" parameter indicating the returned value if the local-number-portability lookup returned a value different from the dialed number.

7.1.3 Procedures at a Terminating CMS

No specific procedures are defined.

7.1.4 Procedures at Proxy

A SIP URI containing a "npdi" in the telephone-subscriber part MUST NOT appear other than in an initial INVITE Request-URI or the URI of a Refer-To header in a REFER request sent to another proxy or CMS_T.

A proxy that performs the local-number-portability lookup and passes the REFER or initial INVITE request to another proxy or CMS_T MUST, in each of these cases, generate a SIP URI containing "npdi". A proxy that performs the local-number-portability lookup and passes the REFER or initial INVITE request to another proxy or CMS_T MUST include the "rn" parameter indicating the returned value if the local-number-portability lookup returned a value.

7.2 Reliability of Provisional Responses

CMSS compliant implementations MUST support the extensions defined in [15] except as noted in this section.

CMSS compliant implementations MUST send any non-100 provisional response reliably as defined in [15].

7.3 SIP UPDATE Method

CMSS compliant implementations MUST support the extensions defined in [19] except as noted in this section.

CMSes MUST include the method "UPDATE" in the Allow header field in the relevant requests and responses as described in section 6.

7.4 Integration of Resource Management and SIP

CMSS compliant implementations MUST support the extensions defined in [14] except as noted in the subsections below.

7.4.1 Procedures at an Originating CMS

The session originator (CMS_O) MUST include a mandatory end-to-end QoS precondition (including the desired direction) for each media flow in the SDP sent with the INVITE. The precondition-type MUST be "qos", the desired status-type MUST be "e2e", and the direction-tag MUST be "sendrecv".

The session originator (CMS_O) MUST include a Require header with option tag "precondition" in the initial INVITE.

7.4.2 Procedures at a Terminating CMS

On receipt of an INVITE request containing preconditions, a terminating CMS (CMS_T) MUST generate a 183-Session-Progress response with an SDP. For each media flow with precondition in the SDP sent, the precondition-type MUST be "qos", the desired status-type MUST be "e2e", and the direction-tag MUST be "sendrecv". The terminating CMS

MUST request a confirmation of the QoS reservation for CMS$_O$ by adding “a=conf:”. The precondition-type MUST be “qos”, the status-type MUST be “e2e”, and the direction-tag MUST be “recv”.

CMS$_T$ MUST wait for the UPDATE message from the originator containing the success/failure indication of each precondition as determined by the originator. If that confirmation indicates a failure for a mandatory precondition, CMS$_T$ MUST send a 580-Precondition-Failure response with the outcome of the preconditions to CMS$_O$.

Once the preconditions are met, CMS$_T$ alerts the user, and the SIP transaction completes normally.

7.5 SIP-Specific Event Notification

CMSS compliant implementations MUST support the extensions defined in [18].

7.6 The REFER Method

CMSS compliant implementations MUST support the extensions defined in Appendix IV except as noted in this section. Note that this method makes use of the Notify mechanism defined in 7.5.

In CMSS, the use of the REFER method is specified only within an existing dialog. The REFER method MAY be used outside of a dialog; however, the details of Event Messaging and hence the use of P-DCS-Billing-Info headers is not specified and would need to be defined by the implementer.

7.6.1 Procedures at an Originating CMS

The CMS originating a REFER MUST include additional header parameters for P–DCS-Billing-Info, P–DCS-Laes, and PDCS-Redirect, as specified in Section 7.7.

The Accept header MUST be present in a REFER request and the value MUST include “message/sipfrag”.

7.6.2 Procedures at a Terminating CMS

If the action requested by the REFER is a SIP INVITE, then the NOTIFY sent when it completes MUST include the call-leg identification for the newly established session (*i.e.*, the From, To, and Call-ID headers).

The uri-parameters in a SIP URI in a Refer-To header MUST be present in the generated INVITE request as described in sections 6.19 and 7.7.

7.7 SIP Proxy to Proxy Extensions for Supporting DCS

CMSS compliant implementations MUST support the extensions defined in Appendix III except as defined in this section.

7.7.1 P-DCS-Trace-Party-ID

Support for the P-DCS-TRACE-PARTY-ID header is not required.

7.7.2 P-DCS-Gate

The P-DCS-Gate header is not required to be present in the initial INVITE request sent between CMSes.

7.7.3 P-DCS Option Tag

The extensions defined in Appendix III do not define any option tags; however the CMSS specification defines the option tag "P-DCS" to indicate support of the extension headers P-DCS-OSPS, P-DCS-Billing-Info, P-DCS-Laes and P-DCS-Redirect as specified in Appendix III.

CMSS compliant implementations may use the option tag "P-DCS" in the Supported and Require headers, and they MUST accept requests with a Require value of “P-DCS”. CMSS compliant implementations MUST include a Require header containing the option tag "P-DCS" when performing an INVITE for Busy-Line-Verify or Emergency-Interrupt as described in Section 8.4.4.4 and 8.4.4.5.

7.8 The SIP "Replaces" Header

CMSS compliant implementations MUST support the extensions defined in Appendix V.

7.9 Private Extensions to the SIP Protocol for Asserted Identity within Trusted Networks

CMSS compliant implementations MUST support the asserted identity extensions defined in [17], except as defined in this section. Note that support of the asserted identity extensions not only implies support of the P-Asserted-Identity header, but also implies support of the Privacy header with a value of “id” as described in [16] and [17] and a value of "critical" as described in [16], as well as the Proxy-Require option tag "Privacy".

For an on-net originated call, there MUST be a single P-Asserted-Identity header present. For an off-net originated call, there MUST be a single P-Asserted-Identity header present if the calling party number is available; otherwise, the P-Asserted-Identity header MUST NOT be present. The P-Asserted-Identity header MUST NOT use a SIPS URI.

The URI MUST contain the number of the calling party as defined in 7.1, either as a tel-URI or as a SIP-URI with telephone-subscriber syntax and "user=phone". If calling name Privacy is requested, the display-name “Anonymous” MUST be used for this header. If the call is initiated on-net and calling name Privacy was not requested, the display-name MUST be set to the name of the calling party. If the call originated off-net and calling-name Privacy was not requested, the display-name MAY be omitted. If calling number Privacy is requested, a Privacy header with priv-values "id" and "critical" MUST be included and a Proxy-Require containing the option tag "Privacy" MUST be included, as described in [16] and [17].

In order to support the asserted identity extension, a Spec(T) is specified, as described in [17]. IPCablecom’s Spec(T) is defined as follows:

1. Protocol requirements

Implementations must adhere to this document.

2. Authentication requirements

For calls that originate on-net, the procedure specified in [2] must be followed.

For calls that originate off-net, calling party information present in the PSTN signalling messages MUST be used, unless it is user-provided or the PSTN is not trusted, in which case it MUST NOT be used.

3. Security requirements

Connections between nodes within the Trust Domain and between UAs and nodes in the Trust Domain MUST use secure signalling as described in [2].

4. Scope of Trust Domain

The CMSS Trust Domain consists of all CMSS hosts that can communicate either directly or indirectly, subject to the security requirements described in [2].

The CMSS Trust Domain also includes the adjacent PSTN network unless configured otherwise.

MTAs MUST NOT be part of the Trust Domain.

It should be noted that the trust boundary here described for signalling is different from the trust boundary described in Section 5.3, which deals with trust for event messages customer premise equipment and third parties.

5. Implicit handling when no Privacy header is present

The CMSS elements in the Trust Domain MUST support the "id" Privacy service; therefore, absence of a Privacy header is assumed to indicate that the user is not requesting any Privacy. If no Privacy header field is present in a request, elements in this Trust Domain MUST act as if no Privacy is requested.

Note that since CMSS (and [2] together) define(s) a single Trust Domain where all CMSes trust each other, a P-Asserted-Identity header will currently never be removed before being forwarded to another CMS.

7.10 A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)

In order to support the message waiting indicator feature when, for example, a voice-mail server is controlled by a different CMS from the one serving the subscriber, CMSS compliant implementations MUST support the extensions defined in this section. This section is an excerpt from [34].

7.10.1 Background and Appropriateness

Messaging Waiting Indication is a common feature of telephone networks. It typically involves playing a special dial tone (called message-waiting dial tone), lighting a light or indicator on the phone, or both. Message-waiting dial tone is similar but distinct from stutter dial tone.

The methods in the SIP [8] base specification were only designed to solve the problem of session initiation for multimedia sessions, and rendezvous. Since Message Waiting Indication is really status information orthogonal to a session, it was not clear how an IP telephone acting as a SIP User Agent would implement comparable functionality. Members of the telephony community viewed this as a shortcoming of SIP.

Users want the useful parts of the functionality they had using traditional analog and PBX telephones. It is also desirable to provide comparable functionality in a flexible way that allows for more customization and new features.

SIP Specific Event Notification SIP Events [18] is an appropriate mechanism to use in this environment, as it preserves the user mobility and rendezvous features which SIP provides.

Using SIP-Specific Event Notification, A Subscriber User Agent (typically an IP phone or SIP software User Agent) subscribes to the status of their messages. A SIP User Agent acting on behalf of the user's messaging system then notifies the Subscriber whenever the account's messages have changed. The Notifier sends this message summary information in the body of the NOTIFY, encoded in a new MIME type defined below. A User Agent can also explicitly fetch the current status.

A SIP User Agent MAY subscribe to multiple accounts (distinguished by the Request URI). Multiple SIP User Agents MAY subscribe to the same account.

Before any subscriptions or notifications are sent, each interested User Agent must be made aware of its messaging server(s). This MAY be manually configured on interested User Agents, or manually configured on an appropriate SIP Proxy.

7.10.2 Event Package Formal Definition

7.10.2.1 Event Package Name

This document defines a SIP Event Package as defined in SIP Events [8]. The event-package token name for this package is:

"message-summary"

7.10.2.2 Event Package Parameters

This package does not define any event package parameters.

7.10.2.3 SUBSCRIBE Bodies

This package does not define any SUBSCRIBE bodies.

7.10.2.4 Subscription Duration

Subscriptions to this event package MAY range from minutes to weeks. Subscriptions in hours or days are more typical and are RECOMMENDED.

7.10.2.5 NOTIFY Bodies

A simple text-based format is used to prevent an undue burden on low-end user agents, (e.g. , inexpensive IP phones with no display). Although this format is text-based, it is intended for machine consumption only.

If no "Accept" header is present in the SUBSCRIBE, the body type defined in this document MUST be assumed.

The format specified in this package attempts to separate orthogonal attributes of messages as much as possible. Messages are separated by media type (audio, text, image, and video); by message status (new and old); and by urgent and non-urgent type.

The text format begins with a simple status line, and optionally a summary line per media type. Valid media types are Voicemail (audio), Email (text), Fax (image), and Video (video). For each media type the total number of new and old messages is reported in the new and old fields.

In some cases, detailed message summaries are not available. The status line allows messaging systems or messaging gateways to provide the traditional boolean message waiting notification.

Messages-Waiting: yes

In the example that follows, more functionality is available to the User Agent. There are one new and three old voice messages.

Voicemail: 1/3

After the summary, the format can optionally list a summary count of urgent messages. Of the one new and three old voice messages, none of the new messages are urgent, but one of the old messages is. All counters have a maximum value of 4,294,967,295 ((2^32) - 1). Notifiers MUST NOT generate a request with a larger value. Subscribers MUST ignore a larger value.

Voicemail: 1/3 (0/1)

Optionally, after the summary counts, the messaging systems MAY append additional message headers, which further describe newly added messages. If any of these headers are not understood by the CMS, they MUST be ignored. A messaging system which includes message headers in a NOTIFY, should provide an administrator configurable mechanism for selecting which headers are sent. Likely headers for inclusion include To, From, Date, Subject, Message-ID, and Priority. Note that the syntax for these headers is more restrictive than for SIP. For example, line folding is prohibited.

Implementations which generate large notifications are reminded to follow the message size restrictions for unreliable transports articulated in Section 18.1.1 of SIP.

Mapping local message state to new/old message status and urgency is an implementation issue of the messaging server.

Messaging systems MAY use any algorithm for determining the appropriate media type for a message. Such algorithms are out of scope of this document.

7.10.2.6 Subscriber generation of SUBSCRIBE requests

Subscriber User Agents will typically SUBSCRIBE to message summary information for a period of hours or days, and automatically attempt to re-SUBSCRIBE well before the subscription is completely expired. If re-subscription fails, the Subscriber SHOULD periodically retry again until a subscription is successful, taking care to backoff to avoid network congestion. If a subscription has expired, new re-subscriptions MUST use a new Call-ID.

The Subscriber SHOULD SUBSCRIBE to that user's message summaries whenever a new user becomes associated with the device (a new login). The Subscriber MAY also explicitly fetch the current status at any time. The subscriber SHOULD renew its subscription after a reboot, or when the subscriber's network connectivity has just been re-established.

The Subscriber MUST be prepared to receive and process a NOTIFY with new state immediately after sending a new SUBSCRIBE, a SUBSCRIBE renewal, an unSUBSCRIBE or a fetch; or at any time during the subscription.

When a user de-registers from a device (logoff, power down of a mobile device, etc.), subscribers SHOULD unsubscribe by sending a SUBSCRIBE message with an Expires header of zero.

7.10.2.7 Notifier processing of SUBSCRIBE requests

When a SIP Messaging System receives SUBSCRIBE messages with the message-summary event-type, it SHOULD authenticate the subscription request in accordance with the requirements in [2]. If authentication is successful, the Notifier MAY limit the duration of the subscription to an administrator defined amount of time as described in SIP Events.

7.10.2.8 Notifier generation of NOTIFY requests

Immediately after a subscription is accepted, the Notifier MUST send a NOTIFY with the current message summary information. This allows the Subscriber to resynchronize its state. This initial synchronization NOTIFY MUST NOT include the optional message headers.

When the status of the messages changes sufficiently for a messaging account to change the number of new or old messages, the Notifier SHOULD send a NOTIFY message to all active subscribers to that account.

A Messaging System MAY send a NOTIFY with an "Expires" header of "0" and a "Subscription-State" header of "terminated" before a graceful shutdown.

7.10.2.9 Subscriber processing of NOTIFY requests

Upon receipt of a valid NOTIFY request, the subscriber SHOULD immediately render the message status and summary information to the end user in an implementation specific way.

The Subscriber MUST be prepared to receive NOTIFYs from different Contacts corresponding to the same SUBSCRIBE (the SUBSCRIBE may have been forked).

7.10.2.10 Handling of Forked Requests

Forked requests are allowed for this event type and may install multiple subscriptions. The Subscriber MAY render multiple summaries to the user, or MAY merge them as described below.

If any of the "Messages-Waiting" status lines report "yes", then the merged state is "yes"; otherwise the merged state is "no".

```
status-line  = "Messages-Waiting: " status CRLF

status = "yes" / "no"
```

The Subscriber MAY merge summary lines in an implementation-specific way if all notifications contain at least one summary line.

```
summary-line = media-type ":" SP new "/" old [ SP urgent ] CRLF

urgent = "(" new-urgent "/" old-urgent ")"
```

7.10.2.11 Rate of Notifications

A Notifier MAY choose to buffer NOTIFY responses for a short administrator-defined period (seconds or minutes) when the message status is changing rapidly. This buffering behavior is RECOMMENDED. Note that timely notification of a newly added message is probably more significant to the end user than notifications of newly deleted messages which do not affect the overall message waiting state.

Notifiers SHOULD NOT generate NOTIFY requests more frequently than once per second.

7.10.2.12 State Agents

Implementers MAY create a service which consolidates and summarizes NOTIFYs from many Contacts. This document does not preclude implementations from building state agents which support this event package.

7.10.2.13 Behavior of a Proxy Server

There are no additional requirements on a SIP Proxy, other than to transparently forward the SUBSCRIBE and NOTIFY methods as required in SIP. Proxies and Redirect servers SHOULD be able to direct the SUBSCRIBE request to an appropriate messaging server User Agent.

7.10.3 Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in an RFC [21].

7.10.3.1 New event-package definition

This document defines a new event-package with the package name:

message-summary

7.10.3.2 Body Format Syntax

The formal syntax for application/simple-message-summary is below:

```
message-summary      = status-line  [*(summary-line)] [ *msg ]
status-line          = "Messages-Waiting:" SP status CRLF
status               = "yes" / "no"
summary-line         = media-type ":" SP new "/" old  [ SP urgent ] CRLF
urgent               = "(" new-urgent "/" old-urgent ")"
media-type           = "Voicemail" / "Email" / "Fax" / "Video"
msg                  = CRLF 1*(mheader)
mheader              = mhname ":" SP mhvalue CRLF
mhname               = alphanum 1*( alphanum / "-" / "_" )
mhvalue              = *( HTAB / %x20-7E / UTF8-NONASCII )
new                  = 1*DIGIT
old                  = 1*DIGIT
new-urgent           = 1*DIGIT
old-urgent           = 1*DIGIT
```

8 CMS-CMS SIGNALLING

In this chapter, the CMS to CMS signalling that takes place between CMSes within a domain, and the signalling that takes place between domains is presented. The primary difference between intra-domain signalling and inter-domain signalling is the use of External Border Proxies, which are described in Section 5.4.

8.1 CMS Interfaces

Signalling between two CMSes is simply referred to as CMS-CMS signalling or CMSS signalling. From a CMS-CMS signalling perspective, the Media Gateway Controller (MGC), Bridge Server, Announcement Server, and other media service nodes are analogous to the CMS, although they do not interface with a Gate Controller. In the following, therefore, the use of the term CMS is to mean any of these devices, unless otherwise noted.

Figure 6 shows the interfaces to a CMS for an on-net to on-net call (Figure 6(a)) and a call involving an off-net endpoint (Figure 6(b)). For on-net calls, the CMS (Call Agent) contains a Gate Controller (GC) function in order to control access to Dynamic Quality of Service on the access network. Initial signalling between the CMS originating a session and the CMS handling termination may be routed through intermediate tandem proxies, but subsequent signalling typically will be sent directly. Both the signalling through tandem proxies and the direct signalling are considered CMS-CMS signalling.

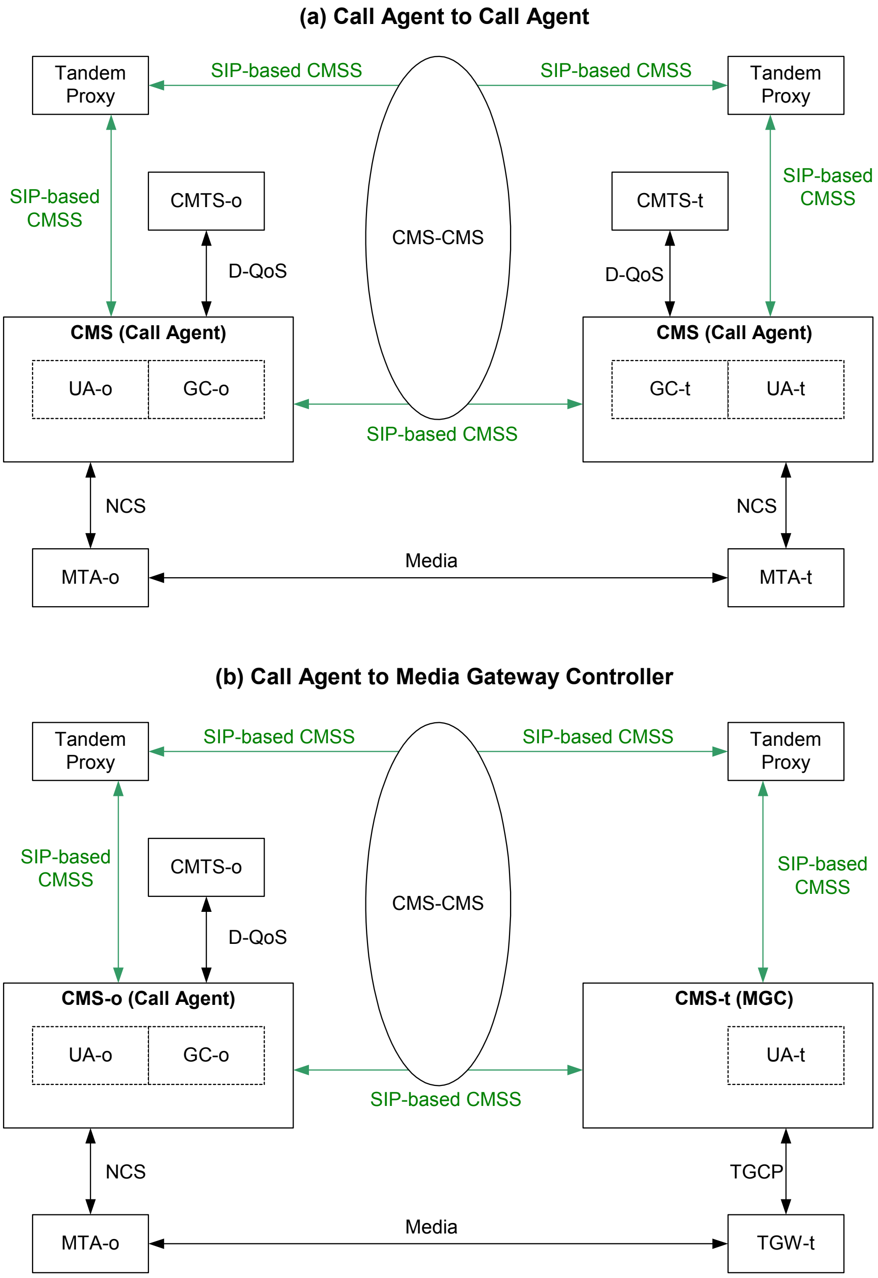

Figure 6: CMS to CMS Signalling

Although all the CMSes in a given domain should be able to communicate, the set of CMSes may not form a fully connected mesh for routing and security reasons. In those cases, some of the CMSes may be reachable only through one or more tandem proxies (*e.g.*, interior or exterior border proxies).

Numerous other interfaces to the CMS exist. These are not shown in Figure 6, and include interfaces to devices such as translation servers, local-number-portability databases, SS7 signalling interfaces, and anonymizers.

The procedures for inter-domain sessions are similar to the procedures for intra-domain sessions with only a few differences. In particular, the initial INVITE message, any interim response messages, and the final response message of the initial INVITE transaction pass through a EBPs in the originating and terminating domains. In addition, call features involving redirection are treated differently; refer to the following sub-sections for further details.

Below, an overview of an example interdomain telephony call flow is presented – the call flow is similar to the intra-domain telephony call flow, except that external border proxies are used for the initial INVITE request its responses:

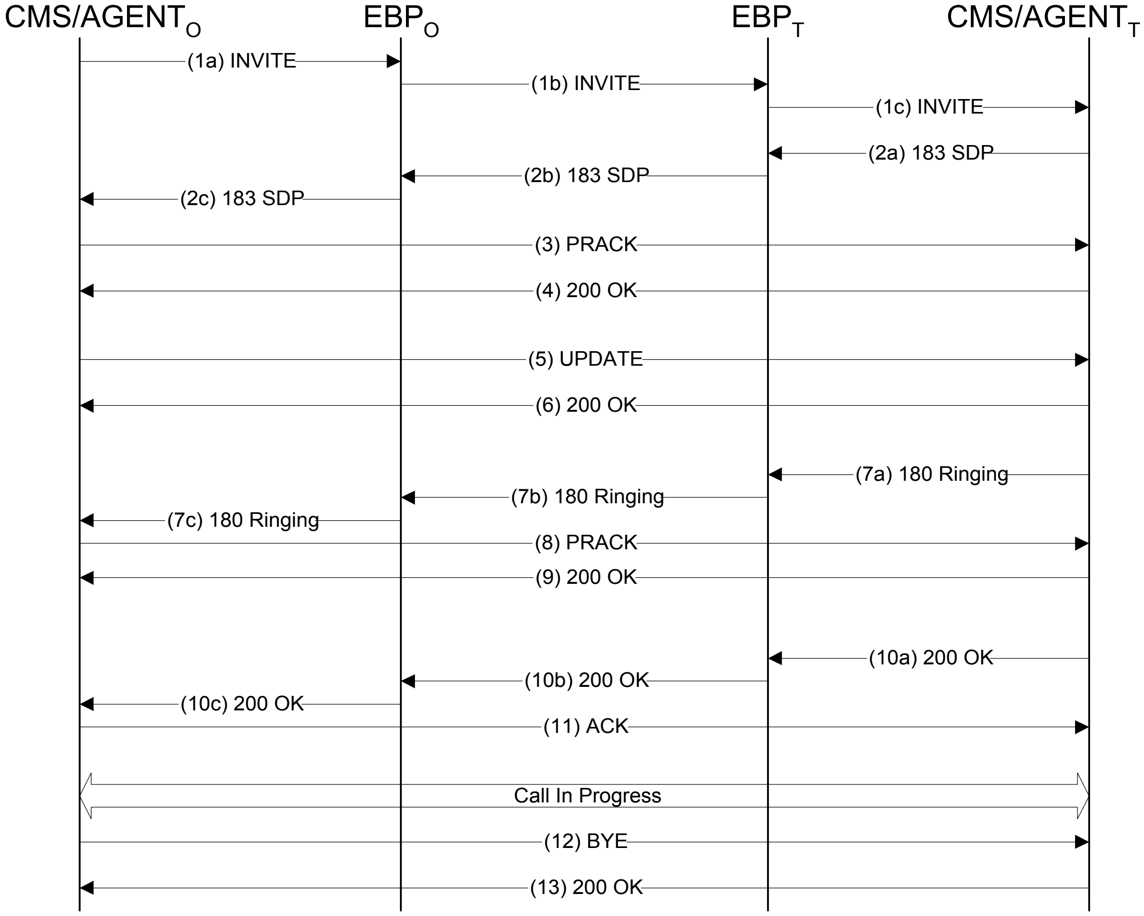

Figure 7: Overview of Interdomain Telephony Call Flow.

End-to-End QoS remains as described in [7].

8.1.1 Overview of CMS Behavior

The CMS contains a trusted SIP User Agent Client (UAC) and User Agent Server (UAS). It maintains call state during the life of the call, and monitors the endpoint device for state changes that affect the call. The interface between the CMS and the endpoint device is outside the scope of this specification, but the particular case of Network-based Call Signalling (NCS) is used when necessary in the examples.

The Call Management Server (CMS) complex includes the CMS and, if needed, Gate Controller functions. The CMS participates in the CMS-CMS signalling; the Gate Controller participates, if needed, in the DQoS signalling (see [4]). Together, they control the coordination of the signalling for call setup and resource management.

DQoS signalling can be used as a secondary fail-safe mechanism to detect call termination. If necessary, the CMS can use the DQoS Gate-Delete message to remove access to QoS resources for a call (see [4] for details).

Messages for setting up a new call, or changing the attributes or participants of an active call, are initiated by the CMS.

8.1.1.1 CMS Behavior in Support of Call Originator

Through a mechanism outside the scope of this specification, the CMS becomes aware that the endpoint device desires to initiate a call, and determines the destination address of that desired call. This may be done, for example, through a Notify message in NCS, where the MTA detected the off-hook condition and collected a complete dial string from a sequence of touchtone button pushes. Alternatively, it may be done through a Notify message in TGCP, where the MG detected a trunk seizure and received the destination address through MF signalling. Or it could be done through an IP-IAM message from a SS7 signalling gateway, or any one of a number of other mechanisms.

The originating CMS (CMS_O) translates the destination address, and then takes the role of a trusted SIP UAC and initiates an INVITE request to the terminating CMS (CMS_T), possibly through one or more proxies. Included in this INVITE request are the SDP definition of the desired media stream(s), the billing/accounting information, the endpoint identification, and the indication of Privacy requested by the call originator.

On receipt of the response to this INVITE request, CMS_O authorizes the resources needed for the media stream(s), directs the endpoint to initiate any resource reservation needed, and informs the destination when the resources are reserved by sending an UPDATE. CMS_O instructs the endpoint to play any media received, and if a provisional response indicating local alerting is received, CMS_O causes the endpoint device to play a local ringback tone. In response to a final 200-OK response, CMS_O cuts through the call and enables the bi-directional media flow(s).

8.1.1.2 CMS Behavior in Support of Call Destination

CMS_O sends an INVITE request to CMS_T, where the dialed number is translated into the address of the terminating endpoint. Through negotiation with the terminating endpoint, CMS_T determines the media stream properties, and authorizes the QoS resources needed. CMS_T responds to the INVITE request with a provisional 183-Session-Progress message, giving the SDP, destination identity information, and billing information if the destination is overriding that given by the call originator, *e.g.*, for reverse charging. Note that, in order to support reverse charging, CMS_O should not generate any event messages that determine the charged party until the 183-Session-Progress response has been received.

CMS_T directs the terminating endpoint to reserve the resources necessary for the media stream(s). On receipt of the UPDATE message from the originating endpoint, CMS_T alerts the destination user. If CMS_T wants to use remote ringback, it sends back a 183-Session-Progress[11]. If CMS_T wants to use local ringback, it sends back a 180-Ringing message. When the terminating endpoint answers the call, CMS_T sends a 200-OK message, cuts through the call and enables the bi-directional media flow(s).

Call features such as call-forwarding-unconditional, call-forwarding-busy, call-waiting, and call-forward-no-answer are controlled and implemented by CMS_T by generating the proper SIP responses as part of the basic call setup procedures. CMS_T, locally storing information about the previous call (on a per-line basis), also implements features such as return-call and call-trace.

8.1.1.3 CMS Behavior in Support of Mid-call Changes

For the duration of the call, CMSO and CMST are available to their respective endpoints, and they respond to any mid-call changes requested by the endpoints. Examples of such changes are: hold/resume; codec change; call transfer; three-way-calling; busy-line verification; and emergency interrupt. CMSO and CMST initiate and perform the CMS-CMS signalling exchanges necessary to make these and similar changes.

8.1.1.4 CMS Behavior in Support of Event Messaging

The IPCablecom Event Messaging specification [3] requires a stream of events to be generated on behalf of each endpoint involved in a call, i.e., for each half of the call (originating and terminating). Each of the originating and terminating event streams is identified by its own Billing-Correlation-ID, and is further identified as to its originating or terminating role. Certain accounting information is needed for the Sig-Start event message [3], and that information is carried in CMS-CMS signalling in the P-DCS-Billing-Info header of the INVITE request. It is further required that each CMS knows the Billing-Correlation-ID and Financial-Entity-ID of the other event message stream, and that information is carried in the first initial INVITE and reliable 1xx, 2xx or 3xx response (typically 183-Session-Progress).

If the call originator and destination are in different IPCablecom Domains (i.e., an inter-domain call), it is necessary for each CMS to generate the complete event stream for the call.

8.1.1.5 CMS Behavior in Support of Call Forwarding

Requirements for event generation for the Call Forwarding services are based on the billing model of the PSTN. A forwarded call is considered to consist of several call-segments, each of which is billed to the party that initiated that call-segment. For instance, a call from party A to party B that is forwarded to party C results in event streams for two separate “calls” – first from A to B (typically charged to A), and a second from B to C (typically charged to B).

[11] The purpose of this 183-Session-Progress message is to aid in PSTN interworking as described in [35], Section 8.2.3.

Continue to consider the case in which party A calls party B and party B forwards the call to party C. Assume that all three parties are in the same IPCablecom domain, and served by the same Record Keeping Server.

We are concerned here with accounting information and with the event messages sent to the Record Keeping Server. CMS B provides accounting information to CMS A concerning the call segment from B to C. This accounting information is used by CMS A to create the INVITE message it sends to C.

Two event streams will be generated, and the information in them will be correlated by a service instance event. The process is as follows: CMS A generates an origination event stream for the call segment from A to B. CMS C generates a termination event stream for the call segment from B to C. CMS B generates a service instance event that correlates these two streams. The correlation information is carried in the P-DCS-Billing-Info headers in the 302-Redirect response. See diagram below.

Note: Refer to Section 8.4.1.8.2 for a more detailed description of this scenario.

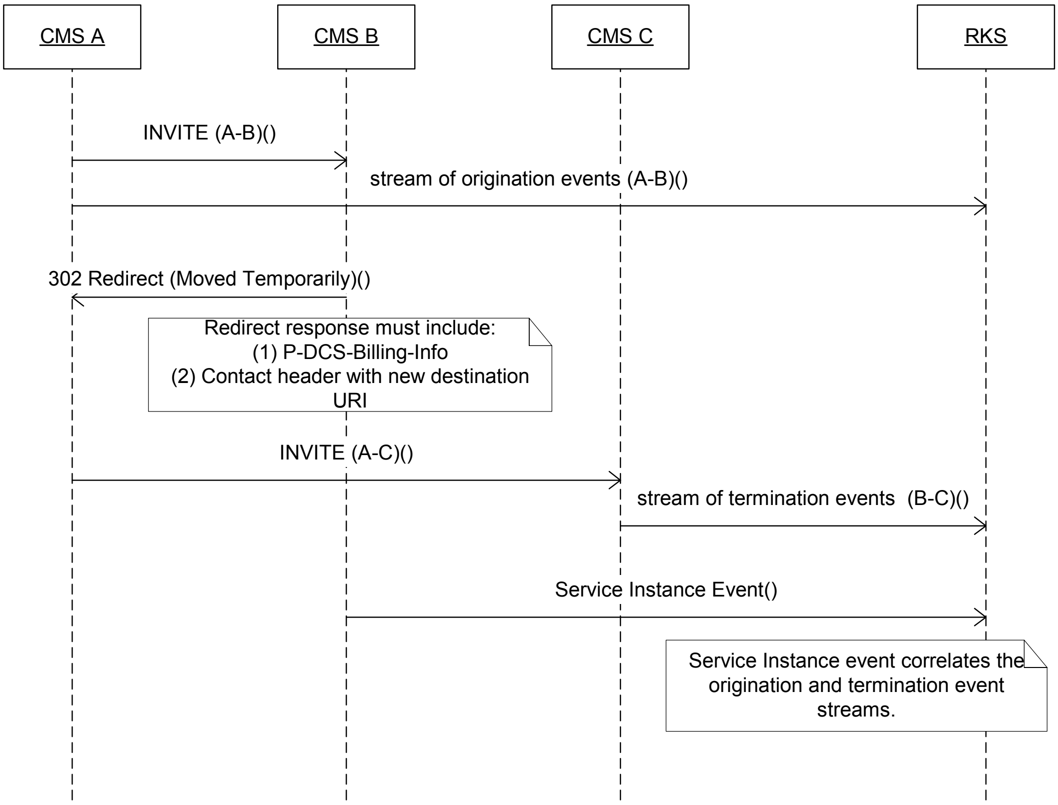
CMS A
CMS B
CMS C
RKS
INVITE (A-B)()
stream of origination events (A-B)()
302 Redirect (Moved Temporarily)()
Redirect response must include:
(1) P-DCS-Billing-Info
(2) Contact header with new destination URI
INVITE (A-C)()
stream of termination events (B-C)()
Service Instance Event()
Service Instance event correlates the origination and termination event streams.

Figure 8: Call Forwarding Support

8.1.1.6 CMS Behavior in Support of Inter-Domain Call Forwarding

When calling and forwarding is performed between IPCablecom domains (or within a domain when the parties are served by different Record Keeping Servers), it is required that all event messages related to a call segment be recorded by the Record Keeping Server within the domain of the party being charged for that call segment. It is therefore necessary for a CMS in each IPCablecom domain to remain on the signalling path for the duration of the resulting call in order to generate the necessary event streams.

There are three separate cases of inter-domain Call Forwarding. There may be three different domains (i.e., A, B, and C are each in a different IPCablecom domain). The first call may be intra-domain and the forwarding may be inter-domain (i.e., A and B in the same domain, C in a different domain). The first call may be inter-domain and the forwarding may be intra-domain (i.e., A in one domain, B and C in a different domain). We consider each case separately.

With A, B, and C each in different IPCablecom domains, CMS-A will generate an origination event message stream for a call from A to B, CMS-B will generate the termination event message stream for the call from A to B, and will also generate an origination event message stream for a call from B to C, and CMS-C will generate a termination event message stream for the call from B to C. CMS-B performs the forwarding operation by proxying an INVITE request to CMS-C, rather than returning a 3xx response to CMS-A. It will include a Record-Route header in the INVITE request, so that CMS-B stays in the signalling path for the duration of the call. The media however, flows directly from A to C.

With A and B in one domain, and C possibly in another domain, CMS-B will return a 302-Redirect response and generate a "service-instance" event for the call forwarding service. CMS-A will generate a new INVITE request to C, and the P-DCS-Billing-Info header will contain the information about the B-to-C leg. The resulting call will involve only CMS-A and CMS-C; CMS-A will generate an origination event stream and CMS-C will generate a termination event stream, just as in the intra-domain call forwarding case.

With A in one domain, and B and C in another domain, CMS-B will proxy the INVITE request to CMS-C and CMS-B will request that CMS-C generate the termination event message stream. CMS-B will generate a "service-instance" event, and will not remain on the signalling path. CMS-A will generate an origination stream for a call from A to B; CMS-C will generate the termination stream for the call from B to C.

When the forwarding is done in support of a Call-Forward-No-Answer (CFNA) service, it is necessary for the CMS to respond to the INVITE with a 3xx response. If the procedures described above would have resulted in the CMS proxying the INVITE to the new destination, the CMS instead generates a private URL (as described in Appendix III) for the Contact header in the 3xx response. All CMSes that had proxied an INVITE for this call that see the 3xx response also generate a private URL and update the Contact header of the 3xx response (as described in Appendix III). The end result is that the signalling path for the resulting call, and event streams generated for the resulting call (except for the service-instance for the CFNA), will be just as if the forwarding had occurred due to "Call-Forward-Unconditional" or "Call-Forward-Busy".

8.1.1.7 CMS Behavior in Support of REFER

Requirements for event generation for the REFER-based services (e.g., Call Transfer, Three-Way Calling) are based on the billing model of the PSTN. Consider the case in which party A calls party B and party B REFERs the call to party C. Assume that all three parties are in the same IPCablecom domain, and served by the same Record Keeping Server.

The REFER could be used to implement a call transfer or a three way call. In the call-transfer scenario, CMS B has been programmed to transfer calls destined for Party B to Party C. In the three-way call scenario, CMS B is instructed to add Party C onto the call. In both cases, a call is set up between A and C at the initiative of B.

We are concerned here with accounting information and with the event messages that will be sent to the Record Keeping Server. CMS B provides accounting information to CMS A. CMS A uses this information to create the INVITE message it sends to C.

Two call segments will be recorded and two event streams will be generated. The first call segment is from A to B, the second is from B to C. The event streams will be as follows: CMS A generates an origination event stream for the call segment from A to B, typically billed to party A. CMS C generates a termination event stream for the call segment from B to C, billed to party B. See diagram below.

Note: Refer to Section 8.4.3 for details and limitations as to the scope of the refer messages covered in this specification.

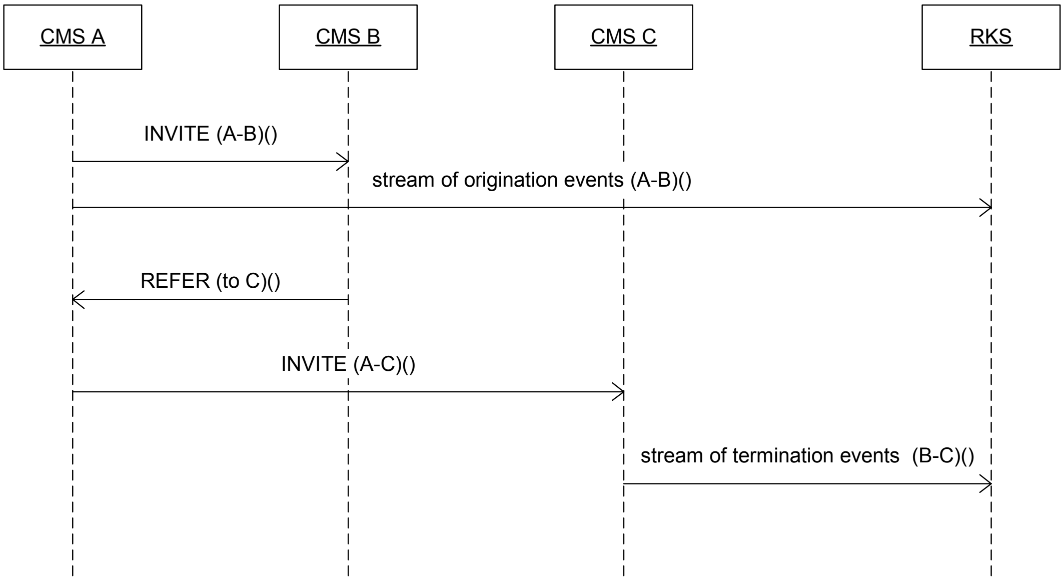
CMS A
CMS B
CMS C
RKS
INVITE (A-B)()
stream of origination events (A-B)()
REFER (to C)()
INVITE (A-C)()
stream of termination events (B-C)()

Figure 9: REFER Support

8.1.1.8 CMS Behavior in Support of Inter-Domain REFER

When REFER operations are performed between IPCablecom domains, it is required that all event messages related to a call segment be recorded within the domain of the party being charged for that call segment. It is therefore necessary for a CMS in each IPCablecom domain to remain on the signalling path for the duration of the resulting call in order to generate the necessary event streams.

8.1.2 Overview of Tandem Proxy

In theory, all CMSes may communicate with each other, both within a given domain, as well as between different domains. In practice, the need for scalable and manageable security and routing implies that one or more levels of indirection may be needed. The tandem proxy provides this indirection. Tandem proxies, which may be stateless proxies, act as call routers and aggregation points for security associations. They may also provide additional functions, such as signalling transformation gateways, signalling firewalls, etc. Depending on its role, a tandem proxy may remain in the call-signalling path for the duration of the call. In the alternative, a tandem proxy may complete its activities and allow the signalling to be passed directly between the CMSes that are managing the endpoints. Two specific types of tandem proxies, known as border proxies, have been defined:

> *Interior border proxy (IBP):* Proxy involved in intra-domain communication. Interior border proxies are used between two realms in the same domain. This type of proxy is optional and not required for signalling.
>
> *Exterior border proxy(EBP):* Proxy involved in inter-domain communication. Exterior border proxies are used to communicate between different domains. Every non-isolated domain has interfaces with one or more other domains via one or more EBPs.

See [36] for further information.

The tandem proxy has a limited role in so far as the CMS-CMS signalling is concerned. Its only concern is to ensure that messages for a given call are routed consistently throughout the call. The proxy simply follows the rules specified in Section 6.16 in order to ensure this.

8.2 CMS Retransmission, Reliability, and Recovery Strategies

SIP [8] defines a retransmission scheme based on two timer values, T1 and T2. The retransmission interval starts at T1 seconds, and is doubled, with each attempt (up to a limit of T2 seconds), up to some maximum number of retransmissions.

The CMS MUST implement an additional application level timer for each dialog (on a call-by-call basis). This timer is termed T3. Requirements on the conditions for setting T3, and actions on its expiration, are given in section 8.4.1. On expiration of this timer, the CMS aborts the current request and returns to a known idle state.

CMS_O sets T3 to T-setup on receipt of the first provisional response to an INVITE. CMS_O cancels T3 for all dialogs on receipt of a final response to any dialog.

CMS_T sets T3 to T-ringing on receipt of an INVITE request, and cancels T3 upon receipt of the final ACK message.

Default values for these timers (T-setup, and T-ringing) are given in Appendix I.

When the provisioned number of message retransmissions is exceeded for an INVITE without any response being received, the CMS MUST try a different CMS address, if available. If multiple CMSes are available, the procedures defined in [20], Section 4.3, MUST be used. When a provisioned number (which may be infinite) of CMS addresses have been tried, the CMS MUST clear the call and return to an idle state.

The behavior of Tandem Proxies depends on their role in the network and is not further specified in this document. Tandem Proxies follow standard SIP processing/retransmission.

8.3 CMS to CMS Routing

CMS to CMS routing is concerned with routing requests to their destination. CMSS supports routing of general SIP-URIs as well as telephone numbers in the form of tel-URIs as defined in Section 7.1. Routing of general SIP-URIs simply follows the procedures in Section 6.8.1. Routing of telephone numbers follows the procedures described here.

The originating CMS receives a telephone number from the originating user; this identifies the destination for the session. In order to route the session setup messages to the correct destination, the number needs to ultimately be resolved to the address of a terminating CMS.

The originating CMS generates a Request-URI based on the destination telephone number. If the destination telephone number may be a ported number, the originating CMS MUST either perform a local-number-portability (LNP) database query or it MUST send the request to another CMS (UA or Proxy) that can perform the LNP query[12]. In the latter case, the Request-URI MUST be a tel-URI.[13]

The CMS that performs the LNP query MUST generate a Request-URI containing either a SIP-URI or a tel-URI as a result of the query. For SIP-URI, the userinfo MUST be in the telephone-subscriber format and the URI MUST contain a “user=phone” parameter. Both tel-URI and SIP-URI MUST contain the "npdi" parameter and the "rn" parameter set to the result of the query as specified in Section 7.1.Furthermore, if the number was ported, the CMS that performed the LNP query MUST include the Location Routing Number (LRN) in the P-DCS-Billing-Info header in the first reliable response.

The mechanism by which the CMS routes the request to the terminating CMS is beyond the scope of this specification. This involves either routing based purely on the tel-URI, or conversion of the tel-URI into a SIP-URI. The SIP-URI format is preferred over keeping the tel-URI, and therefore the CMS SHOULD attempt to form a SIP-URI for the destination. These cases are discussed separately below.

8.3.1 Forming a SIP-URI from a tel-URI

The FQDN ("hostport") part of a SIP-URI identifies the intended recipient of the request. The recipient may or may not be the final destination for the request. The method by which the CMS determines the FQDN part of a SIP-URI formed from a tel-URI is outside the scope of this specification. For example, the CMS may have access to a database that can resolve a telephone number into the FQDN to which the request should be addressed. This may be only the next in a series of hops, or it may be the terminating CMS.

[12] Note that if the originating CMS does not perform the LNP query, then it must wait for the first reliable response before generating event messages that contain the Location Routing Number.

[13] By using the tel-URL format, intermediate proxies can perform LNP queries and modify the URL accordingly. Using a SIP-URI would not allow for this.

If the CMS is able to determine a FQDN, the CMS MUST include that FQDN in a SIP-URI in the Request-URI. Generation of the "userinfo" part of the SIP-URI is as described above. If it is unable to determine a FQDN, the CMS MAY leave the Request-URI as a tel-URI, and forward the request to a tandem proxy that is able to perform this translation.

The CMS SHOULD send the request directly to the FQDN identified in the SIP-URI following the procedures specified in 6.8.1. If the CMS is unable to send the request directly to the FQDN identified in the SIP-URI, it determines a tandem proxy to handle the request. Choice of a tandem proxy may be based on static configuration information, provisioning information, query of a routing function, or other methods. The CMS MUST check that it does not generate a request to a host identified in the Via header(s).

8.3.2 Routing a SIP-URI at Tandem CMSes

If a CMS receives a request with a SIP-URI in the Request-URI that identifies a destination other than the CMS, the CMS considers itself a tandem and attempts to send the request to its intended destination. The CMS MAY now operate as a stateless proxy as defined in Section 6.16. The Request-URI of the forwarded request SHOULD remain unchanged.

If a CMS receives a request identifying itself as the intended destination, performs the Request-URI translation, and determines it is not the CMS serving the destination, the CMS considers itself a tandem and attempts to send the request to its intended destination. The Request-URI MUST be rewritten to address the desired destination.

8.3.3 Routing based on tel-URI

Routing based on tel-URIs is performed hop-by-hop. The Request-URI may be rewritten as a result of Local Number Portability lookup, Freephone Number translation, etc. as described above.

8.4 CMS Procedures

The following subsections contain sample procedures for a basic call from an originating CMS to a terminating CMS, and for various mid-call changes that may be initiated by either endpoint. The call flow diagrams are informative only and are intended to provide guidance to developers. Specific processing required for IPCablecom CMSS-compliant systems, beyond that previously specified in Sections 7 and 6, is noted using the specification language of Section 1.2.

8.4.1 CMS Messages and Procedures for Basic Call Setup

The basic INVITE message sequence for a CMSS call setup includes the INVITE/183-Session-Progress/18x/200-OK/ACK exchange, an UPDATE/200-OK exchange, and one or two PRACK/200-OK message exchanges. These are shown in Figure 4 (section 5.6), and discussed in the following subsections. When it is known that the far-end is being alerted, the 18x will be a 180-Ringing. A 183-Session-Progress will be used instead of 180-Ringing when there is call progress but it is not known whether the called party is being alerted[14].

[14] For example, when interworking with MF trunks, it is not known whether in-band media is ringback or an announcement, and hence a 183-Session -Progress with early media would be used. Please refer to [35] for details.

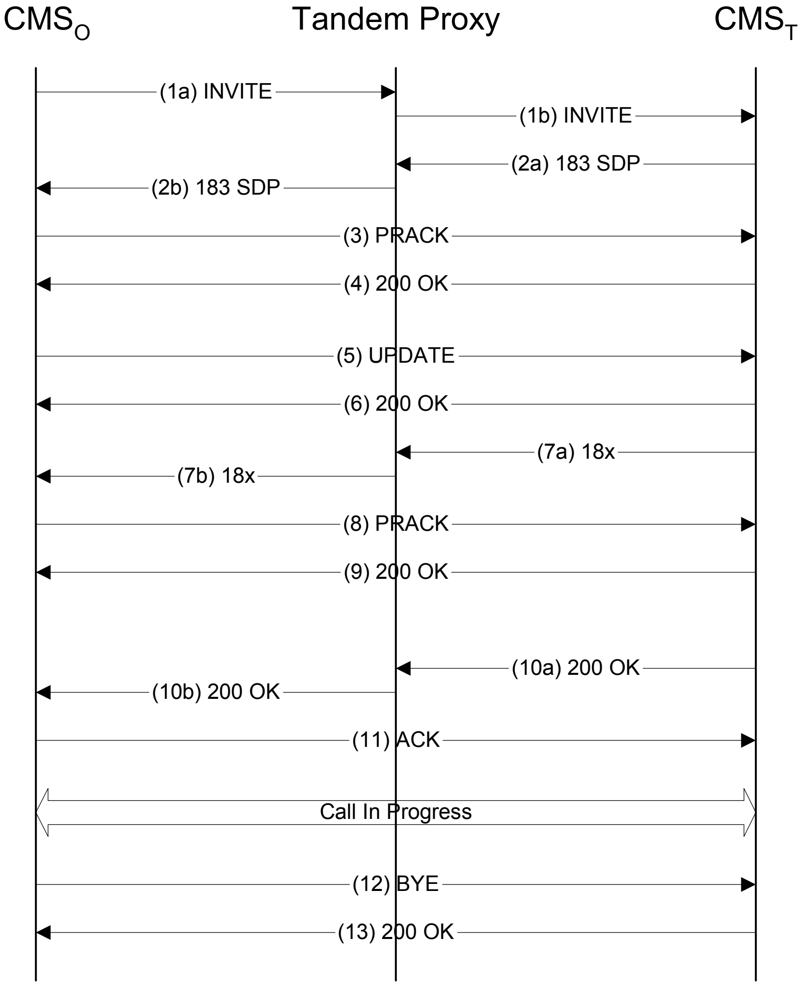

Figure 10: CMS Messages for Basic Call Setup

The following sections trace a basic call from origination to completion, and give the requirements for each message exchange. It therefore switches viewpoints from origination to termination and back. For procedures followed by CMS$_O$ (*i.e.*, originating a call) see sections 8.4.1.1, 8.4.1.3, and 8.4.1.6. For procedures followed by CMS$_T$ (*i.e.*, terminating a call) see sections 8.4.1.2, 8.4.1.4, and 8.4.1.5. A typical CMS implements the procedures in all of these subsections, while specialized CMSs implement only the portions needed in their application.

The behavior below also shows the procedures for call forwarding (unconditional and busy) and call forwarding (no answer).

8.4.1.1 CMS$_O$ initiating Invite

CMS$_O$ becomes aware of a call origination attempt when it receives a Notify message from the MTA. A Media Gateway Controller (MGC) becomes aware of a call origination attempt when it receives a Notify message from the media gateway, or an IP-IAM message from the signalling gateway. A CMS also becomes aware of a call origination attempt when it receives a REFER request from another CMS.

CMS$_O$ MUST check that the indicated line is authorized for outgoing service to the destination phone number.

The following call characteristics are determined by CMS$_O$, and used to generate the INVITE message:

URI of the destination endpoint, either as a tel-URI, or a SIP-URI as specified in section 8.3.

Originating endpoint identification: both the originating phone number (or, in general, a URI of the originator), and the originating account name.

The level of privacy requested by the call originator.

Call leg identification, in the form of SIP From:, To:, and Call-ID: header values.

Charging number, routing number, and location routing number as defined in [3].

Session Description (SDP) for the media flow(s) to the originating endpoint including QoS preconditions and all the acceptable choices of codecs (with appropriate rtpmap and bandwidth parameters).

This information is used to generate SIP headers as follows:

Header:	Requirements for CMS_O
Request URI	The URI of the destination endpoint. MUST conform to the rules for Request-URIs as given in Section 8.3.
P-Asserted-Identity	Originating account name and originating phone number (or URI in general) as described in Section 8.4.1.1.1.
Privacy	If calling number Privacy is requested, this header MUST contain the “id” and "critical" tag values.
Proxy-Require	If calling number Privacy is requested, this header MUST contain the option tag "Privacy".
From	Originating endpoint identification. MUST follow the requirements of section 6.20.20.
To	URI of the destination endpoint. MUST follow the requirements of section 6.20.39.
Call-ID	If Privacy is requested, the Call-ID MUST be generated as specified in section 6.20.8.
Contact	If Privacy is requested, the Contact MUST be generated as specified in section 6.20.10.
P-DCS-Billing-Info	Charging number, routing number, and location routing number as defined in [3]. If the INVITE is being generated as a result of a REFER request, see also section 8.4.3.2.
SDP	The SDP may be generated by interactions between the CMS and the endpoint beyond the scope of this specification. The CMS MUST add QoS preconditions to the SDP as needed in accordance with Section 7.4.

8.4.1.1.1 CMS_O Authentication and Authorization of Originator

Two different cases are considered here:

- a call originating on-net, and
- a call originating off-net.

Except as specified below, if the call originates on-net, CMS_O MUST provide a validated originating phone number for the active line on MTA_O in the P-Asserted-Identity header. CMS_O MUST also provide a validated originating calling name for the active line on MTA_O, unless the originator has requested calling name Privacy, in which case the display-name “Anonymous” MUST be used. See [2] for further detail.

CMS_O MAY permit a call to an emergency service or other special numbers even if provisioned information is not available to generate the calling number P-Asserted-Identity header.

If the call originates off-net and no calling party number is available, then the P-Asserted-Identity header MUST be omitted. Otherwise, the CMS_O MUST provide the calling party number received from the PSTN in the P-Asserted-Identity header. If calling name Privacy is requested, the display-name MUST be set to “Anonymous”.

8.4.1.1.2 Address Translation

CMS_O MUST resolve the destination number into either:

- the address of a destination endpoint served by this CMS; or
- the address of another CMS or proxy (subsequent routing procedures are described in section 8.3).

If it cannot resolve the destination number, it MUST consider the request to be in error.

8.4.1.1.3 IP Address Privacy Support

If the caller requested IP Address Privacy, then CMS_O MUST provide IP address Privacy through the use of an anonymizer as described in Section 9. Since there is no CMSS signalling support for IP address Privacy, CMS_O MUST either provide the IP address Privacy itself or route the request to an anonymizer. The anonymizer MUST provide IP address Privacy for media and MUST ensure that the SDP "c=" line points to a media anonymizer prior to crossing a trust boundary[15].

The anonymizer is described in more detail in Section 9.

8.4.1.1.4 INVITE message generation

If the destination endpoint is not served by CMS_O, CMS_O generates a SIP INVITE message and sends it to CMS_T, the CMS that manages the terminating endpoint.

If the INVITE is being generated in response to a REFER request, and the Refer-To URI contained a P-DCS-Laes header, CMS_O SHOULD remove that header element and perform the electronic surveillance itself. If CMS_O is unable to perform the requested electronic surveillance, then it MUST include the P-DCS-Laes header in the INVITE request. The semantics of P-DCS-Laes are further described in Section 7.7.

CMS_O MUST add the P-DCS-Billing-Info header, which is defined in 7.7. The semantics of the contents of P-DCS-Billing-Info are described in [3].

Finally, CMS_O MAY add a "Require P-DCS" header.

INVITE (CMS_O -> CMS_T) Header:	Requirements on CMS_O For Message Generation
INVITE URI SIP/2.0	As described in 8.3.
Via:	As described in 6.20.42.
Proxy-Require:	As described in 6.20.29.
Supported:	As described in 6.20.37.
Require:	MUST include "100rel", and "precondition".
Allow:	As defined in 6.20.5. MUST include "UPDATE"
P-Asserted-Identity:	As described above
Privacy:	As described above.
P-DCS-Billing-Info:	As defined in 7.7.
Max-Forwards:	As defined in 6.20.22.
From:	As defined in 6.20.20.
To:	As defined in 6.20.39.
Call-ID:	As defined in 6.20.8.
Cseq:	As defined in 6.20.16.
Contact:	As defined in 6.20.10.
Content-Type::	MUST be present and MUST contain "application/SDP".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= m= a=	c= line MAY be modified in support of IP address Privacy. a= line MUST be present and MUST indicate mandatory send and receive precondition as described in Section 7.4.

CMS_O MUST accept a 100-Trying message as described in the following table.

[15] Note that if the terminating endpoint is an NCS MTA then a trust boundary will be crossed no later than between CMS_T and MTA_T. For a PSTN gateway, a trust boundary may not be crossed.

100-Trying (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 100 Trying	As described in 6.13.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On receipt of a 100-Trying provisional response, the transaction timer (T3) for this exchange MUST be set to T-setup. The default value of (T-setup) is given in Appendix I. On expiration of T3, CMS_O clears the call attempt, and sends a CANCEL message to CMS_T with the same values of Request-URI, From, To, and Call-ID for this call attempt, as specified in Section 8.4.1.9.

8.4.1.2 Invite from CMS_O arrives at CMS_T

CMS_T MUST resolve the destination number from the Request-URI into either:

- the address of a destination endpoint served by this CMS; or
- the address of another CMS (subsequent routing procedures are described in section 8.3).

If it cannot resolve the destination number, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code. 404-Not-Found is the recommended error code.

If, by following the procedure described above, CMS_T determined it serves the destination endpoint, processing continues as specified in this section.

CMS_T MUST determine the local endpoint being addressed by this call. CMS_T MUST check to see if this endpoint is authorized to receive this call. If translation or authorization fails, CMS_T MUST return an appropriate 4xx, 5xx, 6xx error code. 404-Not-Found and 403-Forbidden respectively are recommended error codes.

On receiving the INVITE message, CMS_T MUST start the transaction timer (T3) with value T-ringing. The default value of (T-ringing) is given in Appendix I. Timer T3 is canceled by receipt of an ACK message acknowledging the final response from CMS_T. On expiration of timer T3, CMS_T MUST send a 408-Request-Timeout or 302-Redirect response (for call-forwarding-no-answer service) to CMS_O.

CMS_T determines, possibly by communicating with the endpoint, whether it will accept the call, forward to another destination, or return an error. The mechanism by which this is done by CMS_T is outside the scope of this specification.

The following subsections give the detailed procedures for each of the possible cases.

- If CMS_T determines that the endpoint is able to accept the incoming call request, then the procedures in Section 8.4.1.2.1 are followed.
- If CMS_T determined that the call is to be forwarded (either through a provisioned or temporary call-forward-unconditional, or because of a provisioned or temporary call-forward-busy and the line is currently busy), and the RKS-Group-ID of CMS_T is equal to the RKS-Group-ID of CMS_O (as contained in the P-DCS-Billing-Info header in the INVITE request), then CMS_T MAY return a 3xx response to CMS_O through the procedures of Section 8.4.1.2.2.
- If, on the other hand, CMS_T determines that the call is to be forwarded but the special conditions given in the previous paragraph are not met, or if CMS_T does not want to use the optimized procedure above, then the procedures of Section 8.4.1.2.3 MUST be followed to propagate the INVITE request.
- If CMS_T determines the endpoint is not available to accept the call, or if the endpoint returns an error, an appropriate SIP error code is returned to CMS_O. 486-Busy (if the user is already on another call and is not able to take a new call) and 480-Temporarily-Unavailable (otherwise) are recommended error codes. The procedures of Section 8.4.1.2.4 MUST be followed.

8.4.1.2.1 CMS_T Sending 183-Session-Progress Status Response

If the destination endpoint is able to accept the call, CMS_T sends the 183-Session-Progress provisional response reliably (see Section 7.2) to CMS_O.

The following call characteristics are determined by CMS_T and are used to generate the 183-Session-Progress response:

Contact address for direct CMS-CMS signalling messages;

Session Description (SDP) for the media flow(s) to the destination endpoint. This SDP includes all the required fields for precondition as defined in Section 7.4, as well as the choice of codecs (with appropriate rtpmap and bandwidth parameters) that are acceptable to the destination endpoint.

The response's session description MUST indicate a set of codecs that the destination endpoint is willing to support.

If the terminating user subscribes to calling name delivery, CMS_T checks the INVITE for a P-Asserted-Identity header. If the P-Asserted-Identity header does not contain a display-name, but the P-Asserted-Identity does contain a telephone number, CMS_T MUST obtain the calling name by querying a CNAM database by means outside the scope of this document (*e.g.*, by use of TCAP over ISTP, an HTTP query, etc.).

CMS_T checks for an outstanding lawfully authorized surveillance order for the terminating subscriber. If such an order exists, CMS_T includes this information in the authorization for Quality of Service. If the terminating equipment is unable to perform the required surveillance (*e.g.*, if the destination is a voicemail server), CMS_T MUST include a P-DCS-Laes header in the 183-Session-Progress response to CMS_O requesting it to perform the surveillance. The P-DCS-Laes header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call's event messages, and MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be intercepted, and MUST include a random string as specified in [2] for use as a security key between the Delivery Functions.

If the terminating endpoint is an MTA, CMS_T uses the information in the SDP description, the electronic surveillance indication, and the P-DCS-Billing-Info header values to signal the terminating Gate Controller (GC_T) to send a GATE-SET command defining the envelope of the authorized QoS parameters to the terminating CMTS ($CMTS_T$).

CMS_T MUST check to see if the called party has requested IP address Privacy. If IP address Privacy has been requested, then CMS_T MUST provide IP address Privacy through the use of an anonymizer. The anonymizer MUST ensure IP address Privacy for both signalling and media. At a minimum, CMS_T MUST ensure the SDP "c=" line points to the anonymizer prior to crossing a trust boundary[16]. CMS_T MUST also ensure that signalling messages crossing a trust boundary will not reveal any IP address information for the endpoint, (*e.g.*, the Contact header would have to point to an anonymizer). Please refer to Section 9 for additional detail on anonymizers.

CMS_T MUST include a P-DCS-Billing-Info header in the response. This header MUST contain the Billing-Correlation-ID, Financial-Entity-ID, and RKS-Group-ID for the termination event message stream for the call leg between CMS_O and CMS_T. If CMS_T performed the LNP query for this call, the P-DCS-Billing-Info header MUST include the results of that query. The semantics of the contents of P-DCS-Billing-Info are described in [3].

The 183-Session-Progress provisional response sent by CMS_T to CMS_O MUST be as follows:

183-Session-Progress (CMS_T -> CMS_O) Header:	**Requirements On CMS_T for Message Generation**
SIP/2.0 183 Session Progress	Status line with status code 183 MUST be present.
Via:	As described in 6.13.
Require:	As defined in 6.20.32. MUST include "100rel" as described in 7.2.
Supported:	As described in 6.20.37.
P-DCS-Billing-Info	As described above, and in 7.7
From:	As described in 6.13.
To:	
Call-ID:	
Cseq:	
Contact:	As defined in 6.20.10.
Rseq:	As defined in 7.2.
Content-Type:	MUST be present and MUST contain "application/SDP".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.

[16] Note that if the originating endpoint is an NCS MTA then a trust boundary will be crossed no later than between CMS_O and MTA_O. If the originating endpoint is a PSTN gateway, a trust boundary may not be crossed.

183-Session-Progress (CMS$_T$ -> CMS$_O$) Header:	Requirements On CMS$_T$ for Message Generation
v= o= s= c= b= t= a= m=	SDP MUST be present. SDP description of media streams acceptable to the destination endpoint. The a= line MUST be present, MUST indicate mandatory send and receive preconditions, and MUST request confirmation, as described in 7.4.

8.4.1.2.2 CMS$_T$ Sending 3xx REDIRECT Status Response

Procedures in this section are invoked when CMS$_T$ determines (by methods beyond the scope of this specification) that the incoming call is to be forwarded. CMS$_T$ MUST verify that the called party is a subscriber to the Call Forwarding service. If not, CMS$_T$ MUST send a 480 Temporarily Unavailable error response to CMS$_O$.

Further, procedures in this section are invoked only when CMS$_T$ determines that it is not necessary to remain in the signalling path for this call for event message generation [3], by the conditions stated in section 8.4.1.2.

CMS$_T$ MUST check for an outstanding lawfully authorized surveillance order for the terminating subscriber. If found, CMS$_T$ MUST include a P-DCS-Laes header in the 3xx-Redirect response. The P-DCS-Laes header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call's event messages, MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be delivered, and MUST include a random string for use as a security key between the Delivery Functions.

CMS$_T$ MUST include a P-DCS-Billing-Info header with the information about the call-leg from CMS$_T$ in the response to the new destination. This header MUST include the Billing-Correlation-ID assigned by CMS$_T$, the calling number (same as the called number of the INVITE request), the called number (the new destination for the call), and the charge number (typically the same as the called number of the INVITE request). The semantics of the parameter values for P-DCS-Billing-Info are described in [3].

CMS$_T$ MUST send the following 3xx-Redirect response to CMS$_O$.

302-Redirect (CMS$_T$ -> CMS$_O$) Header:	Requirements On CMS$_T$ for Message Generation Requirements On CMS$_O$ for Message Checking
SIP/2.0 302 Moved Temporarily	Status line with status code 3xx MUST be present.
Via:	As described in 6.13.
P-DCS-Billing-Info:	MUST be present, as described above, and in 7.7.
P-DCS-Laes:	MAY be present, as described above and in 7.7.
From:	As described in 6.13.
To:	
Call-ID:	
Cseq:	
Contact:	MUST be inserted by CMS$_T$ and carry the new destination information. It MUST be a valid URI. If the new destination is a telephone number, then the format of the URI MUST be a tel-URI where the URI contains a telephone number as defined in 7.1.
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this response MUST cease on receipt of the following ACK message.

ACK (CMS_O -> CMS_T) Header:	Requirements On CMS_T for Message Checking
ACK URI SIP/2.0	As described in 6.17.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On receipt of the ACK message, CMS_T MUST cancel the transaction timer T3.

8.4.1.2.3 CMS_T Sending INVITE Request to CMS_F

Procedures in this section are invoked when CMS_T determines (by methods beyond the scope of this specification) that the incoming call is to be forwarded. CMS_T MUST verify that the called party is a subscriber to the Call Forwarding service. If not, CMS_T MUST send a 480 Temporarily Unavailable error response to CMS_O.

Further, procedures in this section are invoked only when CMS_T determines that the special conditions permitting optimized behavior, given in Section 8.4.1.2, are not present.

If CMS_T is able to determine, by methods beyond the scope of this specification, that the forwarded destination is served by a CMS (CMS_F) which is part of the same RKS-Group-ID, then CMS_T forwards the INVITE to CMS_F. Otherwise, i.e. when CMS_F is part of a different RKS-Group-ID or CMS_T is unable to determine the RKS-Group-ID of CMS_F, CMS_T MUST include a Record-Route entry in the INVITE request, remain on the signalling path for the call, and generate its stream of event messages for the call.

CMS_T MUST check for an outstanding lawfully authorized surveillance order for the forwarding subscriber. If found, CMS_T MUST include a P-DCS-Laes header in the INVITE request. The P-DCS-Laes header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call's event messages, MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be delivered, and MUST include a random string for use as a security key between the Delivery Functions.

CMS_T MUST replace the P-DCS-Billing-Info header in the request with the proper information regarding the new leg of the forwarded call, so that it can be charged to the forwarding party. Further information on the semantics of the parameter values for P-DCS-Billing-Info are described in [3].

Finally, CMS_T MUST decrement the Max-Forwards value received by one, and include the resulting Max-Forwards in the generated INVITE.

The rest of the INVITE message MUST be identical to that which was received by CMS_T, as prescribed by the proxy behavior specified in Section 6.

The format of the resulting INVITE message as sent by CMS_T to CMS_F, and the associated requirements on the header fields are as follows:

INVITE (CMS_T -> CMS_F) Header:	Additional Requirements for Message Generation
INVITE URI SIP/2.0	As described above
Via:	As described in 6.16 and 6.20.42.
Record-Route:	MAY be present, as described above.
Require:	As described in 6.16
Proxy-Require:	As described in 6.16
Supported:	As described in 6.16
Allow:	As described in 6.16
P-Asserted-Identity:	As described in 6.16
Privacy:	As described in 6.16
P-DCS-Billing-Info:	As described above
P-DCS-Laes:	As described above
P-DCS-Redirect:	As described above
Max-Forwards:	As described above
From:	As described in 6.16
To:	As described in 6.16
Call-ID:	As described in 6.16
CSeq:	As described in 6.16
Contact:	As described in 6.16
Content-Type:	As described in 6.16
Content-Length:	As described in 6.16
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	As described in 6.16

The behavior and processing of the INVITE at CMS_F is identical to that described in section 8.4.1.2, with CMS_F taking the role identified in that section as CMS_T.

CMS_T MUST handle all the responses to this INVITE request, and process as required by Section 6.16, except as follows:

If CMS_T receives a 3xx response to the INVITE, it MUST re-write the Contact header value with a private URL (as defined in Section 7.7), with the following information encoded in the userinfo portion: 1) the value of the Contact header received in the 3xx response; 2) the contents of the P-DCS-Billing-Info headers in the 3xx response; and 3) the value of Billing-Correlation-ID assigned for the event message stream(s) generated by CMS_T.

CMS_T MUST remove the P-DCS-Billing-Info header in the first reliable response, and replace it with a P-DCS-Billing-Info header containing the Billing-Correlation-ID and Financial-Entity-ID for the terminating event stream of the call-leg from CMS_O to CMS_T.

If CMS_T receives a REFER request as part of a dialog created by this INVITE, it MUST re-write the Refer-To header value with a private URL (as defined in Section 7.7), with the following information encoded in the userinfo portion: 1) the value of the Refer-To header received in the REFER request; 2) the contents of the P-DCS-Billing-Info headers in the REFER request; and 3) the value of Billing-Correlation-ID assigned for the event message stream(s) generated by CMS_T.

8.4.1.2.4 CMS_T Sending Other Status Response to INVITE request

A final error response (4xx, 5xx, or 6xx) MUST be sent per Section 6. This includes, but is not limited to, 486-Busy Here. The error response MUST be formatted as follows.

Error (CMS_T -> CMS_O) Header:	Requirements On CMS_T for Message Generation Requirements On CMS_O for Message Checking
SIP/2.0 xxx	Status line header MUST be present. It MUST include the SIP version number and the three digit status code.
Via:	As described in 6.13.
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this response MUST cease on receipt of the following ACK.

ACK (CMS_O -> CMS_T) Header:	Requirements On CMS_T for Message Checking
ACK URI SIP/2.0	As described in 6.17.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

8.4.1.3 CMS_O Receives Initial Status Response

In response to the initial INVITE request, CMS_O MUST be prepared to receive a 183-Session-Progress provisional response (in a normal call establishment), a 3xx-Redirect response (if the call was forwarded), or a 4xx, 5xx, or 6xx error response (error cases, such as busy). Final responses, including 4xx, 5xx, and 6xx, are described in section 8.4.1.8.3.

8.4.1.3.1 CMS_O handling of 183-Session-Progress Response

The 183-Session-Progress provisional response received by CMS_O MUST be checked to ensure that it conforms to the following format:

183-Session-Progress (CMS$_T$ -> CMS$_O$) Header:	**Requirements On CMS$_O$ for Message Checking**
SIP/2.0 183 Session Progress	Status line with status code 183 MUST be present.
Via:	As described in 6.13.
Require:	As defined in 6.20.32 and 7.2. Note that the option tag "100rel" MUST be present.
Supported:	As described in 6.20.37.
P-DCS-Billing-Info:	MUST be present as defined in Section 7.7.
From:	As described in 6.13.
To:	
Call-ID:	
CSeq:	
Contact:	As defined in 6.20.10.
Rseq:	As defined in 7.2.
Content-Type:	MUST be present. MUST contain "application/SDP". The response to the INVITE must contain the SDP description of the media stream to be sent to the destination endpoint.
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	SDP MUST be present. SDP description of media streams acceptable to the destination endpoint. a= line MUST be present, MUST indicate mandatory send and receive preconditions, and MUST request confirmation, as described in 7.4.

If the received provisional response does not conform to the above format, then CMS$_O$ MAY ignore the message. Otherwise, CMS$_O$ checks for an outstanding lawfully authorized surveillance order for the originating subscriber, and, if present, includes this information in the Authorization for Quality of Service or signals this information to the device performing the intercept (*e.g.*, a Media Gateway).

If the P-DCS-Laes header is present in the 183-Session-Progress response (indicating surveillance is required on the terminating subscriber, but that the terminating equipment is unable to perform that function), CMS$_O$ MUST include this information in the Authorization for Quality of Service, or MUST signal this information to the device performing the intercept (*e.g.*, a Media Gateway).

If the 183-Session-Progress provisional response was the first response to the sent INVITE, CMS$_O$ MUST set the transaction timer (T3) for this exchange to T-setup. The default value of (T-setup) is given in Appendix I. On expiration of T3, CMS$_O$ MUST clear the call attempt and send a CANCEL message to CMS$_T$ with the same values of Request-URI, From, To, and Call-ID for this call attempt, as shown in 8.4.1.9.

CMS$_O$ stores the Contact header and the SDP description for the duration of the call.

If CMS$_O$ did not perform the LNP query when sending the INVITE, CMS$_O$ MUST check the P-DCS-Billing-Info header for the presence of a Location Routing Number. If present, the Location Routing Number MUST be used for event messaging.

CMS$_O$ MUST send a PRACK to acknowledge receipt of the reliable 183-Session-Progress. The PRACK message MUST be sent directly to the address specified in the Contact header of the received 183-Session-Progress.

If the originator's SDP is different from that in the initial INVITE, an SDP body MUST be included in the PRACK message. Otherwise, an SDP body SHOULD NOT be included.

PRACK (CMS_O -> CMS_T) Header:	Requirements On CMS_O for message generation
PRACK URI SIP/2.0	As described in 7.2.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22
From:	As described in 6.12.
To:	
Call-ID:	
Cseq:	
Rack:	As defined in 7.2.
Content-Type:	MUST be present if a body is included.
Content-Length:	As described in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	MUST be present if there are changes, SHOULD NOT be present otherwise. Contains the SDP description as modified by CMS_O after processing the SDP returned by CMS_T.

The 200-OK response to the PRACK request MUST be as follows. If an SDP offer was included in the PRACK message, then an SDP body MUST be included in the 200-OK response to it. Otherwise, an SDP body SHOULD NOT be included:

200-OK (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Type:	MAY be present.
Content-Length:	As described in Section 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	If an SDP offer was present in the PRACK, then an answering SDP MUST be present in the 200-OK response, as described in 7.2. SDP SHOULD NOT be present otherwise.

Following receipt of the 183-Session-Progress response, or following receipt of the 200-OK response to the PRACK if an SDP is included in the PRACK message, CMS_O tells the originating endpoint device to attempt to reserve access network resources based on the most recently received SDP parameters.

CMS_O MUST apply operator defined policy if any to the list of codecs specified in the SDP payload to authorize maximum resources that can be used during this call at the originating CMTS ($CMTS_O$). The remaining codec information is used in a GATE-SET command to the originating CMTS it defines the envelope of the authorized QoS parameters. The GATE-SET message also includes any required electronic surveillance information.

After successful completion of the resource reservation, CMS_O MUST send an UPDATE message to CMS_T. This informs the destination that resources are available and that it may proceed to alert the end user (assuming that the terminating side successfully reserved resources). The UPDATE message MUST be formatted as follows:

UPDATE (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
UPDATE URI SIP/2.0	As described in 7.4.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22
From:	As described in 6.12.2.
To:	
Call-ID:	
Cseq:	
Content-Type:	MUST be present, and MUST be as defined in 7.4.
Content-Length:	As described in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	SDP MUST be present as defined in 7.4. Contains the SDP description as modified after processing the SDP returned by the terminating endpoint and with status of the QoS precondition, as described in 7.4.

Retransmissions of this request MUST cease on receipt of a 200-OK.

The originating endpoint must be prepared to receive bearer channel packets once CMS_O has transmitted the UPDATE.

The 200-OK response to the UPDATE MUST be formatted as follows:

200-OK (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Type:	As described in 7.3.
Content-Length:	As described in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	MUST be present. Contains the SDP description response to the QoS confirmation sent in the UPDATE request.

If the resource reservation fails, CMS_O SHOULD send a CANCEL to CMS_T:

CANCEL (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
CANCEL URI SIP/2.0	As described in 6.9.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22.
From:	As described in 6.9.
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this request MUST cease on receipt of a final response to the CANCEL. Normally, the final response will be a 200-OK[17] which MUST be formatted as follows:

200-OK (CMS_T -> CMS_o) Header:	Requirements On CMS_o for Message Checking
SIP/2.0 200 OK	As defined in 6.9.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

8.4.1.3.2 302-Redirect Status Response Handling at CMS_O

Note that the procedures defined in this section are identical to the procedures defined in section 8.4.1.8.2.

CMS_O MUST check that the headers of a received 302-Redirect response are as follows:

302-Redirect (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 302 Moved Temporarily	Status line header MUST be present. It MUST include the SIP version number and the three digit status code.
Via:	As described in 6.20.42.
P-DCS-Billing-Info:	MUST be present as described in 7.7.
P-DCS-Laes:	MAY be present.
From:	As described in 6.13.
To:	
Call-ID:	
Cseq:	
Contact:	As defined in 6.20.10.
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If a received 302-Redirect does not meet the above requirements, CMS_O MAY ignore the message. Otherwise, CMS_O MUST match the 302-Redirect response to the corresponding INVITE. CMS_O MUST return an ACK to CMS_T, using the Request-URI from the earlier INVITE.

ACK (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
ACK URI SIP/2.0	As described in 6.17.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Following transmission of the ACK message to CMS_T, CMS_O MUST issue an INVITE request to the party indicated in the Contact header in the 3xx response. CMS_O MUST generate a Request-URI from the Contact header value as described in 8.3.

If the destination endpoint is not served by CMS_O, CMS_O generates an INVITE message and sends it to CMS_F, the CMS that manages the forwarded-to destination.

[17] A 481 (Call Leg/Transaction Does Not Exist) would be returned if the INVITE transaction had already completed (successfully or not) at the terminating side.

If a P-DCS-Laes header is present in the 3xx response, CMS$_O$ MUST include that header unchanged in the reissued INVITE. CMS$_O$ MUST also include a P-DCS-Redirect header containing the original dialed number, the new destination number, and the number of redirections that have occurred.

CMS$_O$ MUST copy the contents of the P-DCS-Billing-Info header in the 3xx response to a P-DCS-Billing-Info header in the new INVITE.

The rest of the INVITE message SHOULD be identical to that which was sent to CMS$_T$, with the exception of an updated Cseq value.

The format of the resulting INVITE message as sent by CMS$_O$ to CMS$_F$, and the associated requirements on the header fields are as follows:

INVITE (CMS$_O$ -> CMS$_F$) Header:	**Additional Requirements for Message Generation**
INVITE URI SIP/2.0	As described above
Via:	As described in 8.4.1.1.
Require:	As described in 8.4.1.1.
Proxy-Require:	As described in 8.4.1.1.
Supported:	As described in 8.4.1.1.
P-Asserted-Identity:	As described in 8.4.1.1.
Privacy:	As described in 8.4.1.1.
P-DCS-Billing-Info:	As described above
P-DCS-Laes:	As described above
P-DCS-Redirect:	As described above
Max-Forwards:	As defined in 6.20.22
From:	As described in 8.4.1.1.
To:	As described in 8.4.1.1.
Call-ID:	As described in 8.4.1.1.
CSeq:	As described in 8.4.1.1.
Contact:	As described in 8.4.1.1.
Content-Type:	As described in 8.4.1.1.
Content-Length:	As described in 8.4.1.1.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	As described in 8.4.1.1. c= line MAY be modified in support of IP address Privacy.

On receipt of this INVITE message, CMS$_F$ uses the combination of From, To, Call-ID, and Request-URI as described in Section 6 to recognize this as a new call and not a retransmission from a previous call.

The behavior and processing of the INVITE at CMS$_F$ is identical to that described in section 8.4.1.2.

CMS$_O$ MUST accept a 100-Trying message as described in the following table:

100-Trying (CMS$_F$ -> CMS$_O$) Header:	**Requirements On CMS$_O$ for Message Checking**
SIP/2.0 100 Trying	As described in 6.13.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On receipt of a 100-Trying provisional response, the transaction timer (T3) for this exchange CMS$_O$ MAY ignore the message. Otherwise, it MUST be set to T-setup. The default value of (T-setup) is given in 0. On expiration of T3,

CMS_O clears the call attempt and sends a CANCEL message to CMS_T with the same values of Request-URI, From, To, and Call-ID for this call attempt, as specified in Section 8.4.1.9.

Processing of responses to this INVITE request is as given in Section 8.4.1.3.

8.4.1.4 CMS_T Receiving Acknowledgement of 183-Session-Progress

After sending the 183-Session-Progress response to the INVITE, CMS_T MUST wait for the PRACK message acknowledging the Session-Progress. The PRACK message headers MUST be checked as follows:

PRACK (CMS_O -> CMS_T) Header:	Requirements On CMS_T for Message Checking
PRACK URI SIP/2.0	As described in 7.2.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22
From:	As described in 6.12.2.
To:	
Call-ID:	
Cseq:	
Rack:	As described in 7.2.
Content-Type:	MAY be present.
Content-Length:	As described in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	MAY be present. Contains the SDP description as modified by CMS_O after processing the SDP returned by CMS_T.

If the PRACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code. Otherwise,, CMS_T MUST respond with a 200-OK. The 200-OK response MUST be formatted as follows.

200-OK (CMS_T -> CMS_O) Header:	Requirements On CMS_T for Message Generation
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Type:	MAY be present.
Content-Length:	As described in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	If an SDP offer was present in the PRACK, then answering SDP MUST be present in the 200-OK response, as described in 7.2. SDP SHOULD NOT be present otherwise.

Following receipt of the PRACK message, CMS_T instructs the endpoint to reserve network resources. The resource reservation request is based on the SDP parameters received in the PRACK request (if provided), otherwise it is based on the SDP parameters received in the INVITE request. Note that in both cases interactions with the terminating endpoint may lead to only a subset of the SDP parameters actually being accepted and reserved.

After the originating endpoint successfully completes the resource reservation, CMS_O sends an UPDATE message to CMS_T. This informs CMS_T that resources are available at the originator and that it may proceed and alert the end user (assuming resources were reserved successfully at the terminating end). CMS_T MUST check and verify the UPDATE message as follows.

UPDATE (CMS_O -> CMS_T) Header:	**Requirements On CMS_T for Message Checking**
UPDATE URI SIP/2.0	As described in 7.4.
Max-Forwards:	As defined in 6.20.22
Via:	As described in 6.20.42.
From:	As described in 6.12.2.
To:	
Call-ID:	
Cseq:	
Content-Type:	MUST be present. MUST be as defined in 7.4
Content-Length:	As described in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	SDP MUST be present as defined in 7.4. Contains the SDP description as modified after processing the SDP returned by the terminating endpoint and with status of the QoS precondition, as described in 7.4.

If the UPDATE message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code. Otherwise, CMS_T MUST respond to the UPDATE request with a 200-OK, unless an error has occurred as described in 7.3. The 200-OK response to the UPDATE MUST be formatted as follows.

200-OK (CMS_T -> CMS_O) Header:	**Requirements On CMS_T for Message Generation**
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On receipt of the UPDATE message, and the terminating endpoint having successfully reserved the network resources needed for its media flows, CMS_T continues with the alerting procedures of section 8.4.1.5.

If the resource reservation fails, CMS_T MUST send a 580-Precondition-Failure response to CMS_O:

580-Precondition-Failure (CMS_T -> CMS_O) Header:	**Requirements On CMS_T For Message Generation**
SIP/2.0 580 precondition failure	Status line header MUST be present. It MUST include the SIP version number and the three digit status code.
Via:	As described in 6.20.42.
From:	As described in 6.13.
To:	
Call-ID:	
Cseq:	
Content-Type:	MUST be present, and MUST be as defined in 7.4.
Content-Length:	As described in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	SDP MUST be present as defined in 7.4. MUST contain the status of the QoS precondition.

Retransmissions of this response MUST cease on receipt of an ACK.

8.4.1.5 CMS_T sends 180-Ringing or 183-Session-Progress

Once CMS_T receives the UPDATE message, and any applicable resource reservation for the terminating endpoint has completed successfully, CMS_T MUST send a provisional or final response to CMS_O, through the proxy path taken by the initial INVITE request.

When the terminating endpoint is on-net, CMS_T determines, by mechanisms beyond the scope of this specification, whether alerting is necessary. If alerting of the destination user is necessary, CMS_T sends a 180-Ringing response. Otherwise, CMS_T sends a final response as described in section 8.4.1.7.

When the terminating endpoint is off-net, CMS_T waits for an off-net indication to determine what response to generate, as described in [31]. If the response from the PSTN indicates that alerting is being performed, CMS_T generates a 180-Ringing response. If the response indicated progress or in-band information available, the CMS_T generates a 183-Session-Progress instead and ensures that the terminating endpoint can send media to the originating side. A 181 Call is Being Forwarded or 182 Queued could also be generated as described in [31]. In all other cases, CMS_T sends a final response as described in section 8.4.1.7.

The 180-Ringing or 183-Session-Progress message MUST be formatted as follows:

180 Ringing / 183 Session Progress: (CMS_T -> CMS_O) Header:	**Requirements on CMS_T For Message Generation**
SIP/2.0 180 Ringing/183 Session Progress	Status line with status code 180 or 183 MUST be present.
Via:	As described in 6.13.
Require:	As defined in 6.20.32. MUST include "100rel"
From:	As described in 6.13.
To:	
Call-ID:	
Contact:	As defined in 6.20.10.
Cseq:	As described in 6.13.
Rseq:	As defined in 7.2.
Content-Length:	MUST be present if the transport protocol is stream-based (TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this response MUST cease on receipt of PRACK.

After sending the 180-Ringing or 183-Session-Progress response to the INVITE, CMS_T MUST wait for the PRACK message acknowledging the response. The PRACK message headers MUST be checked as follows:

PRACK (CMS_O -> CMS_T) Header:	Requirements On CMS_T for Message Checking
PRACK URI SIP/2.0	As described in 7.2.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22.
From:	As described in 6.12.2.
To:	
Call-ID:	
Cseq:	
Rack:	As described in 7.2.
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the PRACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code. Otherwise, on receipt of this PRACK, CMS_T MUST respond with a 200-OK. The 200-OK response MUST be formatted as follows.

200-OK (CMS_T -> CMS_O) Header:	Requirements On CMS_T for Message Generation
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

8.4.1.6 CMS_O receives 180-Ringing or 183-Session-Progress

After the originating endpoint has completed the resource reservation, and CMS_O has sent the UPDATE message to the destination CMS, CMS_O will receive one of: (1) a provisional response of 180-Ringing or 183-Session-Progress; (2) a final response of 200-OK; or (3) an error. This section covers the procedures for the provisional responses 180 and 183, and section 8.4.1.8 covers the procedures for the final responses. Handling of other responses by CMS_O, in particular 181, 182 and additional 183-Session-Progress responses with preconditions (as in Section 8.4.1.3.1) is OPTIONAL; however, CMS_O MUST NOT fail on receiving such responses.

CMS_O MUST verify the headers of the provisional response according to the following table:

180 or 183 Provisional Response (CMS_T -> CMS_O) Header:	Requirements On CMS_O For Message Checking
SIP/2.0 180 Ringing / 183 Session Progress	Status line with status code 180 or 183 MUST be present.
Require:	As defined in 6.20.32. MUST contain “100rel”
Via:	As described in 6.13.
From:	As described in 6.13.
To:	
Call-ID:	
Contact:	As described in 6.20.10.
Cseq:	As described in 6.13.
Rseq:	As defined in 7.2.
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the received provisional response does not conform to the above format, then CMS_O MAY ignore the message. Otherwise, the 180-Ringing response indicates to the originating CMS that the terminating party is being alerted and that local ringback SHOULD be generated. CMS_O, by methods outside the scope of this specification, informs the originating endpoint of the desired actions. Note that, in accordance with Section 6.13, the originating endpoint must be prepared to receive media based on the offer/answer exchange performed earlier. If media is received while generating local ringback, the originating endpoint SHOULD stop the local ringback tone[18].

The 183-Session-Progress response indicates to the originating CMS that the terminating party is providing some kind of unspecified early media, and hence local ringback SHOULD NOT be generated. CMS_O, by methods outside the scope of this specification, informs the originating endpoint of any desired actions.

CMS_O MUST acknowledge the 180/183 provisional response with a PRACK message:

PRACK (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
PRACK URI SIP/2.0	As described in 7.2.
Via:	As described in 6.20.42.
Max-Forwards	As defined in 6.20.22
From:	As described in 6.12.2.
To:	
Call-ID:	
Cseq:	
Rack:	As described in 7.2.
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this request MUST cease on receipt of a 200-OK. The 200-OK response MUST be as follows:

200-OK (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Contact:	As described in 6.20.10.
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

8.4.1.7 CMS_T Sending final Response

After the destination endpoint has successfully reserved resources, CMS_T has received the UPDATE message from CMS_O (indicating it had also successfully reserved resources), and the destination endpoint has completed whatever alerting procedures were required, CMS_T sends a final response. For a typical telephony service, this is indicated by the user 'going off-hook' and 'answering the phone', and means the endpoint is ready to begin media transfers. The case of a successful completion of a call is covered in section 8.4.1.7.1, and the various error cases are covered in section 8.4.1.7.2 and 8.4.1.7.3.

8.4.1.7.1 CMS_T sending 200-OK Final Response

Once CMS_T determines that the destination endpoint accepts the incoming call (*e.g.*, off-hook, or hook-flash, or by other methods beyond the scope of this specification), it MUST send a 200-OK final response to the originating CMS. The message sent by CMS_T to CMS_O MUST be formatted as follows:

200-OK (CMS_T -> CMS_O) Header:	Requirement
SIP/2.0 200 OK	As described in 6.13.
Via:	

[18] In NCS [5], this can for example be achieved by use of the "media start" event defined in the Line package.

200-OK (CMS$_T$ -> CMS$_O$) Header:	Requirement
From:	
To:	
Call-ID:	
CSeq:	
Contact:	As described in 6.20.10.
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On sending the 200-OK, CMS$_T$ MUST stop timer T3, tell the endpoint device to commit to resources that have been reserved for this call, and tell the endpoint device that it MAY begin sending bearer channel packets.

The terminating device SHOULD be prepared to receive bearer channel packets once it has sent a final response.

Retransmissions of this response MUST cease on receipt of ACK.

The ACK message, which is sent directly between CMS$_O$ and CMS$_T$ MUST be verified as follows:

ACK (CMS$_O$ -> CMS$_T$) Header:	Requirements On CMS$_T$ For Checking Message
ACK URI SIP/2.0	As described in 6.13.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the ACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.1.7.2 CMS$_T$ sending 3xx-Redirect Final Response

If the terminating endpoint wishes to forward the call (*e.g.*, if call-forwarding-no-answer is enabled), a final 3xx-Redirect status response MUST be sent by CMS$_T$, the contact header contains the new URI of the forwarded to destination. CMS$_T$ determines this by means beyond the scope of this specification.

CMS$_T$ MUST check for an outstanding lawfully authorized surveillance order for the terminating subscriber. If found, CMS$_T$ MUST include a P-DCS-Laes header in the 3xx-Redirect response. The P-DCS-Laes header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call's event messages, MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be delivered, and MUST include a random string as specified in [2] for use as a security key between the Delivery Functions.

Two different procedures are defined for handling the call forward case. In the first procedure, CMS$_T$ does not remain on the signalling path for the resulting call. In the second procedure, CMS$_T$ does remain on the signalling path for the resulting call. Use of the first procedure is OPTIONAL; however, its use requires certain conditions to be met as described below. If the first procedure is not used, the second procedure MUST be used.

In order to use the first procedure, the RKS-Group-ID of CMS$_O$ (as given in the P-DCS-Billing-Info header in the INVITE request) MUST be the same as the RKS-Group-ID of CMS$_{T.}$ In this procedure, CMS$_T$ MUST add a P-DCS-Billing-Info headers to the response to allow the new leg of the forwarded call to be charged to the terminating party. CMS$_T$ MUST include in this P-DCS-Billing-Info header the Correlation-ID and Financial-Entity-ID of CMS$_T$, the calling number (same as the called number of the INVITE request), the called number (the new destination for the call), and the charge number (typically the same as the called number in the INVITE request).

In the second procedure, CMS$_T$ MUST generate a private URL (as defined in Appendix III), causing the redirected call attempt to be routed through CMS$_T$ for generation of the proper event messages and billing support. The private URL contains the following information encoded in the userinfo portion: 1) the new forwarded destination; 2) the contents of the P-DCS-Billing-Info headers in the INVITE request; and 3) the values of Billing-Correlation-ID assigned for the event message streams being generated by CMS$_T$. CMS$_T$ MUST also add a P-DCS-Billing-Info header containing the Correlation-ID and Financial-Entity-ID of CMS$_T$ to the 3xx response.

CMS$_T$ MUST send the following 3xx–Redirect response to CMS$_O$:

302-Redirect (CMS$_T$ -> CMS$_O$) Header:	**Requirements On CMS$_T$ for Message Generation**
SIP/2.0 302 Moved Temporarily	Status line with status code 3xx MUST be present.
Via:	As described in 6.13.
P-DCS-Billing-Info:	MUST be present, as described above and in 7.7
P-DCS-Laes:	MAY be present, as described above and in 7.7.
From:	As described in 6.13.
To:	
Call-ID:	
Cseq:	
Contact:	MUST be inserted by CMS$_T$, and carries the new destination information, which MUST be a valid URI. If the new destination is a telephone number, then the format of the URI MUST be a tel-URI where the URI contains a telephone number as defined in 7.1.
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g*., TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this response MUST cease on receipt of an ACK.

ACK (CMS$_O$ -> CMS$_T$) Header:	**Requirements On CMS$_T$ For Message Checking**
ACK URI SIP/2.0	As described in 6.13.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the ACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.1.7.3 Other Status Response to INVITE Request

A final error response (4xx, 5xx, or 6xx) MUST be sent as per [8]. This includes, but is not limited to, 480-Temporarily-Unavailable. The error response MUST be formatted as follows:

Error (CMS_T -> CMS_O) Header:	Requirements On CMS_T for Message Generation
SIP/2.0 xxx	As described in 6.13.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this response MUST cease on receipt of the ACK.

ACK (CMS_O -> CMS_T) Header:	Requirements On CMS_T for Message Checking
ACK URI SIP/2.0	As described in 6.13.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the ACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.1.8 CMS_O Receives Final Response from CMS_T

8.4.1.8.1 CMS_O Receiving 200-OK

Once the terminating endpoint accepts the incoming call (*e.g.*, off-hook or hook-flash), it sends a 200-OK status message to the originating CMS_O. The message sent by CMS_T to CMS_O MUST be formatted as follows:

200-OK (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 200 OK	Status line with status code 200 MUST be present.
Via:	As described in 6.13.
From:	
To:	
Call-ID:	
Cseq:	
Contact:	As described in 6.20.10.
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the 200-OK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

On receiving the final response, CMS_O MUST stop timer T3, tell the endpoint device to commit to resources that have been reserved for this call, and tell the endpoint device that it SHOULD begin sending bearer channel packets.

CMS_O MUST acknowledge the 200-OK response with an ACK message. The header fields MUST be generated as follows:

ACK (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
ACK URI SIP/2.0	As described in 6.13.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the ACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.1.8.2 CMS_O receiving 302-Redirect

Note that the procedures defined in this section are identical to the procedures defined in section 8.4.1.3.2.

If the terminating device wished to forward the call (*e.g.*, if call-forwarding-no-answer was enabled at the destination), a 302-Redirect status response with the forwarded-to destination URI in the contact header is returned. The message sent by CMS_T to CMS_O MUST be formatted as follows:

302-Redirect (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 302 Moved Temporarily	As described in 6.13.
Via:	
P-DCS-Billing-Info:	MUST be present.
P-DCS-Laes:	MAY be present.
From:	As described in 6.13.
To:	
Call-ID:	
Cseq:	
Contact:	MUST be present as described in 6.20.10. Carries the new destination information. MUST be a valid URI.
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

CMS_O MUST match the 302-Redirect response to the earlier INVITE. CMS_O MUST send an ACK message to CMS_T. The required fields of the message are:

ACK (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
ACK URI SIP/2.0	As described in 6.17.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the ACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

Following transmission of the ACK message to CMS_T, CMS_O MUST issue an INVITE request to the party indicated by the Contact header in the 3xx response. CMS_O MUST attempt to resolve the SIP-URI from the Contact header into a destination IP address. CMS_O MUST generate a Request-URI form the Contact header value as described in 8.3.

If the destination endpoint is not served by CMS_O, CMS_O MUST generate a Request-URI form the Contact header value as described in 8.3. CMS_O generates an INVITE message and sends it to CMS_F, the CMS that manages the forwarded-to destination.

If a P-DCS-Laes header is present in the 3xx response, CMS_O MUST include that header unchanged in the reissued INVITE. CMS_O MUST also include a P-DCS-Redirect header containing the original dialed number, the new destination number, and the number of redirections that have occurred.

CMS_O MUST copy the contents of the P-DCS-Billing-Info header in the 3xx response to a P-DCS-Billing-Info header in the new INVITE.

The rest of the INVITE message MUST appear identical to that which was sent to CMS_T, with the exception of an incremented Cseq value.

The format of the resulting INVITE message as sent by CMS_O to CMS_F, and the associated requirements on the header fields are as follows:

INVITE (CMS_O -> CMS_F) Header:	**Additional Requirements for Message Generation**
INVITE URI SIP/2.0	As described above
Via:	As described in 8.4.1.1.
Require:	As described in 8.4.1.1.
Proxy-Require:	As described in 8.4.1.1.
Supported:	As described in 8.4.1.1.
Allow:	As described in 8.4.1.1.
P-Asserted-Identity:	As described in 8.4.1.1.
Privacy:	As described in 8.4.1.1.
P-DCS-Billing-Info:	As described above
P-DCS-Laes:	As described above
P-DCS-Redirect:	As described above
Max-Forwards:	As described in 8.4.1.1.
From:	As described in 8.4.1.1.
To:	As described in 8.4.1.1.
Call-ID:	As described in 8.4.1.1.
CSeq:	As described in 8.4.1.1.
Contact:	As described in 8.4.1.1.
Content-Type:	As described in 8.4.1.1.
Content-Length:	As described in 8.4.1.1.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	As described in 8.4.1.1. c= line MAY be modified in support of IP address Privacy.

On receipt of this INVITE message, CMS_F uses the combination of From, To, Call-ID, Cseq, and Request-URI headers to recognize this as a new call and not a retransmission from a previous call.

The behavior and processing of the INVITE at CMS_F is identical to that described in section 8.4.1.2.

CMS_O MUST accept a 100-Trying message as described in the following table.

100-Trying (CMS_F -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 100 Trying	As described in 6.13.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On receipt of a 100-Trying provisional response, the transaction timer (T3) for this exchange MUST be set to T-setup. The default value of (T-setup) is given in Appendix I. On expiration of T3, CMS_O clears the call attempt and sends a CANCEL message to CMS_T with the same values of Request-URI, From, To, and Call-ID for this call attempt, as specified in Section 8.4.1.9.

Processing of responses to this INVITE request is as given in Section 8.4.1.2.

8.4.1.8.3 CMS_O receiving other error response

A final error response (4xx, 5xx, or 6xx) MAY be sent as per 6.13. This includes, but is not limited to, 480-Temporarily-Unavailable. The error response MUST be verified as follows:

Error (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0	As described in 6.13.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

CMS_O MUST send an ACK message to acknowledge the error response:

ACK (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
ACK URI SIP/2.0	As described in 6.17.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the ACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.1.9 Session Timer expiration at CMS_O

On expiration of timer T3, CMS_O MUST send a CANCEL request to CMS_T and MUST release all resources reserved for this connection. The CANCEL request MUST be as described below. CMS_O MUST also be prepared to send a BYE message in the case that it receives a final response after sending the CANCEL.

CANCEL (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
CANCEL URI SIP/2.0	As described in 6.9.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this request MUST cease on receipt of a 200-OK.

The 200-OK response to the CANCEL MUST be formatted as follows:

200-OK (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
SIP/2.0 200 OK	As described in 6.9.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the 200-OK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.2 Initiating an Emergency Call

A call for emergency services, *e.g.*, 9-1-1, MUST follow the procedures given for a basic call, as given in section 8.4.1, with the following exceptions.

As described in Section 7.1, the emergency services telephone number is not an international number and hence cannot be supplied as a global-number. Instead, the local-number form MUST be used and a “phone-context” parameter set to the relevant prefix, e.g. “+1” MUST be added as illustrated here:

tel:911;phone-context=+1

If the originating endpoint is not authorized for outgoing service, CMS_O MAY permit the call to the emergency services number.

If CMS_O is unable to establish the identity of the originator of the call, CMS_O MAY permit the call to the emergency services number. Otherwise the P-Asserted-Identity header MUST identify the originator of the call as described in Section 8.4.1.1.1.

CMS_O, receiving a 183-Session-Progress response for a 9-1-1 call, MUST indicate enhanced priority for access network admission control in the GATE-SET command to the originating CMTS, using the mechanisms described in [4].

A 9-1-1 call SHOULD NOT be put on hold or disconnected due to feature interaction. CMS_O MUST disable all call features on any line that is placing a call to emergency services.

CMS_O MUST NOT send a BYE request to the Emergency Services Center; rather, CMS_O MUST keep the call up until it receives a BYE request from the Emergency Services Center.

8.4.3 CMS Procedures for REFER

The SIP REFER method is described in 7.6, with further specification text in section 7.5. This section details the procedures that a CMS follows in generating and responding to a REFER request.

In the following sections, CMS_I is the CMS that initiated the REFER request, CMS_O is the target of the REFER (who also initiates the action requested by the REFER), and CMS_T is the CMS that receives the action requested by the REFER. One typical application is three-way-calling (one implementation described in Section 8.4.7), in which case CMS_O is a Bridge Server that receives the REFER request and initiates INVITEs to parties to be added to a conference.

The basic REFER message sequence for a CMS includes the REFER request, a 202-Accepted response, the request initiated by CMS_O, a NOTIFY request, and a 200-OK response.

8.4.3.1 CMS_I Initiates REFER Request

This specification only defines the use of REFER within a dialog. As stated in section 7.6, defining its use outside a dialog requires additional specification of Event Messages and the corresponding use of the contents of the P-DCS-Billing-Info header.

When the REFER is generated within an established call-leg, the call-leg identification (From tag, To tag, and Call-ID) MUST match those of the call-leg between CMS_I and CMS_O. The CSeq MUST be higher than the value of the last-transmitted request (*e.g.*, the ACK). The Request-URI of the REFER MUST be the value of the most recently received Contact header from CMS_O, and the Route header (if one is present for the existing dialog) MUST be included in the REFER request.

By initiating a REFER request, the Initiator is agreeing to be billed for a logical call-leg from himself to CMS_T for the duration of the resulting session. Hence the REFER includes the appropriate billing information so that it can be included in the INVITE sent by CMS_O.

Two different procedures are defined for generating the Refer-To header value. In the first procedure, CMS_I does not remain on the signalling path for the resulting call. In the second procedure, CMS_I does remain on the signalling path for the resulting call. Use of the first procedure is OPTIONAL; however, its use requires certain conditions to be met, as described below. If the first procedure is not used, the second procedure MUST be used.

In order to use the first procedure, the RKS-Group-ID of CMS_O (as given in the P-DCS-Billing-Info header in the INVITE request for this dialog) MUST be the same as the RKS-Group-ID of $\text{CMS}_{\text{I}.}$ In this procedure, the Refer-To header is set up to point to CMS_T. The basic URL is the same as would be used in the Request-URI, were CMS_I sending an INVITE directly to CMS_T; it is constructed according to the procedures described in section 8.3. The method parameter is added, with a method of INVITE. CMS_I MUST add a P-DCS-Billing-Info header to the Refer-To URL to allow the additional leg of the resulting call to be charged to the party that initiated the REFER. CMS_I MUST include in this P-DCS-Billing-Info header the Correlation-ID and Financial-Entity-ID of CMS_I, the calling number (initiator of the REFER request), the called number (the new destination for the call), and the charge number (typically the initiator of the REFER request).

In the second procedure, CMS_I MUST generate a private URL (as defined in Appendix III), and place it in the Refer-To header of the REFER request. This causes the resulting call attempt to be routed through CMS_I for generation of the proper event messages and billing support. The private URL contains the following information encoded in the userinfo portion: 1) the new destination; and 2) the values of Billing-Correlation-ID assigned for the event message streams being generated by CMS_I.

Any additional header parameters appended to the Refer-To URL (*e.g.*, Refer-To: URI ? header=value & header=value) will be copied into the INVITE issued by CMS_O, subject to the procedures given in 6.19. The headers which need to be included in the Refer-To URL are described in the following paragraphs.

CMS_I MUST check for an outstanding lawfully authorized surveillance order for the initiating subscriber. If found, CMS_I MUST include a "P-DCS-Laes=" header parameter in the Refer-To. The P-DCS-Laes header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call's event messages, MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be delivered. CMS_I MUST include a "P-DCS-Redirect=" header parameter in the Refer-To. The P-DCS-Redirect header MUST include the original destination (taken from the P-DCS-Redirect header in the initial INVITE, if any, or else from the To header of that INVITE message), the forwarding subscriber (CMS_I), and the number of redirections (1 plus the number in the P-DCS-Redirect header in the initial INVITE, if any, or else 1).

An additional Replaces header MAY be attached to the Refer-To URI in specific cases.

The REFER request MUST NOT contain an SDP description.

The requirements on the headers which CMS_I MUST include in the message are shown below:

REFER (CMS_I->CMS_O) Header:	Requirements On CMS_I For Message Generation
REFER URI SIP/2.0	MUST be as described in 7.6.

REFER (CMS_I->CMS_O) Header:	Requirements On CMS_I For Message Generation
Via:	As described in 6.20.42.
Require:	MUST include “100rel”, “precondition”.
Proxy-Require:	As described in 6.20.29.
Supported:	As described in 6.20.37.
Refer-To: URI;method=INVITE ? [P-DCS-Billing-Info=yy] [&] [P-DCS-Redirect=mm & P-DCS-Laes=nn]	MUST be as described in 7.6, and identifies the new address of the destination to which the recipient of this REFER is to issue an INVITE. Identifies new call leg to be created. Attached header parameters MUST be as described above.
Max-Forwards:	As defined in 6.12.2.
From:	
To:	
Call-ID:	
CSeq:	
Accept:	MUST include "message/sipfrag"
Contact:	As defined in 6.20.10 and 7.6.
Content-Length:	MUST be present, and MUST indicate a zero-length body.
	An empty line (CRLF) MUST be present.

If the REFER message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code. Otherwise, CMS_I sets an application-level timer (T3) associated with the REFER, with value of T-setup. This timer is canceled on receipt of either a final response to the REFER or a NOTIFY to the REFER indicating a successful session setup. If timer T3 expires, CMS_I MUST clear the REFER attempt. Thus, if no 2xx response was received, CMS_I MUST send a CANCEL to CMS_O with the same values of Request-URI, From tag, To tag, and Call-ID as in the original REFER request.

8.4.3.2 CMS_O Receives REFER

CMS_O receives the REFER request and verifies the requirements shown in the previous sub-section. If acceptable, it returns a 202 Accepted final response and goes on to send an INVITE to the Refer-To party (see section 8.4.1.1). If the request is not acceptable, CMS_O returns an appropriate 4xx response; 406-Not-Acceptable or 486-Busy-Here are recommended.

202-Accepted (CMS_O -> CMS_I) Header:	Requirements On CMS_O for Message Generation
SIP/2.0 202 Accepted	As described in 7.6.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the REFER is received within an existing session at a Bridge Server performing conferencing, the Bridge Server MUST assume that the new call-leg that the REFER will create is intended to use the same conference bridge as the existing call-leg. Before responding, it also verifies that a free port is available on the bridge.

The REFER creates an implicit subscription to the “refer” event package as described in 7.6. Hence, CMS_O MUST send an immediate NOTIFY request to CMS_I upon accepting the REFER. The format of the NOTIFY is:

NOTIFY (CMS_O -> CMS_I) Header:	Requirements On CMS_O for Message Generation

NOTIFY (CMS$_O$ -> CMS$_I$) Header:	Requirements On CMS$_O$ for Message Generation
NOTIFY URL SIP/2.0	MUST be present. Method MUST be NOTIFY. The value of URL MUST be copied from the Contact header previously received in the REFER.
Via:	As described in 7.6.
Max-Forwards:	As defined in 6.20.22
From:	As defined in 6.12.2.
To:	
Call-ID:	
Cseq:	
Event: refer	As described in 7.6.
Contact:	As described in 7.6.
Subscription-State:	As described in 7.6.
Content-Type: message/sipfrag	MUST be present. Type MUST be "message/sipfrag".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
<Message body>	Message body MUST be present. MUST contain the minimal information specified in 7.6.

If the Notify message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.3.3 CMS$_I$ Receives Final Response to REFER

CMS$_I$ stops the transaction timer T3.

If the response is 202-Accepted, CMS$_I$ waits for notification of the final result of the request. As described below, other events may precede receipt of this notification, in which case CMS$_I$ will act on those other events.

8.4.3.4 CMS$_I$ Receives Initial NOTIFY for REFER

Upon receiving the initial NOTIFY for the REFER, CMS$_I$ sends a 200-OK to CMS$_O$:

200-OK (CMS$_I$ -> CMS$_O$) Header:	Requirements On CMS$_I$ for Message Generation
SIP/2.0 200 OK	As described in 7.6.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the 200-OK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.3.5 CMS$_O$ Sends INVITE to Target

CMS$_O$ creates an INVITE request based on the contents of the REFER. The Request-URI and To header are populated with the URI from the Refer-To header. Additional headers appended to the Refer-To URI (*e.g.*, Replaces) are copied to the INVITE.

If the Refer-To URL did not contain a P-DCS-Billing-Info header, then CMS$_O$ MUST include in the generated INVITE a P-DCS-Billing-Info header that is identical to the P-DCS-Billing-Info header that appeared in the INVITE that created the existing dialog. If CMS$_O$ initiated that dialog as a UAC, then this is the header value sent in that INVITE; if CMS$_O$ terminated that dialog as a UAS, then this is the header value received in the INVITE. In this way, the billing arrangements of the previous dialog (between CMS$_I$ and CMS$_O$) are maintained for the first segment of the new call.

The contents of the INVITE are summarized in the following table:

INVITE (CMS_O -> CMS_T) Header:	**Additional Requirements For Message Generation**
INVITE URI SIP/2.0	URI taken from Refer-To
Via:	As described in 6.20.42.
Proxy-Require:	As described in 6.20.29.
Require:	MUST include “100rel”, “precondition”.
Supported:	As described in 6.20.37.
Allow:	As defined in 6.20.5. MUST include “UPDATE”
P-Asserted-Identity:	As described in 8.4.1.1. The identity provided is that of the entity issuing the INVITE, as opposed to the identity of the entity that issued the REFER.
Privacy:	As described in 8.4.1.1.
P-DCS-Billing-Info:	Copied from Refer-To, if present. Otherwise, MUST contain billing information identical to the original call between CMS_O and CMS_I.
Max-Forwards	As defined in 6.13 and 6.20.22
From:	As defined in 6.13 and 6.20.20.
To:	MUST be present. URI taken from Refer-To.
Call-ID:	As defined in 6.13.
Cseq:	
Contact:	As defined in 6.13 and 6.20.10.
Content-Type:	MUST be present and MUST contain “application/ SDP”.
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= m= a=	As described in section 8.4.1.1 c= line MAY be modified in support of IP address Privacy. a= line MUST be present and MUST indicate mandatory send and receive precondition as described in 7.4.

Subsequent steps in setting up this second call-leg at CMS_O introduce nothing new compared with sections 8.4.1.1 through 8.4.1.5.

In the specific case when CMS_O is a Bridge Server performing conferencing services, there are two changes from the usual procedures:

- When the Bridge Server receives 180-Ringing, it instructs the conference bridge to play out ringback tone on all ports except that held by the new call-leg.
- When the final response is received from CMS_T, the Bridge Server instructs the conference bridge to discontinue the ringback tone. Receipt of media from the alerted party will also discontinue the ringback tone.

8.4.3.6 CMS_O Sends Final NOTIFY To CMS_I

CMS_O MUST send a NOTIFY request to CMS_I when it receives the final response to the INVITE.

The NOTIFY request is described in section 7.5. The format of the message is:

NOTIFY (CMS_I -> CMS_O) Header:	Requirements on CMS_I for Message Generation
NOTIFY URL SIP/2.0	MUST be present. Method MUST be NOTIFY. The value of URL MUST be copied from the Contact header previously received in the REFER.
Via:	As described in 6.12.2.
Max-Forwards:	As defined in 6.20.22.
From:	As defined in 6.12.2.
To:	
Call-ID:	
Cseq:	
Event: refer	As described in 7.6.
Contact:	
Subscription-State:	
Content-Type: message/sipfrag	MUST be present. Type MUST be "message/sipfrag".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
<Message body>	Message body MUST be present. At a minimum, MUST contain the information specified in 7.6.

If the NOTIFY message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.3.7 CMS_I Receives NOTIFY

When CMS_I receives a NOTIFY it matches the From, To, and Call-ID headers to an existing call-leg, checks to see that the call-leg has at least one outstanding REFER, and verifies that the value of the Cseq parameter in the Event header of the NOTIFY matches the Cseq header of an outstanding REFER. If all of these checks succeed, CMS_I returns a 200-OK final response:

200-OK (CMS_I -> CMS_O) Header:	Requirements On CMS_I for Message Generation
SIP/2.0 200 OK	As described in 7.6.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the NOTIFY matches an outstanding REFER, CMS_I cancels the corresponding timer T3 and determines the outcome of the triggered INVITE from the status line provided in the NOTIFY body. If the encapsulated status line indicates a result other than 200-OK, the session attempted with the REFER request has failed, and CMS_I SHOULD take action to recover appropriate to the service being requested.

If CMS_I is unable to match the NOTIFY to an outstanding REFER within an existing call-leg, it returns the final response 481 Subscription Does Not Exist, and takes no further action.

8.4.3.8 CMS_O Receives Final Response To NOTIFY

CMS_O terminates the retransmission timer for the NOTIFY. It takes no other action based on the final response.

8.4.4 CMS handling of Mid-Call Changes

Mid-call changes include call-hold, call-resume, call replacement, operator services, and dynamic codec changes.

The initiator of a mid-call change in this section is referred to as CMS$_I$, and the recipient of a mid-call change is referred to as CMS$_R$. Another type of mid-call change involves changing the endpoints of sessions; these are usually referred to as call control services. The REFER method, for which example procedures were given in 8.4.3 and example applications are given in 8.4.6 and 8.4.7, provides tools by which many call control services may be built. Implementation of the REFER method is REQUIRED by this specification. For purposes of this document, three uses of REFER are given as examples: blind transfer, consultative transfer, and ad-hoc conferencing. Based on knowledge of the recipient behavior, the originator MAY perform many other complex call control operations beyond those shown here.

8.4.4.1 CMS$_I$ Initiating Call Hold: UPDATE(hold)

To place a call on hold, an UPDATE(hold) message is sent on the signalling channel to the party that is to be put on hold. This is a standard SIP UPDATE request, with an additional "a=sendonly" attribute for the media stream in the SDP. The format of the UPDATE message sent by the initiating CMS$_I$ and the requirements on the header fields checked at the receiving CMS$_R$ are as follows:

UPDATE(Hold) (CMS$_I$ -> CMS-Agent$_R$) Header:	Requirements On CMS$_I$ for Message Generation Requirements On CMS$_R$ for Message Checking
UPDATE URI SIP/2.0	As described in 7.3.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22
From:	As defined in 6.12.2.
To:	
Call-ID:	
CSeq:	
Content-Type :	MUST be present and MUST contain "application/SDP".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	MUST be present. a= line MUST be present and MUST indicate "sendonly".

If the UPDATE(hold) message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code. Otherwise, CMS$_R$ MUST send the 200-OK with its SDP description to CMS$_I$ and MUST direct the endpoint on hold to stop sending bearer channel packets. Note that this only holds the media stream in one direction. CMS$_R$ MAY decide to return a held SDP as well, however it SHOULD NOT automatically do this in response to an UPDATE(hold).

200-OK (CMS_R -> CMS_I) Header:	**Requirement for CMS_r for Message Generation Requirement for CMS_O for Message Checking**
SIP/2.0 200 OK	As defined in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Content-Type:	MUST be present and MUST contain "application/SDP".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	MUST be present. a= line MAY be present and indicate "sendonly"

After sending the UPDATE(hold), the initiator MUST wait for a 200-OK response. If the 200-OK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.4.2 CMS_I Resuming a held call: UPDATE(resume)

The party that placed the call on hold MUST be the one to take it off hold. To take a call off hold, an UPDATE(resume) is sent. An UPDATE(resume) is an UPDATE(hold) message with the SDP description of the call being reinstated. Note that if this SDP has changed from the pre-hold SDP then QoS may have to be renegotiated. It is consequently RECOMMENDED that the pre-hold SDP be reused for the resumed session.

The format of the INVITE message sent by the initiating endpoint (CMS_I) and the requirements on the header fields checked at the receiving endpoint (CMS_R) are as follows:

UPDATE(Resume) (CMS_i -> CMS_r) Header:	**Requirements On CMS_I for Message Generation Requirements On CMS_R for Message Checking**
UPDATE URI SIP/2.0	As described in 7.3.
Via:	As described in 6.20.42.
From:	As described in 6.12.2.
To:	
Call-ID:	
CSeq:	
Max-Forwards:	As defined in 6.12.2.
Content- Type:	MUST be present and MUST contain "application/SDP".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	MUST be present. SHOULD be same as the last SDP sent before the UPDATE(hold).

The CMS_R sends a 200-OK with an SDP description to CMS_I. Note that if this SDP has changed from the pre-hold SDP then QoS may have to be renegotiated. It is consequently RECOMMENDED that the pre-hold SDP be reused for the resumed session. Note that this only resumes the media stream in one direction. If CMS_R had held the media stream as well, CMS_R MAY decide to return resume SDP as well, however it SHOULD NOT automatically do this in response to an UPDATE(resume).

The 200-OK response MUST be as follows:

200-OK (CMS_R -> CMS_I) Header:	Requirements On CMS_R for Message Generation Requirement On CMS_O for Message Checking
SIP/2.0 200 OK	As defined in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Content- Type:	MUST be present and MUST contain "application/SDP".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	MUST be present.

If the 200-OK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.4.3 CMS_R Receiving Call Hold: UPDATE(hold) and UPDATE(resume)

On receiving an UPDATE(hold), CMS_R sends a 200-OK with SDP description to the party requesting the hold, and instructs the endpoint to stop sending bearer channel packets.

On receiving an UPDATE(resume), which is an UPDATE message with the SDP description of the call being reinstated, the CMS sends the party requesting the resume a 200-OK with SDP description to the party requesting the resume is sent back. Note that if either of the SDPs has changed from the pre-hold SDP then QoS may have to be renegotiated. It is consequently RECOMMENDED that the pre-hold SDP be reused for the resumed session.

CMS_R MUST not initiate an UPDATE(resume) during an UPDATE(hold). See section 8.4.4.1 and 8.4.4.2 for description of the header fields in each message.

8.4.4.4 Operator Services: Initiating INVITE(BLV) and INVITE(EI)

Operator Services (Busy Line verification and Emergency Interrupt) are initiated from the CMS on behalf of a PSTN gateway connecting to special MF trunks groups from the OSPS system. The SIP messages INVITE(BLV) and INVITE(EI) are initiated by CMS_O. These messages include the P-DCS-OSPS header with parameters as defined below.

The INVITE(BLV) message sent by CMS_O to initiate a busy line verification MUST be formatted as follows:

INVITE (BLV) (CMS_O -> CMS_T) Header:	**Requirements On CMS_O for Message Generation**
INVITE URI SIP/2.0	As described in 8.4.1.1.
Via:	As described in 6.20.42.
Require:	As described in 6.20.32. MUST include "100rel", "precondition", and "P-DCS".
Proxy-Require:	As described in 6.20.29.
Supported:	As described in 6.20.37.
P-DCS-OSPS: BLV	MUST be present.
--all other headers--	All other headers are unchanged from INVITE as in 8.4.1.1.
	An empty line MUST be present between the headers and the message body.
--SDP description--	MUST be an SDP description, as specified in 8.4.1.1.

Retransmissions of this request MUST cease on receipt of any response.

The remainder of the call establishment, from the view of the originator, proceeds identically to that of a basic call given in 8.4.1.

On receipt of an indication from the Media Gateway that an intercept tone is present on the trunk, CMS_O initiates an INVITE(EI) request, or an UPDATE(EI) request, to convert the call to an emergency interrupt session. The INVITE(EI), if used for this purpose, is sent over the signalling path, as follows:

INVITE(EI) (CMS_O -> CMS-$Agent_T$) Header:	**Requirements On CMS_O for Message Generation**
INVITE URI SIP/2.0	The request method MUST be set to INVITE. The Request URI MUST be the value of the most recent Contact header received.
Via:	As described in 6.20.42.
Require:	MUST include "P-DCS", and "100rel".
From:	As described in 6.12.2.
To:	
Call-ID:	
CSeq:	
P-DCS-OSPS: EI	MUST be present, as described in 7.7.
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If an UPDATE(EI) request is used instead, it is identical except that the request line contains UPDATE instead of INVITE.

Retransmissions of this request MUST cease on receipt of any response. The expected response is a 200-OK, as follows:

200-OK (CMS$_T$ -> CMS$_O$) Header:	Requirements On CMS$_O$ for Message Checking
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On receipt of the 200-OK, if an INVITE(EI) request was used, CMS$_O$ MUST respond with an ACK message. No ACK message is needed if an UPDATE(EI) request was used.

ACK (CMS$_O$ -> CMS$_T$) Header:	Requirements On CMS$_O$ for Message Generation
ACK URI SIP/2.0	As described in 6.13.
Via:	
Max-Forwards:	
From:	
To:	
Cal l-II:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the ACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.4.5 Operator Services: Receipt of INVITE(BLV) and INVITE(EI)

Operator Services (Busy Line verification and Emergency Interrupt) are initiated from the CMS on behalf of a PSTN gateway connecting to special MF trunks groups from the OSPS system. The SIP messages INVITE(BLV) and INVITE(EI) are initiated by the CMS. These messages include the P-DCS-OSPS header with parameters as defined below.

CMS$_T$ MUST be prepared to receive an INVITE(BLV) at any time. If not received from another CMS by security procedures specified in [2], it SHOULD be rejected. Receipt of an INVITE(BLV) SHOULD NOT result in a busy error response. Receipt of an INVITE(BLV) MUST NOT result in alerting the user.

INVITE(BLV) (CMS$_O$->CMS$_T$) Header:	Requirements On CMS$_T$ for Message Checking
INVITE URI SIP/2.0	MUST be present. Identifies the line that is to be verified as busy.
P-DCS-OSPS: BLV	MUST be present. MUST be set to BLV.
--all other headers, including SDP---	MUST be as specified for INVITE.

CMS$_T$ MUST respond to INVITE(BLV) with a 183-Session-Progress, and the call completes as in Sections 8.4.1.2.1, 8.4.1.4, and 8.4.1.7.

The SDP describes the media flow from the endpoint to the PSTN gateway; CMS$_T$ SHOULD cause a packet stream to be sent to that address. The endpoint MAY perform a mixing operation between the two ends of an active call, and send the mixed stream to the OSPS system. The endpoint MAY check for voice activity locally, and if there is none it MAY send a copy of the received voice stream. The endpoint MAY send a duplicate copy of the locally-generated voice stream.

If the telephone line is idle, CMS$_T$ SHOULD cause a stream of silence packets to be sent to the OSPS system. If the telephone line is ringing, or if it is locally generating a ringback tone, CMS$_T$ SHOULD cause a ringback sequence to be sent to the OSPS system.

The operator may decide to interrupt the call after confirming that the line is busy, and signals this intention by placing an interrupt tone on the voice path to the endpoint. The MG at the PSTN Gateway detects this tone and the CMS for that MG formulates an INVITE(EI) message. This message is a variant of the INVITE with P-DCS-OSPS header set to EI. This INVITE(EI) message is sent over the end-to-end signalling channel from CMS$_O$ to CMS$_T$.

CMS$_T$ MUST be prepared to accept an INVITE(EI) or UPDATE(EI) at any time that a BLV call is active. The INVITE(EI) is defined in the following table:

INVITE(EI) (CMS$_O$->CMS$_T$) Header:	**Requirements On CMS$_T$ for Message Checking**
INVITE URI SIP/2.0	Request line MUST be present.
Require:	MUST be present. MUST include "P-DCS" and "100rel"
Max-Forwards:	As defined in 6.20.22
From:	As described in 6.12.2.
To:	
Call-ID:	
CSeq:	
P-DCS-OSPS: EI	MUST be present. MUST be equal to EI.
Content-Length:	MUST be present if the transport protocol is stream-based (e.g., TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

UPDATE(EI) is identical to INVITE(EI) except that the request line contains UPDATE instead of INVITE.

If CMS$_T$ receives an INVITE(EI) or an UPDATE(EI) but has not previously received INVITE(BLV) with identical call-leg identification, it MUST reject the message.

CMS$_T$ responds to INVITE(EI) or UPDATE(EI) with a 200-OK final response.

200-OK (CMS$_T$ -> CMS$_O$) Header:	**Requirements On CMS$_T$ for Message Generation**
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this response MUST cease on receipt of the following ACK message. Note that no ACK message is needed for UPDATE(EI).

ACK (CMS$_O$ -> CMS$_T$) Header:	**Requirements On CMS$_T$ for Message Checking**
ACK URI SIP/2.0	As described in 6.13.
Via:	
Max-Forwards:	
From:	
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On acceptance of a valid INVITE(EI) or UPDATE(EI), CMS$_T$ MUST enable communication between the operator and the local user. CMS$_T$ MAY place the existing call on hold and switch to the operator call (*e.g.*, call-waiting). In the alternative, if resources are available, CMS$_T$ could establish a three-way call with the operator and the current party or parties.

8.4.4.6 SIP Messages for Codec Changes – INVITE/UPDATE(Codec-change)

A codec change can occur automatically when two or more codecs are negotiated in the SDP "m=" line. This does not involve any SIP signalling and hence it is not addressed here. However, changing to one or more codecs that were not negotiated in the SDP requires SIP signalling described below.

A signalling message may be sent by either endpoint to initiate a such change in the codec(s). There are two separate cases described. The first is a change to one or more codecs that fall within the existing resource authorization, *e.g.*, as established by the set of codecs listed in the initial INVITE request. Resource authorization for those codecs has already been performed, and the message exchange between the CMSes occurs only to synchronize the change. This signalling exchange SHOULD be an UPDATE(Codec-Change) request, as described in 8.4.4.6.1. An INVITE(Codec-change) MAY be used instead, as described in 8.4.4.6.2.

The second case is a change to one or more codecs that require network resources above and beyond the existing resource authorization, *e.g.*, because they were not previously specified in the initial INVITE. The Gate Controller component of the CMSes must be involved in this procedure in order to increase the resource authorization; therefore, the message exchange follows the proxy-proxy signalling path. This signalling exchange MUST be an INVITE(Codec-change), as described in 8.4.4.6.2.

8.4.4.6.1 Codec Change within Previous Authorization

If the new codec(s) that CMS$_I$ wishes to adopt not require additional network resources compared to the codecs included in the SDP of the initial INVITE transaction (or authorized by a subsequent INVITE(codec-change) request), the codec(s) are considered authorized by the network.

In this case, CMS$_I$ initiating the codec change SHOULD send an UPDATE request to the other endpoint with the new codec description. Alternatively, an INVITE request MAY be sent, as described in 8.4.4.6.2; however, this involves a greater number of messages and requires more time to complete.

36

Retransmission of this request MUST cease on receipt of a final response.

On receiving an UPDATE(codec-change), CMS$_R$ MUST match it to the existing call by the use of the From, To, and Call-ID headers. If there is no match, CMS$_R$ sends a 481-Call-does-not-Exist error response.

If a matching call is found, but the codec change is not acceptable, CMS$_R$ MUST send a 488-Not-Acceptable-Here error response.

If a matching call is found, and the codec change is acceptable, CMS$_R$ MUST send a 200-OK response, giving the agreed codec(s).

200-OK (CMS$_R$ -> CMS$_I$) Header:	**Requirements On CMS$_R$ for Message Generation** **Requirements On CMS$_I$ for Message Checking**
SIP/2.0 200 OK	As defined in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Content-Type:	MUST be present and MUST contain "application/SDP".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.

200-OK (CMS_R -> CMS_I) Header:	Requirements On CMS_R for Message Generation Requirements On CMS_I for Message Checking
v= o= s= c= b= t= a= m=	SDP MUST be present

On sending the 200-OK, CMS_R instructs the endpoint to commit the network resources. The endpoint MAY start sending using the new codec.

On receiving a 200-OK response, CMS_I instructs the endpoint to commit network resources. The endpoint MAY start using the new codec.

8.4.4.6.2 Codec Change Requiring New Authorization

The format of the INVITE message sent by CMS_I and the requirements on the header fields checked at the receiving CMS (CMS_R) are:

INVITE(codec-change) (CMS_I->CMS_R) Header:	Requirements on CMS_I for message generation Requirements on CMS_R for message checking
INVITE URI SIP/2.0	As described in 6.12.2. The Request URI MUST be the value of the most recent contact header received for this call.
Require:	MUST include “100rel”, and “precondition”.
Proxy-Require:	As described in 6.20.29.
Supported:	As described in 6.20.37.
Via:	As described in 6.20.42.
P-Asserted-Identity:	As described in section 8.4.1.1
Privacy:	As described in section 8.4.1.1
Max-Forwards:	As defined in 6.20.22
From:	As defined in 6.12.2.
To:	
Call-ID:	
CSeq:	
Content -Type:	MUST be present and MUST contain “application/SDP”.
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	a= line MUST be present and MUST indicate mandatory send and receive precondition as described in 7.4.

Retransmission of this request MUST cease on receipt of a final response.

On receiving an INVITE(Codec-change), CMS_R MUST match it to the existing call by use of the From, To, and Call-ID headers. If there is no match, CMS_R considers this a new call attempt, and the procedure continues as described in 8.4.1.2.

If a matching call is found, CMS$_R$ MUST send a 183-Session-Progress response, giving the agreed codec(s):

183-Session-Progress (CMS$_R$ -> CMS$_I$) Header:	**Requirements On CMS$_R$ for Message Generation Requirements On CMS$_I$ for Message Checking**
SIP/2.0 183 Session Progress	Status line with status code 183 MUST be present.
Via:	As described in 6.12.2.
Require:	As defined in 6.20.32.
From:	As described in 6.12.2.
To:	
Call-ID:	
CSeq:	
Contact:	As defined in 6.20.10.
RSeq:	As defined in 7.2.
Content-Type:	MUST be present and MUST contain “application/SDP”.
Content-Length:	As described in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
v= o= s= c= b= t= a= m=	a= line MUST be present, MUST indicate mandatory send and receive preconditions, and MUST request confirmation, as described in 7.4.

Retransmissions of this response MUST cease on receipt of the PRACK.

CMS$_I$ MUST send a PRACK to acknowledge receipt of the 183-Session-Progress. The PRACK message MUST be sent directly to the address specified in the most recent Contact header:

PRACK (CMS$_I$ -> CMS$_R$) Header:	**Requirements On CMS$_I$ for Message Generation Requirements On CMS$_R$ For Message Checking**
PRACK URI SIP/2.0	As described in 7.2.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22
From:	As described in 6.12.2.
To:	
Call-ID:	
Cseq:	
Rack:	As described in 7.2.
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

Retransmissions of this request MUST cease on receipt of a 200-OK.

CMS$_I$ MUST instruct the endpoint to reserve the resources required. CMS$_I$ sends a UPDATE message to CMS$_R$ when the outcome of the resource reservation is known. This is as shown in 8.4.1.3.

CMS$_R$ MUST send a 200-OK acknowledgement to the PRACK (as in section 8.4.1.4), and use the SDP description in the INVITE message to instruct the endpoint to reserve access network resources. If successful, and after receiving a UPDATE message from CMS$_I$, CMS$_R$ MUST send to CMS$_I$ a 200-OK acknowledgement to the UPDATE (as in section 8.4.1.4) and a 200-OK final response to the INVITE(codec-change).

On sending the 200-OK, CMS$_R$ instructs the endpoint to commit the network resources (assuming the UPDATE indicated success). The endpoint MAY start sending using the new codec.

200-OK (CMS$_R$ -> CMS$_I$) Header:	**Requirements On CMS$_R$ for Message Generation** **Requirements On CMS$_I$ for Message Checking**
SIP/2.0 200 OK	As described in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On receiving a 200-OK response, CMS$_I$ instructs the endpoint to commit network resources and MAY start using the new codec. CMS$_I$ MUST send out an ACK directly to CMS$_R$. The ACK follows the rules for an ACK sent in response to 200-OK for an INVITE message:

ACK (CMS$_I$ -> CMS$_R$) Header:	**Requirements On CMS$_I$ for Message Generation** **Requirements On CMS$_R$ For Message Checking**
ACK URI SIP/2.0	As described in 6.13.
Via:	As described in 6.20.42.
Max-Forwards:	As described in 6.20.22
From:	As described in 6.12.2.
To:	
Call-ID:	
Cseq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the ACK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.5 CMS handling of Call Teardown

To terminate a call, the CMS MUST send a BYE message on the signalling channel to instruct the endpoint to stop transmitting bearer data to the other endpoint. It MUST also instruct the endpoint to release network resources used for the call.

The endpoint that has detected local hangup is denoted by CMS$_I$; the other endpoint in the call is CMS$_R$:

BYE (CMS$_I$ -> CMS$_R$) Header:	**Requirements on CMS$_I$ For Message Generation** **Requirements on CMSR For Message Checking**
BYE URI SIP/2.0	As described in 6.15.
Max-Forwards:	As described in 6.20.22.
From:	As described in 6.15.
To:	
Call-ID:	
CSeq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present

Upon receipt of the BYE message, CMS$_R$ MUST instruct the endpoint to release network resources that have been used for this call, and it MUST send the following 200-OK message in response to the BYE:

200-OK (CMS$_R$ -> CMS$_I$) Header:	**Requirements on CMS$_R$ For Message Generation** **Requirements on CMS$_I$ For Message Checking**

SIP/2.0 200 OK	As described in 6.15.
From:	
To:	
Call-ID:	
CSeq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the 200-OK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.6 Sample Implementation of Call Transfer

The user interface to initiate call transfer and ad hoc conferencing is fundamentally different for NCS MTAs as opposed to intelligent MTAs. The procedures in this document assume NCS-controlled MTAs; intelligent MTAs are outside the scope of this document. In the procedural description that follows, the following roles are identified:

- Initiator: the user who begins the call transfer process, often termed the transferor;
- CMS$_{\text{I}}$: the CMS serving the Initiator's MTA;
- Party B: the party with whom the Initiator is initially in conversation, often termed the transferee;
- CMS$_{\text{B}}$: the CMS serving Party B's MTA;
- Party C: the party to whom the Initiator wishes to transfer the call, often termed the call transfer target;
- CMS$_{\text{C}}$: the CMS serving Party C 's MTA;
- Bridge Server: a call server which owns and control conference bridges.

The call transfer procedure is as follows:

1. A first call is set up between the Initiator and Party B in the usual way. This call may have been originated by either party.

2. The Initiator performs a hook-flash, which is reported to CMS$_{\text{I}}$. The latter recognizes that the Initiator has subscribed to conferencing/call transfer and issues an UPDATE(hold) to CMS$_{\text{B}}$.

3. The Initiator is given dial tone and dials the number of Party C.

4. CMS$_{\text{I}}$ initiates a new call to a Bridge Server by sending an initial INVITE. (The call goes to the Bridge Server rather than CMS$_{\text{C}}$ because CMS$_{\text{I}}$ does not yet know whether the Initiator is invoking ad hoc conferencing or call transfer.) Steps 1-4 are shown in Figure 11.

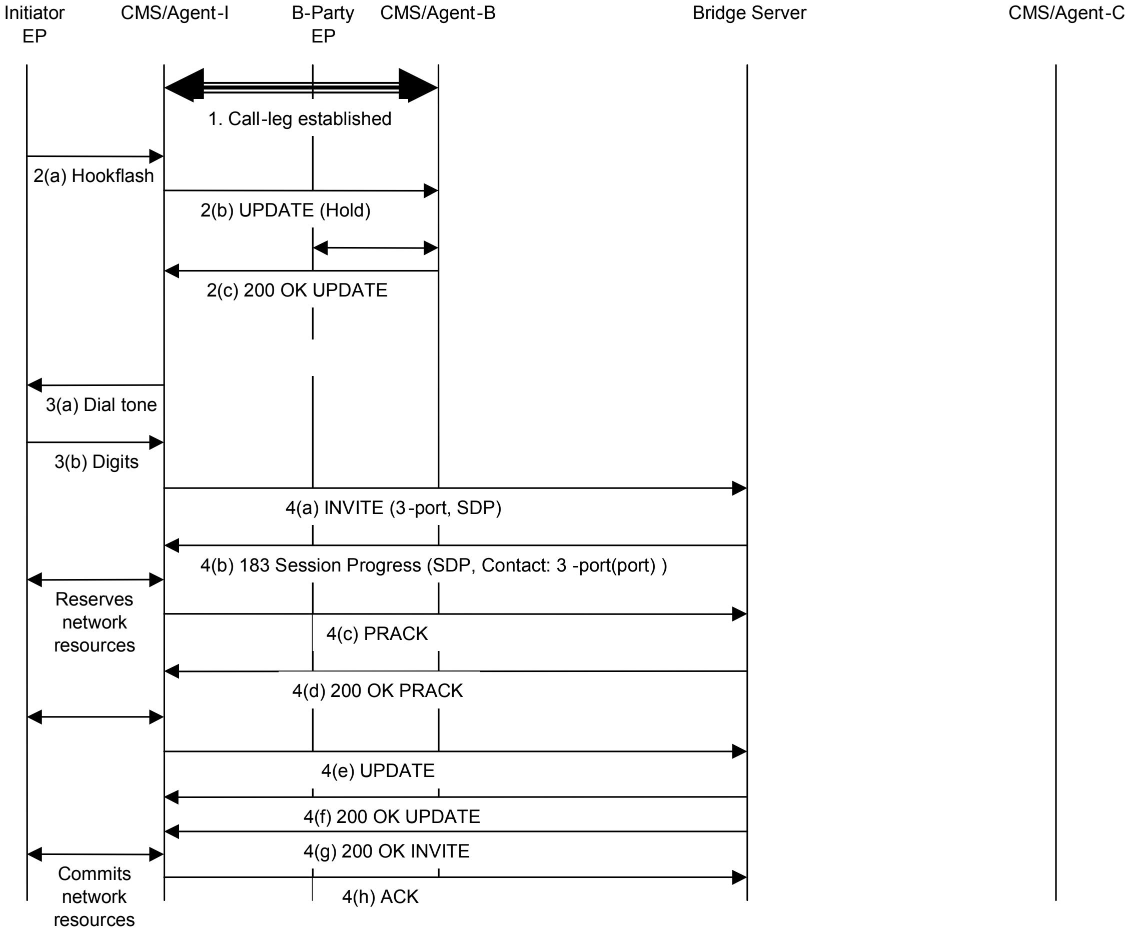

Figure 11: Initiation of Call Transfer/Ad Hoc Conference

There is a failure case (F1) where the Bridge Server is unable to accept the INVITE because no free conference circuits are available. In that case, CMS$_I$ resumes the original call between the Initiator and Party B, thereby allowing these parties to discover that the transfer attempt has failed. This failure case is shown in Figure 12 and is representative of any failure case that prevents the initial connection to the conference bridge.

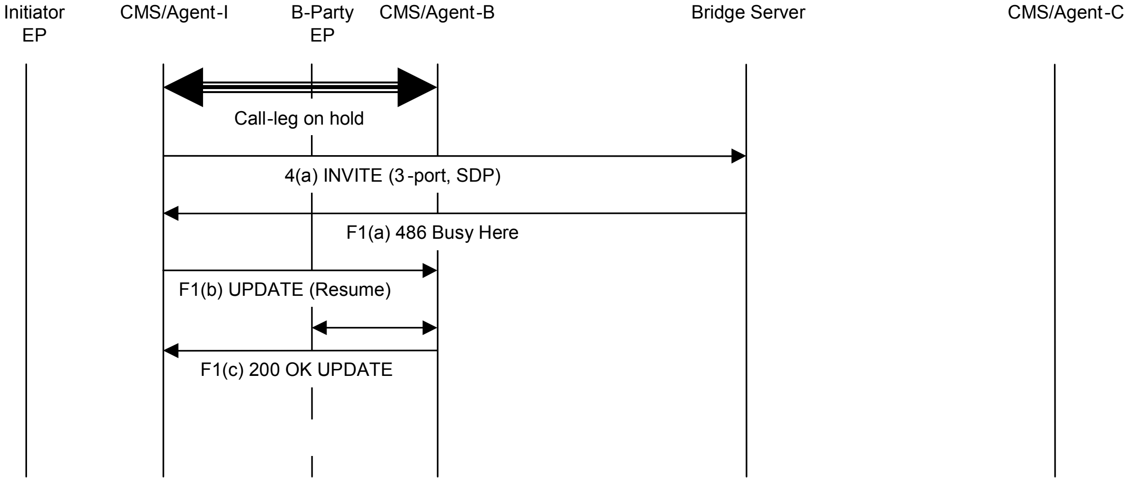

Figure 12: Failure ca– F1 – no free conference circuits

5. Once the Bridge Server has accepted the call, CMS$_I$ issues REFER to the Bridge Server, requesting that it establish a call to Party C. The Bridge Server sends a NOTIFY to CMS$_I$ and establishes the new call on the same conference bridge as the first call. During alerting, it plays ringing audio tone through the bridge to the Initiator. When Party C

answers (which could be at any one of a number of points in the following sequence of steps), the Bridge Server sends a final NOTIFY to CMS$_I$. This step is shown in Figure 13.

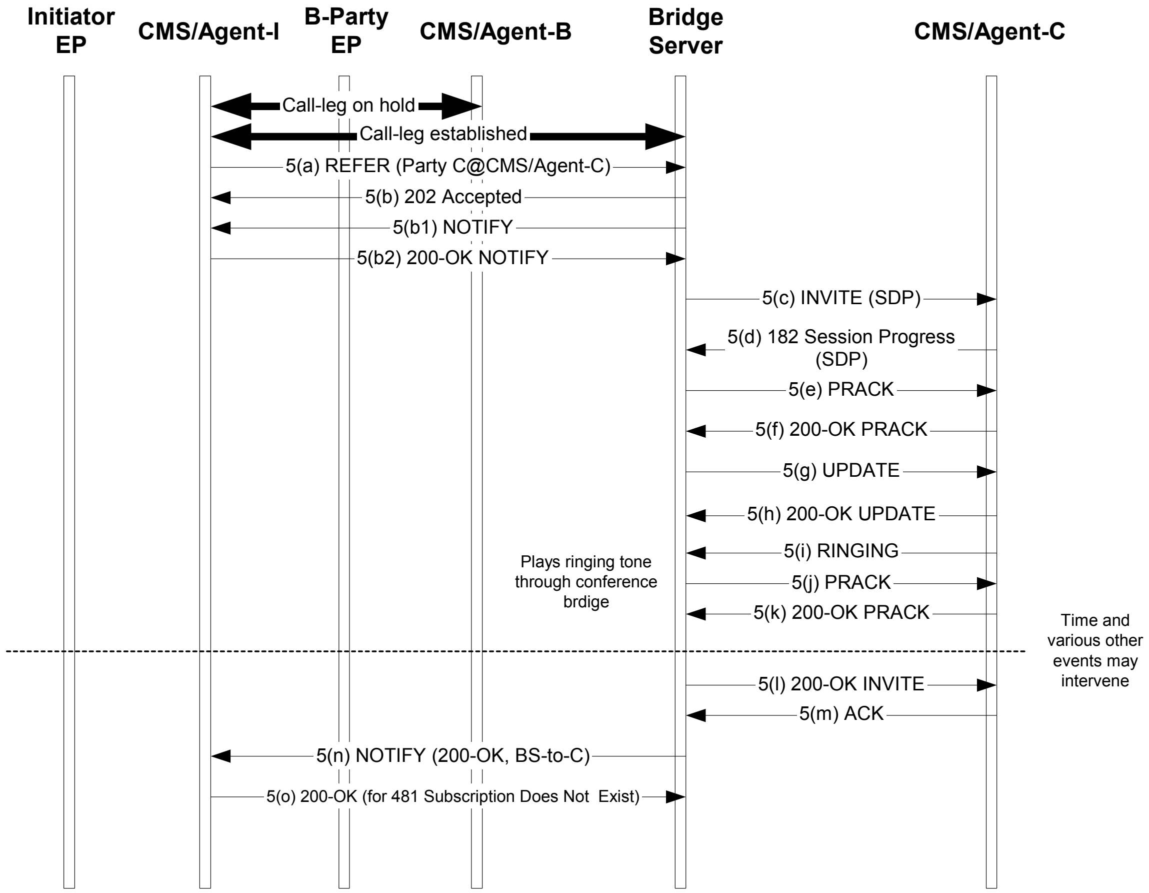

Figure 13: Establishing the leg from Bridge Server to Party C

There are several failure cases possible in this step, most of them due to abnormalities which should be handled properly but are too numerous to document here. However, there is a significant probability that Party C is busy. This case is shown as failure case F2 in Figure 14. As in the success case, the Bridge Server MUST return an immediate NOTIFY after accepting the REFER as well as a NOTIFY request to CMS$_I$, with a body containing the status line of the final response from CMS$_C$. Since this final response indicates that Party C is busy, CMS$_I$ recognizes that the transfer has failed. It tears down the connection to the Bridge Server and causes a busy tone to be played out to the Initiator. When the Initiator performs a second hook flash, CMS$_I$ restores the original direct connection to Party B.

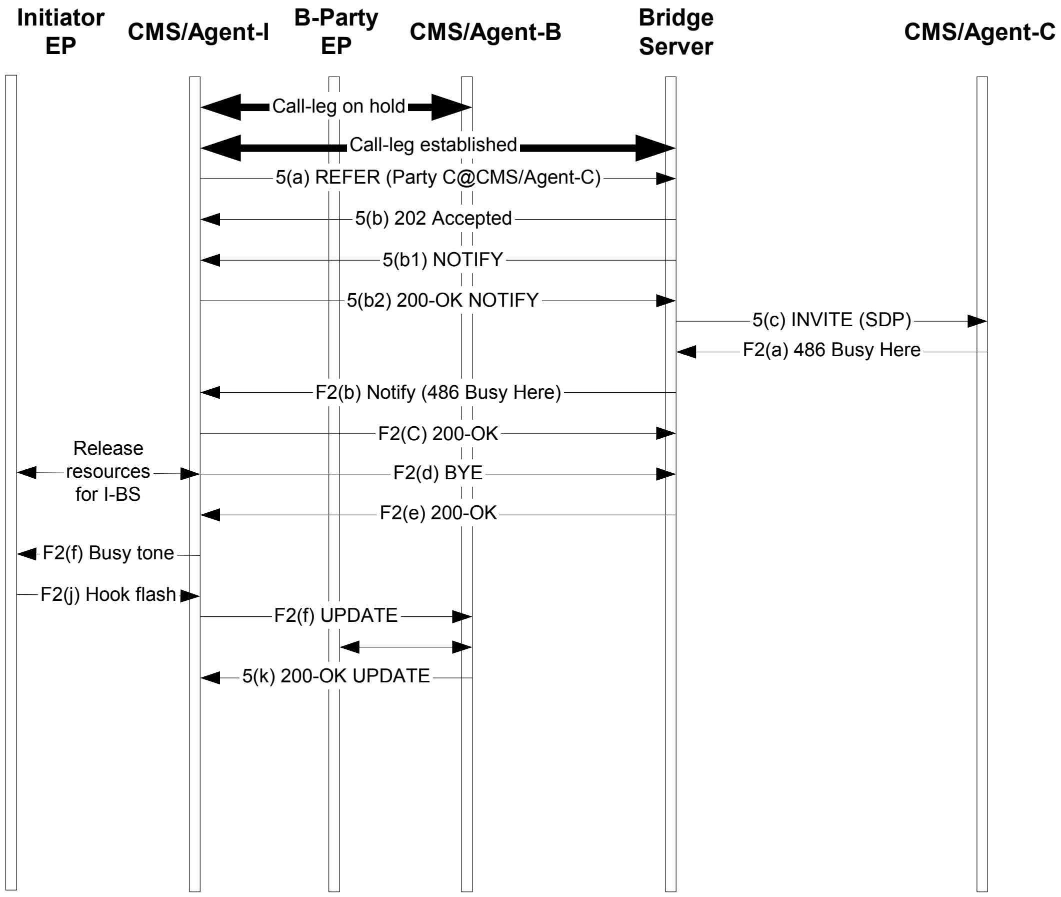

Figure 14: Failure ca– F2 – Party C busy

6. The Initiator hangs up before the Target has answered (blind call transfer) or after talking to the Target (consultative call transfer), and this is reported to the CMS$_I$. (A transfer is termed “blind” because the Transferor does not know whether the call to target will complete successfully.)

7. CMS$_I$ accepts the Initiator’s on-hook as the signal to carry out a call transfer. As a first step, it sends a REFER request to the Bridge Server to establish a call-leg to Party B on the same conference bridge as the others. The Refer-To header within the REFER request contains a Replaces header which is to be sent to Party B. Steps 6 and 7 are shown in Figure 15.

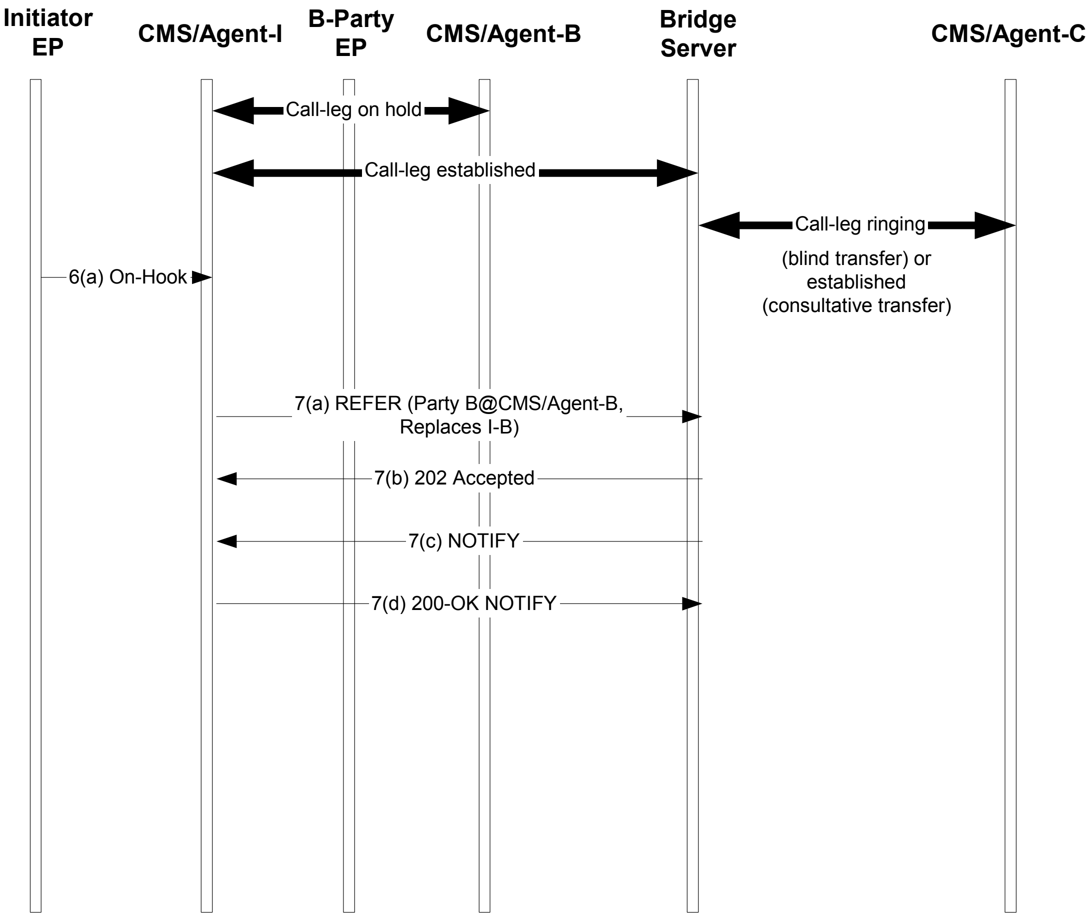

Figure 15: On-hook initiates transfer action

8. As requested, the Bridge Server sends an INVITE to CMS$_B$, containing the Replaces header. CMS$_B$ accepts the new call and drops the direct call between the Initiator and Party B. There is no alerting stage because of the call replacement. When the new call-leg is up, the Bridge Server notifies CMS$_I$ via a NOTIFY request. If the call-leg to Party C is still in the alerting stage, the Bridge Server continues to play ringing tone through the conference bridge, adding Party B as a listener. This step is shown in Figure 16.

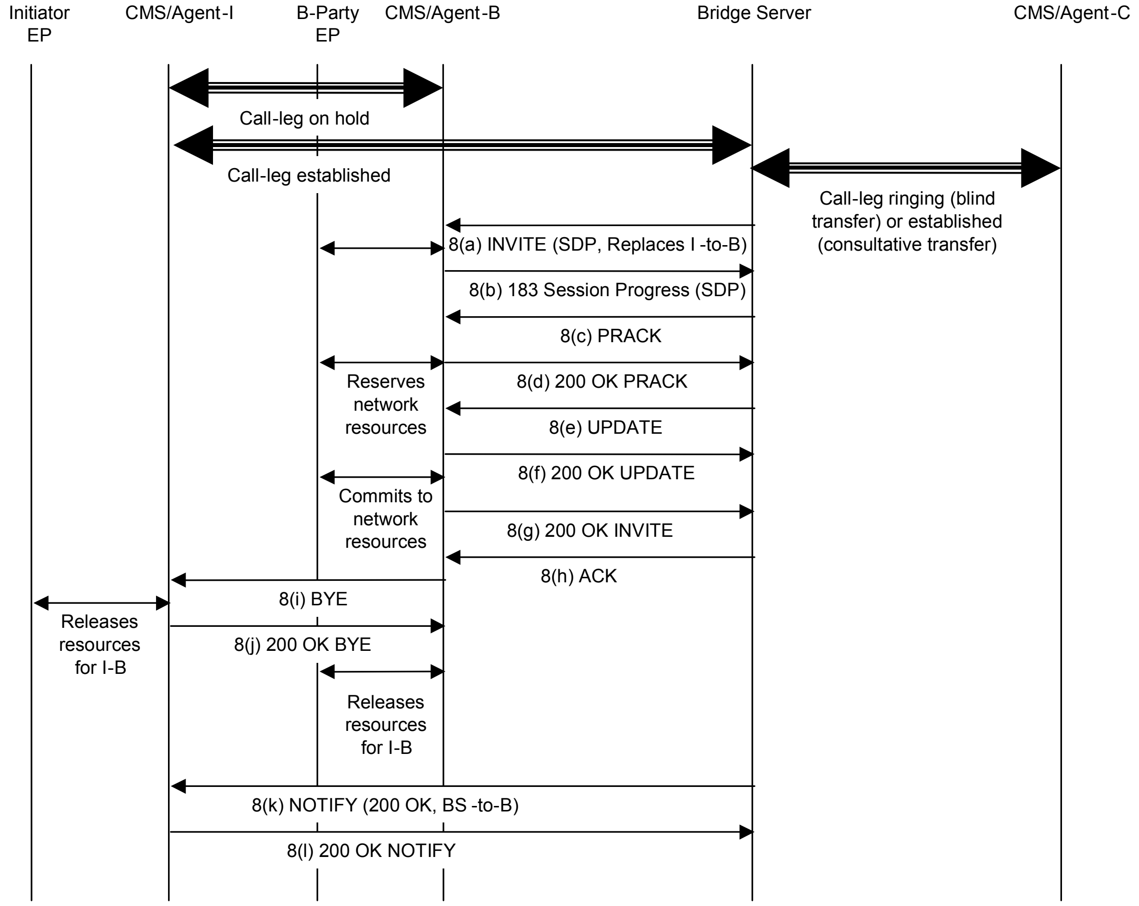
Initiator EP
CMS/Agent-I
B-Party EP
CMS/Agent-B
Bridge Server
CMS/Agent-C
Call-leg on hold
Call-leg established
Call-leg ringing (blind transfer) or established (consultative transfer)
8(a) INVITE (SDP, Replaces I -to-B)
8(b) 183 Session Progress (SDP)
8(c) PRACK
Reserves network resources
8(d) 200 OK PRACK
8(e) UPDATE
8(f) 200 OK UPDATE
Commits to network resources
8(g) 200 OK INVITE
8(h) ACK
8(i) BYE
Releases resources for I-B
8(j) 200 OK BYE
Releases resources for I-B
8(k) NOTIFY (200 OK, BS -to-B)
8(l) 200 OK NOTIFY

Figure 16: Relocation of Party B to Bridge

There are many potential failure points in this step, but all of them are due to abnormalities. The general principle should be to ensure that all call legs are cleared if communication between Party B and Party C is not possible, or to clean up all resources associated with the Initiator but leave the call between Party B and Party C via the Bridge Server in place if steps through 8(h) in Figure 16 have succeeded.

9. When CMS$_I$ receives the NOTIFY from the previous step, it tears down the call-leg to the Bridge Server. The call between Party B and Party C continues through the Bridge Server and conference bridge.[19] This step is shown in Figure 17.

[19] Possibly the Bridge Server could take intelligent action to join the two parties and leave the call, but an appropriate trigger for this action must be identified. Moreover, this introduces a race condition between call rerouting and onset of conversation between Party B and Party C.

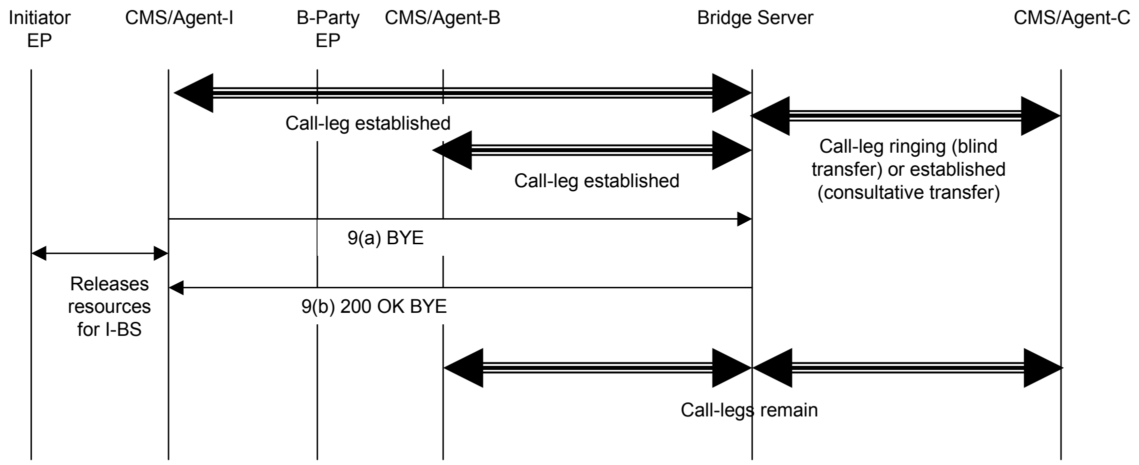

Figure 17: Transfer completed, CMS$_I$ ends involvement in call

10. When one of the remaining parties leaves the call, the Bridge Server also clears the call to the other party. This step is shown in Figure 18.

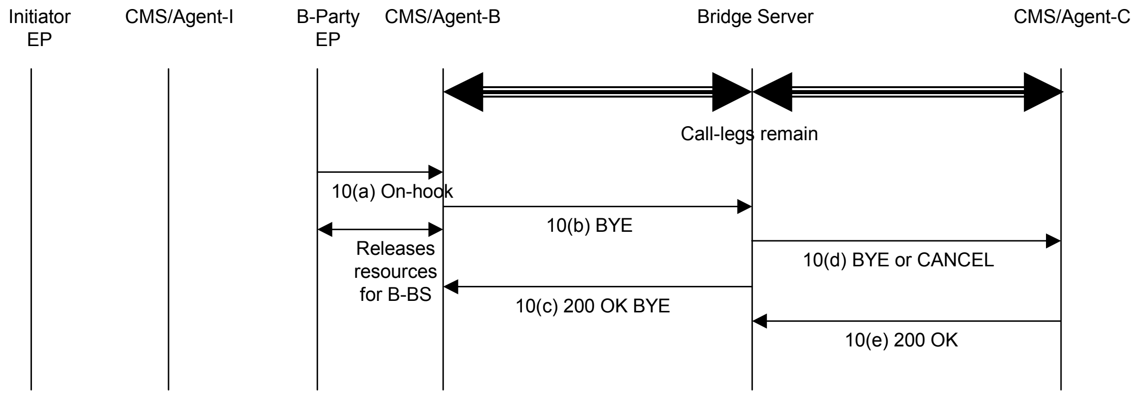

Figure 18: End of transferred call

8.4.7 Sample Implementation of Ad-hoc Conference

An ad-hoc conference is formed when the Initiator has two simultaneous active calls, one to Party B and one to Party C, and desires to connect them together. The beginning of an ad-hoc conference is as described in steps 1 to 5 and Figure 11 to Figure 14 in section 8.4.6. The difference comes in the next step:

6. The Initiator performs another hook-flash.

7. CMS$_I$ accepts the Initiator's hook-flash as the signal to create an ad hoc conference. Its first action is exactly the same as in step 7 of the call transfer procedure: CMS$_I$ sends a REFER request to the Bridge Server to establish a call-leg to Party B on the same conference bridge as the others. The Refer-To header within the REFER request contains a Replaces header which is to be sent to Party B. Except for the use of hook-flash instead of on-hook, the messaging is the same as in Figure 15.

8. The actions of the Bridge Server and CMS$_B$ in response to the REFER are identical to step 8 (Figure 16) of the call transfer procedure. The one exception to this is that when CMS$_I$ receives the NOTIFY (message 8(k) in Figure 16), it does nothing further until the Initiator goes on-hook or the Bridge Server terminates the call-leg.

8.4.8 Automatic Callback

In support of automatic callback, CMSS allows a CMS to send an OPTIONS request in order to determine the line status of the called party.

To determine the line status of an endpoint, an OPTIONS message is sent to the party whose line status is to be determined. This is a standard SIP OPTIONS request. The format of the OPTIONS message sent by the initiating CMS$_I$ and the requirements on the header fields checked at the receiving CMS$_R$ are:

OPTIONS (CMS$_I$ -> CMS$_R$) Header:	**Requirements On CMS$_I$ for Message Generation** **Requirements On CMS$_R$ for Message Checking**
OPTIONS URI SIP/2.0	As described in 6.11.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22.
From:	As described in 6.11.
To:	
Call-ID:	
CSeq:	
Content-Length:	MUST be present if the transport protocol is stream-based (e.g. TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

On receiving an OPTIONS message, CMS$_R$ MUST determine the line status of the called party. If the called party is able to accept an incoming call request, CMS$_R$ MUST return a 200-OK to CMS$_I$:

200-OK (CMS$_R$ -> CMS$_I$) Header:	**Requirement On CMS$_R$ for Message Generation** **Requirement On CMS$_O$ for Message Checking**
SIP/2.0 200 OK	As defined in 6.11.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Allow:	SHOULD be present as described in 6.11
Accept:	
Accept-Encoding:	
Accept-Language:	
Supported:	
Content-Length:	MUST be present if a body is included. Otherwise, MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.
v= o= s= c= b= t= a= m=	SDP SHOULD be present as described in 6.11.

If CMS_R determines that the endpoint is not available or is unable to accept an incoming call, an appropriate SIP error code is sent back to CMS_I. 486-Busy (if the user is already on another call and is not able to take a new call) and 480-Temporarily-Unavailable are recommended error codes. An example using 486 is shown below:

486-Busy (CMS_R -> CMS_I) Header:	Requirement for CMS_r for Message Generation Requirement for CMS_O for Message Checking
SIP/2.0 486 Busy	As defined in 6.11.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

8.4.9 Message Waiting Indicator

In support of message waiting indicator, CMSS supports the extensions defined in Section 7.10.

If a subscriber's MTA is controlled by a CMS that is different from the one controlling the subscriber's messaging system (*e.g.*, voice-mail), then CMS-to-CMS interaction is required in order to communicate the message waiting indicator to the CMS controlling the MTA.

To determine the message waiting indicator status, $CMS_{I,}$ which is controlling the subscriber's MTA, sends a SUBSCRIBE message to CMS_R, which is controlling the messaging system for that subscriber. CMS_R in turn sends a NOTIFY to CMS_I indicating the message waiting indicator status. The interface between CMS_I and the MTA, as well as the interface between CMS_R and the messaging system, is outside the scope of this document.

8.4.9.1 CMS_I Sends SUBSCRIBE to CMS_R

To subscribe to the message waiting status of a subscriber, a SUBSCRIBE message is sent to the CMS of the subscriber's messaging system. This is a standard SIP SUBSCRIBE [18] request using the "message-summary" event package defined in Section 7.10. The format of the SUBSCRIBE message sent by the initiating CMS_I and the requirements on the header fields checked at the receiving CMS_R are:

SUBSCRIBE (CMS_I -> CMS_R) Header:	Requirements On CMS_I for Message Generation Requirements On CMS_R for Message Checking
SUBSCRIBE URI SIP/2.0	As described in 7.5.
Via:	As described in 8.4.1.1.
Max-Forwards:	As defined in 8.4.1.1.
From:	As described in 8.4.1.1.
To:	
Call-ID:	
CSeq:	As defined in 8.4.1.1.
Contact:	As defined in 8.4.1.1
Event:	MUST contain "message-summary" as defined in 7.5.
Expires:	As described in 7.5.
Accept:	MUST contain "application/simple-message-summary" as defined in 7.5 and 7.10.
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present

CMS_R, upon receiving the SUBSCRIBE request, determines whether the request is for a valid subscriber. If it is not, CMS_R returns an appropriate error response and stops further processing. Otherwise, CMS_R returns a 200-OK response and continues processing as described below:

200-OK (CMS_R -> CMS_I) Header:	**Requirements On CMS_R for Message Generation Requirements On CMS_I for Message Checking**
SIP/2.0 200 OK	As described in 7.5.
Via:	
From:	
To:	
Call-ID:	
Cseq:	
Expires:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present

If the 200-OK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

8.4.9.2 CMS_R Sends NOTIFY to CMS_I

CMS_R then sends an immediate NOTIFY with the message-waiting status of the subscriber. Whenever the message-waiting status of the subscriber changes, CMS_R sends an updated NOTIFY with the message-waiting status. The NOTIFY request is described in section 7.5 and 7.10. The format of the message is:

NOTIFY (CMS_R -> CMS_I) Header:	**Requirements On CMS_R for Message Generation Requirements On CMS_I for Message Checking**
NOTIFY URL SIP/2.0	MUST be present. Method MUST be NOTIFY. The value of URL MUST be copied from the Contact header previously received in the SUBSCRIBE.
Via:	As described in 6.20.42.
Max-Forwards:	As defined in 6.20.22.
From:	As defined in 6.12.2.
To:	
Call-ID:	
Cseq:	
Contact:	As described in 6.20.10 and 7.5.
Event:	MUST contain “message-summary” as described in 7.10.
Subscription-State:	As described in 7.5.
Content-Type:	MUST be present. Type MUST be “application/simple-message-summary ".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
<Message body>	Message body MUST be present. At a minimum, MUST contain the information specified in 7.10.

If the NOTIFY message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code. Otherwise, CMS_I MUST responds with a 200-OK.

200-OK (CMS_I -> CMS_R) Header:	Requirement for CMS_I for Message Generation Requirement for CMS_R for Message Checking
SIP/2.0 200 OK	As defined in 6.12.2.
Via:	
From:	
To:	
Call-ID:	
CSeq:	
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the 200-OK message does not meet the above requirements, it MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code.

9 APPLICATION LAYER ANONYMIZER

In this section, additional detail about an application-level anonymizer that is used to support Privacy is provided.

As described earlier, a user may request three different forms of Privacy: user, name, and IP-address Privacy. In order to provide these three different types of Privacy, an application-layer anonymizer is defined, which serves the role of a trusted intermediary, as illustrated below:

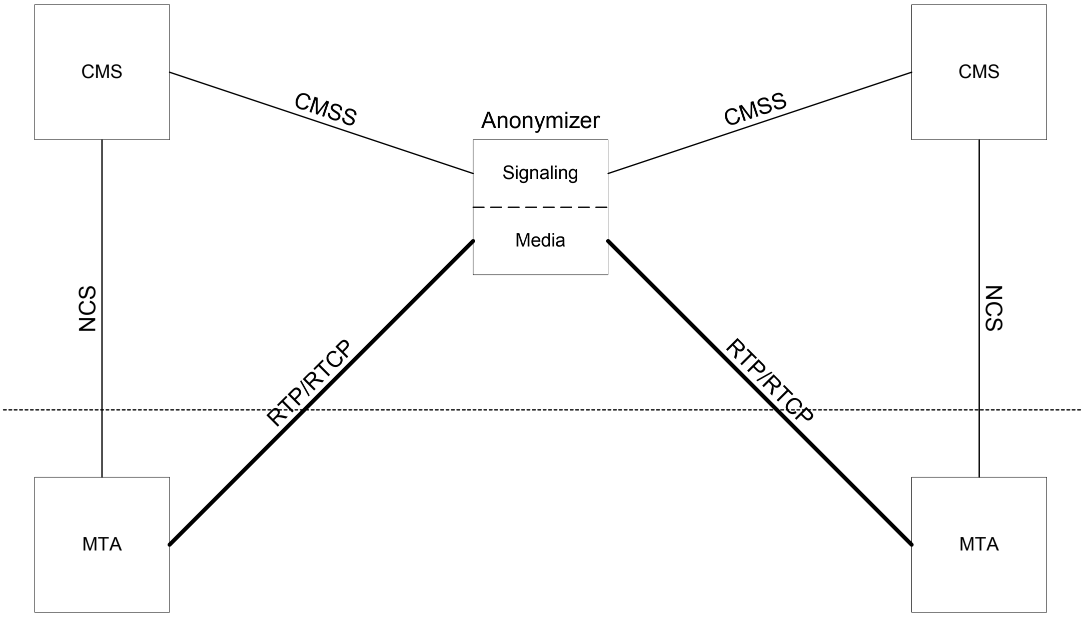

Figure 19: Application Layer Anonymizer

The anonymizer provides three Privacy functions:

Signalling content Privacy

Signalling IP address Privacy

Media IP address Privacy

all of which are illustrated in Figure 20 below

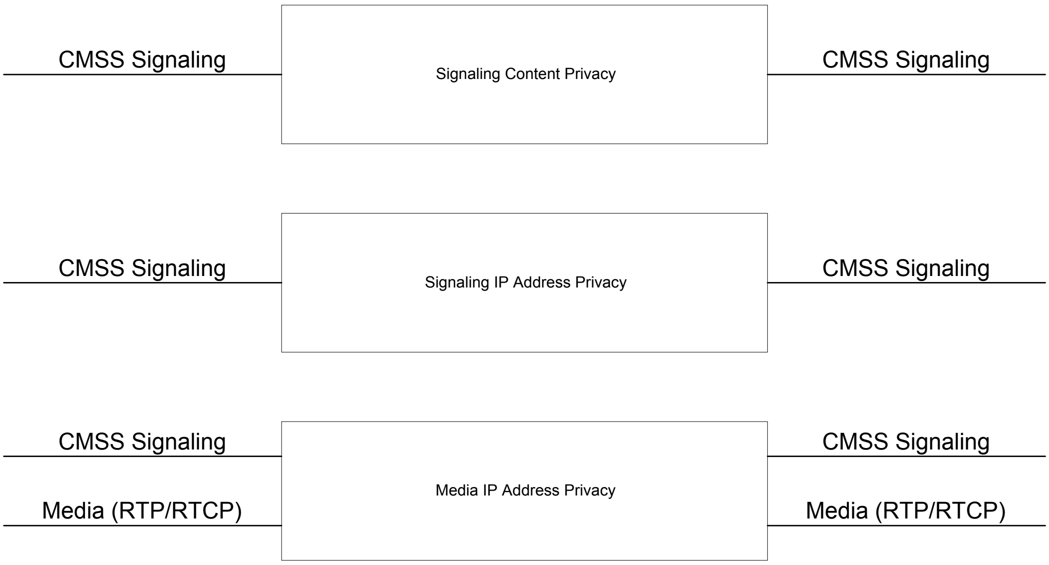

Figure 20: Anonymizer Functions.

9.1 Signalling Content Privacy

The Signalling Content Privacy function serves the role of modifying the SIP signalling messages to handle calling name and calling number Privacy as specified in Section 8.4. More specifically, the Signalling Content Privacy function MUST ensure that the Privacy requirements pertaining to the headers below are met:

Header:	Requirements for CMS$_O$
From	See Section 6.20.20
To	See Section 6.20.39
Call-ID	See section 6.20.8
Contact	See Section 6.20.10
P-Asserted-Identity	See Section 7.9

The Signalling Content function SHOULD be implemented as part of the CMS, thereby avoiding an extra hop as well as the need for a back-to-back UA in order to modify some of the above headers in accordance with the Privacy requirements.

Note that headers other than those required by CMSS, *e.g.*, Call-Info, can have Privacy implications as well. Consequently, such headers SHOULD NOT be used when Privacy is requested.

9.2 IP Address Privacy

The IP address Privacy function serves the role of modifying the SIP signalling messages to honor IP-address Privacy requests in CMSS as specified in Section 8.4. The IP address Privacy function considers IP address Privacy for both media and signalling. This allows the IP address Privacy function to be used in a variety of environments, including ones where the SIP signalling endpoints do not trust each other.

The Signalling IP address Privacy function provides IP address Privacy for the SIP signalling messages themselves. This can be achieved in a number of different ways, and the solution considered here is one where the Signalling IP address Privacy function acts as a back-to-back SIP User Agent.

All signalling IP address information will thus point to, or be based on, the Signalling IP address Privacy function rather than the requesting User Agent (CMS) itself. Calling party IP address Privacy can thus be achieved trivially by routing the session setup through the anonymizer. Called party IP address Privacy, however, is more complicated, since the calling party needs to be referred to the anonymizer and the anonymizer needs to know to forward the messages to the called party. This can be solved in a variety of ways:

> In an NCS architecture, the signalling IP address Privacy is obtained by exchanging signalling with the CMS rather than the MTA. Depending on locality of CMSes, this may or may not provide adequate Privacy. The

media IP address Privacy function is then provided by a separate entity controlled by the IP signalling Privacy function (see below).

The called party can refer subsequent transactions to the anonymizer. The anonymizer needs to be informed of the actual destination and call information for the call. The anonymizer may be informed of this through some unspecified protocol between the anonymizer and CMS.

The called party can redirect the call to the anonymizer, and provide an encrypted blob with the actual destination and call information. The encryption key used must be known to both the anonymizer and the called party unless public key cryptography is used.

Finally, the Media IP address Privacy function provides IP address Privacy for the media stream(s). This involves having the media streams going through the media IP address Privacy function, as well as modifying the SDP provided in signalling to ensure that media is actually routed through the media IP address Privacy function. As before, considered here is a solution where the media IP address Privacy function acts as a back-to-back SIP User Agent.

It should be noted that there is a tight relationship between Signalling and Media IP address Privacy. In particular, there is little reason to provide Signalling IP address Privacy without also providing Media IP address Privacy. However, in a decomposed gateway architecture (such as NCS), it is possible to provide Media IP address Privacy without Signalling IP address Privacy when the SIP signalling endpoints trust each other; this is currently the case in CMSS. In this case, the media IP address Privacy function can be further decomposed, thereby enabling it to stay out of the signalling path. This is illustrated in Figure 21 below, where it is assumed the existence of some unspecified control protocol "anon" between the control function provided in the CMS and the media anonymizer part.

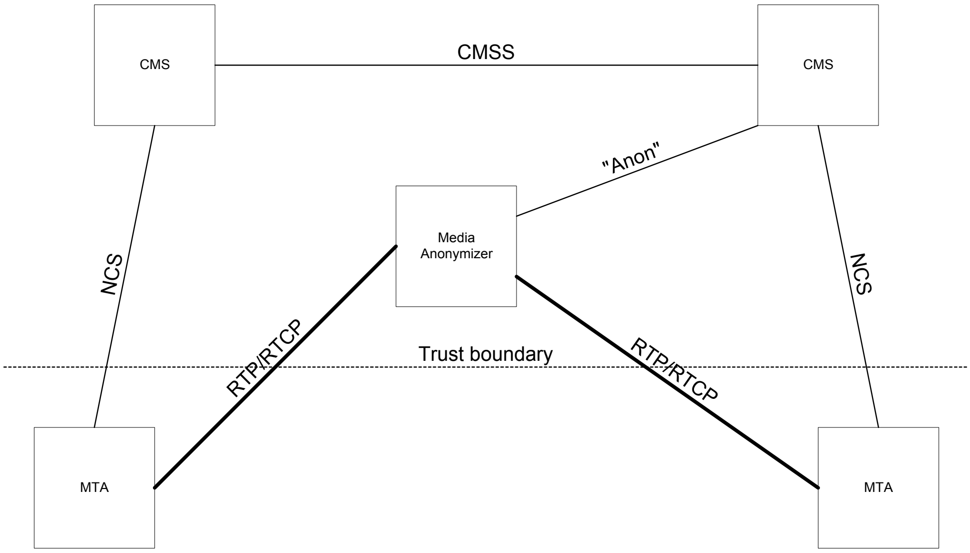

Figure 21: Privacy Issues.

The following table lists the CMSS headers and message bodies that have IP address Privacy implications and describes the IP Privacy function provided, assuming that service providers trust each other:

Header/Body Name	Requirements, Comments
From	The hostname MUST either follow the requirements in Section 6.20.20 or contain the hostname of the anonymizer. If it does not, a new conforming From header MUST be generated as described in Section 6.20.20.
Call-ID	The Call-ID MUST follow the requirements in Section 6.20.8. If it does not, a new non-revealing Call-ID MUST be generated as specified in Section 6.20.8.
Contact	The contact header MUST contain a SIP-URI of the Signalling IP Address Privacy function.
Via	The Via header MUST NOT reveal the IP-address or FQDN of any previous hop. The Via MUST contain the IP address or FQDN of the Signalling IP Address Privacy function.
Record-Route	The Record-Route header MUST NOT reveal the IP-address or FQDN of any previous hop. The Record-Route header MUST contain the IP address or FQDN of the Signalling IP Address Privacy function.
P-DCS-Billing-Info	Because of trust relation between providers, this header is not changed. An untrusted entity MUST not see this information.
SDP	The "c=" line MUST contain the IP address or FQDN of the Media IP Address Privacy function.

Since the SDP "c=" line is set to point to the "MEDIA anonymizer", RTP and RTCP messages will be directed to the anonymizer where they will be forwarded towards their destination with a source IP address of the anonymizer. Neither the RTP nor the RTCP messages will be examined for content. It is assumed that an entity seeking Privacy will not reveal any Privacy information in these messages. This implies that entities for example do not supply their name, e-mail address, etc. in RTCP.

Note that headers other than those required to be used by CMSS can have Privacy implications as well. Consequently, such headers SHOULD NOT be used when Privacy is requested, and, if they are, Privacy concerns must be addressed.

APPENDIX I - TIMER SUMMARY

CMS-CMS signalling uses a timer to provide for cleanup of call state in the event of application-level failure.

The application-level timer is used at the originating and terminating CMS during call setup to ensure that call state advances even if the other party suffers application-level failure. Action to be taken if the application-level timer expires is indicated in the tables below.

Timer Label	Approximate Duration	Timer Description
Session timers (T3) at originating CMSs and CMSs		
T-setup	5 to 6 minutes	Timer between receiving a provisional response to an INVITE and receiving a 200-OK final response. If T-setup expires before receiving the ring or final response, the CMS sends a CANCEL and aborts the call attempt.
Session timers (T3) at terminating CMSs and CMSs		
T-ringing	3 to 4 minutes	Timer between receiving an INVITE request and connect. If T-ringing expires before connect, the CMS sends a 480-Temporarily-Unavailable response and releases the reserved resources, or invokes features such as call forwarding no-answer.

APPENDIX II - CMSS MESSAGE AND HEADER OVERVIEW

This section provides an informative overview of all the SIP messages and headers that CMSS compliant implementations must support.

The first column lists the message or header in question

The second column indicates the level of support required in terms of sending the message or including the header in a message (request or response), using the following:

- Mandatory (M): There is at least one instance where the message or header must be sent by a CMSS compliant implementation.
- Recommended (R): There is no absolute requirement to send this message or header, but there is at least one instance where it is recommended to send this message or header. Note that this does not mean that support for the header is optional.
- Optional (O): A CMSS compliant implementation may send this header or message if it wants to.
- Forbidden (F): A CMSS compliant implementation must not send this message or header.

The third column indicates the level of support required in terms of receiving the message or header. The following codes are used:

- Mandatory (M): The message or header must be supported if received.
- Recommended (R): It is not absolutely required, that the message or header is supported if received, but there is at least one instance where it is recommended to support it if received. Note that this does not mean that support for the header is optional.
- Optional (O): A CMSS compliant implementation may support receiving this message or header if it wants to. In the case of an unsupported method, a 501 must be returned as specified in [40]. In the case of an unsupported header, the header is simply ignored (as specified in [40], Section 8.2.2), assuming that there were no indications to the contrary, *e.g.*, a Require or Proxy-Require field implying support was needed. In the case of an unsupported response, the response is treated as an unrecognized response as defined in [40], Section 8.1.3.2.
- Forbidden (F): If a CMSS compliant implementation receives this message or header, it must not support it. The handling is similar to unsupported optional messages and headers.

The fourth column provides a reference to the Section providing the message or header definition in CMSS. For some entries, additional comments are provided as well.

Please note that the tables below provide only an informative overview intended as a convenient reference for the reader. The tables do not define any formal requirements for CMSS compliant implementations.

RFC 3261 Requests

Request	Send	Recv	Reference and Comments
INVITE	M	M	See Section 6
ACK	M	M	See Section 6
CANCEL	M	M	See Section 6
BYE	M	M	See Section 6
OPTIONS	M	M	See Section 6
REGISTER	O	O	See Section 6

Extension Requests

Request	Send	Recv	Reference and Comments
PRACK	M	M	See Section 7.2
UPDATE	M	M	See Section 7.3
SUBSCRIBE	M	M	See Section 7.5
NOTIFY	M	M	See Section 7.5
REFER	M	M	See Section 7.6

RFC 3261 Responses

CMSS compliant implementations must support all the response codes defined in RFC 3261, except as shown below:

Response	Send	Recv	Reference and Comments
401	F	O	See Section 6.21
407	F	O	See Section 6.21

Extension Responses

Response	Send	Recv	Reference and Comments
580	M	M	See Section 7.4
687	M	M	See Section 7.8

RFC 3261 Header Fields

Header	Send	Recv	Reference and Notes
Accept	M	M	See Section 6.20.1
Accept-Encoding	O	M	See Section 6.20.2
Accept-Language	S	M	See Section 6.20.3
Alert-Info	O	O	See Section 6.20.4
Allow	M	M	See Section 6.20.5. The header value must list all supported methods, *i.e.*, at a minimum, “INVITE”, “ACK”, “CANCEL”, “BYE”, “OPTIONS”, “PRACK”, “UPDATE”, “REFER”, and “NOTIFY”.
Authentication-Info	O	O	See Section 6.20.6
Authorization	O	O	See Section 6.20.7
Call-ID	M	M	See Section 6.20.8
Call-Info	O	O	See Section 6.20.9
Contact	M	M	See Section 6.20.10
Content-Disposition	O	M	See Section 6.20.11
Content-Encoding	O	M	See Section 6.20.12
Content-Language	O	M	See Section 6.20.13
Content-Length	M	M	See Section 6.20.14
Content-Type	M	M	See Section 6.20.15 The values “application/sdp”, “message/sipfrag”, and “application/simple-message-summary” MUST be supported.
CSeq	M	M	See Section 6.20.16
Date	O	O	See Section 6.20.17
Error-Info	O	O	See Section 6.20.18
Expires	M	M	See Section 6.20.19 and Section 7.5
From	M	M	See Section 6.20.20
In-Reply-To	O	O	See Section 6.20.21
Max-Forwards	S	M	See Section 6.20.22
Min-Expires	O	O	See Section 6.20.23
MIME-Version	O	M	See Section 6.20.24
Organization	O	O	See Section 6.20.25
Priority	O	O	See Section 6.20.26
Proxy-Authenticate	O	O	See Section 6.20.27
Proxy-Authorization	O	O	See Section 6.20.28
Proxy-Require	M	M	See Section 6.20.29 The option tag "Privacy" MUST be supported in accordance with Section 7.9
Record-Route	M	M	See Section 6.20.30
Reply-To	O	O	See Section 6.20.31
Require	M	M	See Section 6.20.32. The option tags “precondition”, “replaces”, and “100rel” MUST be supported. Furthermore, the option tag "P-DCS" MAY be sent and MUST be supported if received as described in Section 7.7.3.
Retry-After	O	O	See Section 6.20.33
Route	M	M	See Section 6.20.34
Server	O	O	See Section 6.20.35
Subject	O	O	See Section 6.20.36
Supported	M	M	See Section 6.20.37 The values “precondition”, “replaces”, “100rel”, and "P-DCS" MUST be supported. However, a value present in the "Require" header SHOULD NOT also be present in the Supported header.

Header	Send	Recv	Reference and Notes
Timestamp	O	M	See Section 6.20.38
To	M	M	See Section 6.20.39
Unsupported	M	M	See Section 6.20.40
User-Agent	O	O	See Section 6.20.41
Via	M	M	See Section 6.20.42
Warning	O	O	See Section 6.20.43
WWW-Authenticate	O	O	See Section 6.20.44

Extension Header Fields

Header	Send	Recv	Reference and Notes
RAck	M	M	See Section 7.2
RSeq	M	M	See Section 7.2
Refer-To	M	M	See Section 7.5
P-DCS-OSPS	M	M	See Section 7.7
P-DCS-Billing-Info	M	M	See Section 7.7
P-DCS-Laes	M	M	See Section 7.7
P-DCS-Redirect	M	M	See Section 7.7
Replaces	M	M	See Section 7.8
P-Asserted-Identity	M	M	See Section 7.9
Privacy	M	M	See Section 7.9 The values "id" and "critical" MUST be supported.

APPENDIX III - PRIVATE SESSION INITIATION PROTOCOL (SIP) PROXY-TO-PROXY EXTENSIONS FOR SUPPORTING DCS

Status of this Memo

This document is an Internet-Draft and is in full compliance with all provisions of Section C-10 of RFC 2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

In order to deploy a residential telephone service at very large scale across different domains, it is necessary for trusted elements owned by different service providers to exchange trusted information that conveys customer-specific information and expectations about the parties involved in the call. This document describes private extensions to the Session Initiation Protocol (RFC3261) for supporting the exchange of customer information and billing information between trusted entities in the architecture described in "Architectural Considerations for Providing Carrier Class Telephony Services Utilizing SIP-based Distributed Call Control Mechanisms" (RFCXXXX) (Note: replace with RFC number of draft-dcsgroup-sipping-arch). The use of the extensions is only applicable in administrative domains, or among federations of administrative domains with previously agreed-upon policies where coordination of charging and other functions is required.

III.1. Applicability Statement

The use of the headers defined in this document is only applicable in administrative domains, or among federations of administrative domains with previously agreed-upon policies where coordination of charging and other functions is required, according to the architecture presented in [4].

III.2. Introduction

In order to deploy a SIP[2]-based residential telephone service at very large scale across different domains, it is necessary for trusted elements owned by different service providers to exchange trusted information that conveys billing information and expectations about the parties involved in the call.

There are many billing models used in deriving revenue from telephony services today. Charging for telephony services is tightly coupled to the use of network resources. It is outside the scope of this document to discuss the details of these numerous and varying methods.

A key motivating principle of the DCS architecture described in [4] is the need for network service providers to be able to control and monitor network resources; revenue may be derived from the usage of these resources as well as from the delivery of enhanced services such as telephony. Furthermore, the DCS architecture recognizes the need for coordination between call signalling and resource management. This coordination ensures that users are authenticated and authorized before receiving access to network resources and billable enhanced services.

DCS Proxies, as defined in [4], have access to subscriber information and act as policy decision points and trusted intermediaries along the call signalling path. Edge routers provide the network connectivity and resource policy enforcement mechanism and also capture and report network connectivity and resource usage information. Edge routers need to be given billing information that can be logged with Record Keeping or Billing servers. The DCS Proxy, as a central point of coordination between call signalling and resource management, can provide this information based on the authenticated identity of the calling and called parties. Since there is a trust relationship among DCS Proxies, they can be relied upon to exchange trusted billing information pertaining to the parties involved in a call. See [4] for a description of the trust boundary and trusted versus untrusted entities.

For these reasons, it is appropriate to consider defining SIP header extensions to allow DCS Proxies to exchange information during call setup. It is the intent that the extensions would only appear on trusted network segments, should be inserted upon entering a trusted network region, and removed before leaving trusted network segments.

Significant amounts of information is retrieved by an originating DCS Proxy in its handling of a connection setup request from a user agent. Such information includes location information about the subscriber (essential for emergency services calls), billing information, and station information (e.g. coin operated phone). In addition, while translating the destination number, information such as the local-number-portability office code is obtained and will be needed by all other proxies handling this call.

For Usage Accounting records, it is necessary to have an identifier that can be associated with all the event records produced for the call. The SIP Call-ID header field cannot be used as such an identifier since it is selected by the originating user agent, and may not be unique among all past calls as well as current calls. Further, since this identifier is to be used by the service provider, it should be chosen in a manner and in a format that meets the service provider’s needs.

Billing information may not necessarily be unique for each user (consider the case of calls from an office all billed to the same account). Billing information may not necessarily be identical for all calls made by a single user (consider prepaid calls, credit card calls, collect calls, etc). It is therefore necessary to carry billing information separate from the calling and called party identification. Furthermore, some billing models call for split-charging where multiple entities are billed for portions of the call.

The addition of two SIP General Header Fields allows for the capture of billing information and billing identification for the duration of the call.

It is the intent that the billing extensions would only appear on trusted network segments, and MAY be inserted by a DCS Proxy in INVITE and REFER requests and INVITE responses in a trusted network segment, and removed before leaving trusted network segments.

III.3. Trust Boundary

The DCS architecture [4] defines a trust boundary around the various systems and servers that are owned, operated by, and/or controlled by the service provider. These trusted systems include the proxies and various servers such as bridge servers, voicemail servers, announcement servers, etc. Outside of the trust boundary lie the customer premises equipment, and various application and media servers operated by third-party service providers.

Certain subscriber-specific information, such as billing and accounting information, stays within the trust boundary. Other subscriber-specific information, such as endpoint identity, may be presented to untrusted endpoints or may be withheld based on subscriber profiles.

The User Agent (UA) may be either within the trust boundary or outside the trust boundary, depending on exactly what function is being performed and exactly how it is being performed.   Accordingly, the procedures followed by a User Agent are different depending on whether the UA is within the trust boundary or outside the trust boundary.

The following sections giving procedures for User Agents therefore are subdivided into trusted user agents and untrusted user agents.

III.4. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

The term "private-URL" used in this document refers to a SIP URL that is generated by a proxy, contains a "hostport" that identifies the proxy, and contains a "userinfo" string that is generated by the proxy. The "userinfo" typically contains (or points to) information that is not to be disclosed outside the trusted domain of the proxies, such as billing account numbers, electronic surveillance indication, electronic surveillance parameters, and call redirection information. This information is either stored locally by the proxy, or encrypted with a private key known only to the proxy and encoded in a character string in the "userinfo" portion of the URL. A checksum is included in the "userinfo" data to detect tampering. The mechanism by which a proxy recognizes a "userinfo" as a private-URL and decodes and recovers the original information is local to the proxy and is not subject to standardization. Some possible implementations include an initial magic cookie (e.g. z9hG4Bk followed by the pointer/information), or use of a reserved "user" name (e.g. "private") with the optional "password" containing the pointer/information.

III.5. P-DCS-TRACE-PARTY-ID

In the telephone network, calling identity information is used to support regulatory requirements such as the Customer Originated Trace service, which provide the called party with the ability to report obscene or harassing phone calls to law enforcement. This service is provided independent of caller-id, and works even if the caller requested anonymity. The calling party is here identified as the station originating the call. In order for this service to be dependable, the called party must be able to trust that the calling identity information being presented is valid.

To initiate a customer-originated-trace from an untrusted UAC, an additional header is defined for the INVITE request sent from the untrusted UAC to its proxy. This header is called P-DCS-Trace-Party-ID, and does not appear in any other request or response. The entity addressed by the Request-URI performs the service-provider-specific functions of recording and reporting the caller identity for law enforcement action. It then forwards the call to either an announcement server or to the service-provider’s business office to collect further information about the complaint. A trusted UAC does not use this header, as it initiates this action locally.

III.5.1 Syntax

The ABNF description of this header is (missing productions can be found in [2]):

```
  P-DCS-Trace-Party-ID = "P-DCS-Trace-Party-ID" HCOLON
                                              LAQUOT addr-spec RAQUOT
```

Addr-spec contains a URL that identifies the remote endpoint. Addr-spec typically contains a tel: URL or SIP URI giving the identity of the remote endpoint, as provided in the signalling messages that established the session to be traced.

III.5.2 Procedures at an Untrusted User Agent Client (UAC)

The UAC MUST insert a P-DCS-Trace-Party-ID header into the initial INVITE message for a customer-originated-trace request. The UAC MUST use a SIP URI in the Request-URI with userinfo set to "call-trace" and hostport identifying the provisioned proxy for the untrusted UA.

III.5.3 Procedures at a Trusted User Agent Client (UAC)

A trusted UAC performs the customer-originated-trace in a manner similar to the originating proxy, described below. A trusted UAC MUST NOT include this header in any request.

III.5.4 Procedures at a User Agent Server (UAS)

This header MUST NOT appear in any response sent by a UAS.

III.5.6 Procedures at Proxy

Two sets of proxy procedures are defined: (1) the procedures at an originating proxy, and (2) the procedures at a terminating proxy.
The originating proxy is a proxy that received the INVITE request from a non-trusted endpoint.

The terminating proxy is a proxy that sends the INVITE request to a non-trusted endpoint.

A proxy that both receives the INVITE request from an untrusted endpoint, and sends the INVITE request to an untrusted endpoint, performs both sets of procedures.

III.5.6.1 Procedures at Originating Proxy

If the P-DCS-Trace-Party-ID header is present in the initial INVITE request from the UAC, and the Request-URI of the INVITE has userinfo set to "call-trace" and hostport set to the originating proxy, the originating proxy MUST perform the service-provider-specific functions of recording and reporting the caller identity for law enforcement action. The proxy then MUST direct the call to either an announcement server or to the service-provider’s business office to collect further information about the complaint.

The originating proxy MUST remove the P-DCS-Trace-Party-ID header from the INVITE before sending the request to another proxy or UAS.

III.5.6.2 Procedures at Terminating Proxy

This header MUST NOT appear in any request or response sent by a terminating proxy.

III.6. P-DCS-GATE

The P-DCS-Gate header extension is used only on requests and responses between proxies and trusted user agents. It is never sent to, nor sent by, an untrusted UA.

The proxy-proxy signalling establishes a synchronization path that may be required by the PacketCable Dynamic Quality of Service (D-QoS) specification [5] to coordinate the release of resources of the call. As per the D-QoS specification, the CMTS monitors the packet flow, and generates a Gate-Close message in response to either an explicit close request from the MTA/RGW, or when an equipment or facility failure causes the connection to be broken. This Gate-Close message is directed either to the local CMS/Agent (trusted UA), to the remote CMS/Agent (trusted UA), or to the CMTS serving the remote MTA, depending on the capabilities of the endpoints. When a CMS/Agent receives such a Gate-Close message, it considers it identical to a call termination request.

The P-DCS-Gate header is used between proxies and/or trusted user agents, and conveys the location of the remote gate, identity of the gate, and the security key and ciphersuite to be used in gate coordination messages [5, 6].

III.6.1 Syntax

The ABNF description of the DCS-Gate header is as follows (missing productions can be found in [2] and [3]):

```
P-DCS-Gate          = "P-DCS-Gate" HCOLON hostport
                              SEMI Gate-ID-param *(SEMI Gate-param)
Gate-ID-param       = "gate" EQUAL Gate-ID
Gate-ID             = 1*HEXDIG
Gate-param          = Gate-Key-param / Gate-Cipher-param /
                              Gate-Strength-param / generic-param
Gate-Key-param      = "key" EQUAL Gate-Key
Gate-Key            = 1*HEXDIG
Gate-Cipher-param   = "cipher" EQUAL Gate-CipherSuite
Gate-CipherSuite    = token
Gate-Strength-param = "coord" EQUAL Gate-Strength-token
Gate-Strength-token = "required" / "optional"
```

Hostport gives the IP address or FQDN of the CMTS/EdgeRouter that enforces the QoS, or the trusted endpoint system (CMS/Agent) that simulates the gate coordination exchange on behalf of an edge router.

Gate-ID is a token used at the system named in the hostport parameter to identify the particular session. For DCS systems, it is a 32-bit quantity encoded as an 8-character string.

Gate-Key is a character string that provides keying information to the system named in the hostport parameter. The method of deriving the actual keys for the gate coordination messages, and the security procedures, are beyond the scope of this document.

Gate-CipherSuite is a character string that gives the type of encryption algorithm that will be used to secure the gate coordination messages [6].

Gate-Strength-Token specifies whether the gate coordination is required or optional for the current session. Its use is described in the following sections.

III.6.2 Procedures at an Untrusted User Agent Client (UAC)

This header MUST NOT be sent to an untrusted UAC, and MUST NOT be sent by an untrusted UAC.

III.6.3 Procedures at a Trusted User Agent Client (UAC)

A UAC located within the trust boundary of the service provider MUST perform the functions given in section III.6.6.1.

III.6.4 Procedures at an Untrusted User Agent Server (UAS)

This header is never sent to an untrusted UAS, and is never sent by an untrusted UAS.

III.6.5 Procedures at a Trusted User Agent Server (UAS)

A UAS located within the trust boundary of the service provider performs the functions given in section III.6.6.2.

III.6.6 Procedures at Proxy

The P-DCS-Gate header MUST NOT appear in any message other than the initial INVITE request, or in the first reliable non-100 response to that request. The proxy MUST remove the P-DCS-Gate header in any request or response sent to an untrusted endpoint.

Two sets of proxy procedures are defined: (1) the procedures at an originating proxy, and (2) the procedures at a terminating proxy.
The originating proxy is a proxy that receives the INVITE request from a non-trusted endpoint.

The terminating proxy is a proxy that sends the INVITE request to a non-trusted endpoint.

A proxy that both receives the INVITE request from an untrusted endpoint, and sends the INVITE request to a untrusted endpoint, performs both sets of procedures.

A proxy that is neither an originating proxy nor a terminating proxy has no function in coordinating the commitment of resources.

III.6.6.1 Procedures at Originating Proxy

The originating proxy MUST insert a P-DCS-Gate header in the initial INVITE message for a new call.

The originating proxy MUST identify the system that will perform gate coordination on the originating side (either the proxy itself, or the CMTS controlling the media flow to the endpoint), and the identification token used at that system to identify the call. It MUST insert the IP address or FQDN of that system in the hostport parameter of the P-DCS-Gate header, and the identification token as the Gate-ID.

The originating proxy MUST pick a security key and cipher suite for the gate coordination message exchange, and MUST insert these values in the P-DCS-Gate header.

If the system that will perform gate coordination is a CMTS, the strength token MUST be set to "required". If the system that will perform gate coordination is the proxy itself, the strength token MAY be set to "optional", or omitted. Gate coordination procedures are described in [5].

III.6.6.2 Procedures at Terminating Proxy

The terminating proxy MUST identify the system that will perform gate coordination on the terminating side (either the proxy itself, or the CMTS controlling the media flow to the endpoint), and the identification token used at that system to identify the call. Gate coordination will be required for this call if (1) the strength token in the P-DCS-Gate header in the initial INVITE indicates "required", or (2) the system that will perform gate coordination at the destination is a CMTS.

If gate coordination is required for this call, the terminating proxy MUST include a P-DCS-Gate header in the first reliable non-100 response to the initial INVITE request. It MUST insert the IP address or FQDN of the system that will perform gate coordination in the hostport parameter of the DCS-Gate header, and the identification token as the Gate-ID.

If gate coordination is not required for this call, the terminating proxy SHOULD NOT include a P-DCS-Gate header in the first reliable non-100 response (or any other response) to the initial INVITE request. Gate coordination procedures are described in [5].

III.7. P-DCS-OSPS

Some calls have special call processing requirements that may not be satisfied by normal user agent call processing. For example, when a user is engaged in a call and another call arrives, such a call might be rejected with a busy indication. However, some PSTN operator services require special call processing. In particular, the Busy Line Verification (BLV) and Emergency Interrupt (EI) services initiated by an operator from an Operator Services Position System (OSPS) on the PSTN network have such a need. Similarly, emergency calls to a 9-1-1 PSAP may result in trunk signalling causing a howling tone or sustained ring on the originating line.

In order to inform the SIP user agent that special treatment should be given to a call, we use a new P-DCS-OSPS header field, which may be set to a value indicating when a special type of call processing is requested. We define three values in this header, namely "BLV" for busy line verification, "EI" for emergency interrupt, and "RING" for howling/sustained ring.

If the user agent decides to honor such a request, the response of the user agent to an INVITE with either "BLV" or "EI" will not be a busy indication. Since "EI" and "RING" only occur on established dialogs, they may also appear in UPDATE requests.

III.7.1 Syntax

The ABNF description of the P-DCS-OSPS header is as follows (missing productions can be found in [2]):

```
P-DCS-OSPS  = "P-DCS-OSPS" HCOLON OSPS-Tag
OSPS-Tag    = "BLV" / "EI" / "RING" / token
```

The OSPS-Tag value of "token" is defined for extensibility, and is reserved for future use.

III.7.2 Procedures at an Untrusted User Agent Client (UAC)

The P-DCS-OSPS header MUST NOT be sent in a request from an untrusted UAC.

III.7.3 Procedures at a Trusted User Agent Client (UAC)

This header is typically only inserted by a Media Gateway Controller [4] that is controlling a Media Gateway with special trunks to a PSTN OSPS system or PSAP. This trunk group is usually referred to as a BLV-trunk group and employs special signalling procedures that prevent inadvertent use. Calls originating at the PSTN OSPS system are sent over this trunk group, and result in an INVITE request with the OSPS header.

This header MAY be sent in an INVITE request, and MUST NOT appear in any message other than those listed below.

OSPS-Tag value "BLV" MUST NOT appear in any request or response other than an initial INVITE request establishing a new dialog.

OSPS-Tag value "EI" MUST NOT appear in any request or response other than (1) a subsequent INVITE within a pre-existing dialog established with the OSPS-Tag value of "BLV", or (2) an UPDATE request within a pre-existing dialog established with the OSPS-Tag value of "BLV".

OSPS-Tag value "RING" MUST NOT appear in any request or response other than (1) a subsequent INVITE within a pre-existing dialog established by a UAC to a PSAP, or (2) an UPDATE request within a pre-existing dialog established by a UAC to a PSAP.

III.7.4 Procedures at an Untrusted User Agent Server (UAS)

If the UAS receives an INVITE request with an OSPS-Tag of "BLV", dialog identification that matches an existing dialog, and the existing call was not established with the OSPS-Tag, it MUST reject the request with a 409-Conflict error code.

If the UAS receives an INVITE/UPDATE request with an OSPS-Tag value of "EI" or "RING", with dialog identification that does not match an existing dialog, it MUST reject the request with a 409-Conflict error code.

If the UAS receives an INVITE that contains an OSPS-Tag value of "BLV" and is not willing to cooperate in offering this service, it MUST reject the request with a 403-Forbidden error code.

The UAS SHOULD NOT reject an INVITE with a BLV OSPS-Tag due to a busy condition. The UAS MUST NOT respond with a 3xx-Redirect error code to an INVITE with a BLV OSPS-Tag. The UAS SHOULD NOT alert the user of the incoming call attempt if the BLV OSPS-Tag is present in the INVITE.

If an INVITE with OSPS-Tag of "BLV" is accepted (meeting all QoS pre-conditions, etc.), the UAS MUST send an audio stream on this connection to the address and port given in the SDP of the INVITE. The UAS MAY perform a mixing operation between the two ends of an active call and send the resulting media stream to the address and port indicated. Alternatively, the UAS MAY send a copy of the local voice stream, and (if no activity on the local voice stream) send a copy of the received voice stream. If the state of the UAS is idle, the UAS SHOULD send a stream of silence packets to

OSPS. If the state of the UAS is ringing or ringback, the UAS SHOULD send a ringback stream to OSPS.

If an INVITE/UPDATE with OSPS-Tag of "EI" is accepted, the UAS MUST enable communication between the UAC and the local user. The UAS MAY put any existing call on hold, or initiate an ad-hoc conference.

If an INVITE/UPDATE with OSPS-Tag of "RING" is accepted, the UAS MUST generate a 3-second howling tone or a sustained ring, depending on the state of the user equipment.

III.7.5 Procedures at a Trusted User Agent Server (UAS)

The procedures at a trusted UAS MUST be identical to those described in III.7.4.

III.7.6 Procedures at Proxy

In the DCS architecture, the OSPS is considered a trusted UAC. If a proxy receives a P-DCS-OSPS header in a request from an untrusted source, it MUST reject the request.

A proxy that implements a call-forwarding service MUST NOT respond to an INVITE request with a 3xx response, if the request contained the P-DCS-OSPS header.

III.8. P-DCS-BILLING-INFO

There are many billing models used in deriving revenue from telephony services today. Charging for telephony services is tightly coupled to the use of network resources. It is outside the scope of this document to discuss the details of these numerous and varying methods.

Proxies have access to subscriber information and act as policy decision points and trusted intermediaries along the call signalling path. Edge routers provide the network connection and resource policy enforcement mechanism and also capture and report network connection and resource usage information. Edge routers need to be given billing information that can be logged with Record Keeping or Billing servers. The proxy, as a central point of coordination between call signalling and resource management, can provide this information based on the authenticated identity of the calling and called parties. Since there is a trust relationship among proxies, they can be relied upon to exchange trusted billing information pertaining to the parties involved in a call.

For Usage Accounting records, it is necessary to have an identifier that can be associated with all the event records produced for the call. The SIP Call-ID header field cannot be used as such an identifier since it is selected by the originating user agent, and may not be unique among all past calls as well as current calls. Further, since this identifier is to be used by the service provider, it should be chosen in a manner and in a format that meets the service provider's needs.

Billing information may not necessarily be unique for each user (consider the case of calls from an office all billed to the same account). Billing information may not necessarily be identical for all calls made by a single user (consider prepaid calls, credit card calls, collect calls, etc). It is therefore necessary to carry billing information separate from the calling and called party identification. Furthermore, some billing models call for split-charging where multiple entities are billed for portions of the call.

The addition of a SIP General Header Field allows for the capture of billing information and billing identification for the duration of the call.

It is the intent that the billing extensions would only appear on trusted network segments, and MAY be inserted by a proxy or trusted UA in INVITE requests in a trusted network segment, and removed before leaving trusted network segments. The P-DCS-Billing-Info header extension is used only on requests and responses between proxies and trusted User Agents. It is never sent to, nor sent by, an untrusted UA.

III.8.1 Syntax

The DCS-Billing-Info header is defined by the following ABNF (missing productions can be found in [2]):

```
P-DCS-Billing-Info   = "P-DCS-Billing-Info" HCOLON
                                Billing-Correlation-ID SLASH FEID
                                *(SEMI Billing-Info-param)
Billing-Correlation-ID     = 1*48(HEXDIG)
FEID                 = 1*16(HEXDIG) "@" host
Billing-Info-param   = RKS-Group-ID-param / Charge-param /
                Calling-param / Called-param /
                Routing-param / Loc-Routing-param /
                generic-param
RKS-Group-ID-param   = "rksgroup" EQUAL RKS-Group-ID
RKS-Group-ID         = token
Charge-param         = "charge" EQUAL Acct-Charge-URI
Acct-Charge-URI            = LAQUOT URI RAQUOT
Calling-param        = "calling" EQUAL Acct-Calling-URI
Acct-Calling-URI     = LAQUOT URI RAQUOT
Called-param         = "called" EQUAL Acct-Called-URI
Acct-Called-URI            = LAQUOT URI RAQUOT
Routing-param        = "routing" EQUAL Acct-Routing-URI
Acct-Routing-URI     = LAQUOT URI RAQUOT
Loc-Routing-param    = "locroute" EQUAL Acct-Loc-Routing-URI
Acct-Loc-Routing-URI= LAQUOT URI RAQUOT
```

The P-DCS-billing-Info extension contains an identifier that can be used by an event recorder to associate multiple usage records, possibly from different sources, with a billable account. It further contains the subscriber account information, and other information necessary for accurate billing of the service. This header is only used between proxies and trusted User Agents.

The Billing-Correlation-ID is specified in [7] as a 24-byte binary structure, containing 4 bytes of NNTP timestamp, 8 bytes of the unique identifier of the network element that generated the ID, 8 bytes giving the time zone, and 4 bytes of monotonically increasing sequence number at that network element. This identifier is chosen to be globally unique within the system for a window of several months. This MUST be encoded in the P-DCS-Billing-Info header as a hexadecimal string of up to 48 characters. Leading zeroes may be suppressed.

The Financial Entity ID (FEID) is specified in [7] as an 8-byte structure, containing the financial identifier for that domain, followed by a domain name. FEID can be associated with a type of service and could be assigned to multiple domains by the same provider. A domain could contain multiple assigned FEIDs. This 8-byte structure MUST be encoded in the P-DCS-Billing-

Info header as a hexadecimal string of up to 16 characters. Trailing zeroes may be suppressed.

The RKS-Group-ID specifies a record keeping server (or group of cooperating servers) for event messages relating to this call. It is used to control certain optimizations of procedures when multiple event message streams are being sent to the same Record Keeping Server.

Additional parameters contain the information needed for generation of event message records. Acct-Charge-URI, Acct-Calling-URI, Acct-Called-URI, Acct-Routing-URI, and Acct-Location-Routing-URI are each defined as URLs; they should all contain tel: URLs with E.164 formatted addresses. These fields are further defined in [7].

III.8.2 Procedures at an Untrusted User Agent Client (UAC)

This header is never sent to an untrusted UAC, and is never sent by an untrusted UAC.

III.8.3 Procedures at a Trusted User Agent Client (UAC)

The UAC MUST generate the Billing-Correlation-ID for the call, and insert it into the P-DCS-Billing-Info header in the initial INVITE message sent to the terminating proxy, along with the charging information for the call. The UAC MUST include its FEID, and the RKS-Group-ID for the Record-Keeping-Server being used by the UAC. If the UAC performed a LNP query, it MUST include the Routing Number and Location Routing Number returned by the query.

If the response to the initial INVITE is a 3xx-Redirect, the UAC generates a new initial INVITE request to the destination specified in the Contact: header, as per standard SIP. If a UAC receives a 3xx-Redirect response to an initial INVITE, the new INVITE generated by the UAC MUST contain the P-DCS-Billing-Info header from the 3xx-Redirect response. If the UAC is acting as a B2BUA, instead of generating a new INVITE it MAY generate a private-URL and place it in the Contact header of a 3xx-Redirect response sent to the originating endpoint. This private-URL MUST contain (or contain a pointer to) the P-DCS-Billing-Info value, which indicates the charging arrangement for the new call, and an expiration time very shortly in the future, to limit the ability of the originator to re-use this private-URL for multiple calls.

A UAC that includes a Refer-to header in a REFER request MUST include a P-DCS-Billing-Info header in the Refer-to's URL. This P-DCS-Billing-Info header MUST include the accounting information of the initiator of the REFER.

III.8.4 Procedures at an Untrusted User Agent Server (UAS)

This header is never sent to an untrusted UAS, and is never sent by an untrusted UAS.

III.8.5 Procedures at a Trusted User Agent Server (UAS)

The UAS MUST include a P-DCS-Billing-Info header in the first reliable non-100 response to an initial INVITE message. This P-DCS-Billing-Info header MUST include the Billing-Correlation-ID generated by the UAS, the FEID of the UAS, and the RKS-Group-ID of the Record-Keeping-Server being used by the UAS. The UAS MAY change the values of Acct-Charge-URI if it wishes to override the billing information that was present in the INVITE (e.g. for a toll-free call). The decision to do this and the contents of the new Acct-Charge-URI MUST be determined by service provider policy provisioned in the

UAS. If the UAS performed a LNP query, it MUST include the Routing Number and Location Routing Number returned by the query.

The UAS MUST add a P-DCS-Billing-Info header to a 3xx-redirect response to an initial INVITE, giving the accounting information for the call forwarder, for the call segment from the destination to the forwarded-to destination.

III.8.6 Procedures at Proxy

Three sets of proxy procedures are defined: (1) the procedures at an originating proxy, (2) the procedures at a terminating proxy, and (3) the procedures at a tandem proxy.

The originating proxy is a proxy that received the INVITE request from a non-trusted endpoint.

The terminating proxy is a proxy that sends the INVITE request to a non-trusted endpoint.

A proxy that is neither an originating proxy, nor a terminating proxy, is a tandem proxy.

For purposes of mid-call changes, such as call transfers, the proxy that receives the request from a non-trusted endpoint is considered the initiating proxy; the proxy that sends the request to a non-trusted endpoint is considered the recipient proxy. Procedures for the initiating proxy are included below with those for originating proxies, while procedures for the recipient proxy are included with those for terminating proxies.

A proxy that both receives the INVITE request from an untrusted endpoint, and sends the INVITE request to a non-trusted endpoint, performs both sets of procedures.

III.8.6.1 Procedures at Originating Proxy

The originating proxy MUST generate the Billing-Correlation-ID for the call, and insert it into the P-DCS-Billing-Info header in the initial INVITE message sent to the terminating proxy, along with the charging information for the call. The originating proxy MUST include its FEID, and the RKS-Group-ID for the Record-Keeping-Server being used by the originating proxy. If the originating proxy performed a LNP query, it MUST include the Routing Number and Location Routing Number returned by the query. Any P-DCS-Billing-Info header present from an untrusted UA MUST be removed.

If the Request-URI contains a private-URL, and the decoded username contains billing information, the originating proxy MUST generate a P-DCS-Billing-Info header with that decrypted information. Otherwise, the originating proxy MUST determine the accounting information for the call originator, and insert a P-DCS-Billing-Info header including that information.

If the response to the initial INVITE is a 3xx-Redirect, received prior to a 18x-Ringing, the originating proxy generates a new initial INVITE request to the destination specified in the Contact: header, as per standard SIP. If an originating proxy receives a 3xx-Redirect response to an initial INVITE prior to a 18x-Ringing response, the INVITE generated by the proxy MUST contain the P-DCS-Billing-Info header from the 3xx-Redirect response.

If the response to the initial INVITE is a 3xx-Redirect, received after a 18x-Ringing, the originating proxy generates a private-URL and places it in the Contact header of a 3xx-Redirect response sent to the originating endpoint. This private-URL MUST contain (or contain a pointer to) the P-DCS-Billing-Info value, which indicate the charging arrangement for the new call, and an expiration time very shortly in the future, to limit the ability of the originator to re-use this private-URL for multiple calls.

An originating proxy that processes a REFER request from an untrusted UA MUST include a P-DCS-Billing-Info header in the Refer-to's URL. This P-DCS-Billing-Info header MUST include the accounting information of the initiator.

III.8.6.2 Procedures at Terminating Proxy

The terminating proxy MUST NOT send the P-DCS-Billing-Info header to an untrusted destination.

The terminating proxy MUST include a P-DCS-Billing-Info header in the first reliable non-100 response to an initial INVITE message. This P-DCS-Billing-Info header MUST include the Billing-Correlation-ID generated by the terminating proxy, the FEID of the terminating proxy, and the RKS-Group-ID of the Record-Keeping-Server being used by the terminating proxy. The terminating proxy MAY change the values of Acct-Charge-URI if it wishes to override the billing information that was present in the INVITE (e.g. for a toll-free call). The decision to do this and the contents of the resulting P-DCS-Billing-Info header MUST be determined by service provider policy provisioned in the terminating proxy. If the terminating proxy performed a LNP query, it MUST include the Routing Number and Location Routing Number returned by the query.

The terminating proxy MUST add P-DCS-Billing-Info headers to a 3xx-redirect response to an initial INVITE, giving the accounting information for the call forwarder, for the call segment from the destination to the forwarded-to destination.

A proxy receiving a mid-call REFER request that includes a Refer-to header generates a private-URL and places it in the Refer-to header sent to the endpoint. This private-URL MUST contain the P-DCS-Billing-Info value, which indicate the charging arrangement for the new call, and an expiration time very shortly in the future, to limit the ability of the endpoint to re-use this private-URL for multiple calls.

III.8.6.3 Procedures at Tandem Proxy

If the tandem proxy performed a LNP query, it MUST insert the Routing Number and Location Routing Number returned by the query into the P-DCS-Billing-Info header in the first reliable non-100 response.

III.9. P-DCS-LAES and P-DCS-REDIRECT

The P-DCS-Laes extension contains the information needed to support Lawfully Authorized Electronic Surveillance. This header contains the address and port of an Electronic Surveillance Delivery Function for delivery of a duplicate stream of event messages related to this call. The header may also contain an additional address and port for delivery of call content. Security key information is included to enable pairs of Delivery Functions to securely exchange surveillance information. This header is only used between proxies and trusted User Agents.

The P-DCS-Redirect extension contains call identifying information needed to support the requirements of Lawfully Authorized Electronic Surveillance of redirected calls. This header is only used between proxies and trusted User Agents.

III.9.1 Syntax

The format of the P-DCS-Laes header is given by the following ABNF (missing productions can be found in [2] and [3]):

```
P-DCS-LAES    = "P-DCS-LAES" HCOLON Laes-sig
          *(SEMI Laes-param)
Laes-sig      = hostport
Laes-param    = Laes-content / Laes-key / generic-param
Laes-content        = "content" EQUAL hostport
Laes-key      = "key" EQUAL token
P-DCS-Redirect      = "P-DCS-Redirect" HCOLON Called-ID
                         *(redir-params)
Called-ID     = LAQUOT URI RAQUOT
redir-params        = redir-uri-param / redir-count-param /
                  generic-param
redir-uri-param     = "redirector-uri" EQUAL Redirector
Redirector    = LAQUOT URI RAQUOT
redir-count-param = "count" EQUAL Redir-count
Redir-count   = 1*DIGIT
```

The values of Laes-sig and Laes-content are addresses of the Electronic Surveillance Delivery Function, and used as the destination address for call-identifying information and call-content, respectively. Laes-key is a string generated by the proxy that is used by the Delivery Function to securely transfer information between them [6].

The P-DCS-Redirect header contains redirection information. The redir-uri-param indicates the original destination requested by the user (e.g. dialed number), the Redirector indicates the new destination, and the Redir-count indicates the number of redirections that have occurred.

III.9.2 Procedures at an Untrusted User Agent Client (UAC)

This header MUST NOT be sent to an untrusted UAC, and MUST NOT be sent by an untrusted UAC.

III.9.3 Procedures at a Trusted User Agent Client (UAC)

The UAC checks for an outstanding lawfully authorized surveillance order for the originating subscriber, and, if present, includes this information in the Authorization for Quality of Service or signals this information to the device performing the intercept (e.g. a Media Gateway).

If the P-DCS-LAES header is present in the first reliable non-100 response (indicating surveillance is required on the terminating subscriber, but that the terminating equipment is unable to perform that function), the UAC MUST include this information in the Authorization for Quality of Service, or MUST signal this information to the device performing the intercept (e.g. a Media Gateway).

If a 3xx-Redirect response is received to the initial INVITE request, and if a P-DCS-LAES header is present in the 3xx response, the UAC MUST include that header unchanged in the reissued INVITE. The UAC MUST also include a P-DCS-Redirect header containing the original dialed number, the new destination number, and the number of redirections that have occurred.

A UAC that includes a Refer-to header in a REFER request, when the originating subscriber has an outstanding lawfully authorized surveillance order, MUST include a P-DCS-Laes header attached to the Refer-to. The P-DCS-

LAES header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call's event messages, MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be intercepted, and MUST include a random string for use as a security key between the Delivery Functions.

The trusted UAC MUST NOT send the P-DCS-Laes and P-DCS-Redirect headers to an untrusted entity.

III.9.4 Procedures at an Untrusted User Agent Server (UAS)

This header MUST NOT be sent to an untrusted UAS, and MUST NOT be sent by an untrusted UAS.

III.9.5 Procedures at a Trusted User Agent Server (UAS)

The UAS checks for an outstanding lawfully authorized surveillance order for the terminating subscriber. If present, the UAS includes this information in the authorization for Quality of Service.

If the terminating equipment is unable to perform the required surveillance (e.g. if the destination is a voicemail server), the UAS MUST include a P-DCS-LAES header in the first reliable non-100 response requesting the originating proxy to perform the surveillance. The P-DCS-LAES header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call's event messages, MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be intercepted, and MUST include a random string for use as a security key between the Delivery Functions.

If the response to the initial INVITE request is a 3xx-Redirect response, and there is an outstanding lawfully authorized surveillance order for the terminating subscriber, the UAS MUST include a P-DCS-Laes header in the 3xx-Redirect response, with contents as described above.

The trusted UAS MUST NOT send the P-DCS-Laes and P-DCS-Redirect headers to an untrusted entity.

III.9.6 Procedures at Proxy

Two sets of proxy procedures are defined: (1) the procedures at an originating proxy, and (2) the procedures at a terminating proxy.
The originating proxy is a proxy that received the INVITE request from a non-trusted endpoint.

The terminating proxy is a proxy that sends the INVITE request to a non-trusted endpoint.

For purposes of mid-call changes, such as call transfers, the proxy that receives the request from a non-trusted endpoint is considered the initiating proxy; the proxy that sends the request to a non-trusted endpoint is considered the recipient proxy. Procedures for the initiating proxy are included below with those for originating proxies, while procedures for the recipient proxy are included with those for terminating proxies.

A proxy that both receives the INVITE request from an untrusted endpoint, and sends the INVITE request to a non-trusted endpoint, MUST NOT generate P-DCS-Laes nor P-DCS-Redirect headers.

A proxy that is neither an originating proxy nor a terminating proxy has no function in manipulating existing calls.

III.9.6.1 Procedures at Originating Proxy

The Originating Proxy MUST remove any P-DCS-Laes and P-DCS-Redirect headers in requests or responses to or from an untrusted proxy or UA.

The originating proxy checks for an outstanding lawfully authorized surveillance order for the originating subscriber, and, if present, includes this information in the Authorization for Quality of Service or signals this information to the device performing the intercept (e.g. a Media Gateway).

If the P-DCS-LAES header is present in the first reliable non-100 response (indicating surveillance is required on the terminating subscriber, but that the terminating equipment is unable to perform that function), the originating proxy MUST include this information in the Authorization for Quality of Service, or MUST signal this information to the device performing the intercept (e.g. a Media Gateway).

If the Request-URI in an initial INVITE request contains a private-URL, the originating proxy MUST decrypt the userinfo information to find the real destination for the call, and other special processing information. If electronic surveillance information is contained in the decrypted userinfo, the originating proxy MUST generate a P-DCS-LAES header with the surveillance information.

If a 3xx-Redirect response is received to the initial INVITE request prior to a 18x-Ringing, and if a P-DCS-LAES header is present in the 3xx response, the originating proxy MUST include that header unchanged in the reissued INVITE. The originating proxy MUST also include a P-DCS-Redirect header containing the original dialed number, the new destination number, and the number of redirections that have occurred.

If a 3xx-Redirect response is received to the initial INVITE request after a 18x-Ringing, the originating proxy generates a private-URL and places it in the Contact header of a 3xx-Redirect response sent to the originating endpoint. If a P-DCS-Laes header is present in the 3xx response, this private-URL MUST contain (1) the electronic surveillance information from the 3xx-Redirect response, (2) the original destination number, (3) the identity of the redirecting party, and (4) the number of redirections of this call.

An originating proxy that processes a REFER request from an untrusted UA, when the originating subscriber has an outstanding lawfully authorized surveillance order, MUST include a P-DCS-Laes header in the Refer-to’s URL. The P-DCS-LAES header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call’s event messages, MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be intercepted, and MUST include a random string for use as a security key between the Delivery Functions.

An initiating proxy that sends a mid-call REFER request including a Refer-to header, when the initiating subscriber has an outstanding lawfully authorized surveillance order, MUST include a P-DCS-Laes header in the Refer-to’s URL.

The originating proxy MUST NOT send the P-DCS-Laes and P-DCS-Redirect headers to an untrusted entity.

III.9.6.2 Procedures at Terminating Proxy

The Terminating Proxy MUST remove any P-DCS-Laes and P-DCS-Redirect headers in requests or responses to or from an untrusted proxy or UA.

The terminating proxy checks for an outstanding lawfully authorized surveillance order for the terminating subscriber. If present, the terminating proxy includes this information in the authorization for Quality of Service.

The terminating proxy MUST NOT send the P-DCS-Laes and P-DCS-Redirect headers to an untrusted entity, either as headers in the request or response, or as headers attached to URIs in the request or response.

If the terminating equipment is unable to perform the required surveillance (e.g. if the destination is a voicemail server), the terminating proxy MUST include a P-DCS-LAES header in the first reliable non-100 response requesting the originating proxy to perform the surveillance. The P-DCS-LAES header MUST include the address and port of the local Electronic Surveillance Delivery Function for a copy of the call’s event messages, MUST include the address and port of the local Electronic Surveillance Delivery Function for the copy of call content if call content is to be intercepted, and MUST include a random string for use as a security key between the Delivery Functions.

If the response to the initial INVITE request is a 3xx-Redirect response, and there is an outstanding lawfully authorized surveillance order for the terminating subscriber, the terminating proxy MUST include a P-DCS-Laes header in the 3xx-Redirect response, with contents as described above.

A proxy receiving a mid-call REFER request that includes a Refer-to header with a P-DCS-Laes header attached MUST generate a private-URL and place it in the Refer-to header sent to the endpoint. This private-URL MUST contain the P-DCS-Laes information from the attached header.

III.10. Security Considerations

QoS gate coordination, billing information, and electronic surveillance information are all considered to be sensitive information that MUST be protected from eavesdropping and furthermore require integrity checking. It is therefore necessary that the trusted UAs and proxies take precautions to protect this information from eavesdropping and tampering. Use of IPSec or TLS between Proxies is RECOMMENDED.

III.11. IANA Considerations

This document defines a number of SIP extension headers, which should be included in the registry of SIP headers defined in [2]. Registration information for new headers is as follows:

```
Header Field Name:   P-DCS-Trace-Party-ID
   RFC Number: XXXX [number of this document]
   Compact Form:     none

Header Field Name:   P-DCS-Gate
   RFC Number: XXXX [number of this document]
   Compact Form:     none
```

```
Header Field Name:   P-DCS-OSPS
  RFC Number: XXXX [number of this document]
  Compact Form:      none

Header Field Name:   P-DCS-Billing-Info
  RFC Number: XXXX [number of this document]
  Compact Form:      none

Header Field Name:   P-DCS-Laes
  RFC Number: XXXX [number of this document]
  Compact Form:      none

Header Field Name:   P-DCS-Redirect
  RFC Number: XXXX [number of this document]
  Compact Form:      none
```

III.12. Notice Regarding Intellectual Property Rights

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

III.13. Normative References

1. Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

2. Rosenberg, J, et al, "The Session Initiation Protocol", RFC 3261, June 2002.

3. Crocker, D., and Overell P., "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

III.14. Informative References

4· DCS Group, "Architectural Considerations for Providing Carrier Class Telephony Services Utilizing SIP-based Distributed Call Control Mechanisms", draft-dcsgroup-sipping-arch-00.txt, June 2002.

5· PacketCable Dynamic Quality of Service Specification, pkt-sp-dqos-i03-020116. January, 2002.

6· PacketCable Security Specification, pkt-sp-sec-i05-020116. January, 2002.

7. PacketCable Event Message Specification, pkt-sp-em-i03-011221. December, 2001.

III.15. Acknowledgements

The Distributed Call Signalling work in the PacketCable project is the work of a large number of people, representing many different companies. The authors would like to recognize and thank the following for their assistance: John Wheeler, Motorola; David Boardman, Daniel Paul, Arris Interactive; Bill Blum, Jon Fellows, Jay Strater, Jeff Ollis, Clive Holborow, Motorola; Doug Newlin, Guido Schuster, Ikhlaq Sidhu, 3Com; Jiri Matousek, Bay Networks; Farzi Khazai, Nortel; John Chapman, Bill Guckel, Michael Ramalho, Cisco; Chuck Kalmanek, Doug Nortz, John Lawser, James Cheng, Tung-Hai Hsiao, Partho Mishra, AT&T; Telcordia Technologies; and Lucent Cable Communications.

Previous versions further acknowledged, as co-authors, several people for providing the text of this document. They are: K. K. Ramakrishnan (kkrama@research.att.com), AT&T; Ed Miller (edward.miller@terayon.com), Terayon; Glenn Russell (G.Russell@Cablelabs.com), CableLabs; Burcak Beser (burcak@juniper.net) Juniper Networks, Mike Mannette (Michael-Mannette@3com.com) and Kurt Steinbrenner (Kurt-Steinbrenner@3com.com), 3Com; Dave Oran (oran@cisco.com), Cisco Systems; John Pickens (jpickens@com21.com), Com21; Poornima Lalwaney (poornima.lalwaney@nokia.com), Nokia; Jon Fellows (jfellows@coppermountain.com), Copper Mountain Networks; Doc Evans (n7dr@arrisi.com) Arris, and Keith Kelly (keith@netspeak.com), NetSpeak.

III.16. Editor's Addresses

Bill Marshall
AT&T
Florham Park, NJ 07932
Email: wtm@research.att.com

Flemming Andreasen
Cisco
Edison, NJ
Email: fandreas@cisco.com

APPENDIX IV - THE SIP REFER METHOD - DRAFT-IETF-SIP-REFER-06

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 15, 2003.

Copyright Notice

Abstract

This document defines the REFER method. This SIP extension requests that the recipient REFER to a resource provided in the request. It provides a mechanism allowing the party sending the REFER to be notified of the outcome of the referenced request. This can be used to enable many applications, including call transfer.

In addition to the REFER method, this document defines the refer event package and the Refer-To request header.

IV.1. Overview

This document defines the REFER method. This SIP [1] extension requests that the recipient REFER to a resource provided in the request. This can be used to enable many applications, including Call Transfer.

IV.2. The REFER Method

REFER is a SIP method as defined by RFC3261 [1]. The REFER method indicates that the recipient (identified by the Request-URI) should contact a third party using the contact information provided in the request.

Unless stated otherwise, the protocol for emitting and responding to a REFER request are identical to those for a BYE request in [1]. The behavior of SIP entities not implementing the REFER (or any other unknown) method is explicitly defined in [1].

A REFER request implicitly establishes a subscription to the refer event. Event subscriptions are defined in [2].

A REFER request MAY be placed outside the scope of a dialog created with an INVITE. REFER creates a dialog, and MAY be Record-Routed, hence MUST contain a single Contact header field value. REFERs occurring inside an existing dialog MUST follow the Route/Record-Route logic of that dialog.

IV.2.1 The Refer-To Header Field

Refer-To is a request header field (request-header) as defined by [1]. It only appears in a REFER request. It provides a URL to reference.

Refer-To = ("Refer-To" / "r") HCOLON (name-addr / addr-spec) *(SEMI generic-param)

The following should be interpreted as if it appeared in Table 3 of RFC 3261.

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Refer-To	R		-	-	-	-	-	-

The Refer-To header field MAY be encrypted as part of end-to-end encryption.

The Contact header field is an important part of the Route/Record-Route mechanism and is not available to be used to indicate the target of the reference.

Examples

Refer-To: sip:alice@atlanta.com
Refer-To:
Refer-To:
Refer-To:
Refer-To: http://www.ietf.org

Long headers field values are line-wrapped here for clarity only.

IV.2.2 Header Field Support for the REFER Method

This table adds a column to tables 2 and 3 in [1], describing header field presence in a REFER method. See [1] for a key for the symbols used. A row for the Refer-To request-header should be inferred, mandatory for REFER. Refer-To is not applicable for any other methods. The proxy column in [1] applies to the REFER method unmodified.

Header	Where	REFER
Accept	R	o
Accept	2xx	-
Accept	415	c
Accept-Encoding	R	o
Accept-Encoding	2xx	-
Accept-Encoding	415	c
Accept-Language	R	o
Accept-Language	2xx	-
Accept-Language	415	c
Alert-Info		-
Allow	Rr	o
Allow	405	m
Authentication-Info	2xx	o
Authorization	R	o
Call-ID	c	m
Call-Info		-
Contact	R	m
Contact	1xx	-
Contact	2xx	m
Contact	3-6xx	o
Content-Disposition		o
Content-Encoding		o
Content-Language		o
Content-Length		o
Content-Type		*
CSeq	c	m
Date		o
Error-Info	3-6xx	o
Expires	R	o
From	c	m
In-Reply-To		-
Max-Forwards	R	m
Min-Expires		-
MIME-Version		o
Organization		o
Priority	R	-
Proxy-Authenticate	401	o

Proxy-Authenticate	407	m
Proxy-Authorization	R	o
Proxy-Require	R	o
Record-Route	R	o
Record-Route	2xx,18x	o
Reply-To		-
Require		c
Retry-After	404,413,480,486	o
Retry-After	500,503	o
Retry-After	600,603	o
Route	R	c
Server	r	o
Subject	R	-
Supported	R,2xx	o
Timestamp		o
To	c(1)	m
Unsupported	420	o
User-Agent		o
Via	c(2)	m
Warning	r	o
WWW-Authenticate	401	m
WWW-Authenticate	407	o

Table 1: Header Field Support

IV.2.3 Message Body Inclusion

A REFER method MAY contain a body. This specification assigns no meaning to such a body. A receiving agent may choose to process the body according to its Content-Type.

IV.2.4 Behavior of SIP User Agents

IV.2.4.1 Forming a REFER request

REFER is a SIP request and is constructed as defined in [1]. A REFER request MUST contain exactly one Refer-To header field value.

IV.2.4.2 Processing a REFER request

A UA accepting a well-formed REFER request SHOULD request approval from the user to proceed (this request could be satisfied with an interactive query or through accessing configured policy). If approval is granted, the UA MUST contact the resource identified by the URI in the Refer-To header field value as discussed in Section IV.2.4.3.

If the approval sought above for a well-formed REFER request is immediately denied, the UA MAY decline the request.

An agent responding to a REFER method MUST return a 400 (Bad Request) if the request contained zero or more than one Refer-To header field values.

An agent (including proxies generating local responses) MAY return a 100 (Trying) or any appropriate 4xx-6xx class response as prescribed by [1].

Care should be taken when implementing the logic that determines whether or not to accept the REFER request. A UA not capable of accessing non-SIP URIs SHOULD NOT accept REFER requests to them.

If no final response has been generated according to the rules above, the UA MUST return a 202 Accepted response before the REFER transaction expires.

If a REFER request is accepted (that is, a 2xx class response is returned), the recipient MUST create a subscription and send notifications of the status of the refer as described in Section IV.2.4.4.

IV.2.4.3 Accessing the Referred-to Resource

The resource identified by the Refer-To URI is contacted using the normal mechanisms for that URI type. For example, if the URI is a SIP URI indicating INVITE (using a method=INVITE URI parameter for example), the UA would issue a new INVITE using all of the normal rules for sending an INVITE defined in [1].

IV.2.4.4 Using SIP Events to Report the Results of the Reference

The NOTIFY mechanism defined in [2] MUST be used to inform the agent sending the REFER of the status of the reference. The dialog identifiers (To, From, and Call-ID) of each NOTIFY must match those of the REFER as they would if the REFER had been a SUBSCRIBE request.

Each NOTIFY MUST contain an Event header field with a value of refer and possibly an id parameter (see Section IV.2.4.6).

Each NOTIFY MUST contain a body of type "message/sipfrag" [3].

The creation of a subscription as defined by [1] always results in an immediate NOTIFY. Analogous to the case for SUBSCRIBE described in that document, the agent that issued the REFER MUST be prepared to receive a NOTIFY before the REFER transaction completes.

The implicit subscription created by a REFER is the same as a subscription created with a SUBSCRIBE request. The agent issuing the REFER can terminate this subscription prematurely by unsubscribing using the mechanisms described in [2]. Terminating a subscription, either by explicitly unsubscribing or rejecting NOTIFY, is not an indication that the referenced request should be withdrawn or abandoned. In particular, an agent acting on a REFER request SHOULD NOT issue a CANCEL to any referenced SIP requests because the agent sending the REFER terminated its subscription to the refer event before the referenced request completes.

The agent issuing the REFER may extend its subscription using the subscription refresh mechanisms described in [2].

REFER is the only mechanism that can create a subscription to event refer. If a SUBSCRIBE request for event refer is received for a subscription that does not already exist, it MUST be rejected with a 403.

Notice that unlike SUBSCRIBE, the REFER transaction does not contain a duration for the subscription in either the request or the response. The lifetime of the state being subscribed to is determined by the progress of the referenced request. The duration of the subscription is chosen by the agent accepting the REFER and is communicated to the agent sending the REFER in the subscription's initial NOTIFY (using the Subscription-State expires header parameter). Note that agents accepting REFER and not wishing to hold subscription state can terminate the subscription with this initial NOTIFY.

IV.2.4.5 The Body of the NOTIFY

Each NOTIFY MUST contain a body of type "message/sipfrag" [3]. The body of a NOTIFY MUST begin with a SIP Response Status-Line as defined in [1]. The response class in this status line indicates the status of the referred action. The body MAY contain other SIP header fields to provide information about the outcome of the referenced action. This body provides a complete statement of the status of the referred action. The refer event package does not support state deltas.

If a NOTIFY is generated when the subscription state is pending, its body should consist only of a status line containing a response code of 100.

A minimal, but complete, implementation can respond with a single NOTIFY containing either the body:

SIP/2.0 100 Trying

if the subscription is pending, the body:

SIP/2.0 200 OK

if the reference was successful, the body:

SIP/2.0 503 Service Unavailable

if the reference failed, or the body:

SIP/2.0 603 Declined

if the REFER request was accepted before approval to follow the reference could be obtained and that approval was subsequently denied (see Section IV.2.4.7).

An implementation MAY include more of a SIP message in that body to convey more information. Warning header field values received in responses to the referred action are good candidates. In fact, if the reference was to a SIP URI, the entire response to the referenced action could be returned (perhaps to assist with debugging). However, doing so could have grave security repercussions (see Section IV.5). Implementers must carefully consider what they choose to include.

Note that if the reference was to a non-SIP URI, status in any NOTIFYs to the referrer must still be in the form of SIP Response Status-Lines. The minimal implementation discussed above is sufficient to provide a basic indication of success or failure. For example, if a client receives a REFER to a HTTP URL, and is successful in accessing the resource, its NOTIFY to the referrer can contain the message/sipfrag body of "SIP/2.0 200 OK". If the notifier wishes to return additional non-SIP protocol specific information about the status of the request, it may place it in the body of the sipfrag message.

IV.2.4.6 Multiple REFER Requests in a Dialog

A REFER creates an implicit subscription sharing the dialog identifiers in the REFER request. If more than one REFER is issued in the same dialog (a second attempt at transferring a call for example), the dialog identifiers do not provide enough information to associate the resulting NOTIFYs with the proper REFER.

Thus, for the second and subsequent REFER requests a UA receives in a given dialog, it MUST include an id parameter[2] in the Event header field of each NOTIFY containing the sequence number (the number from the CSeq header field value) of the REFER this NOTIFY is associated with. This id parameter MAY be included in NOTIFYs to the first REFER a UA receives in a given dialog. A SUBSCRIBE sent to refresh or terminate this subscription MUST contain this id parameter.

IV.2.4.7 Using the Subscription-State Header Field with Event Refer

Each NOTIFY must contain a Subscription-State header field as defined in [2]. The final NOTIFY sent in response to a REFER MUST indicate the subscription has been "terminated" with a reason of "noresource". (The resource being subscribed to is the state of the referenced request).

If a NOTIFY indicates a reason that indicates a re-subscribe is appropriate according to [2], the agent sending the REFER is NOT obligated to re-subscribe.

In the case where a REFER was accepted with a 202, but approval to follow the reference was subsequently denied, the reason and retry-after elements of the Subscription-State header field can be used to indicate if and when the REFER can be re-attempted (as described for SUBSCRIBE in [2]).

IV.2.5 Behavior of SIP Registrars/Redirect Servers

A registrar that is unaware of the definition of the REFER method will return a 501 response as defined in [1]. A registrar aware of the definition of REFER SHOULD return a 405 response.

This specification places no requirements on redirect server behavior beyond those specified in [1]. Thus, it is possible for REFER requests to be redirected.

IV.2.6 Behavior of SIP Proxies

SIP proxies do not require modification to support the REFER method. Specifically, as required by [1], a proxy should process a REFER request the same way it processes an OPTIONS request.

IV.3. Package Details: Event refer

This document defines an event package as defined in [2].

IV.3.1 Event Package Name

The name of this event package is "refer".

IV.3.2 Event Package Parameters

This package uses the "id" parameter defined in [2]. Its use in package is described in Section IV.2.4.6.

IV.3.3 SUBSCRIBE Bodies

SUBSCRIBE bodies have no special meaning for this event package.

IV.3.4 Subscription Duration

The duration of an implicit subscription created by a REFER request is initially determined by the agent accepting the REFER and communicated to the subscribing agent in the Subscription-State header field's expire parameter in the first NOTIFY sent in the subscription. Reasonable choices for this initial duration depend on the type of request indicated in the Refer-To URI. The duration SHOULD be chosen to be longer than the time the referenced request will be given to complete. For example, if the Refer-To URI is a SIP INVITE URI, the subscription interval should be longer than the Expire value in the INVITE. Additional time MAY be included to account for time needed to authorize the subscription. The subscribing agent MAY extend the subscription by refreshing it, or terminate it by unsubscribing. As described in Section IV.2.4.7, the agent accepting the REFER will terminate the subscription when it reports the final result of the reference, indicating that termination in the Subscription-State header field.

IV.3.5 NOTIFY Bodies

The bodies of NOTIFY requests for event refer are discussed in Section IV.2.4.5.

IV.3.6 Notifier processing of SUBSCRIBE requests

Notifier processing of SUBSCRIBE requests is discussed in Section IV.2.4.4.

IV.3.7 Notifier Generation of NOTIFY Requests

Notifier generation of NOTIFY requests is discussed in Section IV.2.4.4.

IV.3.8 Subscriber Processing of NOTIFY Requests

Subscriber processing of NOTIFY requests is discussed in Section IV.2.4.4.

IV.3.9 Handling of Forked Requests

A REFER sent within the scope of an existing dialog will not fork. A REFER sent outside the context of a dialog MAY fork, and if it is accepted by multiple agents, MAY create multiple subscriptions. These subscriptions are created and managed as per "Handling of Forked Requests" in [2] as if the REFER had been a SUBSCRIBE. The agent sending the REFER manages the state associated with each subscription separately. It does NOT merge the state from the separate subscriptions. The state is the status of the referenced request at each of the accepting agents.

IV.3.10 Rate of Notifications

An event refer NOTIFY might be generated each time new knowledge of the status of a referenced requests becomes available. For instance, if the REFER was to a SIP INVITE, NOTIFYs might be generated with each provisional response and the final response to the INVITE. Alternatively, the subscription might only result in two NOTIFY requests, the immediate NOTIFY and the NOTIFY carrying the final result of the reference. NOTIFYs to event refer SHOULD NOT be sent more frequently than once per second.

IV.3.11 State Agents

Separate state agents are not defined for event refer.

IV.4. Examples

IV.4.1 Prototypical REFER callflow

```
Agent A              Agent B
|                    |
|  F1 REFER          |
|------------------->|
|      F2 202 Accepted |
|<-------------------|
|      F3 NOTIFY     |
|<-------------------|
| F4 200 OK          |
|------------------->|
|                    |
|                    |------->
|                    |  (whatever)
|                    |<------
|                    |
|       F5 NOTIFY     |
|<----------------------|
|  F6 200 OK          |
|---------------------->|
```

Here are examples of what the four messages between Agent A and Agent B might look like if the reference to (whatever) that Agent B makes is successful. The details of this flow indicate this particular REFER occurs outside a session (there is no To tag in the REFER request). If the REFER occurs inside a session, there would be a non-empty To tag in the request.

Message One (F1)

```
    REFER sip:b@agentland SIP/2.0
    Via: SIP/2.0/UDP agenta.agentland;branch=z9hG4bK2293940223
    To:
    From: ;tag=193402342
    Call-ID: 898234234@agenta.agentland
    CSeq: 93809823 REFER
    Max-Forwards: 70
    Refer-To: (whatever URI)
    Contact: sip:a@agentland
    Content-Length: 0
```

Message Two (F2)

```
    SIP/2.0 202 Accepted
    Via: SIP/2.0/UDP agenta.agentland;branch=z9hG4bK2293940223
    To: ;tag=4992881234
    From: ;tag=193402342
    Call-ID: 898234234@agenta.agentland
    CSeq: 93809823 REFER
    Contact: sip:b@agentland
    Content-Length: 0
```

Message Three (F3)

```
    NOTIFY sip:a@agentland SIP/2.0
    Via: SIP/2.0/UDP agentb.agentland;branch=z9hG4bK9922ef992-25
    To: ;tag=193402342
    From: ;tag=4992881234
    Call-ID: 898234234@agenta.agentland
    CSeq: 1993402 NOTIFY
    Max-Forwards: 70
    Event: refer
    Subscription-State: active;expires=(depends on Refer-To URI)
    Contact: sip:b@agentland
    Content-Type: message/sipfrag;version=2.0
    Content-Length: 20

SIP/2.0 100 Trying
```

Message Four (F4)

```
    SIP/2.0 200 OK
    Via: SIP/2.0/UDP agentb.agentland;branch=z9hG4bK9922ef992-25
    To: ;tag=193402342
    From: ;tag=4992881234
    Call-ID: 898234234@agenta.agentland
     CSeq: 1993402 NOTIFY
    Contact: sip:a@agentland
    Content-Length: 0
```

Message Five (F5)

```
    NOTIFY sip:a@agentland SIP/2.0
    Via: SIP/2.0/UDP agentb.agentland;branch=z9hG4bK9323394234
    To: ;tag=193402342
    From: ;tag=4992881234
    Call-ID: 898234234@agenta.agentland
    CSeq: 1993403 NOTIFY
    Max-Forwards: 70
    Event: refer
    Subscription-State: terminated;reason=noresource
    Contact: sip:b@agentland
    Content-Type: message/sipfrag;version=2.0
    Content-Length: 16

SIP/2.0 200 OK
```

Message Six (F6)

```
   SIP/2.0 200 OK
   Via: SIP/2.0/UDP agentb.agentland;branch=z9hG4bK9323394234
   To: ;tag=193402342
   From: ;tag=4992881234
   Call-ID: 898234234@agenta.agentland
   CSeq: 1993403 NOTIFY
   Contact: sip:a@agentland
   Content-Length: 0
```

IV.4.2 Multiple REFERs in a dialog

Message One above brings an implicit subscription dialog into existence. Suppose Agent A issued a second REFER inside that dialog:

```
     Agent A              Agent B
       |                    |
       |  F7 REFER          |
       |------------------->|
       |      F8 202 Accepted |
       |<-------------------|
       |      F9 NOTIFY     |
       |<-------------------|
       |  F10 200 OK        |
       |------------------->|
       |                    |------->
       |                    |  (something different)
       |                    |<------
       |                    |
       |       F11 NOTIFY   |
       |<-------------------|
       |  F12 200 OK        |
       |------------------->|
       |                    |
       |                    |
```

Message Seven (F7)

```
   REFER sip:b@agentland SIP/2.0
   Via: SIP/2.0/UDP agenta.agentland;branch=z9hG4bK9390399231
   To: ;tag=4992881234
   From: ;tag=193402342
   Call-ID: 898234234@agenta.agentland
   CSeq: 93809824 REFER
```

```
    Max-Forwards: 70
    Refer-To: (some different URI)
    Contact: sip:a@agentland
    Content-Length: 0
```

Message Eight (F8)

```
    SIP/2.0 202 Accepted
    Via: SIP/2.0/UDP agenta.agentland;branch=z9hG4bK9390399231
    To: ;tag=4992881234
    From: ;tag=193402342
    Call-ID: 898234234@agenta.agentland
    CSeq: 93809824 REFER
    Contact: sip:b@agentland
    Content-Length: 0
```

Message Nine (F9)

```
    NOTIFY sip:a@agentland SIP/2.0
    Via: SIP/2.0/UDP agentb.agentland;branch=z9hG4bK9320394238995
    To: ;tag=193402342
    From: ;tag=4992881234
    Call-ID: 898234234@agenta.agentland
    CSeq: 1993404 NOTIFY
    Max-Forwards: 70
    Event: refer
    Subscription-State: active;expires=(depends on Refer-To URI)
    Contact: sip:b@agentland
    Content-Type: message/sipfrag;version=2.0
    Content-Length: 20

SIP/2.0 100 Trying
```

Message Ten (F10)

```
    SIP/2.0 200 OK
    Via: SIP/2.0/UDP agentb.agentland;branch=z9hG4bK9320394238995
    To: ;tag=193402342
    From: ;tag=4992881234
    Call-ID: 898234234@agenta.agentland
    CSeq: 1993404 NOTIFY
    Contact: sip:a@agentland
    Content-Length: 0
```

Message Eleven (F11)

```
    NOTIFY sip:a@agentland SIP/2.0
    Via: SIP/2.0/UDP agentb.agentland;branch=z9hG4bK2994a93eb-fe
    To: ;tag=193402342
    From: ;tag=4992881234
    Call-ID: 898234234@agenta.agentland
    CSeq: 1993405 NOTIFY
    Max-Forwards: 70
    Event: refer;id=93809824
    Subscription-State: terminated;reason=noresource
    Contact: sip:b@agentland
    Content-Type: message/sipfrag;version=2.0
    Content-Length: 16

SIP/2.0 200 OK
```

Message Twelve (F12)

```
    SIP/2.0 200 OK
    Via: SIP/2.0/UDP agentb.agentland;branch=z9hG4bK2994a93eb-fe
```

```
To: ;tag=193402342
From: ;tag=4992881234
Call-ID: 898234234@agenta.agentland
CSeq: 1993405 NOTIFY
Contact: sip:a@agentland
Content-Length: 0
```

IV.5. Security Considerations

The security considerations of [1] apply to the REFER method.

This mechanism relies on providing contact information for the referred-to resource to the party being referred. Care should be taken to provide a suitably restricted URI if the referred to resource should be protected.

Using message/sipfrag bodies to return the progress and results of a REFER request is extremely powerful. Careless use of that capability will compromise security and privacy. Here are a couple of simple, somewhat contrived, examples to demonstrate the potential for harm.

IV.5.1 Circumventing Privacy

Suppose Alice has a user agent that accepts REFER requests to SIP INVITE URIs, and NOTIFYs the referrer of the progress of the INVITE by copying each response to the INVITE into the body of a NOTIFY.

Suppose further that Carol has a reason to avoid Mallory and has configured her system at her proxy to only accept calls from a certain set of people she trusts (including Alice), so that Mallory doesn't learn when she's around, or what user agent she's actually using.

Mallory can send a REFER to Alice, with a Refer-To URI indicating Carol. If Alice can reach Carol, the 200 OK Carol sends gets returned to Mallory in a NOTIFY, letting him know not only that Carol is around, but also the IP address of the agent she's using.

IV.5.2 Circumventing Security

Suppose Alice, with the same user agent as above, is working at a company that is working on the greatest SIP device ever invented - the SIP FOO. The company has been working for months building the device and the marketing materials, carefully keeping the idea, even the name of the idea secret (since a FOO is one of those things that anybody could do if they'd just had the idea first). FOO is up and running, and anyone at the company can use it, but it's not available outside the company firewall.

Mallory has heard rumor that Alice's company is onto something big, and has even managed to get his hands on a URI that he suspects might have something to do with it. He sends a REFER to ALICE with the mysterious URI and as Alice connects to the FOO, Mallory gets NOTIFYs with bodies containing:

```
Server: FOO/v0.9.7
```

IV.5.3 Limiting the Breach

For each of these cases, and in general, returning a carefully selected subset of the information available about the progress of the reference through the NOTIFYs mitigates risk. The minimal implementation described in Section 2.4.5 exposes the least information about what the agent operating on the REFER request has done, and is least likely to be a useful tool for malicious users.

IV.6. Historic Material

This method was initially motivated by the call-transfer application. Starting as TRANSFER, and later generalizing to REFER, this method improved on the BYE/Also concept of the expired draft-ietf-sip-cc-01 by disassociating transfers from the processing of BYE. These changes facilitate recovery of failed transfers and clarify state management in the participating entities.

Early versions of this work required the agent responding to REFER to wait until the referred action completed before sending a final response to the REFER. That final response reflected the success or failure of the referred action. This was infeasible due to the transaction timeout rules defined for non-INVITE requests in [1]. A REFER must always receive an immediate (within the lifetime of a non-INVITE transaction) final response.

IV.7. IANA Considerations

(Note to RFC Editor: Please fill in all occurrences of XXXX in this section with the RFC number of this specification).

This document defines a new SIP method name (REFER), a new SIP header field name with a compact form (Refer-To and r respectively), and an event package (refer).

The following row should be added to the method sub-registry under http://www.iana.org/assignments/sip-parameters.

REFER	[RFCXXXX]

The following information should be added to the header sub-registry under http://www.iana.org/assignmemts/sip-parameters.

Header Name: Refer-To
Compact Form: r
Reference: RFC XXXX

This specification registers an event package, based on the registration procedures defined in [2]. The following is the information required for such a registration:

Package Name: refer
Package or Package-Template: This is a package.
Published Specification: RFC XXXX
Person to Contact: Robert Sparks, rsparks@dynamicsoft.com

IV.8. Acknowledgments

This draft is a collaborative product of the SIP working group.

Normative References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
[2] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.
[3] Sparks, R., "Internet Media Type message/sipfrag", Internet-Draft

APPENDIX V - THE SESSION INITIATION PROTOCOL (SIP) "REPLACES" HEADER

draft-ietf-sip-replaces-02.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 30, 2002.

Copyright Notice

Abstract

This document defines a new header for use with SIP multi-party applications and call control. The Replaces header is used to logically replace an existing SIP dialog with a new SIP dialog. This primitive can be used to enable a variety of features, for example: "Attended Transfer" and "Retrieve from Call Park". Note that definition of these example features is non-normative.

V.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [2].

This document refers frequently to the terms "confirmed dialog" and "early dialog". These are defined in Section 12 of SIP [1].

V.2. Overview

This document describes a SIP [1] extension header field as part of the SIP multiparty applications architecture framework [6]. The Replaces header is used to logically replace an existing SIP dialog with a new SIP dialog. This is especially useful in peer-to-peer call control environments.

One use of the "Replaces" header is to replace one participant with another in a multimedia conversation. While this functionality is already available using 3rd party call control [8] style call control, the 3pcc model requires a central point of control which may not be desirable in many environments. As such, a method of performing these same call control primitives in a distributed, peer- to-peer fashion is very desirable.

Use of a new INVITE with a new header for dialog matching was chosen over making implicit associations in an incoming INVITE based on call-id or other fields for the following reasons:

An INVITE already has the correct semantics for a new call
Using an explicit Replaces header in a new request makes the intent of the request obvious.
A unique call-id may be given to the replacement call. This avoids call-leg matching problems in any of the clients.
There are no adverse effects if the header is unsupported.

The Replaces header enables services such as attended call transfer, retrieve from park, and transition from locally mixed conferences to two party calls in a distributed peer-to-peer way. This list of services is not exhaustive. Although the Replaces header is frequently used in combination with the REFER [4] method as used in cc-transfer [7], they may be used independently.

For example, Alice is talking to Bob from phone1. She transfers Bob to a Parking Place while she goes to the lab. When she gets there she retrieves the "parked" call from phone2 by sending an INVITE with a Replaces header field to Bob with the dialog information Bob shared with the Parking Place. Alice got this information using some out of band mechanism. Perhaps she subscribed to this information from the Parking Place, or went to a website and clicked on a URI. A short call flow for this example follows. (Via and Max-Forwards headers are omitted for clarity.)

```
Alice      Alice              Parking
phone1     phone2      Bob      Place
|          |           |           |
|<================================>|           |
|          |           |           |
|     Alice transfers Bob to Parking Place     |
|          |           |           |
|------------REFER/200----------->|  *1   *2   |
|          |           |--INVITE/200/ACK-->|
|<-----------NOTIFY/200-----------|<==================>|
|------------BYE/200------------->|            |
|          |           |           |
|          |           |           |
| Alice later retrieves call from another phone     |
|          |           |           |
|        *3 |-INV w/Replaces->|           |
|           |<--200-----------|          |
|           |---ACK---------->|----BYE/200------->|
|           |<================>|         |
|          |           |           |
```

Message *1: Bob-> Parking Place
INVITE sip:parkingplace@sip.org SIP/2.0
To:
From: ;tag=7743
Call-ID: 425928@bobster.sip.org
CSeq: 1 INVITE
Contact:
Referred-By:

Message *2: Parking Place -> Bob
SIP/2.0 200 OK
To: ;tag=6472
From: ;tag=7743
Call-ID: 425928@bobster.sip.org
CSeq: 1 INVITE
Contact:

Message *3: Alice@phone2 -> Bob
INVITE sip:bob@bobster.sip.org
To:
From: ;tag=8983
Call-ID: 09870@phone2.sip.org
CSeq: 1 INVITE
Contact:
Require: replaces
Replaces: 425928@bobster.sip.org;to-tag=7743;from-tag=6472

V.3. User Agent Server Behavior: Receiving a Replaces Header

The Replaces header contains information used to match an existing SIP dialog (call-id, to-tag, and from-tag). Upon receiving an INVITE with a Replaces header, the UA attempts to match this information with a confirmed or early dialog. The to-tag and from-tag are matched as if they were present in an incoming request. In other words the to-tag is compared to the local tag, and the from-tag is compared to the remote tag.

If more than one Replaces header field is present in an INVITE, or if a Replaces header field is present in a request other than INVITE, the UAS MUST reject the request with a 400 Bad Request response.

The Replaces header has specific call control semantics. If both a Replaces header field and another header field with contradictory semantics are present in a request, the request MUST be rejected with a 400 "Bad Request" response.

If the Replaces header field matches more than one dialog, the UA MUST act as if no match is found.

If no match is found, the UAS rejects the INVITE and returns a 481 Call/Transaction Does Not Exist response. Likewise, if the Replaces header field matches a dialog which was not created with an INVITE, the UAS MUST reject the request with an appropriate response (ex: 400, 481, or 501).

If the Replaces header field matches a dialog which has already terminated, the UA SHOULD decline the request with a 603 Declined response.

If the Replaces header field matches a active dialog, the UA SHOULD verify that the initiator of the new INVITE is authorized to replace the matched dialog. If the initiator of the new INVITE has authenticated successfully as equivalent to the user who is being replaced, then the replacement is authorized. In addition, the UA MAY use other authorization mechanisms defined for this purpose in standards track extensions. For example, an extension could define a mechanism for transitively asserting authorization of a replacement.

If authorization is successful, the UA attempts to accept the new INVITE, reassign the user interface and other resources of the matched dialog to the new INVITE, and shut down the replaced dialog. If the UA cannot accept the new INVITE (for example: it cannot establish required QoS or keying, or it has incompatible media), the UA MUST return an appropriate error response and MUST leave the matched dialog unchanged.

If the Replaces header field matches a confirmed dialog, it accepts the new INVITE by sending a 200-class response, and shuts down the replaced dialog by sending a BYE. If the Replaces header field matches an early dialog that was initiated by the UA, it accepts the new INVITE by sending a 200-class response, and shuts down the replaced dialog by sending a CANCEL. If the Replaces header field matches an early dialog that was not initiated by the UA, the UA returns a provisional or final response to the new INVITE which is suitable for the state of the resources used by the matched dialog, and responds to the replaced early dialog with a 687 "Transaction Terminated" response (defined earlier in this document).

V.4. User Agent Client Behavior: Sending a Replaces header

A User Agent that wishes to replace a single existing early or confirmed dialog with a new dialog of its own, MAY send the target User Agent an INVITE request containing a Replaces header field. The UAC places the Call-ID, to-tag, and from-tag information for the target dialog in a single Replaces header field and sends the new INVITE to the target.

Note that use of this mechanism does not provide a way to match multiple dialogs, nor does it provide a way to match an entire call, an entire transaction, or to follow a chain of proxy forking logic. For example, if Alice replaces Cathy in an early dialog with Bob, but he does not answer, Alice's replacement request will not match other dialogs to which Bob's UA redirects, nor other branches to which his proxy forwards.

V.5. Proxy behavior

Proxy Servers do not require any new behavior to support this extension. They simply pass the Replaces header field transparently as described in the SIP specification.

Note that it is possible for a proxy (especially when forking based on some application layer logic, such as caller screening or time-of-day routing) to forward an INVITE request containing a Replaces header field to a completely orthogonal set of Contacts than the original request it was intended to replace. In this case, the INVITE request with the Replaces header field will fail.

V.6. Syntax

V.6.1 The Replaces Header

The Replaces header field indicates that a single dialog identified by the header field is to be shut down and logically replaced by the incoming INVITE in which it is contained. It is a request header only, and defined only for INVITE requests. The Replaces header field MAY be encrypted as part of end-to-end encryption. Only a single Replaces header field value may be present in a SIP request This document adds the following entry to Table 3 of [1]. Additions to this table are also provided for extension methods defined at the time of publication of this document. This is provided as a courtesy to the reader and is not normative in any way. SUBSCRIBE and NOTIFY, REFER, INFO, UPDATE, and PRACK are defined respectively in [10], [4], [11], [12], and [13].

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Replaces	R		-	-	-	o	-	-

			SUB	NOT	REF	INF	UPD	PRA
Replaces	R		-	-	-	-	-	-

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC-2234 [3].

```
Replaces        = "Replaces" HCOLON callid *(SEMI replaces-param)
replaces-param  = to-tag / from-tag / generic-param
to-tag          = "to-tag" EQUAL token
from-tag        = "from-tag" EQUAL token
```

A Replaces header MUST contain exactly one to-tag and exactly one from-tag, as they are required for unique dialog matching. For compatibility with dialogs initiated by RFC2543 [5] compliant UAs, a tag of zero matches both tags of zero and null tags.

Examples:

```
Replaces: 98732@sip.billybiggs.com
;from-tag=r33th4x0r
;to-tag=ff87ff

Replaces: 12adf2f34456gs5;to-tag=12345;from-tag=54321

Replaces: 87134@171.161.34.23;to-tag=24796;from-tag=0
```

V.6.2 New option tag for Require and Supported headers

This specification defines a new Require/Supported header option tag "replaces". UAs which support the Replaces header MUST include the "replaces" option tag in a Supported header field. UAs that want explicit failure notification if Replaces is not supported MAY include the "replaces" option in a Require header field.

Example:

```
Require: replaces, 100rel
```

V.6.3 687 Response Code: "Dialog Terminated"

This specification defines a new SIP response code. The 687 "Dialog Terminated" response code indicates that an early dialog has been completely replaced by a new dialog. A new response code was chosen from the 6xx class to prevent intervening proxies from attempting to fork additional branches of the replaced dialog.

V.7. Usage Examples

The following non-normative examples are not intended to enumerate all the possibilities for the usage of this extension, but rather to provide examples or ideas only. For more examples, please see service-examples [9]. Via and Max-Forwards headers are omitted for clarity and brevity.

V.7.1 Replacing an Early Dialog at the receiver

In this example, a Customer tries calling a call center and for some reason cannot get through properly. The customer calls an Operator and asks for help. The operator calls the contact center, and upon receiving a provisional response, assumes that everything is OK and transfers the Customer to the Call Center, replacing the operator's place in the queue.

Call

```
   Operator            Customer                Center
      |                   |                     |
      |<--INVITE/180/200/ACK--|                      |
      |<=====================>|  "Hello, I'm having    |
      |                   |  trouble calling ..." |
      |"OK, I'll try it and   |                       |
      | transfer you if it    |                       |
      | works for me"         |                       |
      |                   |                     |
   *1 |-----INVITE ---------------------------------->|
   *2 |<----182: You are caller number 7--------------|
      |                   |                     |
      |  completes transfer   |                       |
      |                   |                     |
      |---REFER/200---------->|                       |
      |                   |--INVITE with Replaces->| *3
      |                   |<----182: caller #7-----| *4
      |<----687 Dialog Terminated----------------------| *5
      |-----ACK--------------------------------------->|
      |<--NOTIFY/200----------|                       |
      |---BYE/200------------>|                       |
      |                   |  ...time passes..       |
      |                   |                     |
      |                   |                     |
      |                   |                     |
      |                   |<---200 OK--------------|
      |<--NOTIFY/200----------|----ACK---------------->|
      |                   |                     |
      |                   |                     |
```

```
Message *1: Operator -> Call Center
INVITE sip:helpdesk@clueless.org SIP/2.0
To:
From: ;tag=7743
Call-ID: 425928@dhcp23311.acme.com
CSeq: 1 INVITE
Contact:
Accept-Language: en
```

```
Message *2: Call Center -> Operator
SIP/2.0 182 You are 7th in Queue
To: ;tag=6472
From: ;tag=7743
Call-ID: 425928@dhcp23311.acme.com
CSeq: 1 INVITE
Contact:
```

```
Message *3: Customer -> Call Center
INVITE sip:helpdesk@frontline.clueless.org
To:
From: ;tag=8983
Call-ID: 09870@lobby12.acme.com
CSeq: 1 INVITE
Contact:
Replaces: 425928@dhcp23311.acme.com;to-tag=7743;from-tag=6472
Accept-Language: en
Referred-By:
```

```
Message *4: Call Center -> Customer
SIP/2.0 182 You are 7th in Queue
```

```
To:
From: ;tag=8983
Call-ID: 09870@lobby12.acme.com
CSeq: 1 INVITE
Contact:
```

```
Message *5: Call Center -> Operator
SIP/2.0 687 Dialog Terminated
To: ;tag=6472
From: ;tag=7743
Call-ID: 425928@dhcp23311.acme.com
CSeq: 1 INVITE
Contact:
```

V.7.2 Replacing an Early Dialog at the originator

In this example, Bob just arrived in the lab and hasn't registered there yet. He hears his desk phone ring. He quickly logs into a software UA on a nearby computer. Among other things, the software UA has access to the dialog state of his desk phone. When it notices that his phone is ringing it offers him the choice to take the call there. The software UA sends an INVITE with Replaces to Alice. When Alice's UA receives this new INVITE, it CANCELs her original INVITE and connects Alice to Bob.

```
                    Bob                Bob
    Alice           desk               lab
      |               |                 |
  *1  |-----INVITE----------->|                 |
  *2  |<----180---------------|  Bob hears desk phone  |
      |               |  ringing from lab but  |
      |               |  isn't REGISTERed yet  |
      |               |                 |
      |               |<--fetch dialog state --|
      |               |---response ----------->|
 *3/4 |<-----INVITE with Replaces/200/ACK--------------|
 *5/6 |------CANCEL/200------>|                 |
 *7   |<-----487--------------|                 |
      |------ACK------------->|                 |
      |               |                 |
      |               |                 |
```

```
Message *1: Alice -> Bob's desk phone
INVITE sip:bob@sip.org SIP/2.0
To:
From: ;tag=7743
Call-ID: 425928@phone.sip.org
CSeq: 1 INVITE
Contact:
```

```
Message *2: Bob's desk phone -> Alice
SIP/2.0 180 Ringing
To: ;tag=6472
From: ;tag=7743
Call-ID: 425928@phone.sip.org
CSeq: 1 INVITE
Contact:
```

```
Message *3: Bob in lab -> Alice
INVITE sip:alice@phone.sip.org
To:
From: ;tag=8983
Call-ID: 09870@labpc.sip.org
CSeq: 1 INVITE
```

Contact:
Replaces: 425928@phone.sip.org;to-tag=7743;from-tag=6472

Message *4: Alice -> Bob in lab
SIP/2.0 200 OK
To: ;tag=9232
From: ;tag=8983
Call-ID: 09870@labpc.sip.org
CSeq: 1 INVITE
Contact:

Message *5: Alice -> Bob's desk
CANCEL sip:bob@sip.org SIP/2.0
To:
From: ;tag=7743
Call-ID: 425928@phone.sip.org
CSeq: 1 CANCEL
Contact:

Message *6: Bob's desk -> Alice
SIP/2.0 200 OK
To:
From: ;tag=7743
Call-ID: 425928@phone.sip.org
CSeq: 1 CANCEL
Contact:

Message *7: Bob's desk -> Alice
SIP/2.0 487 Request Terminated
To: ;tag=6472
From: ;tag=7743
Call-ID: 425928@phone.sip.org
CSeq: 1 INVITE
Contact:

V.8. Security Considerations

The extension specified in this document significantly changes the relative security of SIP devices. Currently in SIP, even if an eavesdropper learns the Call-ID, To, and From headers of a dialog, they cannot easily modify or destroy that dialog if Digest authentication or end-to-end message integrity are used.

This extension can be used to disconnect participants or replace participants in a multimedia conversation. As such, invitations with the Replaces header SHOULD only be accepted if the peer requesting replacement has been properly authenticated using a standard SIP mechanism, and authorized to request a replacement of the target dialog.

Some mechanisms for obtaining the dialog information needed by the Replaces header (Call-ID, to-tag, and from-tag) include URIs on a web page, subscriptions to an appropriate event package, and notifications after a REFER request. Use of end-to-end security mechanisms to encrypt this information is also RECOMMENDED.

This extension was designed to take advantage of future signature or authorization schemes defined by the SIP Working Group. In general, call control features would benefit considerably from such work.

V.9. IANA Considerations

V.9.1 Registration of "Replaces" SIP header

Name of Header: Replaces
Short form: none
Normative description: section F-6.1 of this document

V.9.2 Registration of "replaces" SIP Option-tag

Name of option: replaces

Description: Support for the SIP Replaces header
SIP headers defined: Replaces
Normative description: This document

V.9.3 Registration of "687" SIP Response code

Number of response code: 687
Default reason phrase: Dialog Terminated
Normative description: section F-6.3 of this document

V.10. Changes

V.10.1 Changes Since -01

- Removed the to-tag=* matching mechanism, and related proxy requirements and examples based on WG consensus at interim meeting and on the mailing list.
- Reorganized motivational overview material
- Moved extra examples to service-flows
- Added authorization language in UAS behavior section
- Removed allowance to match on one of multiple matching dialogs with no tags
- Updated references

V.10.2 Changes Since -00

- When no dialog matches the Call-ID and tags in a Replaces header, the UAS now returns a 481 instead of silently accepting the INVITE.
- Changed the BNF to match the explicit white space BNF now used by SIP.
- Added the to-tag=* matching mechanism.
- Added requirements for forking proxies and a discussion of the consequences if forking proxies do not support Replaces.
- Added last two examples.
- Split normative and non-normative references

APPENDIX VI - BIBLIOGRAPHY

[26] ITU-T Recommendation J.167 - Media terminal adapter (MTA) device provisioning requirements for the delivery of real-time services over cable television networks using cable modems.

[27] ITU-T Recommendation J.161 - Audio Codec requirements for the provision of bidirectional audio service over cable television networks using cable modems.

[28] ITU-T Recommendation J.175 - IPCablecom Audio Server Protocol, July, 2002.

[29] IETF RFC 791, Transmission Control Protocol, Postal, J., September 1981

[30] Architectural Considerations for Providing Carrier Class Telephony Services Utilizing SIP-based Distributed Call Control Mechanisms, RFC TBD

[31] IETF RFC 2806 - A. Vaha-Sipila, "URLs for Telephone Calls", April 2000

[32] H. Schulzrinne, A. Vaha-Sipila, "The tel URI for Telephone Calls", work in progress, June 2002.

[33] J. Yu, "Extensions to the "tel" URL to Support Number Portability and Freephone Service", work in progress, June 2002.

[34] R. Mahy, “A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)”, work in progress, May 2002

[35] IETF RFC 3398, G. Camarillo, A. Roach, J. Peterson, L. Ong, “Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping”, December 2002

[36] PacketCable 1.2 Architecture Framework, PKT-TR-ARCH1.2-V01-001229, December 29, 2000, www.packetcable.com/specifications

APPENDIX VII - ACKNOWLEDGMENTS

Much of the text in Section 7.1 came from [32] and [33] and much of the text in 7.10 came from [34], which contained the following copyright notice:

“Copyright © The Internet Society (2002). All Rights Reserved. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an “AS IS” basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.”