PacketCable™ 1.5 Specifications

CMS to CMS Signaling

Superseded

PKT-SP-CMSS1.5-I02-050812

ISSUED

Notice

Document Status Sheet

Document Control Number:	PKT-SP-CMSS1.5-I02-050812			
Document Title:	CMS to CMS Signaling			
Revision History:	D01 – First draft released September 30, 2004 D02 – Second draft released December 10, 2004 I01 – Issued Specification January 28, 2005 I02 – Issued Specification August 12, 2005			
Date:	August 12, 2005			
Status:	~~Work in Progress~~	~~Draft~~	Issued	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ PacketCable/ Vendor~~	Public

Key to Document Status Codes:

Work in Progress	An incomplete document, designed to guide discussion and generate feedback, that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Contents

List of Figures

This page left blank intentionally.

1 INTRODUCTION

1.1 Scope

This specification describes the PacketCable Call Management Server (CMS) to CMS Signaling 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 CMSs, and so for purposes of this specification the MGC is considered identical to a CMS. CMSs currently support multimedia endpoints (within the PacketCable infrastructure) that use the Network-based Call Signaling [24] (NCS) protocol and the PSTN Gateway Call Signaling Protocol [25] (TGCP) for communicating signaling 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 Signaling (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 Signaling and PacketCable 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 CMSs belonging to a single service provider as well as between CMSs that belong to different service providers. In the case that the CMSs are owned by multiple service providers, it is assumed that the service providers have a mutual trust relationship.

Other PacketCable documents describe interfaces between other system elements. These documents cover areas such as: Event Message recording for billing and other back office functions [23]; Dynamic Quality of Service [21]; Operations and Provisioning [39]; Electronic Surveillance [22]; and Security [26]. These other specifications indirectly place requirements on the signaling 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-PacketCable-compliant devices.

From time to time this document refers to the voice communications capabilities of a PacketCable network in terms of "IP Telephony." The legal/regulatory classification of IP-based voice communications provided over cable networks and otherwise, and the legal/regulatory obligations, if any, borne by providers of such voice communications, are not yet fully defined by appropriate legal and regulatory authorities. Nothing in this document is addressed to, or intended to affect, those issues. In particular, while this document uses standard terms such as "call," "call signaling," "telephony," etc., it should be recalled that while a PacketCable network performs activities analogous to these PSTN functions, the manner by which it does so differs considerably from the manner in which they are performed in the PSTN by telecommunications carriers, and that these differences may be significant for legal/regulatory purposes. Moreover, while reference is made here to "IP Telephony," it should be recognized that this term embraces a number of different technologies and network architecture, each with different potential associated legal/regulatory obligations. No particular legal/regulatory consequences are assumed or implied by the use of this term.

1.2 Specification Language

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"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

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[1] IETF RFC 1890, RTP Profile for Audio and Video Conferences with Minimal Control, January 1996.

[2] IETF RFC 2234, Augmented BNF for Syntax Specifications: ABNF, November 1997.

[3] IETF RFC 2327, SDP: Session Description Protocol, April 1998.

[4] IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, August 1998.

[5] IETF RFC 2397, The "data" URL Scheme, August 1998.

[6] IETF RFC 3261, SIP: Session Initiation Protocol, February 2002.

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

[8] IETF RFC 3263, Session Initiation Protocol (SIP): Locating SIP Servers, June 2002.

[9] IETF RFC 3265, SIP-Specific Event Notification, Roach A., June 2002.

[10] IETF RFC 3311, The SIP UPDATE Method, Rosenberg, J., September 2002.

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

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

[13] IETF RFC 3325, Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks, November 2002.

[14] IETF RFC 3420, Internet Media Types message/sip and message/sipfrag, November 2002.

[15] IETF RFC 768, User Datagram Protocol, August 1980.

[16] IETF RFC 3603, Private Session Initiation Protocol (SIP) Proxy-to-Proxy Extensions for Supporting the PacketCable Distributed Call Signaling Architecture, October 2003.

[17] IETF RFC 3515, The Session Initiation Protocol (SIP) Refer Method, April 2003.

[18] See Appendix C (The Session Initiation Protocol [SIP] "Replaces" Header) for more details about this reference.

[19] ITU-T Rec. E.123, Notation for national and international telephone numbers, e-mail addresses and Web addresses, February 2001.

[20] ITU-T Rec. E.164, The international public telecommunication numbering plan, May 1997.

[21] PacketCable 1.5 Dynamic Quality of Service Specification, PKT-SP-DQOS1.5-I02-050812, August 12, 2005.

[22] PacketCable 1.5 Electronic Surveillance Specification, PKT-SP-ESP1.5-I01-050128, January 28, 2005.

[23] PacketCable 1.5 Event Messaging Specification, PKT-SP-EM1.5-I02-050812, August 12, 2005.

[24] PacketCable 1.5 Network-Based Call Signaling Protocol Specification, PKT-SP-NCS1.5-I02-050812, August 12, 2005.

[25] PacketCable 1.5 PSTN Gateway Call Signaling Protocol Specification, PKT-SP-TGCP1.5-I02-050812, August 12, 2005.

[26] PacketCable 1.5 Security Specification, PKT-SP-SEC1.5-I01-050128, January 28, 2005.

[27] IETF RFC 3842, A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP), R. Mahy, August 2004.

[28] IETF RFC 3966, The tel URI for Telephone Numbers, December, 2004.

[29] See Appendix D, New Parameters for the "tel" URI to Support Number Portability, draft-ietf-iptel-tel-np-04.

2.2 Informative References

[30] IETF RFC 3398, ISUP to SIP Mapping, December, 2002.

[31] Architectural Considerations for Providing Carrier Class Telephony Services Utilizing SIP-based Distributed Call Control Mechanisms, draft-dcsgroup-sipping-arch-01.txt, January 2003, www.ietf.org/ietf/1id-abstracts.txt.

[32] IETF RFC791, Transmission Control Protocol, Postal, J., September 1981.

[33] New Parameters for the "tel" URI to Support Number Portability, work in progress, draft-ietf-iptel-tel-np-03.txt, November 2004, www.ietf.org/internet-drafts/draft-ietf-iptel-tel-np-03.txt.

[34] PacketCable 1.5 Architecture Framework, PKT-TR-ARCH1.5-V01-050128, January 28, 2005.

[35] PacketCable 1.5 Audio Server Protocol Specification, PKT-SP-ASP1.5-I01-050128, January 28, 2005.

[36] PacketCable 1.5 Audio/Video Codecs Specification, PKT-SP-CODEC1.5-I02-050812, August 12, 2005.

[37] PacketCable Basic Residential Feature Descriptions for PacketCable-Based VoIP Services, PKT-TR-VOIPBRF-R01-000608, June 8, 2000.

[38] PacketCable Extended Residential Feature Descriptions for PacketCable-Based VoIP Services, PKT-TR-VOIPERF-R01-000831, August 31, 2000.

[39] PacketCable 1.5 MTA Device Provisioning Specification, PKT-SP-PROV1.5-I02-050812, August 12, 2005.

[40] PacketCable OSS Overview, PKT-TR-OSS-V02-991201, December 1, 1999.

[41] Yu, J., New Parameter for the "tel" URI to Support Dial Around Indicator, work in progress, April 2004.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone 303-661-9100; Fax 303-661-9199; Internet: http://www.cablelabs.com
- ITU-T Recommendations available at http://www.itu.int
- IETF RFCs available at http://www.ietf.org/rfc.html

3 TERMS AND DEFINITIONS

PacketCable specifications use the following terms:

Access Control	Limiting the flow of information from the resources of a system only to authorized persons, programs, processes or other system resources on a network.
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.
A-link	A-Links are SS7 links that interconnect STPs and either SSPs or SCPs. 'A' stands for "Access".
Announcement Server	An announcement server plays informational announcements in PacketCable network. Announcements are needed for communications that do not complete and to provide enhanced information services to the user.
Asymmetric Key	An encryption key or a decryption key used in a public key cryptography, where encryption and decryption keys are always distinct.
Authentication	The process of verifying the claimed identity of an entity to another entity.
Authenticity	The ability to ensure that the given information is without modification or forgery and was in fact produced by the entity who claims to have given the information.
Authorization	The act of giving access to a service or device if one has the permission to have the access.
Cipher	An algorithm that transforms data between plaintext and ciphertext.
Ciphersuite	A set which must contain both an encryption algorithm and a message authentication algorithm (*e.g.,* a MAC or an HMAC). In general, it may also contain a key management algorithm, which does not apply in the context of PacketCable.
Ciphertext	The (encrypted) message output from a cryptographic algorithm that is in a format that is unintelligible.
Cleartext	The original (unencrypted) state of a message or data.
CNAM	Calling Name
Confidentiality	A way to ensure that information is not disclosed to any one other then the intended parties. Information is encrypted to provide confidentiality. Also known as Privacy.
Cryptoanalysis	The process of recovering the plaintext of a message or the encryption key without access to the key.
Cryptographic algorithm	An algorithm used to transfer text between plaintext and ciphertext.
Decipherment	A procedure applied to ciphertext to translate it into plaintext.
Decryption	A procedure applied to ciphertext to translate it into plaintext.
Decryption key	The key in the cryptographic algorithm to translate the ciphertext to plaintext
Digital certificate	A binding between an entity's public key and one or more attributes relating to its identity, also known as a public key certificate

Digital signature	A data value generated by a public key algorithm based on the contents of a block of data and a private key, yielding an individualized cryptographic checksum
Downstream	The direction from the head-end toward the subscriber location.
Encipherment	A method used to translate information in plaintext into ciphertext.
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
Errored Second	Any 1-sec interval containing at least one bit error.
Event Message	Message capturing a single portion of a connection
F-link	F-Links are SS7 links that directly connect two SS7 end points, such as two SSPs. 'F' stands for "Fully Associated"
Flow [IP Flow]	A unidirectional sequence of packets identified by ISO Layer 3 and Layer 4 header information. This information includes source/destination IP addresses, source/destination port numbers, protocol ID. Multiple multimedia streams may be carried in a single IP Flow.
Flow [DOCSIS Flow]	(a.k.a. DOCSIS-QoS "service flow"). A unidirectional sequence of packets associated with a SID and a QoS. Multiple multimedia streams may be carried in a single DOCSIS Flow.
Gateway	Devices bridging between the PacketCable 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 Signaling Gateway which sends and receives circuit switched network signaling tot he edge of the PacketCable network.
Header	Protocol control information located at the beginning of a protocol data unit.
Integrity	A way to ensure that information is not modified except by those who are authorized to do so.
IntraLATA	Within a Local Access Transport Area
Jitter	Variability in the delay of a stream of incoming packets making up a flow such as a voice communication.
Kerberos	A secret-key network authentication protocol that uses a choice of cryptographic algorithms for encryption and a centralized key database for authentication.
Key	A mathematical value input into the selected cryptographic algorithm.
Key Exchange	The swapping of public keys between entities to be used to encrypt communication between the entities.
Key Management	The process of distributing shared symmetric keys needed to run a security protocol.
Keying Material	A set of cryptographic keys and their associated parameters, normally associated with a particular run of a security protocol.
Key Pair	An associated public and private key where the correspondence between the two are mathematically related, but it is computationally infeasible to derive the private key from the public key.
Keyspace	The range of all possible values of the key for a particular cryptographic algorithm.

Latency	The time, expressed in quantity of symbols, taken for a signal element to pass through a device.
Link Encryption	Cryptography applied to data as it travels on data links between the network devices.
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.
Network Management	The functions related to the management of data across the network.
Network Management OSS	The functions related to the management of data link layer and physical layer resources and their stations across the data network supported by the hybrid fiber/coax system.
Nonce	A random value used only once which is sent in a communications protocol exchange to prevent replay attacks.
Non-Repudiation	The ability to prevent a sender from denying later that he or she sent a message or performed an action.
Off-Net Call	A communication connecting a PacketCable subscriber out to a user on the PSTN
On-Net Call	A communication placed by one customer to another customer entirely on the PacketCable Network
One-way Hash	A hash function that has an insignificant number of collisions upon output.
Plaintext	The original (unencrypted) state of a message or data.
Pre-shared Key	A shared secret key passed to both parties in a communication flow, using an unspecified manual or out-of-band mechanism.
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.
Private Key	The key used in public key cryptography that belongs to an individual entity and must be kept secret.
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 Certificate	A binding between an entity's public key and one or more attributes relating to its identity, also known as a digital certificate.
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.
Root Private Key	The private signing key of the highest level Certification Authority. It is normally used to sign public key certificates for lower-level Certification Authorities or other entities.
Root Public Key	The public key of the highest level Certification Authority, normally used to verify digital signatures that it generated with the corresponding root private key.

RSA Key Pair	A public/private key pair created for use with the RSA cryptographic algorithm.
Secret Key	The cryptographic key used in a symmetric key algorithm, which results in the secrecy of the encrypted data depending solely upon keeping the key a secret, also known as a symmetric key.
Session Key	A cryptographic key intended to encrypt data for a limited period of time, typically between a pair of entities.
Signed and Sealed	An "envelope" of information which has been signed with a digital signature and sealed by using encryption.
Subflow	A unidirectional flow of IP packets characterized by a single source and destination IP address and source and destination UDP/TCP port.
Symmetric Key	The cryptographic key used in a symmetric key algorithm, which results in the secrecy of the encrypted data depending solely upon keeping the key a secret, also known as a secret key.
Systems Management	Functions in the application layer related to the management of various open systems Interconnection (OSI) resources and their status across all layers of the OSI architecture.
Transit Delays	The time difference between the instant at which the first bit of a PDU crosses one designated boundary, and the instant at which the last bit of the same PDU crosses a second designated boundary.
Trunk	An analog or digital connection from a circuit switch which carries user media content and may carry voice signaling (MF, R2, etc.).
Tunnel Mode	An IPSEC (ESP or AH) mode that is applied to an IP tunnel, where an outer IP packet header (of an intermediate destination) is added on top of the original, inner IP header. In this case, the ESP or AH transform treats the inner IP header as if it were part of the packet payload. When the packet reaches the intermediate destination, the tunnel terminates and both the outer IP packet header and the IPSEC ESP or AH transform are taken out.
Upstream	The direction from the subscriber location toward the head-end.
X.509 certificate	A public key certificate specification developed as part of the ITU-T X.500 standards directory

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations and acronyms.

ACR	Anonymous Call Reject
AVP	Audio Video Profile
BLV	Busy Line Verification
CA	Call Agent
CFB	Call Forwarding on Busy
CFNA	Call Forwarding No Answer
CFU	Call Forward Unconditional
CLASS	Custom Local Area Signaling Services
CMS	Call Management System
CMTS	Cable Modem Termination System
CODEC	Coder-DECoder
DCS	Distributed Call Signaling
DOCSIS®	Data Over Cable System Interface Specification
DP	DCS Proxy
EI	Emergency Interrupt
EP	Endpoint
E.164	Telephone number standard of ITU
FQDN	Fully Qualified Domain Name
GC	Gate Controller
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSEC	IP Security
ITU	International Telecommunication Union
LAES	Lawfully Authorized Electronic Surveillance
LNP	Local Number Portability
LRN	Local Routing Number
MF	Multi-Frequency
MG	Media Gateway
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MTA	Multimedia Terminal Adapter
NCS	Network Call Signaling
OSPS	Operator Services Positioning System
OSS	Operations Support System
PSTN	Public Switched Telephone Network
QoS	Quality of Service

RFC	Request for Comments (IETF standard)
RGW	Residential Gateway
RKS	Record Keeping Server
RTP	Real-Time Transport Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SS7	Signaling System #7
TCP	Transmission Control Protocol
TGCP	Trunk gateway control protocol
UAC	User agent – Client
UAS	User agent – Server
UDP	User Datagram Protocol
URI	Universal Resource Identifier
URL	Universal Resource Locator

5 BACKGROUND AND MOTIVATION

The design of the Call Management Server Signaling (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 Signaling (NCS) and PSTN Gateway Call Signaling (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 CMSs residing in one or more service providers' networks. This specification covers the signaling performed between CMSs. 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 specification to take advantage of the widespread availability of packet networks to enable a rich set of applications and services for users.

CMSs may have a need to interconnect with other entities and networks which in turn introduces at least two issues:

- Interoperability - The extensions specified in CMSS may not all be supported.
- Security - When information is sent to or received from an entity, that entity may not be trusted (*e.g.,* a SIP endpoint or VoIP peer network) and special procedures may need to be invoked.

The current CMSS specification has been made flexible to accommodate the interoperability issue identified above. The security issue is resolved as follows; whenever signaling is sent to or received from a particular SIP entity, that entity is assumed to be trusted (see also Section 5.3).

A key element of the CMSS architecture is a recognition of the need for coordination between call signaling, 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 signaling and resource management is also needed to prevent fraud. The coordination between call signaling and Dynamic QoS [21] 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 (indirectly) 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 CMSs, 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 signaling 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:

1. It is essential to provide network-layer Quality of Service while allowing the service provider to derive revenues from the use of such service.
2. The CMSS architecture must allow for communication between CMSs in the network. At a high level, one may regard a CMS as performing complex signaling tasks on behalf of an endpoint. When the network includes multiple CMSs, CMSS should provide the call signaling function between the CMSs 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 CMSs (including the maintaining call state in the CMSs).
3. The CMSS architecture must enable interoperability with SIP entities that do not support all of the extensions specified by CMSS. CMSS compliant implementations will use the extensions and procedures defined in this document. However, when communicating with non-CMSS compliant implementations, CMSS compliant implementations will only use the extensions supported by both implementations. Furthermore, CMSs may be configured to require peer support for certain extensions, and fail calls with peers that do not support those extensions.
4. 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.
5. 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.
6. The architecture must be implementable, cost-effectively, at very large scale.

5.2 PacketCable Architecture

The CMS to CMS Signaling (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 PacketCable elements (Domain, Realm, etc.) are provided in the PacketCable Interdomain Architecture Technical Report

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 CMSs. 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 CMSs, and so for purposes of this specification 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 signaling 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 signaling 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 CMSs, and so for purposes of this specification, the MGC is considered identical to a CMS.

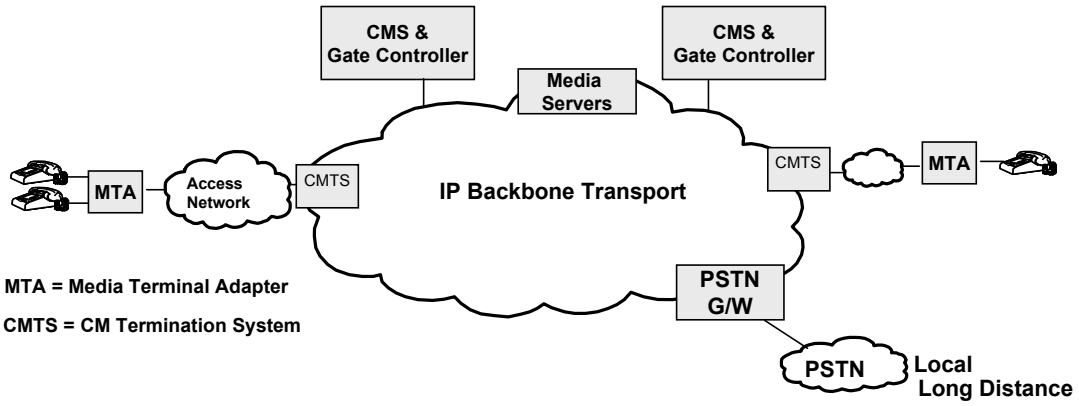


Figure 1. System Architecture – REALM

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 [23].

DQoS [21] 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.

CMSs 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, CMSs authenticate signaling messages and authorize requests for service on a session-by-session basis.
- Name/number translation and call routing: CMSs translate dialed numbers or names to a next-hop IP address based on call routing logic.
- Service-specific admission control: CMSs can implement a broad range of admission control policies for the telephony service. For example, CMSs 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 signaling overload.
- Signaling and service feature support: CMSs maintain and track all signaling activities to ensure 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 CMSs, since the Gate Controllers in the CMSs play a role as a policy server that controls when the CMTS can provide enhanced QoS service. There is also a trust relationship among CMSs. Details of the security model and mechanisms are specified in [26].

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 Signaling 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 signaling gateway acts as the IP peer of a PSTN endpoint for call signaling, providing signaling inter-working between the PacketCable network and conventional telephony signaling protocols such as ISUP/SS7. The MGC uses the PSTN Gateway Call Signaling Protocol (TGCP) to control the operation of the media gateway.

There are additional system elements that may be involved in providing the multimedia service [34]. 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 [26].

This specification 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], CMSs, 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/CMSs at the signaling level. The EBP interfaces to other PacketCable domains. Although these other PacketCable domains are outside the trust boundary, CMSS still trusts call signaling sent to and received from these other PacketCable domains.

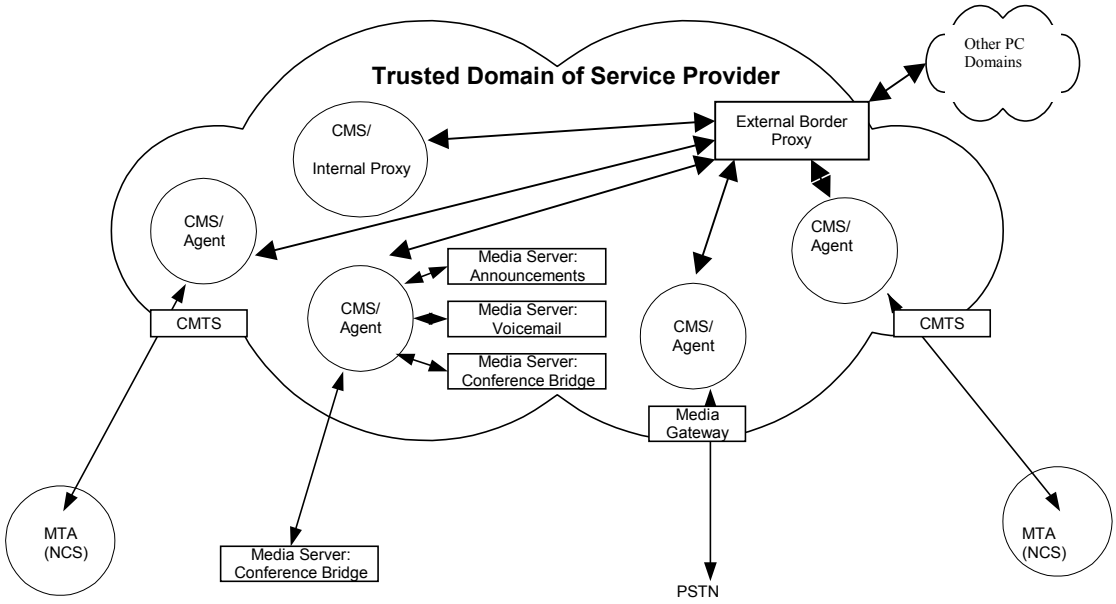


Figure 2. Trusted Domain of PacketCable Service Provider

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 signaling, number translation, and feature support. In addition to processing signaling 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 specification describes the messages required to support IP Telephony between entities that support one or more of the role indicated in the following table:

[1] More generally referred to as tandem proxies or simply proxies.
[2] but not the PSTN GW.

Role	Distinguishing function
Call Agent (CA) or CMS as defined above.	Support of endpoints implementing Network-Based Call Signaling (NCS) [24].
Media Gateway Controller (MGC)	Interworking with the PSTN. Use of PSTN Gateway Call Signaling (TGCP) [25] 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 levels vary between providers, the document assumes CMSs 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 CMSs. 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 signaling 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) is a proxy that can be used for inter-realm (intra-domain) signaling and the Exterior Border Proxy (EBP) is required for inter-domain signaling. See the PacketCable Inter-Domain Architecture Document 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.

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.

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

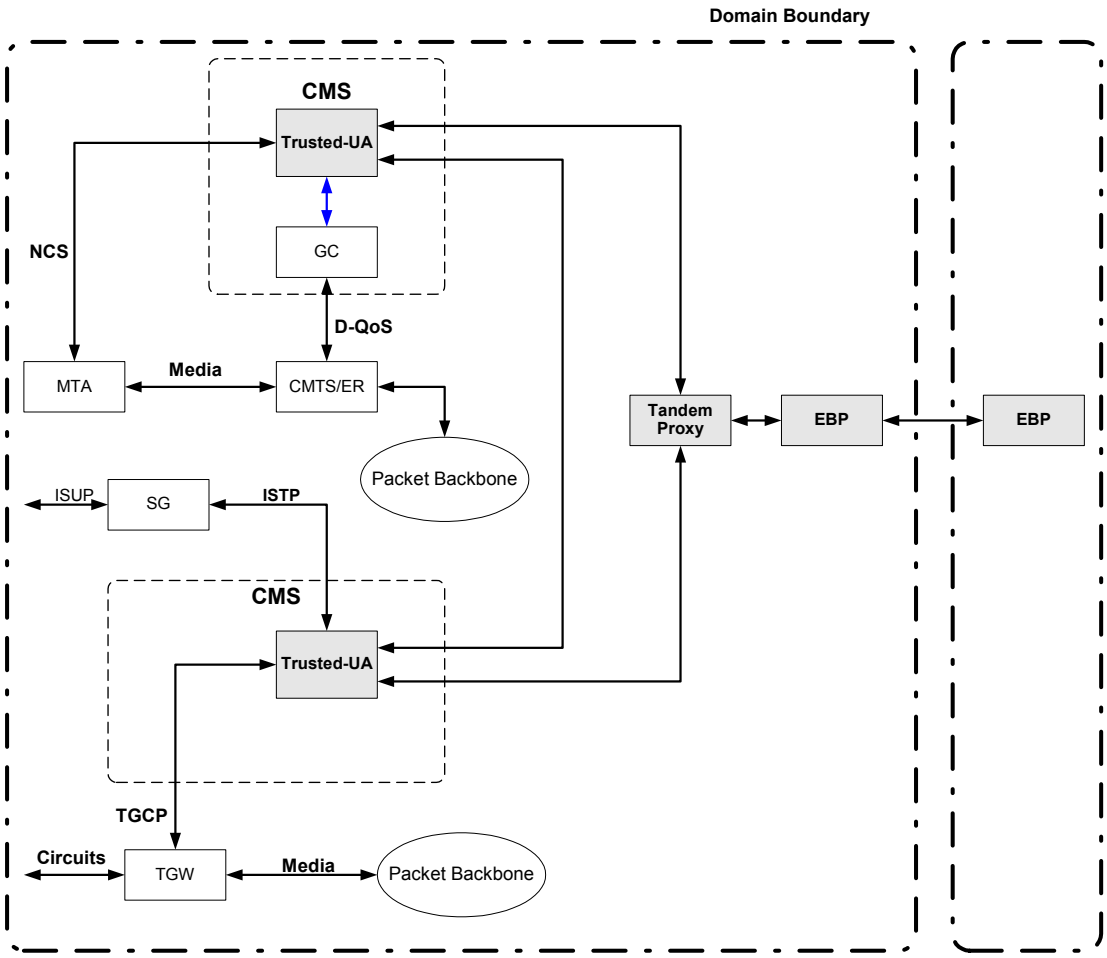


Figure 3. CMS Signaling Model

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 [24], the Media-Gateway-Controller (MGC) described in [25], and the Announcement Controller (ANC) described in [35]; 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 signaling exchanges directly.

The term CMS in this specification 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 [6] and a proxy MUST follow the requirements given for SIP proxies and redirect servers in [6].

Tandem proxies act as call routers and security association aggregation points. They may also provide additional functions such as signaling transformation gateways, signaling firewalls, etc. Depending on its role, a tandem proxy may remain in the call-signaling 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

PacketCable defines the Call Management Server (CMS) as a complex of server functions which support call signaling, number translation, call routing, feature support and admission control. Within the CMS complex, the PacketCable architecture allocates many of these responsibilities to the Proxy/CA/MGC and the Gate Controller (GC) function. In addition to processing session-signaling 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 signaling requests, the Gate Controller is responsible for the policy decision regarding whether a requested QoS level should be admitted. Upon receipt of signaling information, a CMS instructs the Gate Controller to authorize a QoS level in advance of any resource reservation signaling (see [21] 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 signaling. The CMTSes are the policy enforcement points, and ensure that the media path is provided the QoS it is authorized to receive.

The PacketCable 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, then 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 CMSs 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 CMSs 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 QoS "Pre-Condition" 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 CMSs 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 signaling 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 (see [23] for details (BCID) of the terminating event message stream, etc. (see [21] for further detail). 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 signaling 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 signaling 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 signaling with CMS_T. At this point in time, all preconditions have been met, and CMS_T can exchange NCS signaling 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 signaling 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 signaling 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

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

proxy. CMS_O acknowledges the 200-OK directly with the (11) ACK and exchanges NCS signaling 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 signaling with its MTA to delete the connection and release the resources reserved.

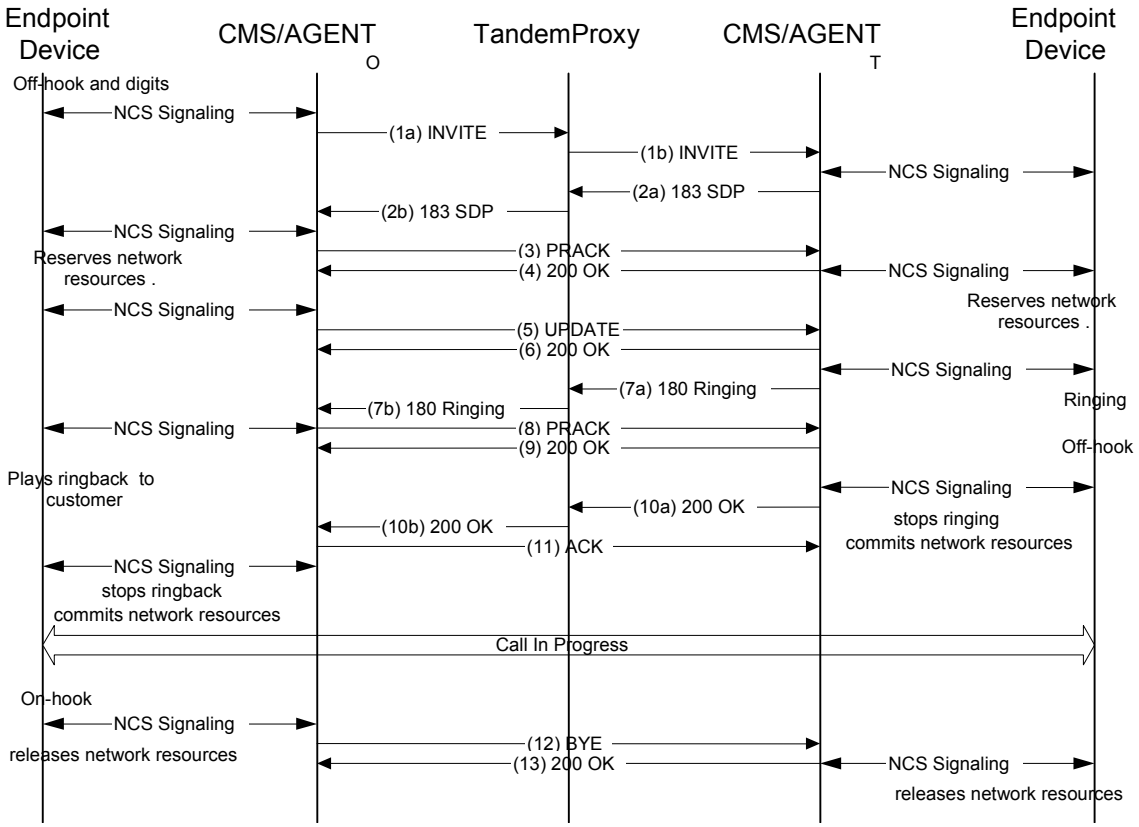


Figure 4. CMS – CMS Signaling Basic Call Flow

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 [30]. Note that CMSS in general is intended to be compatible with the interworking recommendations provided in [30].

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 signaling 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 signaling 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 signaling 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 signaling 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 [30] for details). In this latter case, the MGC would also exchange TGCP signaling with the MG instructing the MG to send packets to the MTA so that in-band media provided by the PSTN can be heard by the subscriber.
19. The CMS exchanges NCS signaling 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 signaling 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 signaling 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 signaling with the CMS to notify the CMS of the on hook condition.
29. The CMS exchanges NCS signaling 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.

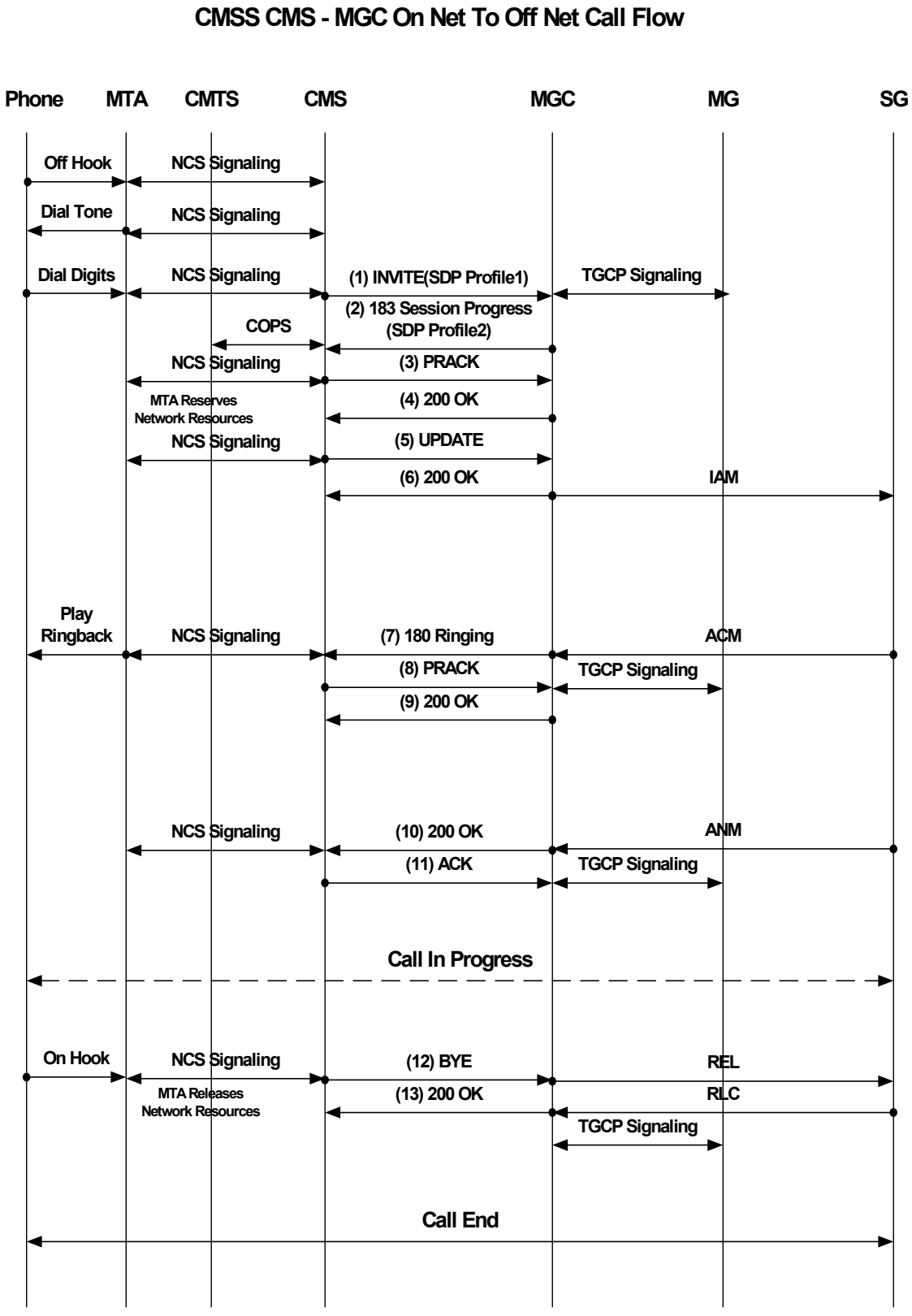


Figure 5. CMS to MGC Signaling

6 SIP PROFILE

This section defines a SIP [6] 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 [6], and section titles at all header levels track the section titles of the SIP specification [6].

This section and Section 7 define the nearly complete set of enhancements and restrictions to a standard SIP implementation based on [6]. 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 [6]. Sections 6 through 9 are considered normative.

6.1 Introduction

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 1.

Since there are no requirements in SIP/2.0 [6] 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 [6] Section 2.

Since there are no requirements in SIP/2.0 [6] Section 2, CMSS implementations are automatically compliant with it.

6.3 Terminology

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 3.

Since there are no requirements in SIP/2.0 [6] 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 [6] Section 4.

Since there are no requirements in SIP/2.0 [6] 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 [6] Section 5.

Since there are no requirements in SIP/2.0 [6] Section 5, CMSS implementations are automatically compliant with it.

6.6 Definitions

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] 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 [6] Section 6, CMSS implementations are automatically compliant with it.

6.7 SIP Messages

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 7, except as noted below.

6.7.1 Requests

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] 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 SIP and SIPS (from here on collectively referred to as SIP(s) URI(s) as defined in [6] and tel URI as defined in Section 7.1 MUST be supported in the Request-URI.

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

6.7.2 Responses

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 7.2.

6.7.3 Header Fields

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] 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 [6] Section 7.3.3.

6.7.4 Bodies

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] 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. Furthermore, the message body type

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

"application/sdp" MUST be supported in success responses to OPTIONS requests, and 488 responses to INVITE requests.

Refer to Appendix B 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 [6] Section 7.4.2.

6.7.5 Framing SIP Messages

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] 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 [6] 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 is OPTIONAL, however if supported, it MUST be as specified in [6] Section 8.1.

When a request is forked, multiple responses may be received, each of which results in the creation of an early dialog. Furthermore, each response may contain an answer and each early dialog may involve early media. Support for multiple simultaneous media streams for a single call, however is OPTIONAL for an MTA. In particular, MTAs may not be able to receive media from multiple different sources simultaneously, *e.g.*, due to resource constraints, or when security services are used on the media stream. Furthermore, having multiple different sources sending media to the MTA at the same time has QoS implications that are outside the scope of this document.

As a result of this, whenever a given request results in multiple early dialogs with multiple simultaneous media streams, the UAC SHOULD NOT enable early media on more than one of these dialogs. The details of how that is achieved are left to the implementation, however below are two options:

- The UAC may provide the MTA with the answer SDP from one of the early dialogs. The MTA in turn will only process media received in accordance with that SDP.

- When answer SDPs are received on the early dialogs, the UAC may issue new offers on all but one of these to suppress early media. Note that this will also suppress final media until a new offer/answer exchange has been performed.

Note that when media stream security is not being used, and the answer SDP is not provided until the final answer, the UAC cannot prevent the MTA from receiving multiple early media streams.

Once a final dialog has been established, media SHOULD be allowed on that dialog only.

6.8.1.1 Generating the Request

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] 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 or SIPS 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.

By default, CMSS compliant implementations MUST NOT require support for any particular extension, however a given deployment MAY be configured to require one or more extensions to be supported. A default CMSS implementation will thus not use the Require or Proxy-Require header fields in requests outside of a dialog. Instead, a list of supported extensions will be included in the SIP Supported header in requests outside of a dialog. Once a dialog has been established (whether early or final), one or more of the supported extensions can then be required.

The above defines the default CMSS implementation, however a particular deployment MAY require that one or more extensions are supported. The set of extensions that are required to be used in a particular deployment can be configured on the CMSS. When one or more of such extensions have been configured as required, requests outside of dialogs will include the relevant option tags in the Require and/or Proxy-Require header fields. It should be noted that signaling with endpoints as well as proxies that do not support a required extension will result in failures

The IETF allows option tags to be defined for their purpose only in standards-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 Processing Responses

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 8.1.3 except as noted in this section.

When receiving a 401 (Unauthorized) or 407 (Proxy Authentication Required) response, the SIP authorization procedure SHOULD only be followed if the UAC has credentials for the realm in question.

When receiving a 420 (Bad Extension) response, the SIP retry procedures SHOULD NOT be followed in the case where the deployment has been configured to require support of any of the extensions listed in the Unsupported header. In all other cases, the SIP retry procedures SHOULD be followed.

6.8.2 UAS Behavior

The CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 8.2 and its subsections.

6.8.3 Redirect Servers

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 8.3.

6.9 Canceling a Request

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 9 and its subsections.

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

6.10 Registrations

Proxies MUST, and UACs MAY, support the SIP REGISTER method in accordance with [6] Section 10. Support for registrars is OPTIONAL; however if supported, it MUST be as specified in [6] Section 10.

6.11 Querying for Capabilities

CMSS compliant applications MUST be in accordance with SIP/2.0 [6], Section 11.

6.12 Dialogs

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 12 and its subsections.

6.13 Initiating a Session

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 13 and its subsections, except as noted in this section.

CMSS compliant implementations SHOULD include a message body of type "application/sdp" with the initial INVITE.

When an initial INVITE is received with an offer SDP, an answer SDP SHOULD be included in the first non-failure response to the INVITE. Note that if the response is not sent reliably, then the same answer SDP must be sent in the final response.

6.14 Modifying an Existing Session

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 14 and its subsections, except as noted in this section.

When a CMSS compliant implementation sends a re-INVITE, it SHOULD include a message body of type "application/sdp" with a new offer. Furthermore, CMSS compliant implementations 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 [6] Section 15 and its subsections.

6.16 Proxy Behavior

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] 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, systems that interact with CMSS compliant implementations should avoid using parallel forking and early media at the same time.

6.17 Transactions

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] 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 [6] Section 17 in case of any conflicts.

6.18 Transport

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 18 and its subsections.

6.19 Common Message Components

6.19.1 SIP and SIPS URI Component

The definition of a SIP and SIPS-URI is as given in [6] 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 [6] 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 [6], and the requirements for supporting each in CMSS are identified.

6.20.1 Accept

The Accept header MUST be supported as specified in [6] Section 20.1.

6.20.2 Accept-Encoding

The Accept-Encoding header MUST be supported as specified in [6] 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 [6] Section 20.3, except as noted below.

CMSS compliant implementations SHOULD include the "Accept-Language" header in requests or responses as defined in [6]. The value “en” for English MUST be supported, other values MAY be supported based on configuration data.

6.20.4 Alert-Info

Support for the Alert-Info header is OPTIONAL; however, if supported, it MUST be as specified in [6] Section 20.4.

It is noted that there are security risks associated with acting on the Alert-Info header as described in [6] Section 20.4.

6.20.5 Allow

The Allow header field MUST be supported as specified in [6] Section 20.5. CMSS compliant implementations MUST include the "Allow" header in the initial INVITE and the 200-OK response to the initial INVITE. When an Allow header is not received, the set of supported methods is unknown.

Refer to Appendix B 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 [6] 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 [6] Section 20.7.

See also Section 6.22.

6.20.8 Call-ID

The Call-ID header MUST be supported as specified in [6] 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 [6] 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 [6] Section 20.9.

It is noted that there are security risks associated with acting on the Call-Info header as described in [6] Section 20.9.

6.20.10 Contact

The Contact header MUST be supported as specified in [6] 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 or SIPS 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 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, SIPS, or tel-URI. If the new destination is a telephone number, it SHOULD contain a tel URI with the number of the new destination as described in 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 [6] 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 [6] 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 [6] Section 20.13.

6.20.14 Content-Length

The Content-Length header MUST be supported as specified in [6] Section 20.14.

It should be noted, that when stream-based protocols (such as TCP) are being used, a Content-Length header field must always be included, even if set to zero.

6.20.15 Content-Type

The "Content-Type" header MUST be supported as specified in [6] Section 20.15.

Refer to Appendix B for a list of supported values.

6.20.16 CSeq

The CSeq header MUST be supported as specified in [6] Section 20.16.

6.20.17 Date

Support for the Date header is OPTIONAL; however, if supported, it MUST be as specified in [6] 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 [6] Section 20.18.

It is noted that there are security risks associated with acting on the Error-Info header as described in [6] Section 20.18.

6.20.19 Expires

Support for the Expires header in the non-REGISTER methods and responses defined in [6] is OPTIONAL[8]; however, if supported, it MUST be as specified in [6] Section 20.19.

6.20.20 From

The From header MUST be supported as specified in [6] 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 [6] Section 20.21.

Note that use of this header is subject to security considerations as described in [6] Section 20.21.

6.20.22 Max-Forwards

The Max-Forwards header MUST be supported as specified in [6] Section 20.22, except as noted below.

When a CMSS compliant implementation of a back-to-back User Agent (B2BUA) forwards a request, it SHOULD 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 [6] Section 20.23.

6.20.24 MIME-Version

The MIME-Version header MUST be supported as specified in [6] 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 [6] Section 20.25.

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

6.20.26 Priority

Support for the Priority header is OPTIONAL; however if supported, it MUST be as specified in [6] 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 [6] Section 20.27.

See also Section 6.22.

6.20.28 Proxy-Authorization

Support for the Proxy-Authorization header is OPTIONAL; however, if supported, it MUST be as specified in [6] Section 20.28.

See also Section 6.22.

6.20.29 Proxy-Require

The Proxy-Require header MUST be supported as specified in [6] 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 B for a list of supported values.

Refer to Section 6.8.1.1 for considerations around the use of required proxy extensions.

6.20.30 Record-Route

The Record-Route header MUST be supported as specified in [6] 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 [6] Section 20.31.

6.20.32 Require

The "Require" header MUST be supported as specified in [6] 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 B for a list of supported values.

Refer to Section 6.8.1.1 for considerations around the use of UserAgent extensions.

6.20.33 Retry-After

Support for the Retry-After header is OPTIONAL; however, if supported, it MUST be as specified in [6] Section 20.33.

6.20.34 Route

The Route header MUST be supported as specified in [6] Section 20.34.

6.20.35 Server

Support for the Server header is OPTIONAL; however, if supported, it MUST be as specified [6] Section 20.35.

6.20.36 Subject

Support for the Subject header is OPTIONAL, however if supported, it MUST be as specified [6] Section 20.36.

6.20.37 Supported

The "Supported" header MUST be supported as specified in [6] 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 B for a list of supported values.

Refer to Section 6.8.1.1 for considerations around the use of UserAgent extensions.

6.20.38 Timestamp

The Timestamp header MUST be supported as specified in [6] 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 [6] Section 20.38.

6.20.39 To

The To header MUST be supported as specified in [6] Section 20.39, except as noted below.

In support of user Privacy, CMSS restricts the allowable contents of the SIP "To" header. The "To" header may indicate 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 (as defined in [28]), then the userinfo part of the addr-spec MUST contain a full E.164 number, including the country code.
- If a local telephone number is used (as defined in [28])), then the userinfo part of the addr-spec must contain a phone-context. When possible, the phone-context should be a country code.

When the call originator does not request privacy, CMSS compliant applications SHOULD generate a "To" header according to the following rules:

- If a global telephone number is used (as defined in [28]), then the userinfo part of the addr-spec must contain a full E.164 number, including the country code.
- If a local telephone number is used (as defined in [28]), then the userinfo part of the addr-spec must contain a phone-context. When possible, the phone-context should be a country code.

6.20.40 Unsupported

The Unsupported header MUST be supported as specified in [6]

6.20.41 User-Agent

Support for the User-Agent header is OPTIONAL; however, if supported, it MUST be as specified [6] Section 20.41.

6.20.42 Via

The Via header MUST be supported as specified in [6] 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 exist 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 [6] 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 [6] Section 20.44.

See also Section 6.22.

6.21 Response Codes

CMSS compliant applications MUST be in accordance with SIP/2.0 [6] Section 21 and its subsections, except as specified below.

CMSS compliant applications SHOULD 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 [26] and [6] Section 22. Where these documents conflict, [26] 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 [26] and [6] Section 23. Where these documents conflict, [26] takes precedence.

6.24 Examples

The examples provided in [6] 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 [6] Section 25.

6.26 Security Considerations: Threat Model and Security Usage Recommendations

CMSS compliant applications MUST comply with the PacketCable Security specification [26].

Support for the SIP Security Considerations specified in [6] Section 26 is considered OPTIONAL, unless they conflict with the PacketCable Security specification [26], in which case they MUST NOT be used.

6.27 Table of Timer Values

CMSS compliant applications MUST comply with SIP/2.0 [6] Appendix A.

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

7 SIP EXTENSIONS

SIP [6] 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 PacketCable 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 Signaling (DCS) framework, as described in [31].

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 PacketCable 1.5 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 [6] 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 CMSs and Proxies.

This specification extends SIP in several ways, which are summarized here. A CMSS compliant implementation MUST support all of these:

- 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 [11].
- 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 [12] and [13].
- 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 [16].
- 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 [7].
- 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 [17], [9] and [18].

- 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 [10].

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

This section, and Section 8, define the nearly complete set of enhancements and requirements to a standard SIP implementation based on [6]. 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 as specified in RFC3966 [28], except as noted in this section.

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

CMSS supports extensions to the tel URI that relate to number portability and freephone service, as specified in draft-ietf-iptel-tel-np-04 [29], except as noted in this section.

CMSS supports additional extensions to the tel URI to define a dial-around-indicator to indicate how the Carrier ID Code (CIC) was derived [41].

These extensions are defined as optional in the tel URI sense. RFC3966 [28] specifies that an implementation MAY ignore optional parameters. However, CMSS compliant applications MUST support the draft-ietf-iptel-tel-np-04 [29], extensions and the dial-around-indicator extension, and hence they MUST NOT be ignored when received.

The dial-around-indicator extension defines a new tel URI parameter that has the following syntax. The syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC2234 [2].

```
dai = *1(dial-around-indicator)

dial-around-indicator = “dai=” 1(dial-around-indicator-value)

dial-around-indicator-value =
                                "presub" /
                                "presub-da" /
                                "presub-daUnkwn" /
                                "da" /
                                "CIC-chrgPty" /
                                "altCIC-chrgPty" /
                                "verbal-clgPty" /
                                "verbal-chrgPty" /
                                "emergency" /
                                "operator" /
```

CMSS compliant implementations will use the draft-ietf-iptel-tel-np-04 [29] 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 and a CIC parameter is included in the response to the freephone database query, 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 and carrier-based routing is to be done for the call, then the CIC MUST identity the pre-subscribed carrier for the caller, unless the caller dialed a carrier access code, in which case the CIC for that carrier MUST be used. Also, if the number dialed is not a freephone number and carrier-based routing is to be done for the call, then the dial-around-indicator parameter MUST be included. The dial-around-indicator parameter SHOULD NOT be included in any other case. When the dial-around-indicator parameter is included, it MUST be set to one of the following values:
  - "presub" - the CIC contains the caller's presubscribed carrier
  - "presub-da" - the CIC contains the caller's dialed carrier-identification-code; the caller has a presubscribed carrier
  - "presub-daUnkwn" - the CIC may contain either a caller dialed carrier-identification-code or the caller's presubscribed carrier
  - "da" - the CIC contains the caller's dialed carrier-identification-code; the caller does not have a presubscribed carrier
  - "CIC-chrgPty" - the CIC is the preferred carrier of the charged party
  - "altCIC-chrgPty" - the CIC is the alternate carrier of the charged party
  - "verbal-clgPty" - the CIC was delivered verbally by the calling party
  - "verbal-chrgPty" - the CIC was delivered verbally by the charged party
  - "emergency" - this is an emergency call
  - "operator" - the carrier was selected by a network operator

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 Tel or SIP(s) 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(s) or Tel URI with the "npdi" parameter in the telephone-subscriber part.

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 Tel or SIP(s) 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(s) or Tel 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 [7], except as noted in this section.

CMSS compliant implementations MUST by default include a Supported header containing the value “100rel” in the initial INVITE request. Alternatively, if a CMS is configured to require use of reliable provisional responses, the initial INVITE request MUST include a Require header containing the value “100rel”.

When a CMSS compliant implementation receives an INVITE request with a Supported header that contains the value “100rel”, the CMS MUST send all non-100 provisional response reliably as defined in [7].

7.3 SIP UPDATE Method

CMSS compliant implementations MUST support the extensions defined in [10] except as noted in this section.

CMSs 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 [11] except as noted in the subsections below.

7.4.1 Procedures at an Originating CMS

CMSS compliant implementations MUST support use of QoS preconditions as defined in the following subsections, however if a given deployment does not want to use QoS preconditions, the session originator (CMS_O) MUST NOT include any QoS precondition attributes in the SDP, and the procedures below do not apply. When QoS preconditions are supported and can be used, one of the following three procedures MUST be followed:

7.4.1.1 Default Operation Using QoS Preconditions Strength "Optional"

Unless configured otherwise, the session originator (CMS_O) MUST include a Supported header containing the value "precondition". For each media flow in the SDP sent with the INVITE, the precondition-type MUST be "qos", the strength-tag MUST be "optional", the desired status-type MUST be "e2e", and the direction-tag MUST be "sendrecv". When CMS_O receives an answer SDP, CMS_O MUST see if the answer SDP contains any QoS preconditions: if it does, then session establishment MUST continue in accordance with the QoS preconditions. Otherwise, session establishment is already progressing and CMS_O MUST simply continue with local QoS operations independent of session progress.

7.4.1.2 QoS Preconditions Strength "None"

If a given deployment has no preference about the use of QoS preconditions, the session originator (CMS_O) SHOULD include a Supported header containing the value "precondition". For each QoS precondition it includes in the SDP, the precondition-type MUST then be "qos", the strength-tag MUST be "none", the desired status-type MUST be "e2e", and the direction-tag MUST be "sendrecv".

When CMS_O receives an answer SDP, CMS_O MUST see if the answer SDP contains any QoS preconditions; if it does, then session establishment MUST continue in accordance with the QoS preconditions. Otherwise, session establishment is already progressing and CMS_O MUST simply continue with local QoS operations independent of session progress.

7.4.1.3 QoS Preconditions Strength "Mandatory"

If a given deployment requires use of QoS preconditions, the session originator (CMS_O) MUST include a Require header containing the value "precondition". For each media flow in the SDP sent with the INVITE, the precondition-type MUST be "qos", the strength-tag MUST be "mandatory", the desired status-type MUST be "e2e", and the direction-tag MUST be "sendrecv".

7.4.2 Procedures at a Terminating CMS

When an INVITE is received without any QoS preconditions, the procedures in this section do not apply; instead normal SIP processing of the call occurs. When an INVITE with QoS preconditions is received, three sets of procedures are defined in the following subsections.

7.4.2.1 QoS Preconditions Strength "None" or "Optional"

Unless configured otherwise, on receipt of an INVITE request containing "optional" or "none" QoS preconditions, a terminating CMS (CMS_T) MUST generate a 183-Session-Progress response with an SDP containing mandatory preconditions. For each media flow with QoS precondition in the SDP sent, the precondition-type MUST be "qos", the strength-tag MUST be "mandatory", 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.

7.4.2.2 QoS Preconditions Strength "Mandatory"

On receipt of an INVITE request containing mandatory QoS preconditions, a terminating CMS (CMS_T) MUST generate a 183-Session-Progress response with an SDP. For each media flow with QoS precondition in the SDP sent, the precondition-type MUST be "qos", the strength-tag "mandatory", 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".

7.5 SIP-Specific Event Notification

CMSS compliant implementations MUST support the extensions defined in [9].

7.6 The REFER Method

CMSS compliant implementations MUST support the extensions defined in [17] 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, and SHOULD include the additional header parameters for P-DCS-Laes, and P-DCS-Redirect, as specified in Section 7.7. **NOTE**: Please refer to section 7.7.2. for additional guidance regarding the usage of P-DCS-Laes and P-DCS-Redirect headers.

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, SIPS, or tel 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 [16] 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-LAES and P-DCS-REDIRECT

For the purposes of this specification, the 4th paragraph in Section 8 of [16] is intended to require that CMSS compliant devices fully implement the P-DCS-LAES and P-DCS-REDIRECT headers as defined in [16] (with the exceptions noted in this specification), and also implement a mechanism that allows the service provider to turn this feature on or off as required by applicable legislation. For example, an operator may be required to include these headers as specified in [16] for calls within the operator's own network, while excluding the headers for calls that terminate on another operator's network.

7.7.2.1 P-DCS-LAES Header

The P-DCS-LAES-Header is used to pass the responsibility for performing electronic surveillance on a call from one CMS to another. For example, if a call terminates to a line that is marked for surveillance, and the line also has call-forwarding activated, then the terminating CMS can include the P-DCS-LAES header in the forwarded INVITE request or 302-Redirect response to inform the forwarded-to CMS that it should perform the surveillance function.

The P-DCS-LAES header contains information to support surveillance of both call-data and call-content. The call-data information, which consists of the BCID assigned to the call-data event stream and the address of the Delivery Function (DF) to receive the call-data events, is always present. The call-content information, which consists of the CCCID assigned to the call-content stream, plus the address of the DF to receive call-content, is optional.

The P-DCS-LAES header has the following syntax:

```
P-DCS-LAES        = "P-DCS-LAES" HCOLON Laes-sig *(SEMI Laes-param)
Laes-sig          = hostport
Laes-param        = Laes-content / Laes-key / Laes-cccid / Laes-bcid /
                    generic-param
Laes-content      = "content" EQUAL hostport
Laes-key          = "key" EQUAL token
Laes-bcid         = "bcid" EQUAL 1*48(HEXDIG)
Laes-cccid        = "cccid" EQUAL 1*8(HEXDIG)
```

The above syntax conforms to and enhances the syntax specified in the SIP Proxy-to-Proxy Extensions RFC 3603 [16]. The Laes-bcid field MUST always be present. The Laes-cccid field MUST be present when the Laes-content field is present. The Laes-key field MUST NOT be included.

7.7.2.2 Surveillance Procedures at Originating CMS

An originating CMS (CMS_O) is required to perform electronic surveillance functions for an originating call if the originating line has an outstanding lawfully authorized electronic surveillance order. An originating CMS may also choose to perform electronic surveillance functions on behalf of a terminating CMS for certain-call forwarding and call-transfer scenarios. The following subsections detail the responsibilities of the originating CMS for these various surveillance scenarios.

7.7.2.2.1 CMS_O Receives REFER Request or Redirect Response

When CMS_O receives a 3XX Redirect response containing a P-DCS-Laes header in response to an INVITE, or receives a REFER request containing a P-DCS-Laes header in the Refer-To header for an active dialog, then it MUST copy the received P-DCS-Laes header into the subsequent INVITE that is generated as a result of the REFER or Redirect. This will enable the new terminating CMS to perform the surveillance on behalf of the CMS that generated the Redirect response or REFER request message.

The following subsections describe the CMS_O behavior when the P-DCS-Leas header cannot be forwarded to the new terminating CMS for some reason.

7.7.2.2.1.1 Redirected Call Ends Early

If CMS_O receives a REFER request or 3XX Redirect response message as described above, but the call ends before the subsequent INVITE is sent (say the call is abandoned), then CMS_O MUST send a SurveillanceStop message to its local DF containing the following information:

- The local BCID already assigned to the call (note, this is a required field in the event message header),
- The remote BCID assigned by CMS_T and received in the P-DCS-Laes header,
- The call-data IP address and port of the remote DF of CMS_T received in the P-DCS-Laes header,
- An indicator specifying that both call-data and call-content surveillance are to be stopped,
- An indicator specifying that the local surveillance session (if active) and remote surveillance session are to be stopped.

This will tell the remote DF (i.e., the DF of CMS_T) that the call has ended, and not to expect further surveillance information.

7.7.2.2.1.2 P-DCS-LAES Header Cannot Be Included in Subsequent INVITE

If CMS_O is unable to include the P-DCS-Laes header in the subsequent INVITE for some reason (see section 7.7.2), then CMS_O may choose either to perform the required surveillance function or to stop the remote surveillance session.

7.7.2.2.1.2.1 CMS_O Chooses To Perform Requested Surveillance

If CMS_O chooses to perform the requested call-data surveillance function, it MUST send a SignalingStart message to its local DF containing the following information:

- The local BCID already assigned to the call (note, this is a required field in the event message header),
- The remote BCID assigned by CMS_T and received in the P-DCS-Laes header,
- The call-data IP address and port of the remote DF of CMS_T received in the P-DCS-Laes header.

This will bind the local BCID to the remote surveillance session, and thus enable the local DF to relay subsequent call-data events for this call to the remote DF and BCID of the terminating CMS. Note that if CMS_O is already monitoring the call (*e.g.,* due to an outstanding lawfully authorized surveillance order on the originating subscriber) when it receives a P-DCS-Laes header, then it MUST send a second SignalingStart message to its local DF, containing the appropriate parameters as specified above. This means that the DF must be able to receive two SignalingStart messages for the same call. The second SignalingStart should be used by the local DF only to establish the local-to-remote binding for relay of call-data and possibly call-content to the remote DF.

If the P-DCS-Laes header received in the 3XX Redirect response or REFER request also indicates that call content surveillance is to be performed (in addition to call data), then CMS_O MUST allocate a local CCCID for the call and request the CMTS of the originating line (or MG of the originating trunk if the originator is off-net) to provide a copy of the call content to the local DF. (Note, if the originating line or trunk is already being surveilled, then CMS_O simply uses the already allocated CCCID). In addition to the call-data information specified above, CMS_O MUST include the following data in the SignalingStart message to the local DF:

- The local CCCID assigned to the call,
- The remote CCCID assigned by CMS_T and received in the P-DCS-Laes header,
- The call-content IP address and port of the remote DF of CMS_T received in the P-DCS-Laes header.

This will enable the local DF to relay subsequent call-content packets received from the originating CMTS or MG for this call to the remote DF and CCCID of the terminating CMS.

When the call ends, CMS_O MUST send a SurveillanceStop message to its local DF containing the local BCID and indicating that both local and remote call-data and call-content surveillance are to be stopped. This will terminate the surveillance session in the remote DF for both call-data and call-content (if applicable), clear the local-to-remote binding information in the local DF, and stop the local surveillance session (if active).

7.7.2.2.1.2.2 CMS_O Chooses to Perform Call-Data But Not Call-Content

If the P-DCS-Laes header received in the 3XX Redirect response or REFER request indicates that both call-data and call-content surveillance are to be performed, but CMS_O chooses to support only call-data (and not call-content), then it MUST send a SignalingStart message to its local DF containing the call-data information specified in section 7.7.2.2.1.2.1.

This will enable the local DF to relay subsequent call-data events for this call to the remote DF and BCID of the terminating CMS.

In addition, CMS_O MUST the send a SurveillanceStop message containing the following information:

- The local BCID assigned by CMS_O to the call (this BCID was bound to the remote surveillance session by the previous SignalingStart message),
- An indicator specifying that only the remote surveillance session is to be stopped (this allows a local surveillance session that may be in progress on the originating endpoint to continue),
- An indicator specifying that (only) call-content surveillance is to be stopped (this allows the remote call-data surveillance to continue).

On receiving this message, the local DF will send a SurveillanceStop message to stop the remote call-content surveillance session.

7.7.2.2.1.2.3 CMS_O Chooses Not To Perform the Requested Surveillance

If CMS_O chooses not to perform any of the requested surveillance functions, then it MUST send a SurveillanceStop message to its local DF containing the following information:

- The local BCID assigned by CMS_O to the call (note that even though the local BCID is a required parameter, it doesn't convey any useful information in this case since the local BCID was not bound to the remote surveillance session by a previous SignalingStart message),
- The remote BCID assigned by CMS_T and received in the P-DCS-Laes header,
- The call-data IP address and port of the remote DF of CMS_T received in the P-DCS-Laes header
- An indicator specifying that only the remote surveillance session is to be stopped (this allows a local surveillance session that may be in progress on the originating endpoint to continue),
- An indicator specifying that both call-data and call-content surveillance are to be stopped.

On receiving this message, the local DF will send a SurveillanceStop message to stop the remote surveillance session.

7.7.2.3 Surveillance Procedures at Terminating CMS

A terminating CMS is required to perform surveillance functions for an incoming call for two cases; when the terminating line has an outstanding lawfully authorized electronic surveillance order, and when the

received INVITE request contains a P-DCS-Laes header requesting the terminating CMS to perform surveillance for this call on behalf of a remote CMS. The following sections detail the responsibilities of the terminating CMS (CMS_T) for these cases.

7.7.2.3.1 Terminating Line is Able to Accept the Call

If the terminating line is able to accept the call, and either a local outstanding lawfully authorized electronic surveillance order exists for the line, or a P-DCS-Laes header is received in the INVITE, then CMS_T MUST send a SignalingStart message to the local DF containing the identity of the terminating line and the local BCID assigned to the call (note, the local BCID is a mandatory field). This will associate the terminating line to the BCID for all subsequent call-data event messages sent to the local DF for this call. If a P-DCS-Laes header was received, then CMS_T MUST include the following additional information in the SignalingStart message:

- The remote BCID assigned by the remote CMS and received in the P-DCS-Laes header,
- The call-data IP address and port of the remote DF received in the P-DCS-Laes header.

This additional data will enable the local DF to relay subsequent call-data events for this call to the remote DF and BCID of the remote CMS.

If either the local electronic surveillance order or the received P-DCS-Laes header indicates that call-content surveillance is to be performed, then CMS_T MUST allocate a local CCCID for the call and request the CMTS of the terminating line (or MG of the terminating trunk if the terminator is off-net) to provide a copy of the call content to the local DF. In addition to the call-data parameters specified above, CMS_T MUST include the local CCCID in the SignalingStart message to the local DF. If a P-DCS-Laes header was received that indicates that call-content surveillance is to be performed, then CMS_T MUST include the following additional information in the SignalingStart message:

- The remote CCCID assigned by the remote CMS and received in the P-DCS-Laes header,
- The call-content IP address and port of the remote DF received in the P-DCS-Laes header.

This additional data will enable the local DF to relay subsequent call-content packets received from the terminating CMTS or MG for this call to the remote DF and CCCID.

When the call ends, CMS_T MUST send a SurveillanceStop message to its local DF containing the local BCID and indicating that both local and remote call-data and call-content surveillance are to be stopped. This will terminate the surveillance session in the remote DF for both call-data and call-content (if applicable), clear the local-to-remote binding information in the local DF, and stop the local surveillance session (if active).

7.7.2.3.2 Terminating Line is Unable to Accept the Call

If CMS_T receives an INVITE request containing a P-DCS-Laes header, and the terminating endpoint is not able to accept the call for some reason (*e.g.*, line is busy, line is unknown), and CMS_T does not need to otherwise initiate a surveillance session, then CMS_T MUST send a SurveillanceStop message containing the following information:

- The local BCID assigned by CMS_T to this call (note, to avoid affecting other surveillance sessions, CMS_T must use the BCID for this call, and not the BCID of any other in-progress call on the same line),
- The remote BCID received in the P-DCS-Laes header,
- The call-data IP address and port of the remote DF received in the P-DCS-Laes header,
- An indicator specifying that both call-data and (if active) call-content surveillance are to be stopped.

On receiving this message, the local DF will send a SurveillanceStop message to stop the remote surveillance session.

Note, there are cases where CMS_T must initiate a surveillance session for the terminating call even though it does not actually offer the call to the terminating line. For example, if the terminating line activates the do-not-disturb feature, then CMS_T must initiate a surveillance session to record a service instance of do-not-disturb, and then immediately stop the surveillance session. These cases are handled as specified in section 7.7.2.3.1.

7.7.2.3.3 Terminating CMS is Unable to Perform Call-Content Surveillance

If CMS_T receives an INVITE containing a P-DCS-Laes header requesting call-data and call-content surveillance, and CMS_T is unable to perform the call-content surveillance for some reason (*e.g.*, call routed to voice mail server), then CMS_T must continue to perform the call-data surveillance as specified in section 7.7.2.3.1. Once this procedure has established the local-to-remote call-data surveillance information in the local DF, CMS_T MUST send SurveillanceStop message containing the following information:

- The local BCID assigned to the terminating call,
- An indication that call-content surveillance is to be terminated.

This will enable the local DF to inform the remote DF that the call-content surveillance session has ended while allowing the call-data surveillance to continue for the duration of the call.

7.7.2.3.4 Terminating CMS Redirects or Transfers the Call

If CMS_T is required to perform surveillance on a call (either as a result of terminating to a subscriber with a lawfully authorized surveillance order, or as specified in the P-DCS-Laes header of the INVITE message from the CMS_O), but the call is redirected or transferred to a new terminating line, then CMS_T MUST send a SignalingStart message to the local DF containing the identity of the terminating line and the local BCID assigned to the call. This will associate the terminating line to the local BCID for all subsequent call-data event messages sent to the local DF for this call. If a P-DCS-Laes header was received, then CMS_T MUST include the following additional information in the SignalingStart message:

- The remote BCID assigned by the remote CMS and received in the P-DCS-Laes header, The call-data IP address and port of the remote DF received in the P-DCS-Laes header.

This additional data will enable the local DF to relay subsequent call-data events for this call to the remote DF and BCID of the remote CMS.

If either the local electronic surveillance order or the received P-DCS-Laes header indicates that call-content surveillance is to be performed, then CMS_T MUST allocate a local CCCID for the call. (Note that if the call is forwarded immediately on termination, then CMS_T does not request the terminating CMTS or MG to provide a copy of the call-content for this call.) In addition to the call-data parameters specified above, CMS_T MUST include the local CCCID in the Signaling Start message to the local DF.

If a P-DCS-Laes header was received indicating that call-content surveillance is to be performed, then CMS_T MUST include the following additional information in the SignalingStart message:

- The remote CCCID assigned by the remote CMS and received in the P-DCS-Laes header,
- The call-content IP address and port of the remote DF received in the P-DCS-Laes header.

This additional data will enable the local DF to relay subsequent call-content packets received from the final terminating CMTS or MG for this call to the remote DF and CCCID.

The remaining action taken by CMS_T depends on whether it redirects the call by sending a REFER request or 3XX Redirect response to the originating CMS, or by remaining in the signaling path as a proxy for the remainder of the call.

7.7.2.3.4.1 CMS_T Sends REFER Request or Redirect Response

If CMS_T transfers or forwards the call by sending a REFER request or Redirect response to the originating CMS, then it MUST include a P-DCS-Laes header in the Redirect response or in the Refer-To header of the REFER request. (Note, the CMS initiating the REFER is sometimes referred to as CMS_I in this specification.) The P-DCS-Laes header MUST contain the following information:

- The local BCID assigned to the call,
- The call-data IP address and port of the local DF.
- If CMS_T is required to perform call-content surveillance for the call, then it MUST include the following additional data in the P-DCS-Laes header:
- The local CCCID assigned to the call,
- The call-content IP address and port of the local DF.

This will enable the DF of the new terminating CMS to relay call-data events and (if required) call-content packets to the local DF.

7.7.2.3.4.2 CMS_T Remains in the Signaling Path as a Proxy

If CMS_T is to remain in the signaling path, and it is allowed to include a P-DCS-Laes header in the new INVITE request per section 7.7.2, then CMS_T MUST include the P-DCS-Laes header in the new INVITE request. The P-DCS-Laes header MUST contain the following information:

- The local BCID assigned to the call,
- The call-data IP address and port of the local DF.

If CMS_T is required to perform call-content surveillance for the call, and it is allowed to include a P-DCS-Laes header in the new INVITE request per section 7.7.2, then it MUST include the following additional data in the P-DCS-Laes header:

- The local CCCID assigned to the call,
- The call-content IP address and port of the local DF.

This will enable the DF of the new terminating CMS to relay call-data events and (if required) call-content packets to the local DF.

Once CMS_T has sent the INVITE containing the P-DCS-Laes header, it will not generate any further call-data events to its local DF for this call.

If CMS_T is to remain in the signaling path, but it is not allowed to include a P-DCS-Laes header in the new INVITE request per section 7.7.2, then it MUST support the call-data surveillance as specified in section 7.7.2.3.1. Furthermore, if call-content surveillance is required, then CMS_T MUST send a SignalingStop message to terminate the call-content surveillance session.

7.7.2.4 Surveillance Procedures at a CMS Proxy

Electronic surveillance does not impose any special requirements on CMS tandem-proxies that act as dedicated routing proxies. Tandem-proxies will pass the P-DCS-Laes header transparently, and will not communicate with the DF.

7.7.3 P-DCS-Billing-Info

CMSS defines a new parameter called JIP-param in the P-DCS-Billing-Info header. The JIP-param identifies the CMS serving the calling party by specifying the NPA-NXX of the originating switch or network node (similar to the Jurisdiction Information Parameter (JIP) used in SS7 ISUP).

The JIP-param has the following syntax:

```
JIP-param               = "jip" EQUAL jip
jip                     = LDQUOT 1*phonedigit-hex jip-context RDQUOT
jip-context             = ";jip-context=" jip-descriptor
jip-descriptor          = global-hex-digits
global-hex-digits       = "+" 1*3(phonedigit) *phonedigit-hex
phonedigit              = DIGIT / [ visual-separator ]
phonedigit-hex          = HEXDIG / "*" / "#" / [ visual-separator ]
visual-separator        = "-" / "." / "(" / ")"
```

7.7.3.1 Procedures at an Originating CMS

An originating CMS that passes a REFER or initial INVITE request to a proxy or a terminating CMS MUST include the JIP-param in the P-DCS-Billing-Info header indicating the NPA-NNX of the originating CMS. In the case of a REFER request, the JIP-param MUST be included in the P-DCS-Billing header in the Refer-To header. Similarly, REQpending3 an originating MGC that receives the originator's jurisdiction information from the PSTN and passes an initial INVITE to a proxy or terminating CMS MUST include the jurisdiction information in the JIP-param in the P-DCS-Billing-Info header.

The following example shows how a JIP-param would be encoded when the calling number is ported to a CMS serving NPA-NXX 202-544:

jip="202554;jip-context=+1"

7.7.3.2 Procedures at a Terminating CMS

A terminating CMS that returns a 3xx-Redirect response to an originating CMS MUST include the JIP-param in the P-DCS-Billing-Info header indicating the NPA-NNX of the "forwarded from" party.

7.7.3.3 Procedures at a Proxy

No specific procedures are defined.

7.7.4 P-DCS Option Tag

The extensions defined in [16] 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 [16].

CMSS compliant implementations may use the option tag "P-DCS" in the Supported header, and they MUST accept requests with a Require value of "P-DCS".

By default, CMSS compliant implementations MUST NOT include a Require header field with the value "P-DCS", except when performing an INVITE for Busy-Line-Verify or Emergency-Interrupt as described

in Section 8.4.4.4 and 8.4.4.5, or when the peer has indicated support for the option tag. It SHOULD be possible to disable use of the "P-DCS" option tag for such calls as well (either by disabling use of the option tag, or by retrying without it upon failure due to lack of peer support).

7.8 The SIP "Replaces" Header

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

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 [13] 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 [12] and [13] and a value of "critical" as described in [12], 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.

CMSS compliant implementations MUST populate the URI in the P-Asserted-Indentity header with the number of the calling party as defined in 7.1, either as a tel-URI or as a SIP or SIPS URI with telephone-subscriber syntax and "user=phone";. however, they MUST be prepared to receive non-telephone-number URIs in incoming messages. 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.

When calling number privacy is requested, a Proxy-Require containing the option tag "Privacy" MUST be included by default, as described in [12] and [13], unless it is known (by means outside the scope of this document), that all proxies that reside on a trust boundary in the domain support the privacy extensions. It should be noted, that reliance on such knowledge is very brittle and can easily lead to unintended disclosure of private information; *e.g.*, when new proxies are added, software upgraded, configurations changed, etc.

In order to support the asserted identity extension, a Spec(T) is specified, as described in [13]. PacketCable'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 [26] must be followed.

For calls that originate off-net, calling party information present in the PSTN signaling 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 signaling as described in [26].

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 [26].

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 signaling 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 [26] together) define(s) a single Trust Domain where all CMSs 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 functionality described in [27] except as indicated in this section.

CMSS compliant implementations MUST support the message-context-class “voice-message”. Support of all other message-context-classes is optional.

CMSS compliant implementations MAY support group or individual message accounts.

CMSS compliant implementations MUST ignore newly introduced message headers in the Notify message body that are not recognized.

8 CMS-CMS SIGNALING

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

8.1 CMS Interfaces

Signaling between two CMSs is simply referred to as CMS-CMS signaling or CMSS signaling. From a CMS-CMS signaling 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 signaling between the CMS originating a session and the CMS handling termination may be routed through intermediate tandem proxies, but subsequent signaling typically will be sent directly. Both the signaling through tandem proxies and the direct signaling are considered CMS-CMS signaling.

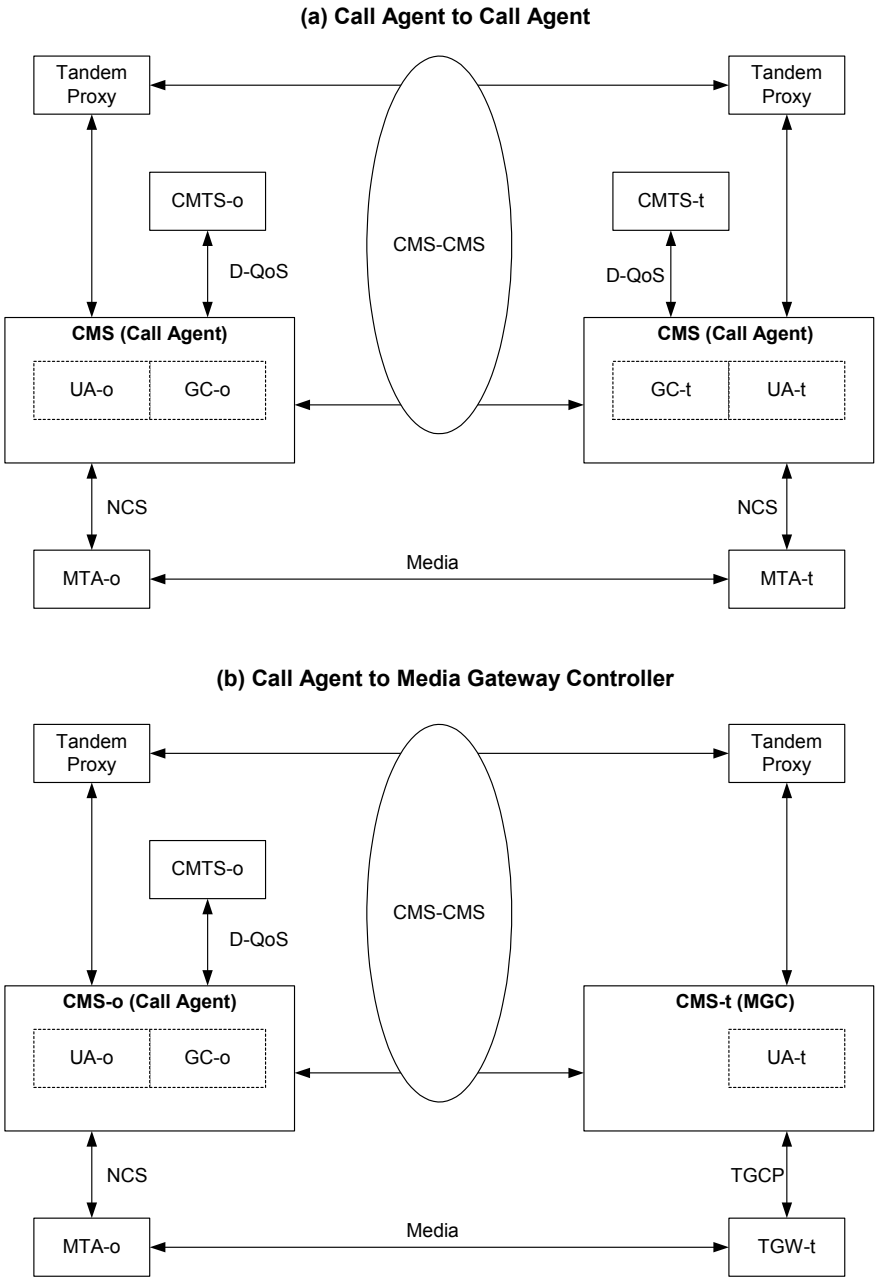


Figure 6. CMS to CMS Signaling

Although all the CMSs in a given domain should be able to communicate, the set of CMSs may not form a fully connected mesh for routing and security reasons. In those cases, some of the CMSs 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 signaling 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:

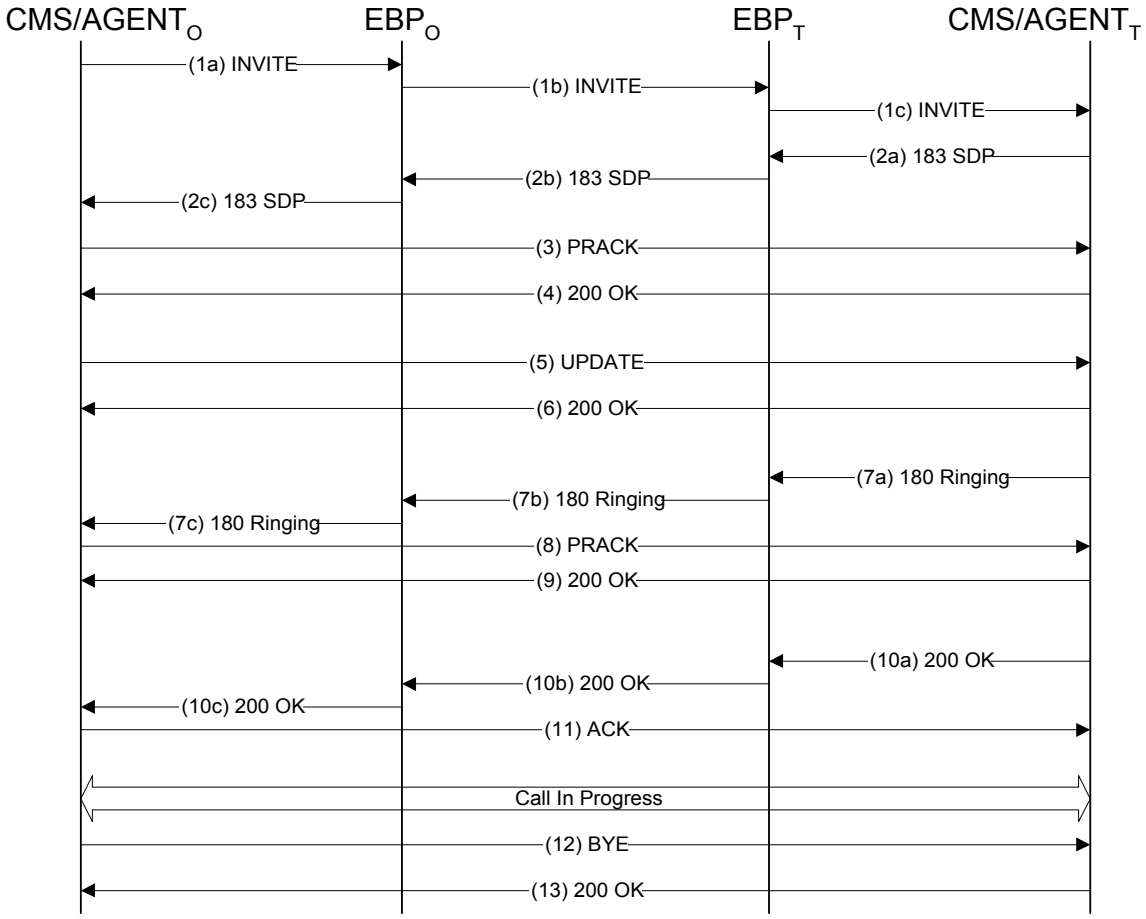


Figure 7. Overview of Interdomain Telephony Call Flow.

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 Signaling (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 signaling; the Gate Controller participates, if needed, in the DQoS signaling (see [21]). Together, they control the coordination of the signaling for call setup and resource management.

DQoS signaling 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 [21] 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 signaling. Or it could be done through an IP-IAM message from a SS7 signaling 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.

In order to support distance sensitive billing, the NPA-NNX of the originating CMS must be known. If the originating telephone number is ported, the NPA-NNX of the originating CMS cannot be inferred from the originating telephone number. Therefore, the NPA-NNX associated with the originating CMS, referred to as Jurisdiction Information Parameter (JIP) in ANSI SS7 ISUP, is included in the JIP-param in the P-DCS-Billing header of the INVITE request. Since JIP is a required ISUP parameter, the originating CMS will always include the JIP-param in the INVITE request.

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[9]. 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).

[9] The purpose of this 183-Session-Progress message is to aid in PSTN interworking as described in [30], Section 8.2.3.

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, CMS_O and CMS_T 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. CMS_O and CMS_T initiate and perform the CMS-CMS signaling exchanges necessary to make these and similar changes.

8.1.1.4 CMS Behavior in Support of Event Messaging

The PacketCable Event Messaging specification [23] 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 [23], and that information is carried in CMS-CMS signaling 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 PacketCable 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).

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 PacketCable 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.

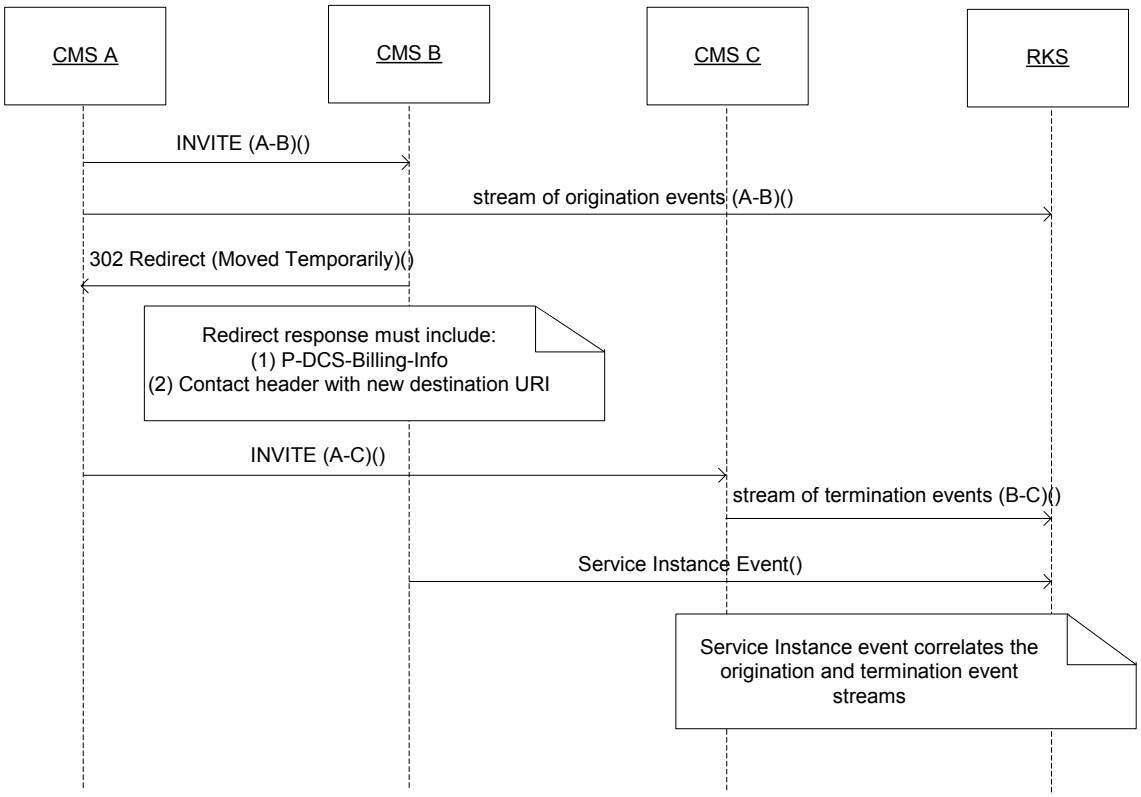


Figure 8. Call Forwarding Support

8.1.1.6 CMS Behavior in Support of Inter-Domain Call Forwarding

When calling and forwarding is performed between PacketCable 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 PacketCable domain to remain on the signaling 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 PacketCable 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 PacketCable 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 signaling 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 signaling 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 [16]) for the Contact header in the 3xx response. All CMSs 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 [16]). The end result is that the signaling 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 PacketCable domain, and served by the same Record Keeping Server. This is the intra-domain scenario (refer to Figure 9).

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 Figure 9 below.

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

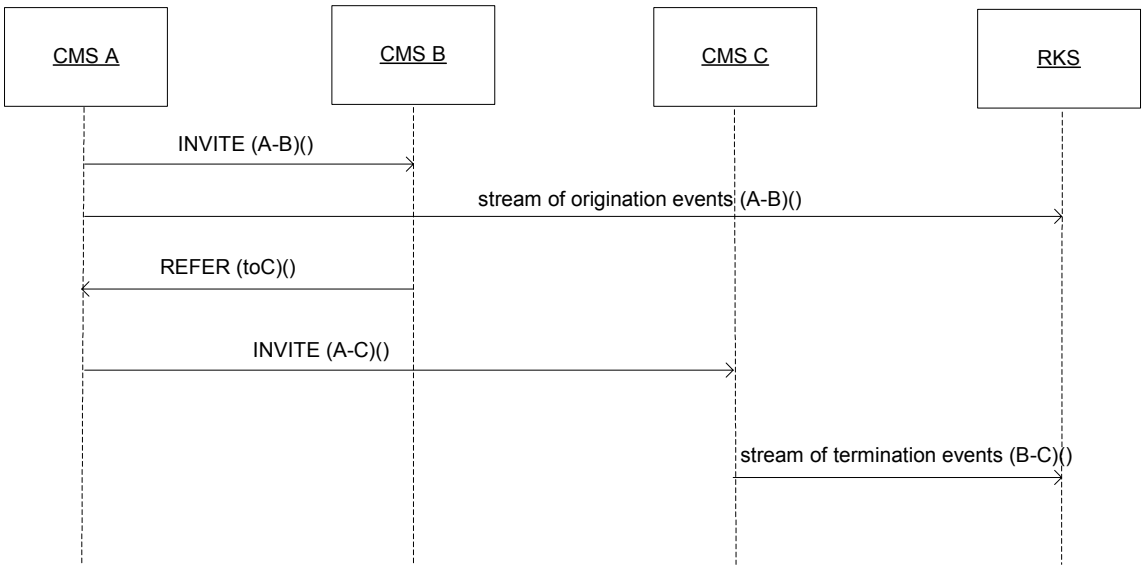


Figure 9. REFER Support

8.1.1.8 CMS Behavior in Support of Inter-Domain REFER

When REFER operations are performed between PacketCable 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 PacketCable domain to remain on the signaling 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 CMSs 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 signaling transformation gateways, signaling firewalls, etc. Depending on its role, a tandem proxy may remain in the call-signaling path for the duration of the call. In the alternative, a tandem proxy may complete its activities and allow the signaling to be passed directly between the CMSs 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 signaling.
- *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 [34] for further information.

The tandem proxy has a limited role in so far as the CMS-CMS signaling 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 [6] 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 A.

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 CMSs are available, the procedures defined in [8], 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(s) URIs as well as telephone numbers in the form of tel-URIs as defined in Section 7.1. Routing of general SIP(s) 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 SHOULD either perform a

local-number-portability (LNP) database query or it SHOULD send the request to another CMS (UA or Proxy) that can perform the LNP query[10]. In the latter case, the Request-URI SHOULD be a tel-URI.[11]

The CMS that performs the LNP query MUST generate a Request-URI containing either a SIP(s) URI or a tel-URI as a result of the query. For SIP(s) URI, the userinfo MUST be in the telephone-subscriber format and the URI MUST contain a "user=phone" parameter. Both tel-URI and SIP(s) 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(s) URI. The SIP (s) URI format is preferred over keeping the tel-URI, and therefore the CMS SHOULD attempt to form a SIP(s) 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(s) 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(s) 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.

If the CMS is able to determine a FQDN, the CMS MUST include that FQDN in a SIP(s) URI in the Request-URI. Generation of the "userinfo" part of the SIP(s) 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(s) 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(s) 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.

8.3.2 Routing a SIP(s) URI at Tandem CMSs

If a CMS receives a request with a SIP(s) 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.

[10] 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.
[11] By using the tel-URL format, intermediate proxies can perform LNP queries and modify the URL accordingly. Using a SIP(s) URI would not allow for this unless the proxy's domain name matches the domain name specified in the SIP(s) URL.

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 in Section 8.3.1.

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 procedures assume that the participating CMSs have been configured to require support of all CMSS extensions. The call flow diagrams are informative only and are intended to provide guidance to developers. Specific processing required for PacketCable 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[12].

[12] 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 [30] for details.

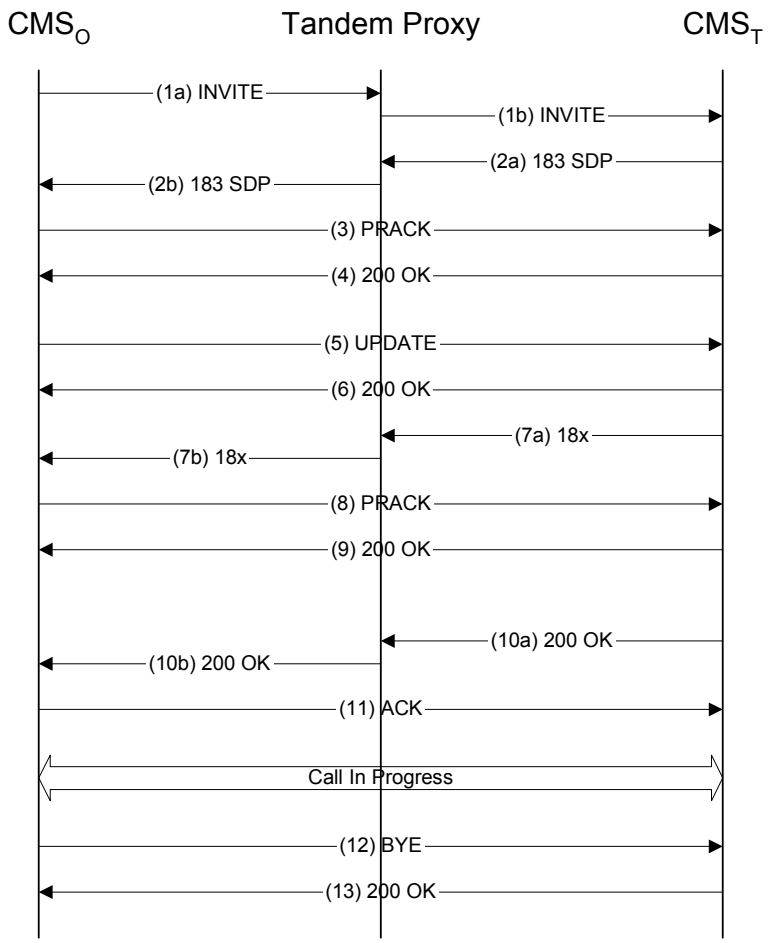


Figure 10. CMS Messages for Basic Call Setup

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, 8.4.1.6 and 8.4.1.8. For procedures followed by CMS_T (*i.e.*, terminating a call) see sections 8.4.1.2, 8.4.1.4, 8.4.1.5 and 8.4.1.7. 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 signaling 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(s) URI as specified in Section 8.3.
- Originating endpoint identification: both the originating phone number (or, in general, a URI of the originator), the originating account name, and the originator's jurisdiction information (JIP NPA-NXX).
- The level of anonymity 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 [23].
- 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 0.
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, calling jurisdiction information, and location routing number as defined in [23]. 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 [26] 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 signaling 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[13].

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.

Please refer to Section 7.7.2.2 for electronic surveillance procedures at the originating CMS.

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 [23].

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

[13] 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.

INVITE (CMS_O -> CMS_T) Header:	Requirements on CMS_O For Message Generation
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.

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 A. 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 A. 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.

Please refer to Section 7.7.2.3 for electronic surveillance procedures at the terminating CMS.

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 signaling 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.)

Please refer to Section 7.7.2.3.1 for electronic surveillance procedures at the terminating CMS when the terminating line is able to accept the call.

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 signaling and media. At a minimum, CMS_T MUST ensure the SDP "c=" line points to the anonymizer prior to crossing a trust boundary[14]. CMS_T MUST also ensure that signaling 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 [23].

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:	

[14] 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
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.
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 signaling path for this call for event message generation [23], by the conditions stated in Section 8.4.1.2.

Please refer to Section 7.7.2.3.4.1 for procedures at the terminating CMS for generating the 3XX Redirect response with a P-DCS-Laes header.

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 calling jurisdiction information (JIP NPA-NXX), 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 [23].

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 signaling path for the call, and generate its stream of event messages for the call.

Please refer to Section 7.7.2.3.4.2 for procedures at the terminating CMS for generating the INVITE message to the forward-to CMS with a P-DCS-Laes header.

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 [23].

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 A. 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.
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.
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.
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[15] 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.

[15] 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.

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.

If a P-DCS-Laes header is present in the 3xx response, CMS_O SHOULD include that header unchanged in the reissued INVITE. CMS_O SHOULD also include a P-DCS-Redirect header containing the original dialed number, the new destination number, and the number of redirections that have occurred. **NOTE**: Please refer to Section 7.7.2. for additional guidance regarding the usage of P-DCS-Laes and P-DCS Redirect headers.

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 Appendix A. 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.
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 in6.12.
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.
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 in6.12.
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 [30]. 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 [30]. 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 in6.12.
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 in6.12.
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 tone16.

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 in6.12.
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.
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.

[16] In NCS [24], this can for example be achieved by use of the "media start" event defined in the Line package.

8.4.1.7.1 CMS_T sending 200-OK Final Response

Once CMST 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:	
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.

Please refer to Section 7.7.2.3.4.1 for procedures at the terminating CMS for generating the 3XX Redirect response with a P-DCS-Laes header.

Two different procedures are defined for handling the call forward case. In the first procedure, CMS_T does not remain on the signaling path for the resulting call. In the second procedure, CMS_T does remain on the signaling 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 calling jurisdiction information (JIP NPA-NXX), 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 [16]) 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 SIP(s) URI or tel-URI. If the new destination is a telephone number, then the format of the URI SHOULD 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 [6]. 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 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 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 SHOULD include that header unchanged in the reissued INVITE. CMS_O SHOULD also include a P-DCS-Redirect header containing the original dialed number, the new destination number, and the number of redirections that have occurred. NOTE: Please refer to Section 7.7.2. for additional guidance regarding the usage of P-DCS-Laes and P-DCS-Redirect headers.

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 Section 8.4.1.2. 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 (or) sip:911;phone-context=+1@dcs-proxy;user=phone

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 0.

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 [21].

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 signaling path for the resulting call. In the second procedure, CMS_I does remain on the signaling 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 CMS_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 calling jurisdiction information (JIP NPA-NXX), 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 [16]), 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.

Please refer to Section 7.7.2.3.4.1 for procedures at the terminating CMS for generating the REFER request with a P-DCS-Laes header.

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.
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.
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 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.
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.
Max-Forwards:	As defined in 6.20.22.
From:	As defined in 6.12.
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 signaling 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.
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.
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.
To:	
Call-ID:	
CSeq:	
Max-Forwards:	As defined in 6.12.
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.
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 signaling 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.
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.
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 [26], 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-Progres, 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 signaling 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.
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.
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 signaling and hence it is not addressed here. However, changing to one or more codecs that were not negotiated in the SDP requires SIP signaling described below.

A signaling 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 CMSs occurs only to synchronize the change. This signaling 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 CMSs must be involved in this procedure in order to increase the resource authorization; therefore, the message exchange follows the proxy-proxy signaling path. This signaling 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.

The format of the UPDATE request sent by the initiating CMS (CMS_I), and the requirements on the header fields checked at the receiving CMS_R are:

UPDATE(codec-change) (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.
Max-Forwards:	As defined in 6.20.22.
From:	As described in 6.12.
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.

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.
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=	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. 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 36.12.
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.
Require:	As defined in 6.20.32.
From:	As described in 6.12.
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.
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 in6.12.
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.
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 signaling 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_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_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_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_I. The latter recognizes that the Initiator has subscribed to conferencing/call transfer and issues an UPDATE(hold) to CMS_B.
3. The Initiator is given dial tone and dials the number of Party C.
4. CMSI initiates a new call to a Bridge Server by sending an initial INVITE. (The call goes to the Bridge Server rather than CMSC because CMSI does not yet know whether the Initiator is invoking ad hoc conferencing or call transfer.) Steps 1-4 are shown in Figure 11.

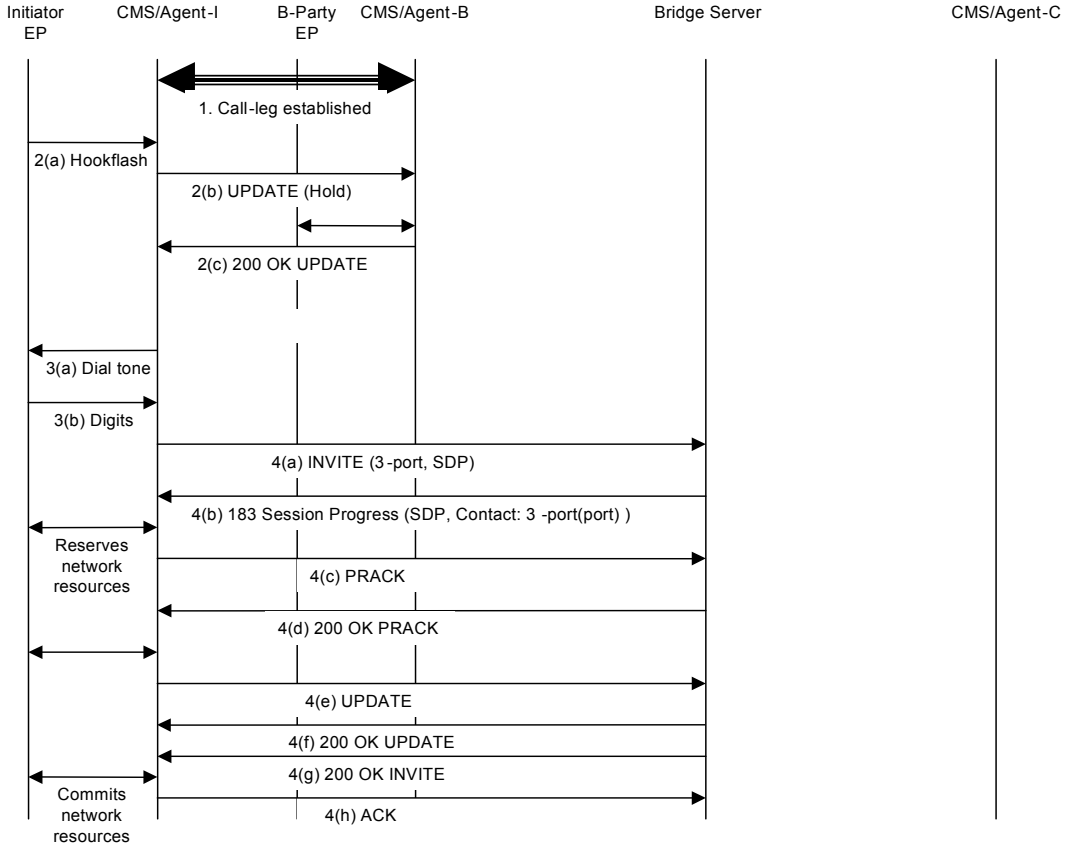


Figure 11. End of transferred call

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, CMSI 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.

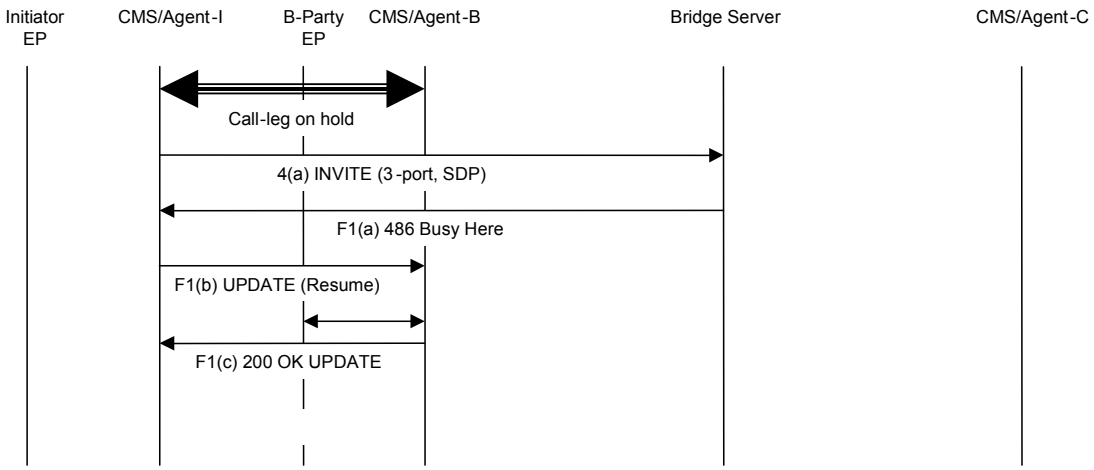


Figure 12. Failure ca– F1 – no free conference circuits

5. Once the Bridge Server has accepted the call, CMSI issues REFER to the Bridge Server, requesting that it establish a call to Party C. The Bridge Server sends a NOTIFY to CMSI 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 CMSI. This step is shown in Figure 13.

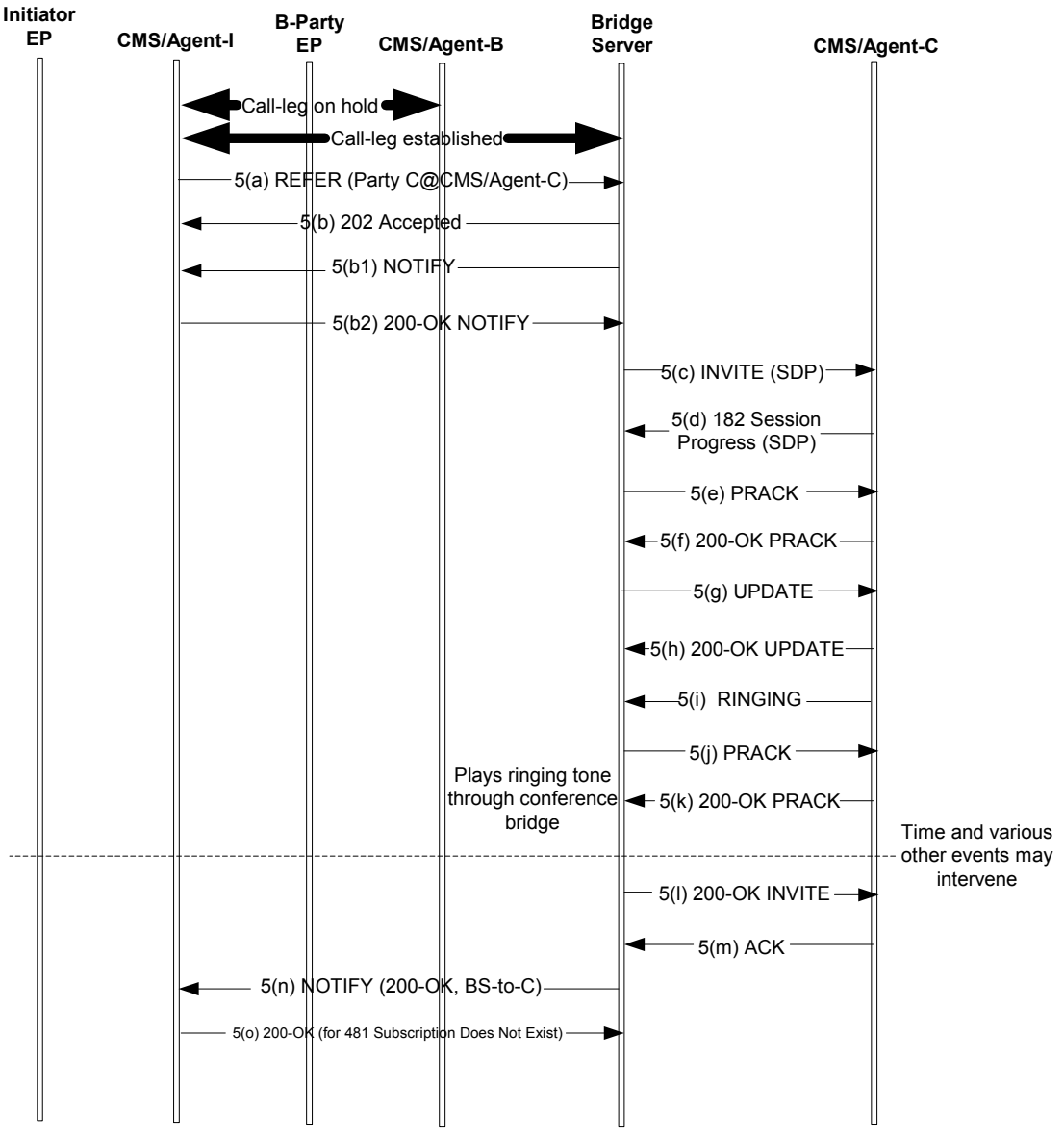


Figure 13. Establishing the leg from Bridge Server to Party C

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.

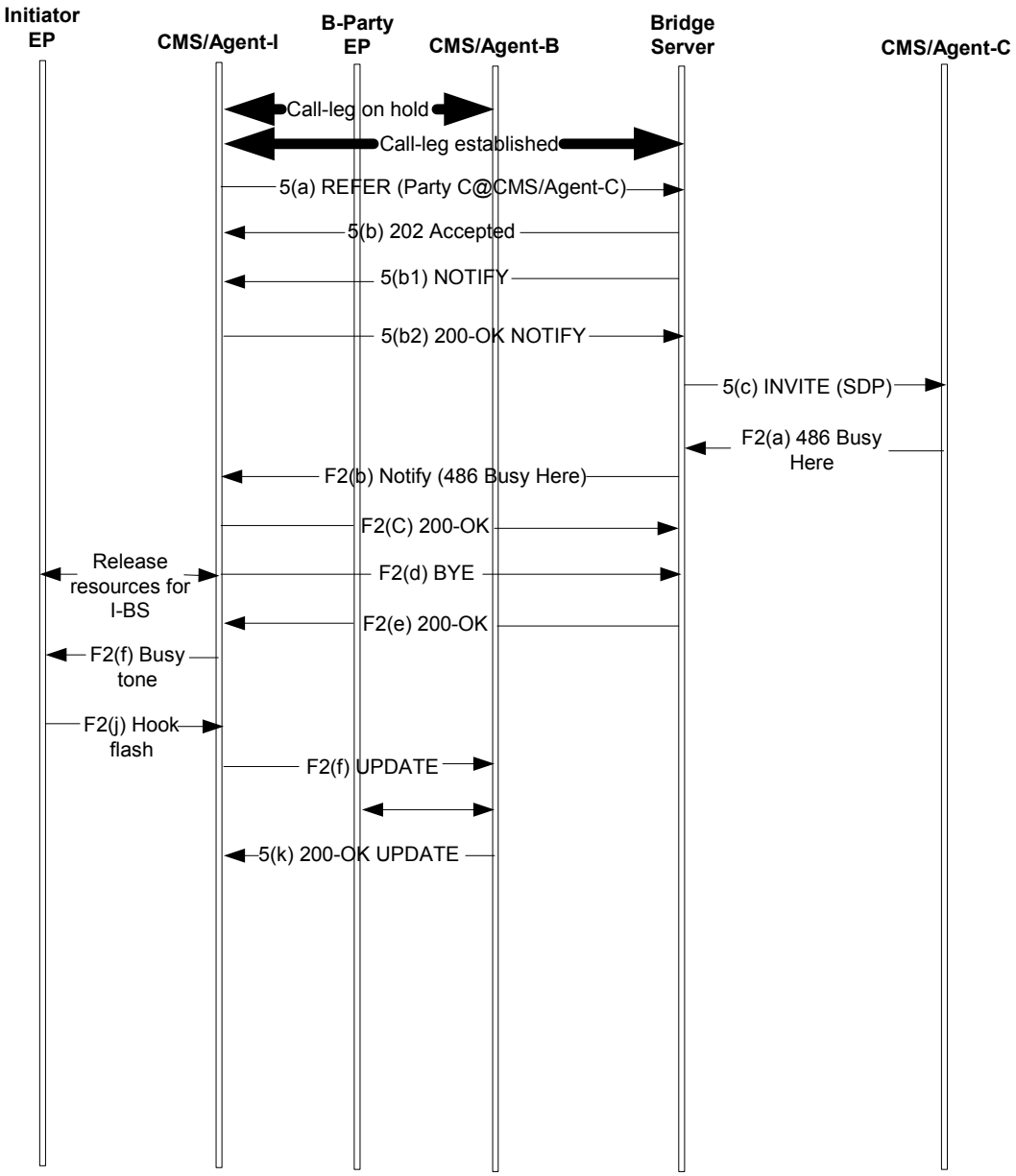


Figure 14. Failure ca– F2 – Party C busy

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 5 are shown in Figure 15.

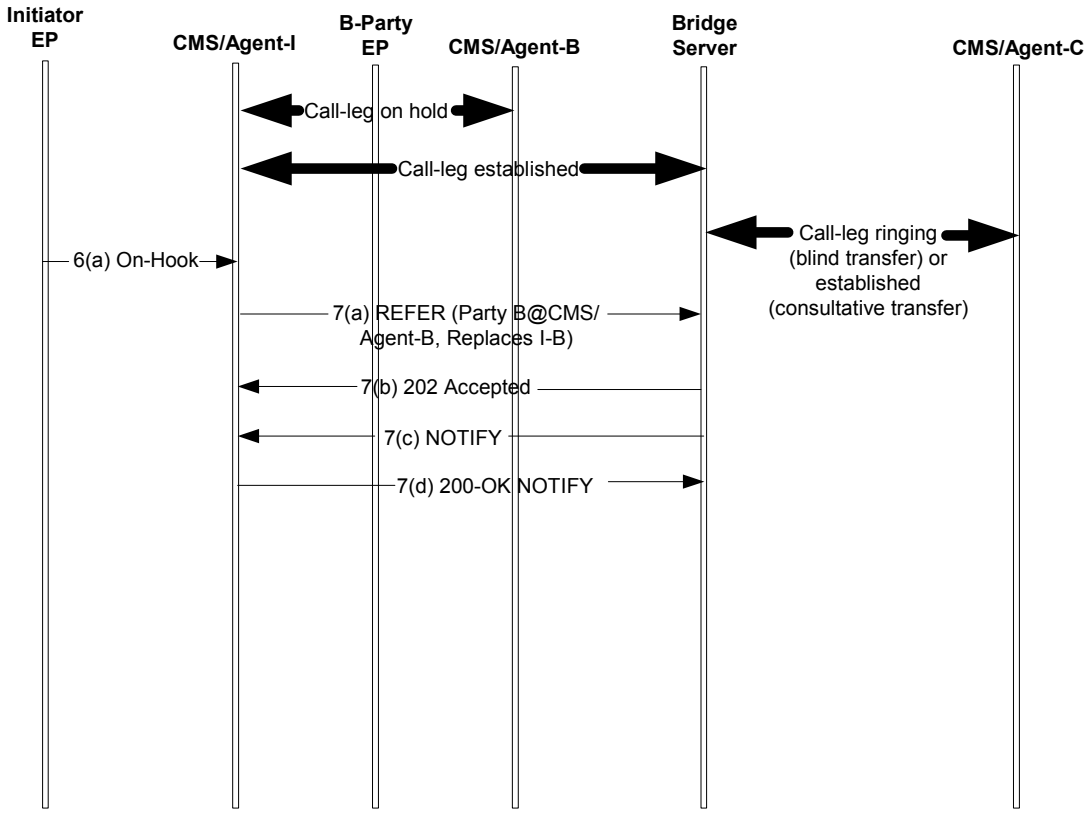


Figure 15. On-hook initiates transfer action

8. As requested, the Bridge Server sends an INVITE to CMSB, containing the Replaces header. CMSB 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 CMSI 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.

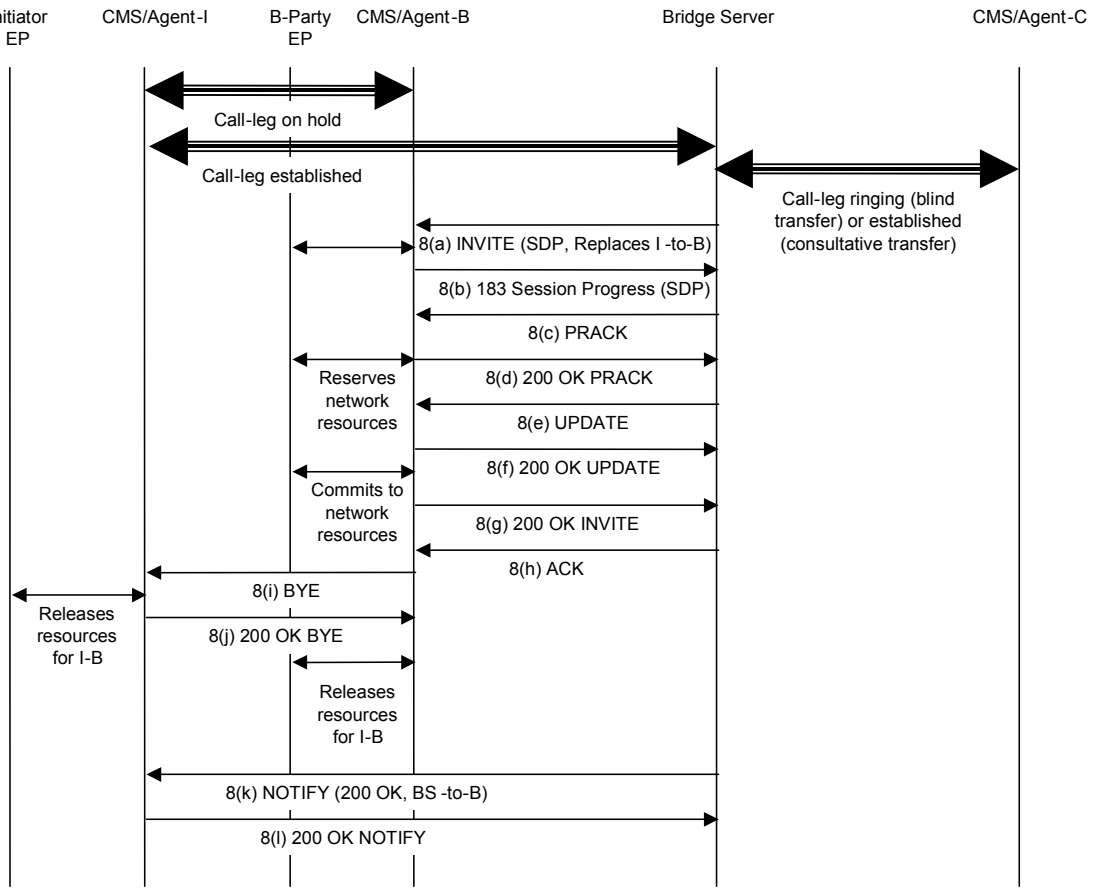


Figure 16. Relocation of Party B to Bridge

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 CMSI 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[17]. This step is shown in Figure 17.

[17] 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.

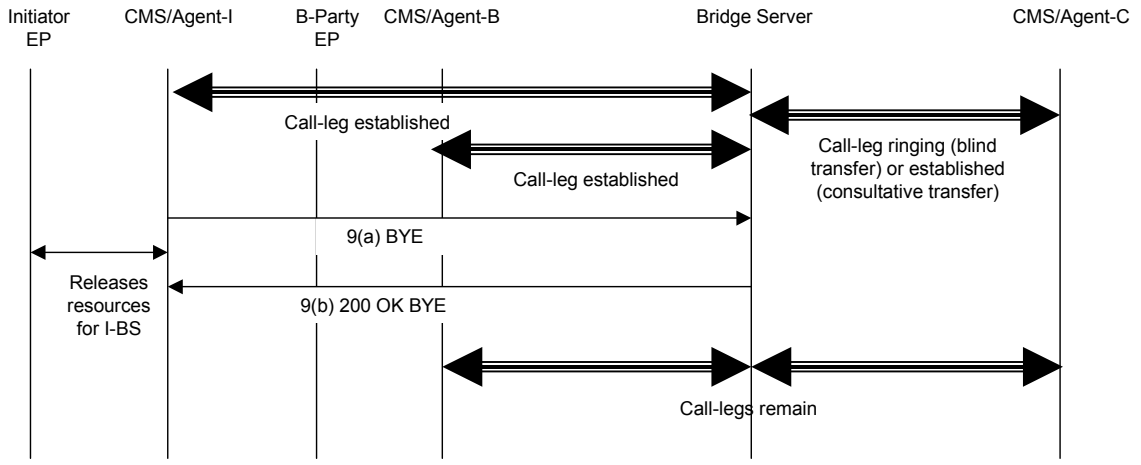


Figure 17. Transfer completed, CMSI ends involvement in call

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.

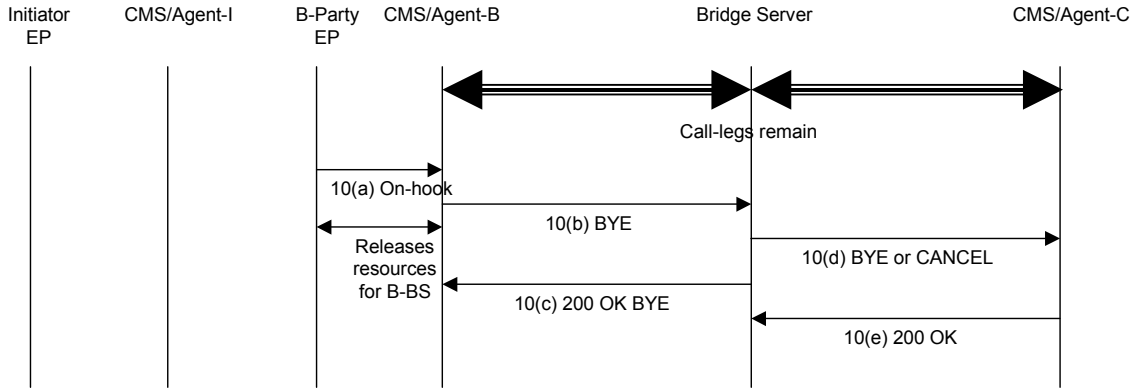


Figure 18. End of transferred call

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 CMSI Sends SUBSCRIBE to CMSR

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 [9] 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 CMSR Sends NOTIFY to CMSI

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 sections 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.
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.
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:

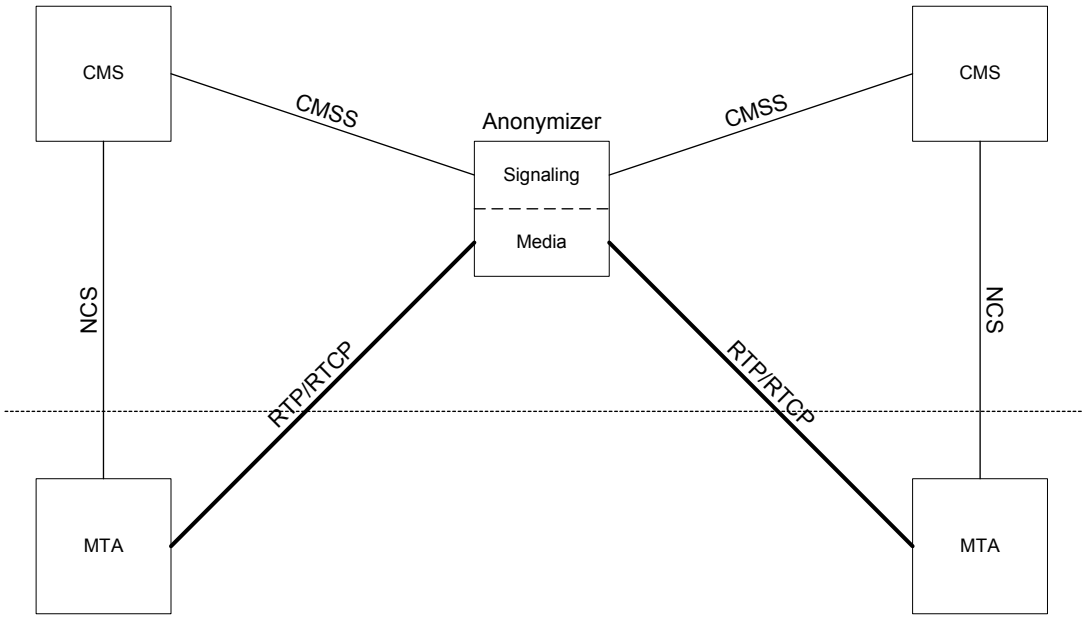


Figure 19. Application Layer Anonymizer

The anonymizer provides three Privacy functions:

Signaling content Privacy
Signaling IP address Privacy
Media IP address Privacy

all of which are illustrated in Figure 20.

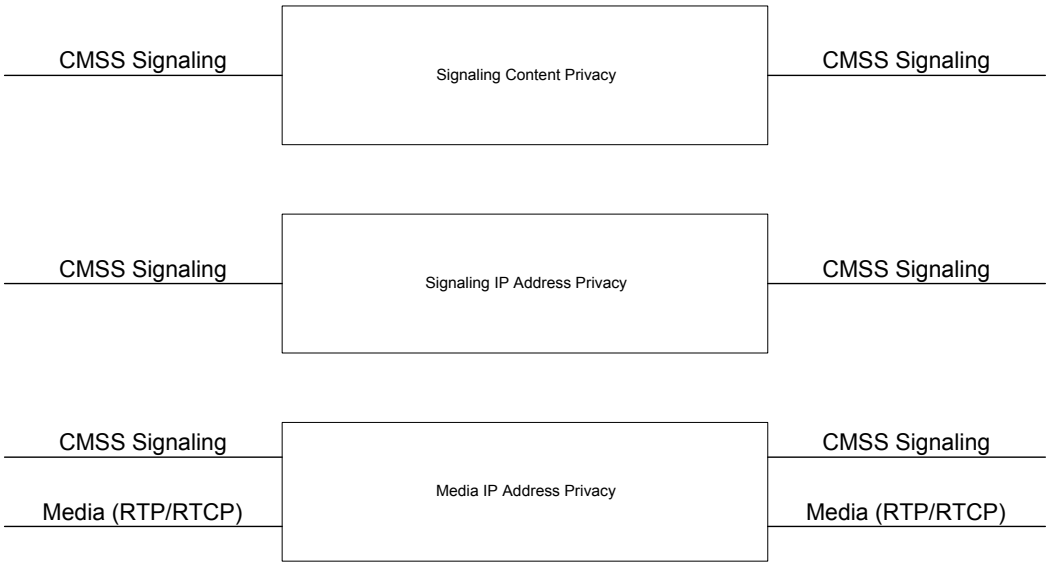


Figure 20. Anonymizer Functions.

9.1 Signaling Content Privacy

The Signaling Content Privacy function serves the role of modifying the SIP signaling messages to handle calling name and calling number Privacy as specified in Section 8.4. More specifically, the Signaling 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 Signaling 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 signaling 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 signaling. This allows the IP address Privacy function to be used in a variety of environments, including ones where the SIP signaling endpoints do not trust each other.

The Signaling IP address Privacy function provides IP address Privacy for the SIP signaling messages themselves. This can be achieved in a number of different ways, and the solution considered here is one where the Signaling IP address Privacy function acts as a back-to-back SIP User Agent.

All signaling IP address information will thus point to, or be based on, the Signaling 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 signaling IP address Privacy is obtained by exchanging signaling with the CMS rather than the MTA. Depending on locality of CMSs, 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 signaling 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 signaling 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 Signaling and Media IP address Privacy. In particular, there is little reason to provide Signaling 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 Signaling IP address Privacy when the SIP signaling 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 signaling 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.

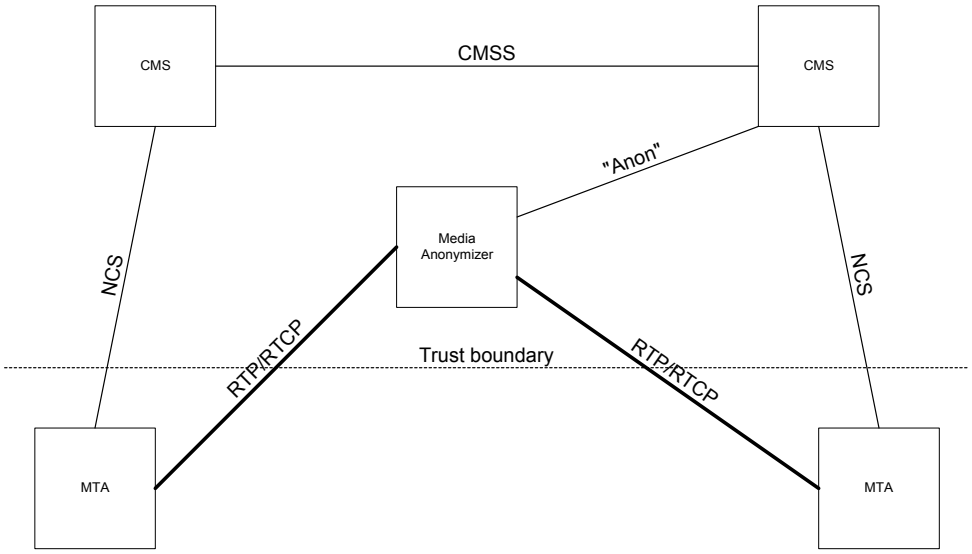


Figure 21. Privacy Issues.

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(s) URI of the Signaling 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 Signaling 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 Signaling 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 A TIMER SUMMARY

CMS-CMS signaling 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 table 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 B 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 [6]. In the case of an unsupported header, the header is simply ignored (as specified in [6], 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 [6], 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

Header	Send	Recv	Reference and Notes
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 7.7.4.
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.
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 C THE SESSION INITIATION PROTOCOL (SIP) "REPLACES" HEADER

(Editor's Note: The cross references used in this appendix apply to those listed in Section C-11, not in Section 2 of this document*.)*

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.

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.

C-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].

C-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.

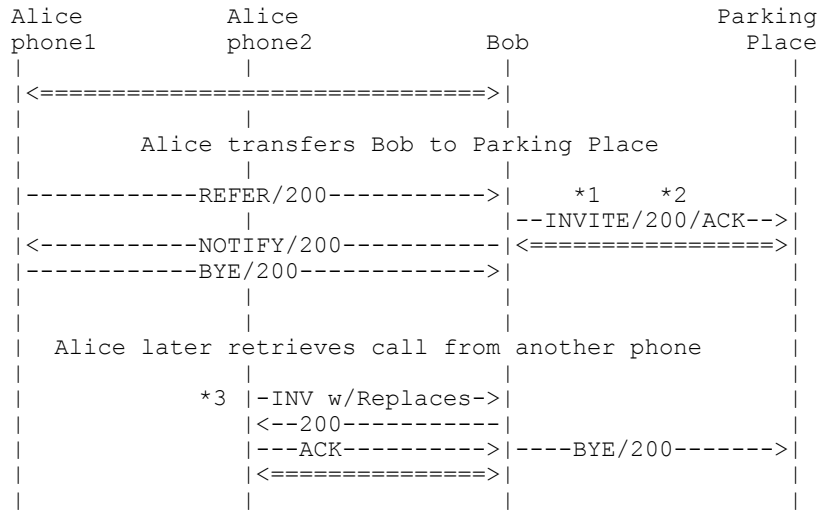
One use of the "Replaces" header is to replace one participant with another in a multimedia conversation. While this functionality is already available using 3rd party call control [8] style call control, the 3pcc model requires a central point of control which may not be desirable in many environments. 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.)



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

C-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).

C-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.

C-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.

C-6. Syntax

C-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

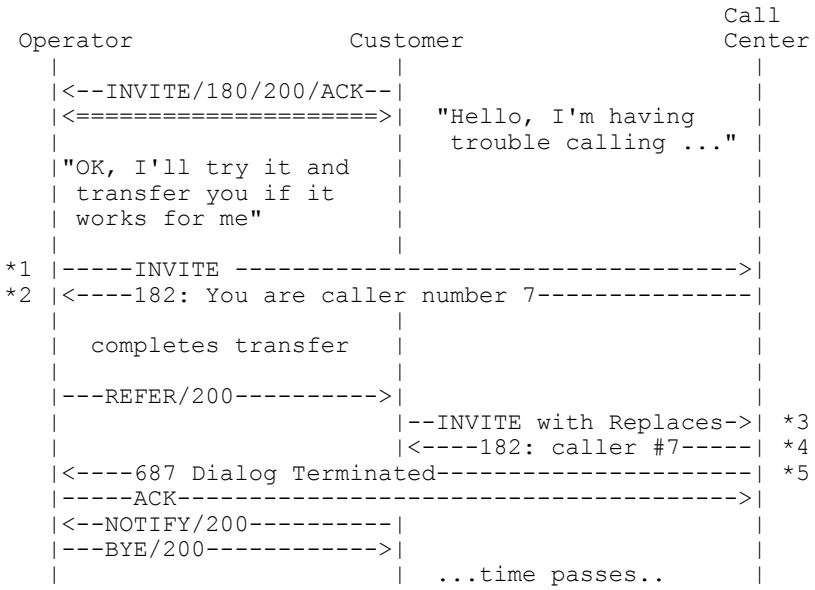
C-6.2 New option tag for Require and Supported headers
This specification defines a new Require/Supported header option tag "replaces". UAs which support the Replaces header MUST include the "replaces" option tag in a Supported header field. UAs that want explicit failure notification if Replaces is not supported MAY include the "replaces" option in a Require header field.

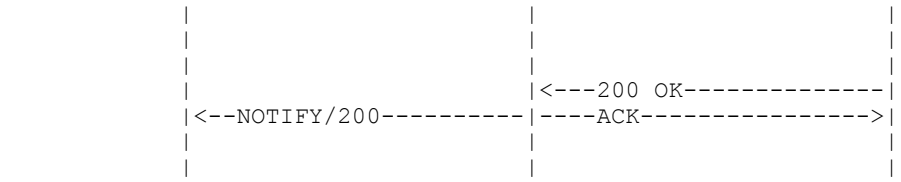
Example:
Require: replaces, 100rel

C-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.

C-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.

C-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.





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:

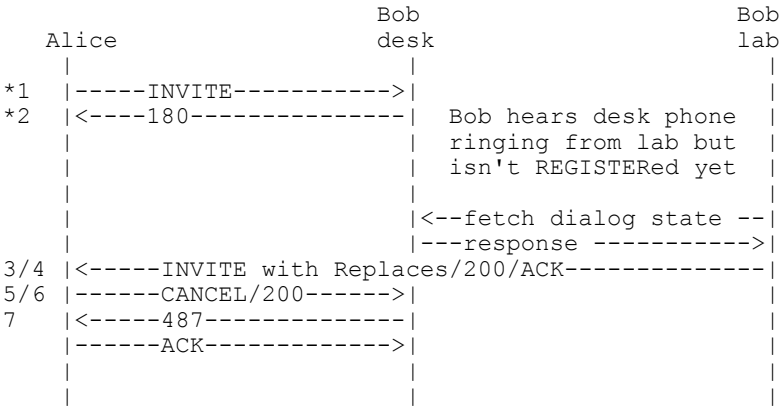
Message *4: Call Center -> Customer
SIP/2.0 182 You are 7th in Queue
To:
From: ;tag=8983
Call-ID: 09870@lobby12.acme.com
CSeq: 1 INVITE
Contact:

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:

C-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.



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:

C-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.

C-9. IANA Considerations

C-9.1 Registration of "Replaces" SIP header

Name of Header: Replaces
Short form: none
Normative description: section F-6.1 of this document

C-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

C-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

C-10. Changes

C-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

11. Updated references

C-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

C-11. Acknowledgments

Thanks to Robert Sparks, Alan Johnston, and Ben Campbell and many other members of the SIP WG for their continued support of the cause of distributed call control in SIP.

Normative References

[1] Rosenberg, J. and H. Schulzrinne, "SIP: Session Initiation Protocol", draft-ietf-sip-rfc2543bis-09 (work in progress), February 2002.
[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[3] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

Informational References

[4] Sparks, R., "The Refer Method", draft-ietf-sip-refer-04 (work in progress), May 2002.
[5] Handley, M., Schulzrinne, H., Schooler, E. and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.
[6] Mahy, R., "A Multi-party Application Framework for SIP", draft-ietf-sipping-cc-framework-00 (work in progress), March 2002.
[7] Sparks, R., "SIP Call Control - Transfer", draft-ietf-sip-cc-transfer-05.txt (work in progress), July 2001.
[8] Rosenberg, J., Schulzrinne, H., Camarillo, G. and J. Peterson, "Best Current Practices for Third Party Call Control in the Session Initiation Protocol", draft-ietf-sipping-3pcc-00 (work in progress), May 2002.
[9] Johnston, A., "SIP Service Examples", draft-ietf-sipping-service-examples-01 (work in progress), April 2002.
[10] Roach, A., "SIP-Specific Event Notification", draft-ietf-sip-events-05 (work in progress), March 2002.
[11] Donovan, S., "The SIP INFO Method", RFC 2976, October 2000.
[12] Rosenberg, J., "The Session Initiation Protocol UPDATE Method", draft-ietf-sip-update-02 (work in progress), May 2002.
[13] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in SIP", draft-ietf-sip-100rel-06 (work in progress), February 2002.

Authors' Addresses

Rohan Mahy
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

EMail: rohan@cisco.com

Billy Biggs
EMail: bbiggs@dumbterm.net

Rick Dean
EMail: rfc@fdd.com

Acknowledgement
Funding for the RFC Editor function is currently provided by the Internet Society.

APPENDIX D NEW PARAMETERS FOR THE "TEL" URI TO SUPPORT NUMBER PORTABILITY

This appendix incorporates the IPTEL Working Group Internet Draft “New Parameters for the "tel" URI to Support Number Portability”, draft-ietf-iptel-tel-np-04.txt, J. Yu, February 17, 2005.

IPTEL Working Group J. Yu
Internet Draft NeuStar
Document: draft-ietf-iptel-tel-np-04.txt February 17, 2005
Category: Standards Track

New Parameters for the "tel" URI to Support Number Portability

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with RFC 3668.

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 August 17, 2005.

Abstract

Number Portability (NP) for the geographical telephone numbers and freephone numbers impacts signaling and routing in the Global Switched Telephone Network (GSTN) and Internet Protocol (IP) domain. At present, there is no mechanism for a network node in the IP domain to pass the NP-related information to the next-hop network node after it has performed an NP database dip. This document defines several new parameters in the "tel" Uniform Resource Identifier (URI) to carry the NP-related information that can be used by the network nodes in the IP domain to correctly set up the calls or sessions.

1. 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 RFC2119 [RFC2119].

2. Introduction

Number portability (NP) [RFC3482] allows the telephony subscribers to keep their telephone numbers when they change service provider (service provider portability), move to a new location (location portability), or change the subscribed services (service portability). The NP implementations in many countries presently support service provider portability for geographic telephone numbers and freephone numbers (e.g., 800 numbers in the North America). NP impacts call signaling and routing. One impact is the need to

carry the NP-related information in the "tel" URI [RFC3966] for protocols such as the Session Initiation Protocol (SIP) [RFC3261] and H.323 [H323] after the NP database dip has been performed. Another impact is for a Voice over IP (VoIP) server to use the NP-related information in a received "tel" URI to determine routing.

A routing number is associated with a geographical telephone number that has been ported out from a donor carrier to another carrier. A donor carrier is the initial carrier where a geographical telephone number was located before ever being ported. A "non-ported" geographical telephone number does not have any routing number associated with it because the first N digits of the geographical telephone number can be used for routing. A routing number can also be used to indicate the switch or network node that originates a call or service similar to the Jurisdiction Information Parameter in Signaling System Number 7 (SS7) Integrated Services Digital Network User Part (ISUP).

The NP database dip indicator is used to inform the downstream servers or switches during call setup that there is no need to perform the NP database dip for a geographical telephone number again.

A "Carrier Identification Code (CIC)" identifies the current freephone service provider for a freephone number. This parameter can also be used to carry the pre-subscribed or dialed long distance carrier information; however, that is outside the scope of this document.

This document defines several new parameters for the "tel" Uniform Resource Identifier (URI) [RFC3966] to support NP. Section 3 lists the abbreviations used in this document. Section 4 provides the formal syntax definition. Section 5 describes the rules for a network node that deals with some or all of the parameters defined in this document for a "tel" URI. Section 6 provides a few examples to show how those parameters defined in this document are added to a "tel" URI after retrieving NP-related information from the NP database. Section 7 discusses the security considerations.

3. Abbreviations

ABNF	Augmented Backus-Naur Form
ANSI	American National Standards Institute
CIC	Carrier Identification Code (also cic)
CIP	Carrier Identification Parameter
FCI	Forward Call Indicator
GAP	Generic Address Parameter
GSTN	Global Switched Telephone Network
IC	Identification Code
IP	Internet Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISUP	Integrated Services Digital Network User Part
JIP	Jurisdiction Information Parameter
NP	Number Portability
NPDB	Number Portability Database
npdi	NP Database Dip Indicator
rn	Routing Number
PNTI	Ported Number Translation Indicator
SIP	Session Initiation Protocol
SS7	Signaling System Number 7
URI	Uniform Resource Identifier
VoIP	Voice over IP

4. Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) as described in RFC-2234 [RFC2234] and defines new parameters in accordance with the ABNF for the "tel" URI provided in RFC 3966 [RFC3966].

```
rn                    = *1(routing-number)
npdi                  = *1(npdb-dip-indicator)
cic                   = *1(carrier-id-code)
routing-number        = "rn=" global-rn / local-rn
global-rn             = "+" 1*global-hex-digits
```

```
local-rn                = 1*phonedigit-hex rn-context
rn-context              = ";rn-context=" rn-descriptor
rn-descriptor           = domainname / global-hex-digits
global-hex-digits       = "+" 1*3(phonedigit) *phonedigit-hex
phonedigit              = DIGIT / [ visual-separator ]
phonedigit-hex          = HEXDIG / "*" / "#" / [ visual-separator ]
visual-separator        = "-" / "." / "(" / ")"
domainname              = *( domainlabel "." ) toplabel ["."]
domainlabel             = alphanum
                           / alphanum *( alphanum / "-" ) alphanum
toplabel                = ALPHA / ALPHA *( alphanum / "-" )
alphanum alphanum        = ALPHA / DIGIT
npdb-dip-indicator      = "npdi"
carrier-id-code         = "cic=" global-cic / local-cic
global-cic              = "+" 1*phonedigit-hex
local-cic               = 1*phonedigit-hex cic-context
cic-context             = ";cic-context=" rn-descriptor
```

The "routing-number", "npdb-dip-indicator" or "carrier-id-code" each can appear in the "tel" URI at most once.

For a "global-rn", the routing number information after "+" MUST begin with a valid E.164 [E164] country code. Hexadecimal digit is allowed after the country code in the "global-rn".

For a "local-rn", the routing number in the "rn" parameter MUST be interpreted according to the "rn-context". For example, if a national routing number is in the "rn" parameter, the "rn-context" MUST contain a valid E.164 country code after "+" if it is in the "global-hex-digits" format. Hexadecimal digit is allowed in the "local-rn".

For a "global-cic", the CIC information after "+" MUST begin with a valid E.164 country code.

For a "local-cic", the CIC value in the "cic" parameter MUST be interpreted according to the "cic-context". For example, if the national CIC value is in the "cic" parameter, the "cic-context" SHALL contain a valid E.164 country code after "+" if it is in the "global-hex-digits" format.

5. Normative Rules

This section discusses how a network node handles a received "tel" URI that contains one or more of the parameters defined in this document or has accessed an NP database for a freephone number or geographical telephone number and needs to add some of the parameters defined in this document to a "tel" URI.

In countries where there is no freephone number portability or geographical telephone number portability, the call routing can be based on the leading digits of the freephone number or geographical telephone number. This document does not describe those scenarios.

Please note that two accesses to the freephone databases are normally done for routing a call to a freephone number. The first one is done by the originating network that queries a freephone database for the CIC information so that the call can be routed to the serving freephone service provider of the called freephone number. When the call reaches the serving freephone provider, the second database access is performed to map the freephone number to a geographical telephone number and/or internal routing information. This document does not address the case where internal routing information is returned.

The first freephone database contains the CIC information for all the active freephone numbers while the second one usually contains mapping information only for those freephone numbers served by a freephone service provider. Because the originating carrier may provide freephone service, its freephone database would contain the CIC information for all the active freephone numbers plus the mapping information for those freephone numbers it serves. This document refers to the two database accesses as "the first freephone database access" and "the second freephone database access".

When handling the "rn" and "cic" parameters and the phone numbers in the "tel" URI for the purposes such as database access and routing, the visual separators in them are removed before using the information in them.

When a network node handles a "tel" URI that contains invalid "rn" or "cic" information, it may release the call or drop the invalid parameter and access the appropriate NP database or freephone database to see if it can retrieve a valid routing number for a geographical telephone number or valid CIC for the freephone number.

5.1 Handling "tel" URI with Defined Parameter or Parameters

If the "tel" URI contains the "npdi" parameter, the network node SHALL NOT retrieve the NP-related information for geographical telephone numbers even if it is set to do so.

If the "tel" URI contains the "cic" parameter whose CIC value is different from the one this network node is associated with, this network node SHALL NOT retrieve the NP-related information for the geographical telephone number or perform the first freephone database access for the freephone number in the "tel" URI.

For the "cic" and "rn" parameters and either a freephone number or geographical telephone number, the order of processing is to look for the "cic" parameter first for call routing. If the CIC information is not useful or the "cic" parameter does not exist, then the next step is to look for the "rn" parameter. If the information in the "rn" parameter is not useful or the "rn" parameter does not exist, then the freephone number or geographical telephone number is used.

If the network node does not know how to route based on the "cic" or "rn" parameter, the local policies SHALL decide whether to stop the call processing or continue the call processing by ignoring the invalid/unknown information.

When looking for the "cic" parameter and that parameter exists in the "tel" URI:

- The network node SHALL ignore the "cic" parameter if the CIC identifies a carrier or service provider associated with that node and look for the "rn" parameter for making the routing decision.

  It SHALL remove the "cic" parameter when it routes the call to the next-hop network node that belongs to another carrier or service provider.

- The network node SHALL invoke special handling process if the "cic" parameter contains a code that requires such a treatment. For example, a CIC value of "0110" in the response to a freephone DB query in the North America indicates "local, translated geographical telephone number provided"). In this particular example, the "cic" parameter is ignored. Please note that this particular CIC value of "+1-0110" normally will not appear in the call setup message. It is given as an example to show that such special CIC values may exist. The exact code values and the handling of them are outside the scope of this document.

- Otherwise, the network node SHALL make the routing decision based on the CIC. The network node SHALL NOT remove the "cic" parameter unless it is handing over the call to the carrier or service provider identified by the CIC and the local policies require it to remove the "cic" parameter. How the call is actually routed based on the CIC value in the "cic" parameter is outside the scope of this document.

When looking for the "rn" parameter and that parameter exists in the "tel" URI:

- If the routing number in the "rn" parameter points to this network node (e.g., the call has reached the intended network node), this network node SHALL look for the freephone number or geographical telephone number for making the routing decision. It SHALL remove the "rn" parameter when setting up the call to the next-hop network node

regardless if that next-hop network node is in the same or different network.

- If the routing number in the "rn" parameter points to a network this network node is in (e.g., in some countries the routing number gets the call to the serving carrier network where another NP database access is required to locate the serving switch), this network node SHALL look for the freephone number or geographical telephone number for making the routing decision. The network node MAY access the NP database for routing information if it is set to do so. It SHALL remove the "rn" parameter if the next-hop network node belongs to another carrier or service provider.

- Otherwise, the network node SHALL make the routing decision based on the routing number in the "rn" parameter. How the call is actually routed based on the routing number in the "rn" parameter is outside the scope of this document.

When the "cic" or "rn" parameter is not used for routing, the network node uses the freephone number or geographical telephone number for making routing decisions. It may access the NP database if it is set to do so, or it may route the call to a designated network node that will access the NP database, or it may route the call based on the local routing table. How the call is handled at this stage is outside the scope of this document. See Section 5.2 for rules in adding the parameter or parameters defined in this document to the "tel" URI if the network node is set to access the NP database.

5.2 Adding Defined Parameter or Parameters to the "tel" URI

There are two cases in terms of NP database access. One is for a geographical telephone number and the other is for a freephone number. They are discussed in Sections 5.2.1 and 5.2.2 for a "tel" URI that is used for routing.

Section 5.2.3 discusses a special case where the "rn" parameter is added to a "tel" URI that is associated with the first network node that handles the call request from the caller. Section 5.3.4 discusses the addition of the parameter or parameters defined in this document to the "tel" URI due to protocol conversion.

5.2.1 Retrieving NP-related information for a geographical telephone number

When a network node accesses an NP database for a geographical telephone number:

- If the network node retrieves a routing number, it SHALL add the "rn" parameter to the "tel" URI to carry the routing number information in the "global-rn" or "local-rn" format. It SHALL also add the "npdi" parameter.

- If the network node does not retrieve a routing number (e.g., for a non-ported geographical telephone number), it SHALL add the "npdi" parameter to the "tel" URI.

The network node SHALL follow the rules described in Section 5.1 for using the information in the "tel" URI to make the routing decision.

5.2.2 Retrieving NP-related information for a freephone number

When a network node performs the first or second freephone database access for a freephone number:

- If the network node retrieves a CIC that identifies a carrier or service provider associated with that network node, or indicates that a geographic number is supplied (e.g., "+1-0110" means "local, translated geographical telephone number provided"), it would have retrieved a geographical telephone number. The network node SHALL NOT add the "cic" parameter and SHALL replace the freephone number in the "tel" URI with

the retrieved geographical telephone number in either the "global-number" or "local-number" format.

Some freephone databases may not return the geographical telephone number but internal routing information in a proprietary format (e.g., switch ID and trunk group ID). That case is outside the scope of this document.

- If the network node retrieves a CIC that belongs to another freephone service provider, the network node SHALL add the "cic" parameter to the "tel" URI that contains the CIC in the "global-cic" or "local-cic" format.

- The originating carrier may have business agreements with a freephone service provider to return the geographical telephone number in addition to the CIC. When a geographical telephone number is returned, the network node SHALL replace the freephone number in the "tel" URI with the returned geographical telephone number in either the "global-number" or "local-number" format.

- If the network node retrieves a geographical telephone number (which is the typical case for the second freephone database access), the network node SHALL replace the freephone number in the "tel" URI with the retrieved geographical telephone number in either the "global-number" or "local-number" format.

  When a geographical telephone number is returned in the response, it is possible that the NP-related information for that geographical telephone number could also be returned. In that case, the network node SHALL add the "npdi" parameter and SHALL add the "rn" parameter to contain the routing number in either the "global-rn" or "local-rn" format only when the routing number is available.

The network node SHALL follow the rules described in Section 5.1 for using the information in the "tel" URI to make the routing decision.

5.2.3 Adding location information about the caller

In SS7 ISUP, the JIP identifies the switch that originates the call and the information in it may be used by the serving carrier to determine the call charge to the caller or by the involved carriers to determine the settlement amount between them.

A network node that is the first to handle the call request from the caller MAY include the "rn" parameter to the "tel" URI associated with the caller, if one exists. For example, if the network node is a Global Switched Telephone Network (GSTN) gateway that receives an ISUP message that contains the JIP, the correct location information in the JIP can be placed in the "rn" parameter of the "tel" URI that is associated with the caller.

Please note that the information in the "rn" parameter may not be authenticated; therefore, the use of the information by the recipient of the "tel" URI for anything related to charging is done at its own risk.

5.2.4 Adding the parameter or parameters defined in this document due to protocol conversion

A GSTN gateway needs to convert between SS7 ISUP and the VoIP protocol such as SIP or H.323. This type of network node SHALL map between the corresponding ISUP parameters and the parameters defined in this document associated with the "tel" URI for routing and MAY map between the corresponding ISUP parameters and the parameters defined in this document that are in the "tel" URI associated with the caller.

Since ISUP support for NP depends on the individual country, the following discussion applies to a situation when a network node is to map between the NP information in the American National Standards Institute (ANSI) ISUP and the NP-related parameters in the "tel" URI.

For a ported geographical telephone number, the network node SHALL convert the routing number in the ISUP Called Party Number parameter to a routing number in either the "global-rn" or "local-rn" format and carry it in the "rn" parameter for a "tel" URI that is used for routing. The network node

SHALL convert the phone number that is marked as the "ported number" in the ISUP Generic Address Parameter (GAP) to a phone number in either the "global-number" or "local-number" format [28] and put it in the global-number-digits or local-number-digits (see [28]) part of the "tel" URI that is used for routing.

For a non-ported geographical telephone number, the network node SHALL convert the phone number in the ISUP Called Party Number parameter to a phone number in either the "global-number" or "local-number" format and put it in the global-number-digits or local-number-digits (see [RFC3966]) part of the "tel" URI that is used for routing. A "rn" SHALL NOT appear in the "tel" URI unless the local policies require the network node to include it. It is outside the scope of this document how to include the "rn" parameter if the local policies require the network node to do so.

The network node SHALL include the "npdi" parameter in the "tel" URI that is used for routing when the Ported Number Translation Indicator (PNTI) bit in the Forward Call Indicator (FCI) parameter is set to "1".

The network node SHALL include the "cic" parameter in either the "global-cic" or "local-cic" format in the "tel" URI that is used for routing when the ISUP Carrier Identification Parameter (CIP) is present.

The network node MAY include the "rn" parameter in the "tel" URI associated with the caller information when the ISUP JIP is present. This may be subject to the network node's local policy and/or the signaling protocol that carries the "tel" URI.

Mapping NP-related parameters in a "tel" URI to the NP-related information in the ISUP message depends on the national ISUP implementation and is outside the scope of this document.

6. Examples

A. A "tel" URI, tel:+1-800-123-4567, contains a freephone number "+1-800-123-4567". Assume that this freephone number is served by a freephone service provide with a CIC "+1-6789". After retrieving the NP-related information, the "tel" URI would be set to

   tel:+1-800-123-4567;cic=+1-6789

B. A "tel" URI, tel:+1-800-123-4567;cic=+1-6789, is handled by a network node in the serving freephone service provider's network. Assume that the freephone number is mapped to a geographical telephone number "+1-202-533-1234". After retrieving the NP-related information, the "tel" URI would be set to

   tel:+1-202-533-1234

C. A "tel" URI, tel:+1-202-533-1234, contains a geographical telephone number "+1-202-533-1234". Assume that this geographical telephone number is ported and is associated with a routing number "1-202-544-0000". After retrieving the NP-related information, the "tel" URI would be set to

   tel:+1-202-533-1234;rn=+1-202-544-0000;npdi

D. A "tel" URI, tel:+1-202-533-6789, contains a geographical telephone number "+1-202-533-6789". Assume that this geographical telephone number is not ported. After accessing the NP database, the "tel" URI would be set to

   tel:+1-202-533-6789;npdi

E. A "tel" URI, tel:+1-202-533-1234;rn=+1-202-000-0000;npdi, contains an invalid routing number (e.g., no routing information on "+1-202-000-0000"), the network node may drop the "rn" parameter and access the NP database again.

F. A "tel" URI, tel:+1-800-123-456, contains a freephone number "+1-800-123-456" that is one digit short. When accessing the freephone database, there

won't be any "cic" information for this freephone number. The call would be released.

G. A "tel" URI, tel:+1-800-123-4567;cic=+1-56789, is handled by a network node in an originating or transit network. The "cic" information is invalid. The network node may drop the "cic" parameter and access the freephone database again. If the same wrong CIC information is received, the network node would release the call because it does not know how to route the call with an invalid CIC. If the valid information is received, the network node will include the "cic" to contain the received CIC and route the call based on the "cic".

7. Security Considerations

In addition to those security implications discussed in the revised "tel" URI [28], there are new security implications associated with the parameters defined in this document.

If the value of the "rn" or "cic" in the "tel" URI is changed illegally when the signaling message carrying the "tel" URI is en route to the destination entity, the signaling message or call may be routed to the wrong network or network node causing the call setup to be rejected.

If the "npdi" is illegally inserted into the "tel" URI when the signaling message carrying the "tel" URI is en route to the destination entity, the call may be routed to the wrong network or network node causing the call setup to be rejected. It is less a problem if the "npdi" is illegally removed. An additional NPDB query may be performed to retrieve the routing number information and have the "npdi" included again.

If the "rn" in the "tel" URI that is associated with the caller is illegally changed or inserted, the call charge based on that "rn" would be incorrect.

It is RECOMMENDED that protocols carrying the "tel" URI ensure message integrity during the message transfer between the two communicating network nodes so as to detect any unauthorized changes to the content of the "tel" URI and other information.

8. Normative References

[RFC2119] S. Bradner, RFC2119, "Key words for use in RFCs to Indicate Requirement Levels", March 1997.
[RFC3966] H. Schulzrinne, RFC3966, "The tel URI for Telephone Numbers", December 2004.
[RFC2234] D. Crocker and P. Overell, RFC2234, "Augmented BNF for Syntax Specifications: ABNF", November 1997.
[E164] ITU-T Recommendation E.164, "The international public telecommunication numbering plan", May 1997.

9. Informative References

[RFC3482] M. Foster, T. McGarry and J. Yu, RFC3482, "Number Portability in the GSTN: An Overview", February 2003.
[RFC3261] J. Rosenberg, et al., RFC3261, "SIP: Session Initiation Protocol", June 2002.
[H323] ITU-T Recommendation H.323, "Packet-Based Multimedia Communications Systems", November 2000.

10. Acknowledgments

The author would like to thank Penn Pfautz, Jon Peterson, Jonathan Rosenberg, Henning Schulzrinne, Antti Vaha-Sipila, Flemming Andreasen and Mike Hammer for their discussions and comments.

11. Author's Address

James Yu NeuStar, Inc. 46000 Center Oak Plaza Sterling, VA 20166 U.S.A.
Phone: +1-571-434-5572 Email: james.yu@neustar.biz

12. Full Copyright Statement

Intellectual Property

Acknowledgements

Funding for the RFC Editor function is currently provided by the Internet Society

APPENDIX E ACKNOWLEDGMENTS

This specification was developed and influenced by numerous individuals representing many different vendors and organizations. PacketCable hereby wishes to thank everybody who participated directly or indirectly in this effort. In particular, PacketCable wants to recognize the following individuals for their significant involvement and contributions to this specification: Burcak Beser, Mike Mannette, Kurt Steinbrenner (3Com); Dave Boardman (Arris), Koan Chong, K.K. Ramakrishnan, Bill Marshall, Doug Nortz, Chuck Kalmanek, Bob Sayko, and Tung-Hai Hsiao (AT&T); Flemming Andreasen, Dave Oran, Bill Guckel, and Michael Ramalho (Cisco); John Pickens (Com21); Anjan Bose (Convergent Networks); Javier Martinez and D.R. Evans (Lucent); Tom Taylor (Nortel); Poornima Lalwaney, Jon Fellows, and John Wheeler (Motorola); Keith Kelly (NetSpeak); Peter Leong and Dayanand Shetty (Syndeo); Edward Miller, Matt Osman, and Glenn Russell (CableLabs).

Much of the text in Section 7.1 came from [1] and [33] and much of the text in 7.10 came from [1], which contained the following copyright notice:

APPENDIX F REVISION HISTORY

The following ECNs have been incorporated into PKT-SP-CMSS1.5-I02-050812.

ECN	ECN Date	Summary
CMSS1.5-N-05.0230-5	3/14/05	Align with Tel-URU RFC-3966
CMSS1.5-N-05.0244-5	3/14/05	Synchronize CMSS with IETF RFC 3842 "A Message Summary and Message Waiting Indication Event Package for the SIP"
CMSS1.5-N-05.0231-4	7/18/05	CMSS Generality
CMSS1.5-N-05.0294-1	8/12/05	Corrects errors in ECNs CMSS1.5-N-05.0231 and CMSS1.5-N-05.0244.