PacketCable™ 1.5 Specifications

CMS to CMS Signaling

PKT-SP-CMSS1.5-I04-070412

ISSUED

Notice

This PacketCable specification is a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. (CableLabs®) for the benefit of the cable industry. Neither CableLabs, nor any other entity participating in the creation of this document, is responsible for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document by any party. This document is furnished on an AS-IS basis and neither CableLabs, nor other participating entity, provides any representation or warranty, express or implied, regarding its accuracy, completeness, or fitness for a particular purpose.

Document Status Sheet

Document Control Number:	PKT-SP-CMSS1.5-I04-070412			
Document Title:	CMS to CMS Signaling			
Revision History:	I01 – Issued Specification January 28, 2005			
	I02 – Issued Specification August 12, 2005			
	I03 – Issued Specification December 28, 2006			
	I04 – Issued Specification April 12, 2007			
Date:	April 12, 2007			
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.

Trademarks

Contents

List of Figures

List of Tables

This page intentionally left blank.

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 [59]; 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 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 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 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 RFC4566, SDP: Session Description Protocol, July 2006.

[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] IETF RFC 3891, The Session Initiation Protocol (SIP) Replaces Header, September 2004.

[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-I03-070412, April 12, 2007, Cable Television Laboratories, Inc.

[22] PacketCable 1.5 Electronic Surveillance Specification, PKT-SP-ESP1.5-I02-070412, April 12, 2007, Cable Television Laboratories, Inc.

[23] PacketCable 1.5 Event Messaging Specification, PKT-SP-EM1.5-I03-070412, April 12, 2007, Cable Television Laboratories, Inc.

[24] PacketCable 1.5 Network-Based Call Signaling Protocol Specification, PKT-SP-NCS1.5-I03-070412, April 12, 2007, Cable Television Laboratories, Inc.

[25] PacketCable 1.5 PSTN Gateway Call Signaling Protocol Specification, PKT-SP-TGCP1.5-I03-070412, April 12, 2007, Cable Television Laboratories, Inc.

[26] PacketCable 1.5 Security Specification, PKT-SP-SEC1.5-I02-070412, April 12, 2007, Cable Television Laboratories, Inc.

[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] IETF RFC 4694, Number Portability Parameters for the "tel" URI, October 2006.

[30] IETF RFC 3761, The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM), 2004.

[31] IETF RFC 3764, Enumservice registration for Session Initiation Protocol (SIP) Addresses-of-Record, 2004.

[32] IETF RFC 4415, IANA Registration for Enumservice Voice, 2006.

[33] IETF RFC 3402, Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm, 2002.

[34] IETF RFC 1034, DOMAIN NAMES - CONCEPTS AND FACILITIES, 1987.

[35] IETF RFC 2671, Extension Mechanisms for DNS (EDNS0), 1999.

[36] IETF Internet-Draft, Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP), draft-ietf-sip-gruu-11, October 23, 2006, work in progress.

[37] IETF RFC 3264, An Offer/Answer Model with Session Description Protocol (SDP), June, 2002.

[38] IETF RFC 3455, Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP), January 2003.

[39] PacketCable IMS Delta Specification, PKT-SP-24.229-I02-061013, Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Stage 3 Specification 3GPP TS 24.229, October 13, 2006, Cable Television Laboratories Inc.

[40] IETF RFC 3959, The Early Session Disposition Type for SIP, G. Camarillo, December, 2004.

[41] IETF RFC 4244, An Extension to the Session Initiation Protocol (SIP) for Request History Information, November, 2005.

[42] IETF RFC 4458, Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR), April 2006.

[43] IETF RFC 4538, Request Authorization through Dialog Identification in the Session Initiation Protocol (SIP), June, 2006.

[44] IETF RFC 3911, The Session Initiation Protocol (SIP) Join Header, October, 2004.

[45] IETF Internet-Draft, Rejecting Anonymous Requests in the Session Initiation Protocol (SIP), draft-ietf-sip-acr-code-03, October 2006, work in progress.

[46] RFC 4235, An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP), November 2005.

[47] Telcordia GR-227, LSSGR: CLASS Feature: Automatic Recall (FSD 01-02-1260), April 2002, FR-64.

[48] Telcordia GR-529 LSSGR: Public Safety (FSDs 15-01-0000, 15-03-0000, 15-07-0000), Issue 1, June 2000.

[49] Telcordia GR-1277, LSSGR: Operator Services: Switching System Generic Requirements Using Integrated Services Digital Network User Part (ISUP), August 2001.

[50] Telcordia GR-2956, LSSGR: CCS SS7 Generic Requirements in Support of E9-1-1 Service, December 2002.

[51] IETF Internet-Draft, DAI Parameter for the "tel" URI, draft-yu-tel-dai-00.txt, October 2006, work in progress.

2.2 Informative References

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

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

[54] PacketCable 1.5 Architecture Framework, PKT-TR-ARCH1.5-V2-070412, April 12, 2007, Cable Television Laboratories, Inc.

[55] PacketCable 1.5 Audio Server Protocol Specification, PKT-SP-ASP1.5-I02-070412, April 12, 2007, Cable Television Laboratories, Inc.

[56] PacketCable 1.5 Audio/Video Codecs Specification, PKT-SP-CODEC1.5-I02-070412, April 12, 2007, Cable Television Laboratories, Inc.

[57] PacketCable Basic Residential Feature Descriptions for PacketCable-Based VoIP Services, PKT-TR-VOIPBRF-R01-000608, June 8, 2000, Cable Television Laboratories, Inc.

[58] PacketCable Extended Residential Feature Descriptions for PacketCable-Based VoIP Services, PKT-TR-VOIPERF-R01-000831, August 31, 2000, Cable Television Laboratories, Inc.

[59] PacketCable 1.5 MTA Device Provisioning Specification, PKT-SP-PROV1.5-I03-070412, April 12, 2007, Cable Television Laboratories, Inc.

[60] PacketCable OSS Overview, PKT-TR-OSS-V02-991201, December 1, 1999, Cable Television Laboratories, Inc.

[61] PacketCable Architecture Framework Technical Report, PKT-TR-ARCH-FRM-V02-061013, October 31, 2006, Cable Television Laboratories, Inc.

[62] Telcordia GR-391, LSSGR: CLASS Feature: Calling Identity Delivery Blocking Features (FSD 01-02-1053), June 2000.

[63] Telcordia GR-246, Specification of Signalling System Number 7, December 2005.

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

[65] IETF RFC 3551, RTP Profile for Auditor and Video Conferences with Minimal Control, July 2003

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; Internet: http://www.cablelabs.com
- ITU-T Recommendations available at http://www.itu.int
- Internet Engineering Task Force (IETF), Internet: http://www.ietf.org/
  Note: 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.
  The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

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.

Cryptanalysis 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

Domain A PacketCable domain is made up of one or more PacketCable zones that are operated and managed by a single administrative entity. A PacketCable domain may also be referred to as an administrative domain.

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

Exterior Border Proxy A 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 Exterior Border Proxies.

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.

Interior Border Proxy A 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.

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.
Realm	A realm is a set of one or more PacketCable Zones managed as a single community of operation for the purposes of security administration and subdomain routing.
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

Zone

A PacketCable zone consists of the set of MTAs in one or more DOCSIS HFC access networks that are managed by a single functional CMS.

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
DND	Do Not Disturb
DOCSIS®	Data Over Cable System Interface Specification
DP	DCS Proxy
EBP	Exterior Border Proxy
EI	Emergency Interrupt
EP	Endpoint
E.164	Telephone number standard of ITU
FQDN	Fully Qualified Domain Name
GC	Gate Controller
IBP	Interior Border Proxy
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
SCF	Selective Call Forwarding
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.

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.

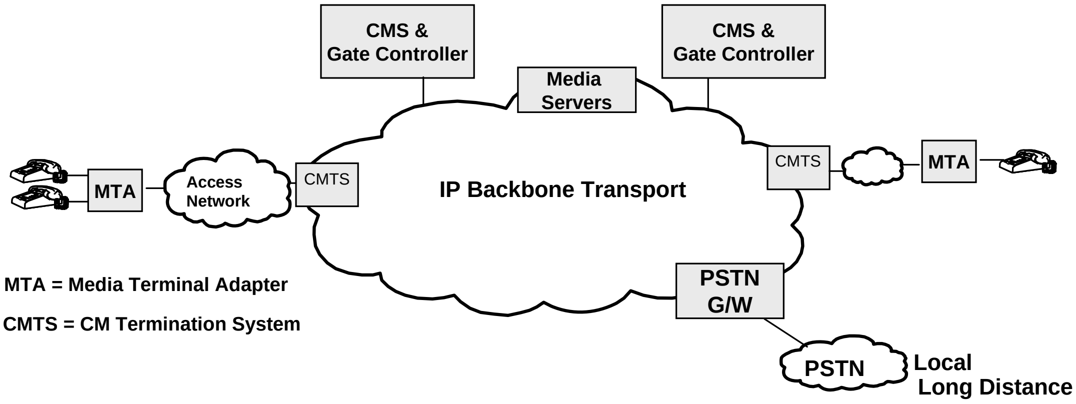


Figure 1. System Architecture – REALM

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 CMTSs. 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.

Additionally CMSS supports inter-working with other PacketCable architectures, in particular PacketCable 2.0. PacketCable 2.0 adds support for SIP-based endpoints, and a SIP-based service platform that may be used to support a variety of services, please see [61] for further details on the PacketCable 2.0 architecture and associated network elements. The CMSS interface support direct inter-working with PacketCable 2.0 Call Session Control Functions (CSCFs) network elements for the majority of session related procedures. There are however a few instances where direct inter-working via CMSS is not possible. As such, specific procedures are documented to cover these special cases. In particular special inter-working procedures are provided for electronic surveillance, accounting and SDP. The procedures for electronic surveillance can be found in Section 7.7.2.5, the procedures for accounting can be found in Section 7.7.3.4, and the procedures for SDP can be found in Section 8.4.10.

There are additional system elements that may be involved in providing the multimedia service [54]. 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, CMTSs 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 CMTSs/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.

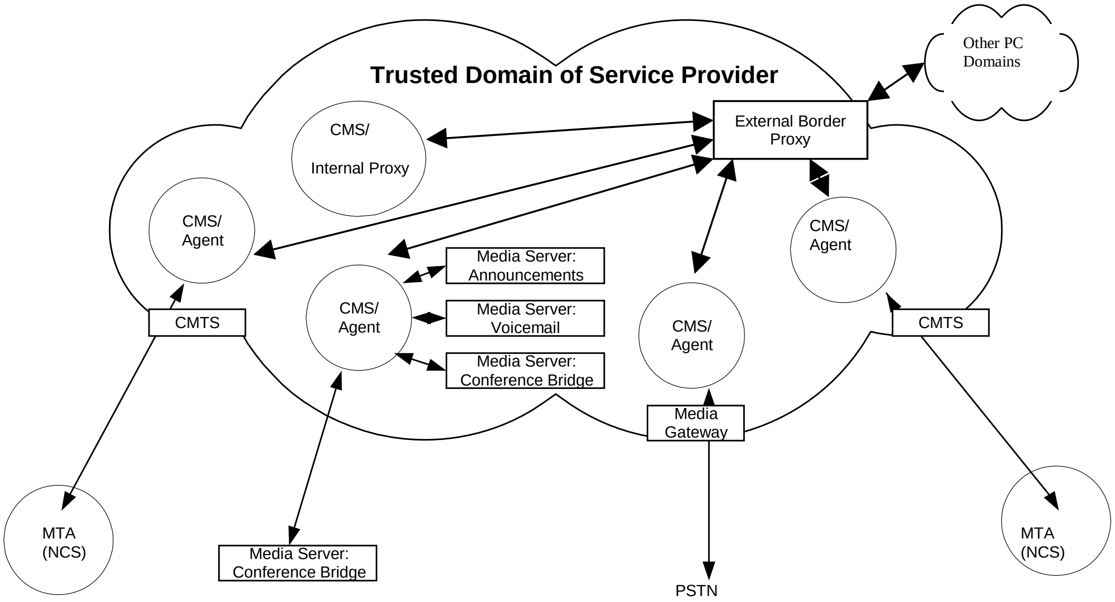


Figure 2. Trusted Domain of PacketCable Service Provider

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. 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.

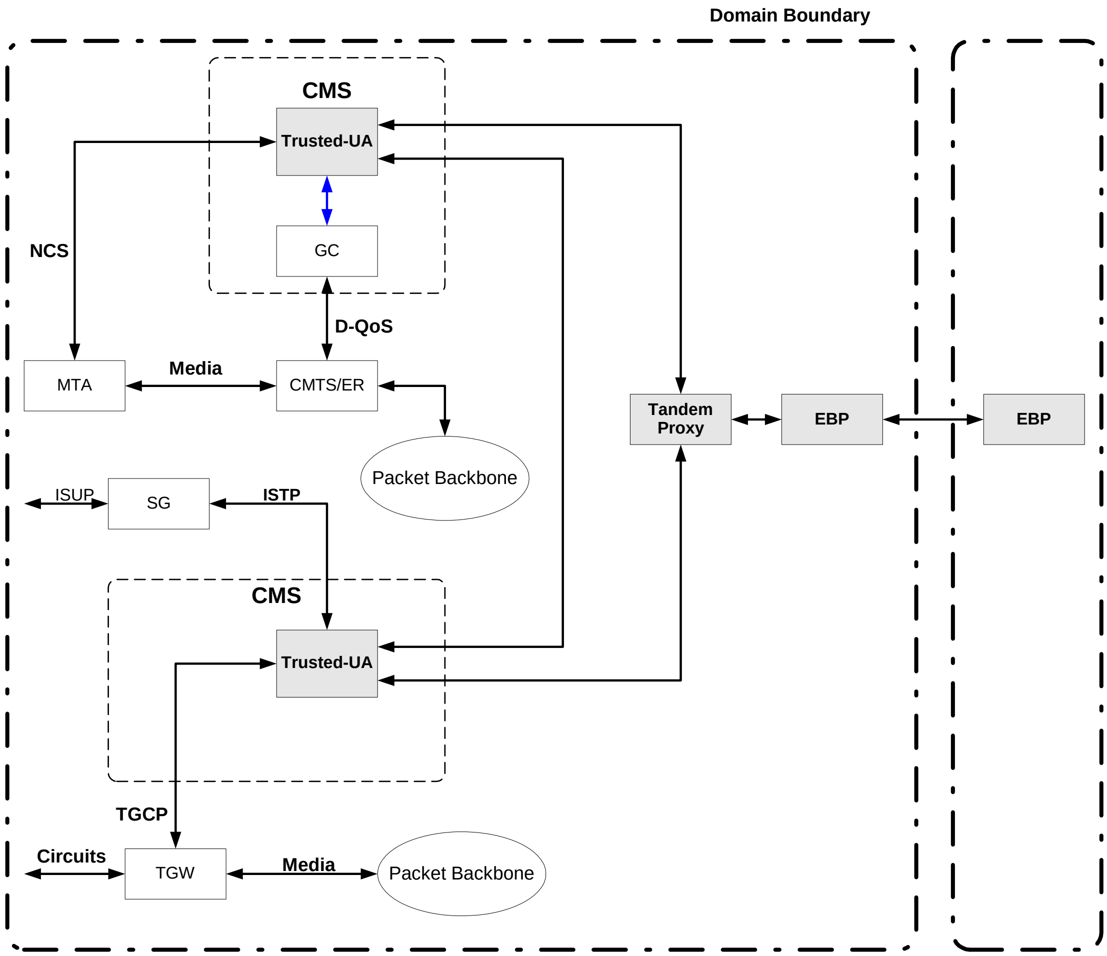


Figure 3. CMS Signaling Model

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 [55]; 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 CMTSs ($CMTS_O$, $CMTS_T$) through DQoS signaling. The CMTSs 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

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

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 contact 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 contact address is in the form of a Globally Routable User-Agent URI (GRUU). 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 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.

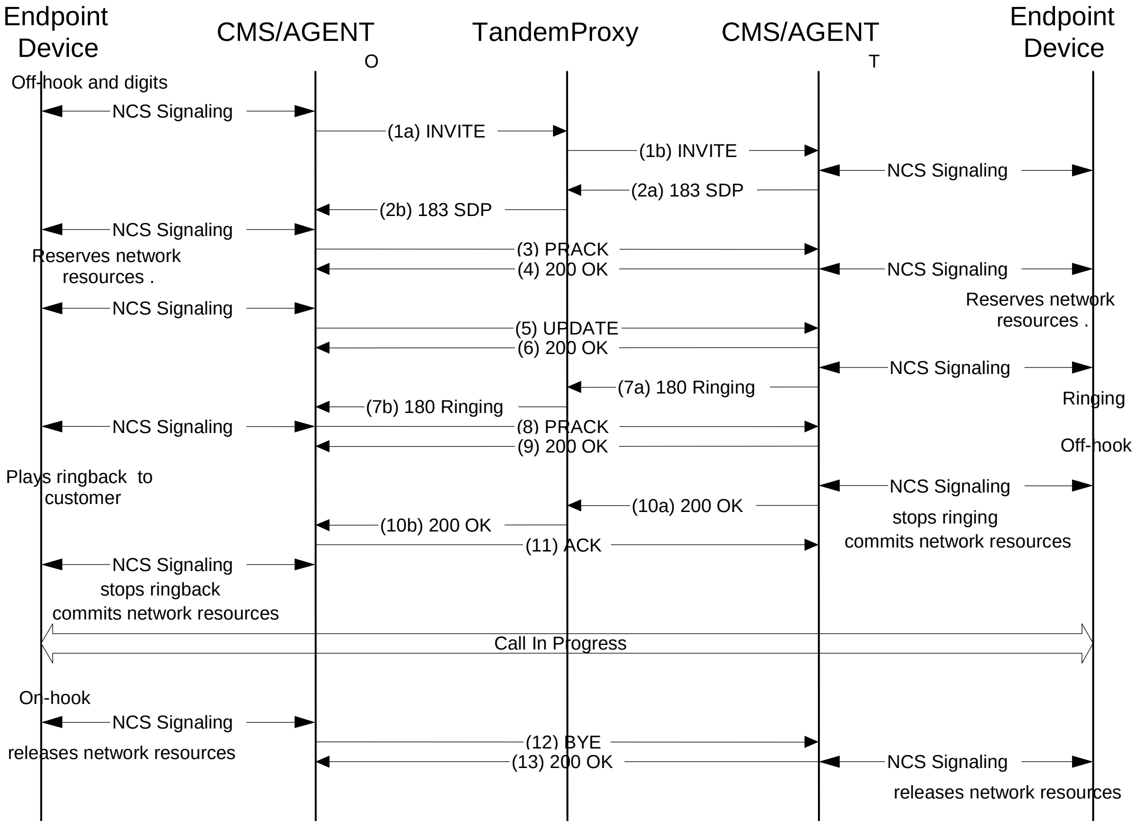


Figure 4. CMS – CMS Signaling Basic Call Flow

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 [52]. Note that CMSS in general is intended to be compatible with the interworking recommendations provided in [52].

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 [52] 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.

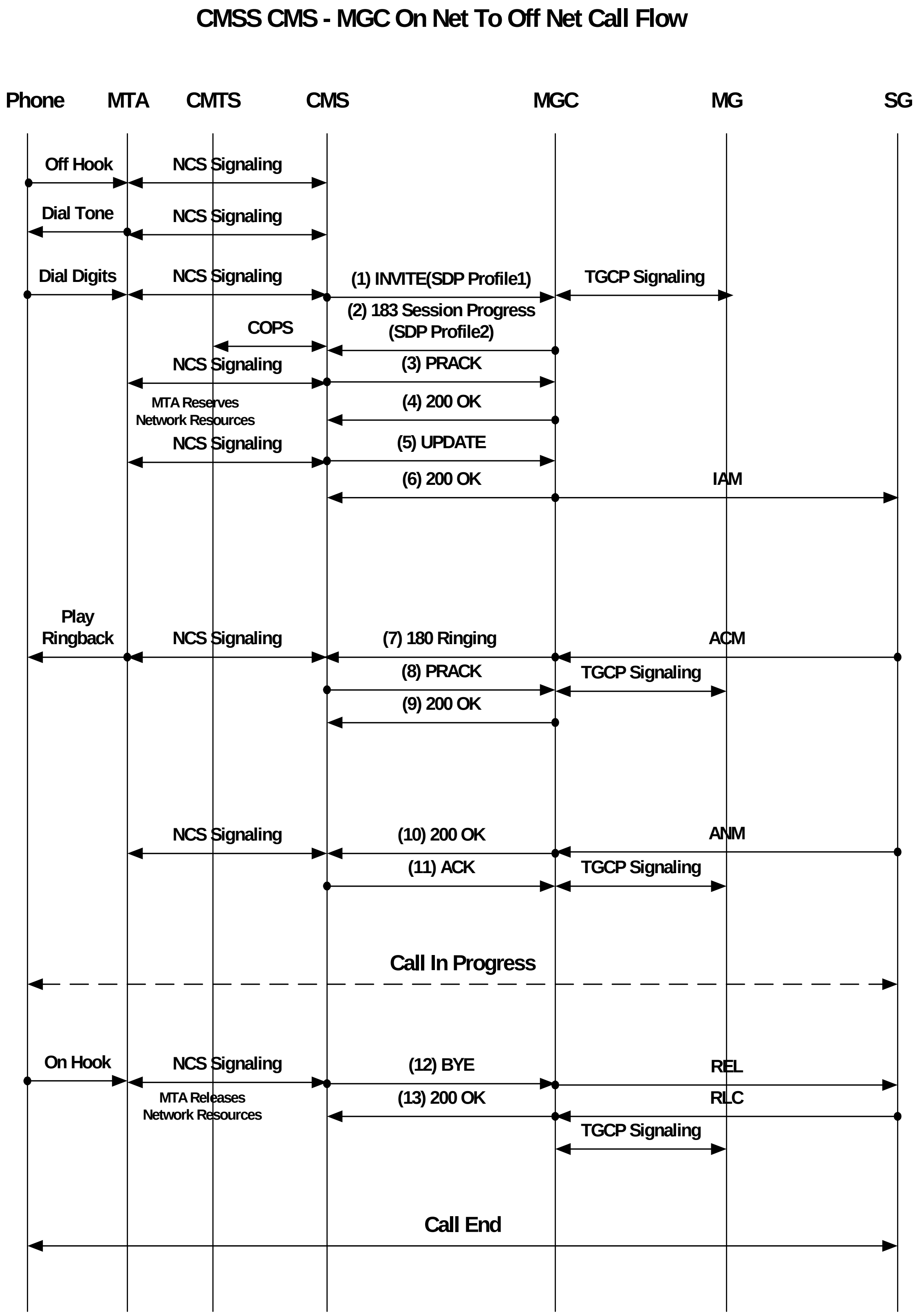


Figure 5. CMS to MGC Signaling

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 URI 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, 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 URI. Refer to Section 8.3 for details on forming the associated Request-URI.

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 "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 Section 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 URI[6]. The From and To fields in the request might contain random strings that protect the Privacy of the session originator.

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

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.

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 URI Component

The definition of a SIP 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

The Alert-Info header MUST be supported for emergency calls (see Section 8.4.2) ; otherwise, it MAY be supported. When included, 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

The Call-Info header MUST be supported as specified in [6] Section 20.9 except as noted below.

The Call-Info header "purpose" parameter value of "answer_if_not_busy" MUST be supported; other encodings MAY be supported.

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 URI. The contact address MUST be in the form of a Globally Routable User-Agent URI (GRUU) [36] as specified in 7.11.

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 exist and reliability is a concern, it is considered a reasonable trade-off to refrain from using the IP address form. When providing a GRUU on behalf of a user that is requesting Privacy the CMSS compliant application MUST provide a GRUU that does not reveal the identity of the user (i.e., a GRUU that has the properties of a temporary GRUU as defined in draft-ietf-sip-gruu [36]).

CMSS compliant applications MUST populate the Contact header field in a 3xx response to an INVITE request with a valid SIP 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[7]; 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".

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

- 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.

6.20.26 Priority

The Priority header MUST be supported for emergency calls (see Section 8.4.2); otherwise, it MAY be supported. When included, 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 FSubject

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.

CMSS compliant implementations MUST be in accordance with reference [45] in their ability to process any 433 Anonymity Disallowed final response that is received.

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 complaint 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.

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 compliant implementations MUST support extensions to the tel URI that relate to number portability and freephone service, as specified in RFC 4694 [29], except as noted in this section.

CMSS compliant implementations MUST support extensions to the tel URI that relate to the subscriber's presubscribed or dialed Carrier ID Code (CIC), as specified in [29] and draft-yu-tel-dai [51], except as noted in this section.

CMSS compliant implementations MUST support extensions to the tel URI to define a dial-around-indicator to indicate how the CIC was derived, as specified in [51] with the exceptions noted in this section.

These extensions are defined as optional in the tel URI sense. RFC3966 [28] specifies that an implementation MAY ignore optional parameters. However, CMSS compliant implementations MUST support the extensions as specified in [29] and [51], and hence they MUST NOT be ignored when received.

Unless stated otherwise in this document, the set of requirements within [51] that apply to a CMSS compliant implementation depend on the role of the CMS as follows:

- a CMS that plays the role of a Call Agent serving NCS endpoints MUST follow the requirements for "Network Node A" specified in [51],
- a CMS that plays the role of a proxy MUST follow the requirements for "Network Node B" specified in [51],
- an MGC MUST follow the requirements for the "GSTN GW" as specified in [51].

CMSS compliant implementations will use the [51] 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 requires routing via an equal access carrier in the PSTN and is either a tel-URI or a Sip URI with user=phone, a carrier-id-code parameter MUST be included in the telephone-subscriber part with a value corresponding to the 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 CMST.

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. In particular, CMSS compliant implementations MUST support segmented QoS preconditions. CMSS compliant implementations MAY support end-to-end QoS preconditions, although the procedures are not specified here.

7.4.1 Procedures at an Originating CMS

CMSS compliant implementations MUST support use of segmented 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", and the status-type MUST be segmented, i.e., there needs to be separate attributes for the local and the remote segment as described in [11]. The strength-tag in the desired-status attributes MUST be "none" and the direction-tag in the local and remote desired-status attributes 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", and the status-type MUST be segmented, i.e., there needs to be separate attributes for the local and the remote segment as described in [11]. The strength-tag in the desired-status attributes MUST be "none",and the direction-tag in the local and remote desired-status attributes 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", and the status-type MUST be segmented, i.e., there needs to be separate attributes for the local and the remote segment as described in [11]. The strength-tag in the desired-status attributes MUST be "mandatory", and the direction-tag in the local and remote desired-status attributes 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 using a segmented status-type, 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 answer SDP, the precondition-type MUST be "qos" and the status-type MUST be segmented, i.e., there needs to be separate attributes for the local and the remote segment as described in [11]. The strength-tag in the desired-status attributes MUST be "mandatory" (i.e., CMS$_T$ must upgrade the strength to "mandatory" in the answer), and the direction-tag in the local and remote desired-status attributes MUST be "sendrecv".

Unless configured otherwise, CMS$_T$ MUST NOT proceed with the session until resources have been reserved in both the local and remote segments. If the desired status of the originating segment (remote from CMS$_T$'s point of view) QoS precondition does not match its current status, then CMS$_T$ MUST request a confirmation of the originating segment QoS reservation from CMS$_O$ by adding an "a=conf:" attribute where the precondition-type MUST be "qos", the status-type MUST be "remote", and the direction-tag MUST be "sendrecv".

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 using a segmented status-type, a terminating CMS (CMS$_T$) MUST generate a 183-Session-Progress response with an SDP. For each media flow with QoS precondition in the answer SDP sent, the precondition-type MUST be "qos", and the status-type MUST be segmented, i.e., there needs to be separate attributes for the local and the remote segment as described in [11]. The strength-tag in the desired-status attributes MUST be "mandatory", and the direction-tag in the local and remote desired-status attributes MUST be "sendrecv".

Unless configured otherwise, CMS$_T$ MUST NOT proceed with the session until resources have been reserved in both the local and remote segments. If the desired status of the originating segment (remote from CMS$_T$'s point of view) QoS precondition does not match its current status, then CMS$_T$ MUST request a confirmation of the originating segment QoS reservation from CMS$_O$ by adding an "a=conf:". attribute where precondition-type MUST be "qos", the status-type MUST be remote, and the direction-tag MUST be "sendrecv".

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 both within an existing dialog and outside of a dialog as a dialog creating request.

7.6.1 Procedures at an Originating CMS

The CMS originating a REFER MUST include additional header parameters in the Refer-To header for a nested P-DCS-Billing-Info header, and SHOULD include the additional header parameters for nested P-DCS-Laes, and P-DCS-Redirect headers, 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".

An originating CMS MAY initiate the REFER request as part of an existing INVITE established dialog or outside of the dialog as a dialog creating request. If the REFER is sent outside of a dialog then the CMS MUST include a Target-Dialog header in the REFER that identifies the associated INVITE established dialog.

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 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. Further, any nested headers (using the '?' syntax defined in RFC 3261 [6]) present in the Refer-To header MUST be present in the INVITE request.

If the REFER is received outside of an existing dialog then CMS_T MUST verify that the REFER contains a Target-Dialog header as defined in 7.14. If the header is absent then CMS_T MUST reject the REFER with a 400-Bad-Request final response. If the Target-Dialog header is present, then CMS_T MUST associate the dialog in the target header with an existing SIP dialog. If the dialog identified by the Target-Dialog does not match an existing SIP dialog at CMS_T then CMS_T MUST reject the REFER with a 481-Transaction-Not-Found final response. If the dialog identified by the Target-Dialog header does match an existing dialog, then CMS_T MUST associate the REFER with that dialog. Furthermore, if the Refer-To header in the REFER does not contain a nested P-DCS-Billing-Info header, then CMS_T MUST use the BCID from the dialog identified in the Target-Dialog header as the BCID for the new dialog that is created as a result of the REFER. Refer to section 7.14 for further details on the Target-Dialog mechanism.

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 CMST 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.

7.7.2.5 Interworking with CSCFs

CSCFs do not generate CCCIDs even when intercepting call content, but populate the CCCID with a null value. Therefore, a CMS that receives a P-DCS-LAES from a CSCF that requires call content intercept will need to generate a CCCID value on behalf of the CSCF. If the CMS_T receives a SIP message containing a P-DCS-LAES header requesting call-data and call-content surveillance from a CSCF that does not include a value for the 'cccid', the CMS_T MUST generate a local 'cccid' value and populate the P-DCS-LAES header with this value, and use this value in event reporting to the DF. This CCCID is generated on behalf of the CSCF for use by the CMS. If the CMS_T receives a SIP message containing a P-DCS-LAES header requesting call-data and call-content surveillance from a CSCF that includes a value for the 'cccid', the CMS_T MUST use the value in event reporting to the DF. A CMS_T that does content tapping and receives a SIP message containing a P-DCS-LAES header SHOULD send a Media-Report message to the DF. The CMS_T follows the balance of procedures as described in section 7.7.2.3.

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, 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.

7.7.3.4 Interaction with CSCF Signaling Nodes

Accounting information is shared between CMS and Call Session Control Functions (CSCFs) using the P-Charging-Vector and P-Charging-Function-Address headers as defined in [38] and [39]. When routing session requests to

CSCFs, it will be necessary for the CMS or MGC to be aware of the fact that it is communicating with a CSCF so that these headers can be processed correctly as specified in the following sections.

For signaling to CSCFs across trust boundaries, an Inter-Operator Identifier (IOI) in the P-Charging-Vector is used to identify the originating and terminating operators. This IOI is equivalent to the Financial Entity ID (FEID) in the PacketCable 1.5 domain, so a mapping is done from FEID to IOI on outgoing signaling, and from IOI to FEID on incoming signaling.

7.7.3.4.1 Sessions initiated from a CSCF to a CMS or MGC

Incoming sessions from a CSCF will contain charging information in a P-Charging-Vector and possibly a P-Charging-Function-Address header.

If the incoming request is from within the same trust domain as the CMS or MGC, the P-Charging-Vector header will be guaranteed to contain the icid-value parameter, and this value will be used to correlate the PacketCable 1.5 event messages with the charging information from the IMS domain. In this case, the ICID will be distinct from the BCID, and the terminating BCID MUST be independently generated by the terminating CMS or MGC as is normally done for inter-CMS calls.

If the incoming request is from outside the trust domain of the CMS or MGC, the P-Charging-Vector will be guaranteed to contain the orig-ioi parameter, and this value will be used to do inter-operator settlements. Since the orig-ioi identifies the other operator's domain, this value will be used as the originating FEID in the PacketCable 1.5 domain.

When a CMS or MGC receives an INVITE from a CSCF with an icid-value parameter in the P-Charging-Vector header, the icid-value MUST be included in Event Messaging as described in [23].

When a CMS or MGC receives an INVITE from a CSCF with an orig-ioi parameter in the P-Charging-Vector header, the orig-ioi MUST be used as the originating FEID for the session, and included in Event Messaging as described in [23].

When a CMS or MGC responds to an INVITE from a CSCF that contained the orig-ioi parameter, it MUST include the term-ioi parameter in the P-Charging-Vector header containing the terminating FEID.

When the CMS or MGC receives an INVITE from a CSCF with a P-Charging-Function-Address header, the CCF values included in the header MAY be used to identify the RKS entities to be used for the session.

7.7.3.4.2 Sessions initiated from a CMS or MGC to a CSCF

Outgoing sessions to a CSCF will contain charging information in a P-Charging-Vector and possibly a P-Charging-Function-Address header.

If the outgoing request is for a CSCF within the same trust domain as the CMS or MGC, the P-Charging-Vector header needs to be populated with the icid-value parameter. Since the originating BCID in the PacketCable 1.5 domain fulfills the uniqueness requirements for ICIDs, the CMS or MGC will use the BCID as the icid-value in the session request.

If the outgoing request is to a CSCF outside the trust domain of the CMS or MGC, the P-Charging-Vector needs to be populated with the orig-ioi parameter, and this value will be used to do inter-operator settlements. Since the FEID identifies operator domain of the CMS or MGC, this value will be used as the orig-ioi.

When a CMS or MGC generates an INVITE to a CSCF that is within the same trust domain as the CMS or MGC, the CMS or MGC MUST include a P-Charging-Vector header in the INVITE containing an icid-value parameter with the originating BCID. The BCID MUST be encoded as a hexadecimal character string of up to 48 bytes as defined in [16]. In this case, the CMS or MGC MAY also include the icid-gen-addr and/or P-Charging-Function-Addresses header.

When a CMS or MGC generates an INVITE to a CSCF that is not in the same trust domain as the CMS or MGC, the CMS or MGC MUST include a P-Charging-Vector header within the INVITE containing an orig-ioi parameter with the value of the originating FEID. In this case, the CMS or MGC MUST NOT include the icid-value, icid-gen-addr, or P-Charging-Function-Addresses header in the INVITE.

When a CMS or MGC receives a response from a CSCF with a term-ioi parameter in the P-Charging-Vector header, the value of the term-ioi MUST be used as the terminating FEID for the session and included in Event Messages to the RKS as described in [23].

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-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".

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-Identity header with the number of the calling party as defined in 7.1, either as a tel-URI or as a SIP URI with telephone-subscriber syntax and "user=phone"; 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.

7.11 Globally Routable User Agent URI (GRUU)

A CMSS compliant implementation MUST provide a GRUU referring to itself in the contact address of dialog-initiating SIP messages, as specified in draft-ietf-sip-gruu [36]; such a GRUU conforms to the definition of a self-made GRUU. A CMSS compliant implementation MUST handle requests received outside of the dialog in which the contact was provided. For example, upon receipt of an INVITE request addressed to a GRUU assigned to a dialog it has active, and containing a Replaces header referencing that dialog, the element will be able to establish the new call replacing the old one.

A CMSS compliant implementation MUST be able to recognize those GRUUs it has assigned and verify their validity. When providing a GRUU on behalf of a user which will be used in subsequent out-of-dialog requests to that user, the CMSS compliant implementation MUST be able to derive the identity of the user from the received GRUU.

This specification does not define how GRUUs are created; they may be provisioned by the operator or obtained by any other mechanism. Note that in some cases a GRUU will need to remain valid beyond the lifetime of the dialog in which it was advertised.

7.12 The Early Session Disposition Type for SIP

CMSS compliant implementations MUST support the early session disposition type as specified in RFC 3959 [40].

The detection and prevention of fraud issues associated with the use of the early session disposition type is considered outside the scope of this document.

7.13 An Extension to the Session Initiation Protocol (SIP) for Request History Information

CMSS compliant implementations MUST support the History-Info header as described in [41].

7.14 Request Authorization through Dialog Identification in the Session Initiation Protocol (SIP)

In order to support call transfer implementations which make use of a REFER sent out of dialog CMSS compliant implementations MUST support the functionality defined in RFC 4538 [43] which specifies the Target-Dialog header used in the out of dialog REFER.

The CMS generating or receiving a REFER outside of an existing dialog MUST associate the dialog in the Target-Dialog header with an existing dialog.

7.15 The Session Initiation Protocol (SIP) Join Header

CMSS compliant implementations MUST support the Join header as described in RFC 3911 [44].

7.16 SUBSCRIBE to Dialog Event Package

In order to allow a CMS to determine the line status of an endpoint, CMSS compliant implementations MUST support the functionality described in RFC 4235 [46] except as indicated in this section.

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

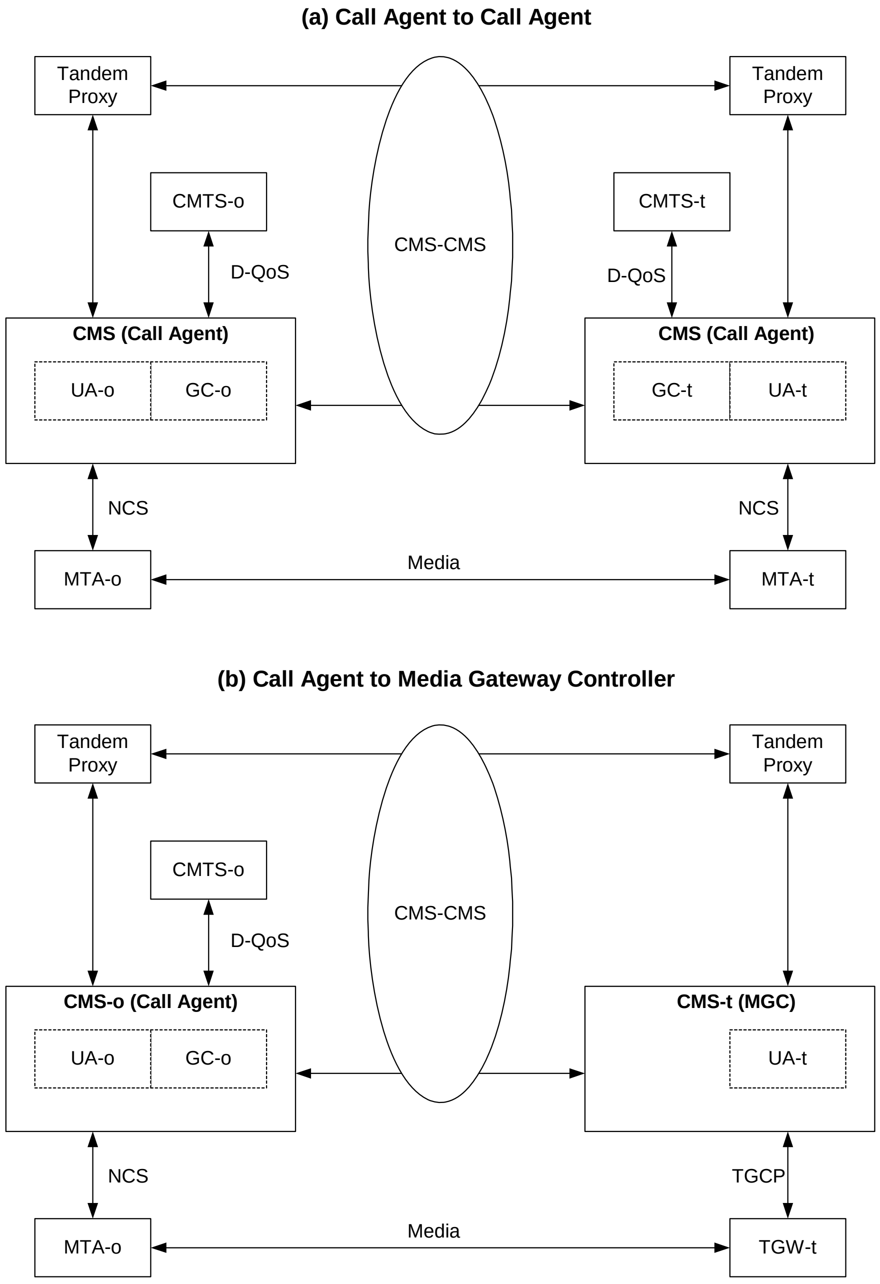
8 CMS-CMS SIGNALING

In this section, 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.


(a) Call Agent to Call Agent

(b) Call Agent to Media Gateway Controller

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 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:

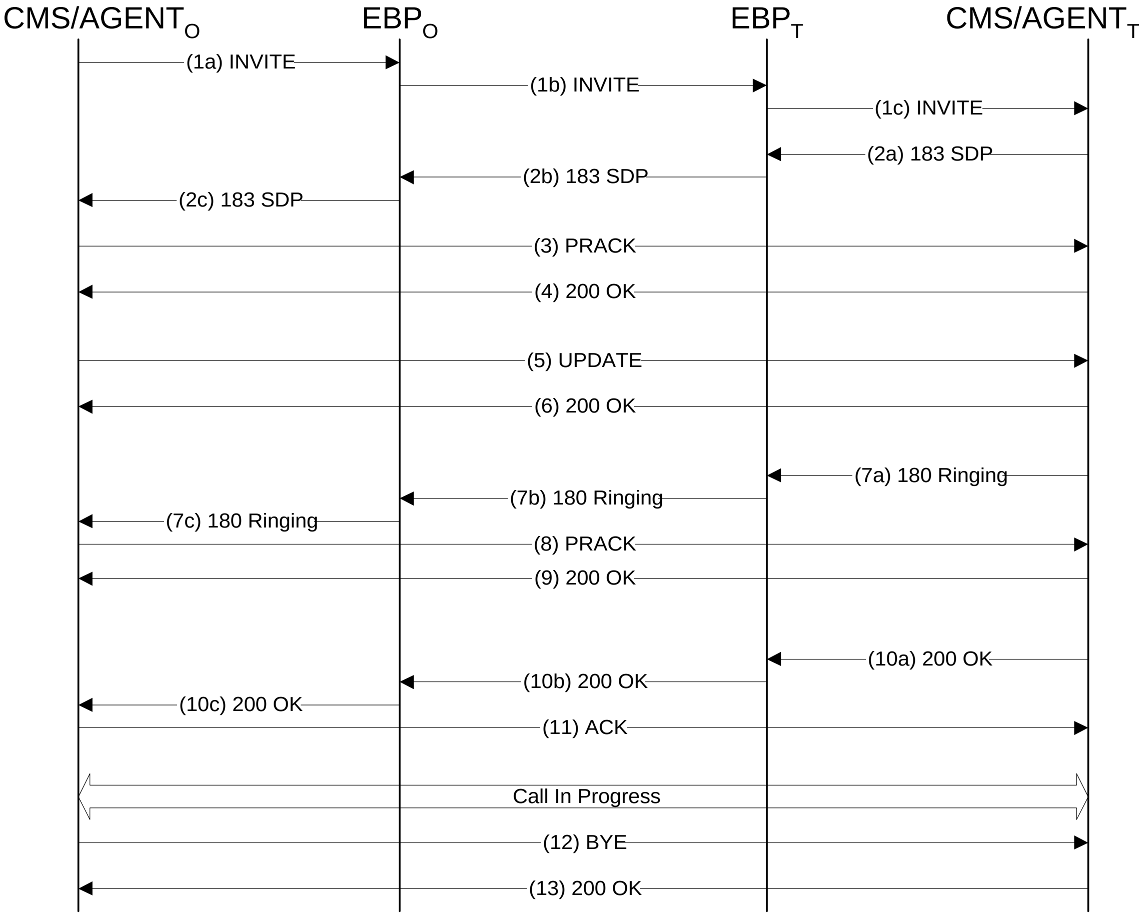


Figure 7. Overview of Interdomain Telephony Call Flow

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[8]. If CMS_T wants to use local ringback, it sends back a 180-Ringing message. When the terminating endpoint answers the call, CMS_T sends a 200-OK message, cuts through the call and enables the bi-directional media flow(s).

Call features such as call-forwarding-unconditional, call-forwarding-busy, call-waiting, and call-forward-no-answer are controlled and implemented by CMS_T by generating the proper SIP responses as part of the basic call setup

[8] The purpose of this 183-Session-Progress message is to aid in PSTN interworking as described in [52], Section 8.2.3.

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.

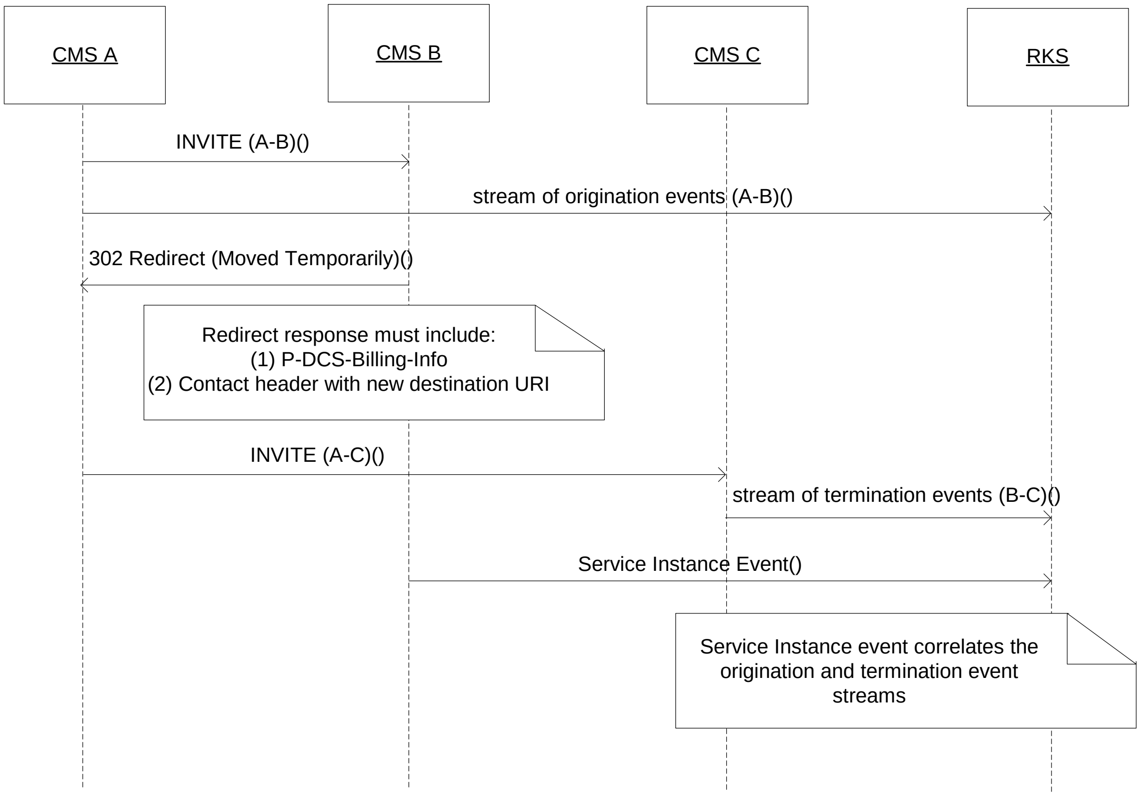

CMS A
CMS B
CMS C
RKS
INVITE (A-B)()
stream of origination events (A-B)()
302 Redirect (Moved Temporarily)()
Redirect response must include:
(1) P-DCS-Billing-Info
(2) Contact header with new destination URI
INVITE (A-C)()
stream of termination events (B-C)()
Service Instance Event()
Service Instance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

Figure 8. Call Forwarding Support

8.1.1.6 CMS Behavior in Support of Inter-Domain Call Forwarding

When calling and forwarding is performed between 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.

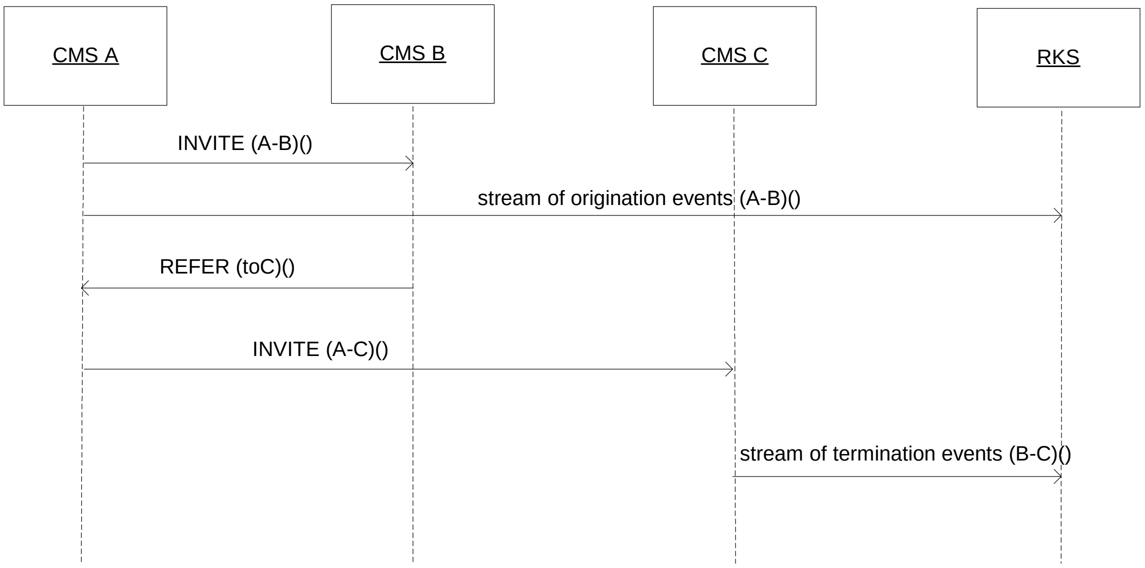

CMS A
CMS B
CMS C
RKS
INVITE (A-B)()
stream of origination events (A-B)()
REFER (toC)()
INVITE (A-C)()
stream of termination events (B-C)()

Figure 9. REFER Support

8.1.1.8 CMS Behavior in Support of Inter-Domain REFER

When REFER operations are performed between 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 [54] 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[9]. In the latter case, the Request-URI SHOULD be a tel-URI. [10]

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.

[9] 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.

[10] 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.

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.

CMS MAY implement the ENUM Client requirements as defined in Appendix C for resolving tel URIs to SIP URIs. The rules for determining when to attempt translation of a Tel URI to a SIP URI are a matter of local policy (e.g., a Tel URI with a carrier ID may be preferred over a SIP URI). When using the procedures defined in Appendix C it is possible that multiple URIs may be returned in response to a single ENUM query. If such a case is encountered, the following requirements apply:

1. The CMS MUST filter the set of returned URIs based on URI type (e.g., SIP:, tel:, etc.), removing those from the list which have service types not supported by the CMS;
2. Absent local policy, the CMS MUST choose the URI with the lowest preference value in the list. The remaining URIs MUST be stored by the CMS;
3. The CMS MUST replace the request-URI with the URI contained in the chosen record and send the request directly to the FQDN identified in the chosen URI following the procedure specified in 6.8.1;
4. If the CMS is unable to determine the next hop for the chosen URI, or there is no response (including an error response) from the next hop, the CMS MUST choose the next URI in the list with the lowest preference value and repeat steps 1-4 until either the list is exhausted or there is a successful response from the next hop.

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.

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.[11]

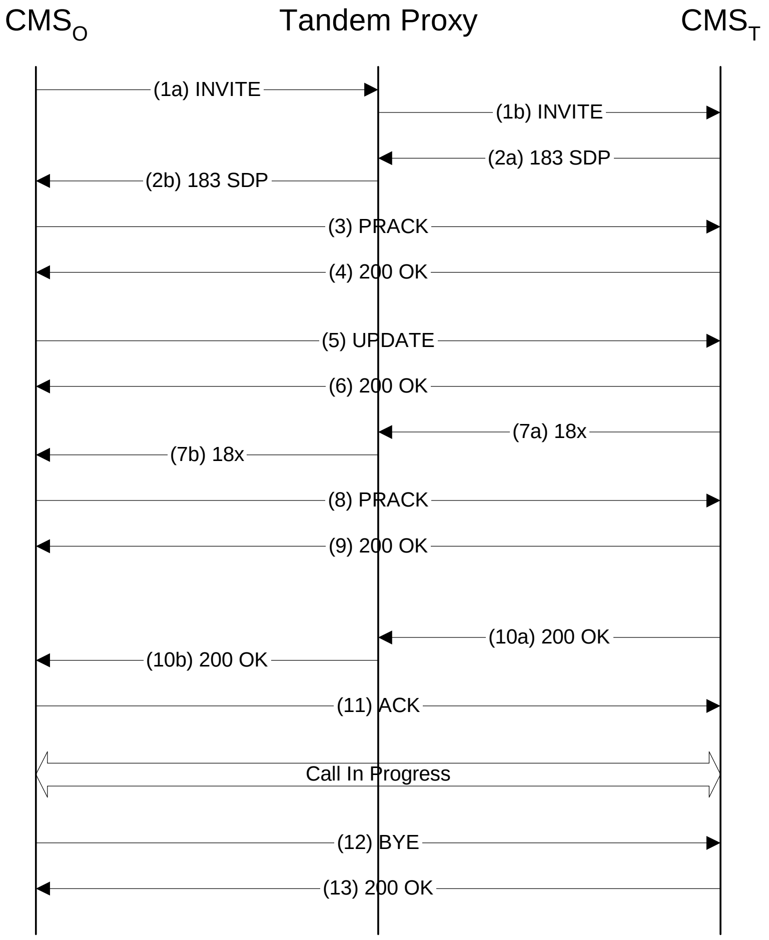


Figure 10. CMS Messages for Basic Call Setup

[11] 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 [52] for details.

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, 0, 0, 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 8.4.1.1.1.
Privacy	If calling number Privacy is requested, this header MUST contain the "id" and "critical" tag values.
Proxy-Require	If calling number Privacy is requested, this header MUST contain the option tag "Privacy".
From	Originating endpoint identification. MUST follow the requirements of Section 6.20.20.
To	URI of the destination endpoint. MUST follow the requirements of Section 6.20.39.
Call-ID	If Privacy is requested, the Call-ID MUST be generated as specified in Section 6.20.8.
Contact	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

Header:	Requirements for CMS_O
	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[12].

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.

[12] 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.

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
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:	

100-Trying (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
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.

- If CMS_T receives an INVITE with a Call-Info header declaring "purpose= answer_if_not_busy" then CMS_T MUST ignore any active CFB service for the target endpoint and not forward the call if the endpoint is busy, and instead return a 486-Busy-Here response. If the endpoint has CFU, SCF (where the calling party is on the SCF screening list), DND, or Solicitor Blocking (where the calling party is not on the solicitor blocking white list) active, CMS_T MUST reject the INVITE with a 480-Temporarily-Unavailable response.

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[13]. 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:

[13] 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**
SIP/2.0 183 Session Progress	Status line with status code 183 MUST be present.
Via:	As described in 6.13.
Require:	As defined in 6.20.32. MUST include "100rel" as described in 7.2.
Supported:	As described in 6.20.37.
P-DCS-Billing-Info	As described above, and in 7.7
From:	As described in 6.13.
To:	
Call-ID:	
Cseq:	
Contact:	As defined in 6.20.10.
Rseq:	As defined in 7.2.
Content-Type:	MUST be present and MUST contain "application/SDP".
Content-Length:	As defined in 6.20.14.
	An empty line (CRLF) MUST be present between the headers and the message body.
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].

The CMS_T is responsible for including and/or updating the History-Info header.

If there is no History-Info header present in the received INVITE, the CMS_T MUST add one to the response in accordance with the procedures in section 8.4.12.

If there is a History-Info header present in the received INVITE, then the CMS_T MUST check the data in the History-Info header against the forward-to address. If this check reveals a forwarding loop and the forward attempt is due to Call Forward Busy Line, the CMS_T MUST respond with 486-User-Busy. If the check reveals a loop and the forward attempt is due to Call Forward Variable or Selective Call Forward, the CMS_T MUST respond with 480-Temporarily-Unavailable.

If the check does not reveal a loop, the CMS_T MUST add itself to the header in conformance with the procedures outlined in section 8.4.12.2 and include the entire updated header in the 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** **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.
History-Info:	MUST be present, as described in 8.4.12.
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:	

ACK (CMS$_O$ -> CMS$_T$) Header:	Requirements On CMS$_T$ for Message Checking
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].

If there is no History-Info header present in the received INVITE, the CMS$_T$ MUST add one to the generated INVITE in accordance with the procedures in section 8.4.12.

If there is a History-Info header present in the received INVITE, then the CMS$_T$ MUST check the data in the History-Info header against the forward-to address. If this check reveals a forwarding loop and the forward attempt is due to Call Forward Busy Line, the CMS$_T$ MUST respond with 486-User-Busy. If the check reveals a loop and the forward attempt is due to Call Forward Variable or Selective Call Forward, the CMS$_T$ MUST respond with 480-Temporarily-Unavailable.

If the check does not reveal a loop, the CMS$_T$ MUST add itself to the header in conformance with the procedures outlined in section 8.4.12 and include the entire updated header in the generated INVITE.

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
History-Info:	As described in 8.4.12
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 or outside of the dialog (because routing of the out-of-dialog REFER by the network results in the request being presented to CMS_T) but containing a Target-Dialog header that identifies the 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 CMSO after processing the SDP returned by CMST.

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:	

UPDATE (CMS_O -> CMS_T) Header:	Requirements On CMS_O for Message Generation
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.

200-OK (CMS_T -> CMS_O) Header:	**Requirements On CMS_O for Message Checking**
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[14] 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.

[14] 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.

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.
History-Info:	MUST be present as described in 8.4.12.
From:	As described in 6.13.
To:	
Call-ID:	
Cseq:	
Contact:	As defined in 6.20.10.
Content-Length:	MUST be present if the transport protocol is stream-based (e.g., TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

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

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

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

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

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 History-Info header in the 3xx response to a History-Info header in the new INVITE.

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
History-Info:	As defined in 8.4.12
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.

INVITE (CMS_O -> CMS_F) Header:	Additional Requirements for Message Generation
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 0.

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.

PRACK (CMS$_O$ -> CMS$_T$) Header:	Requirements On CMS$_T$ for Message Checking
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 CMSO after processing the SDP returned by CMST.

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

200-OK (CMS$_T$ -> CMS$_O$) Header:	Requirements On CMS$_T$ for Message Generation
SIP/2.0 200 OK	As described in 6.12.
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.

200-OK (CMS_T -> CMS_O) Header:	Requirements On CMS_T for Message Generation
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 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 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 [52]. 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 [52]. 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.

PRACK (CMS_O -> CMS_T) Header:	Requirements On CMS_T for Message Checking
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:	

180 or 183 Provisional Response (CMS_T -> CMS_O) Header:	Requirements On CMS_O For Message Checking
Contact:	As described in 6.20.10.
Cseq:	As described in 6.13.
Rseq:	As defined in 7.2.
Content-Length:	MUST be present if the transport protocol is stream-based (*e.g.*, TCP), as described in 6.20.14.
	An empty line (CRLF) MUST be present.

If the received provisional response does not conform to the above format, then CMS_O MAY ignore the message. Otherwise, the 180-Ringing response indicates to the originating CMS that the terminating party is being alerted and that local ringback SHOULD be generated. CMS_O, by methods outside the scope of this specification, informs the originating endpoint of the desired actions. Note that, in accordance with Section 6.13, the originating endpoint must be prepared to receive media based on the offer/answer exchange performed earlier. If media is received while generating local ringback, the originating endpoint SHOULD stop the local ringback tone[15].

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.

If the originating endpoint is off-net then the following requirements apply to the originating MGC. When a SIP 183 provisional response is received from CMS_T, then the MGC MUST set the Interworking Indicator (bit I) of the Backward Call Indicators mandatory parameter in the ACM to a value of '1' meaning Interworking Encountered. This enables cut-through of both forwards and backwards transmission paths in the preceding TDM equipment. If the MGC receives a SIP 180 provisional response from CMS_T, then it MUST set the Interworking Indicator (bit I) of the Backward Call Indicators mandatory parameter in the ACM to a value of '0', meaning No Interworking Encountered. A forward transmission path will already have been cut through at the preceding TDM equipment on the handling of the ISUP Initial Address Message (IAM).

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.

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

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.

8.4.1.7.1 CMS_T sending 200-OK Final Response

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

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

In either case, the CMS_T is responsible for including and/or updating the History-Info header.

If there is no History-Info header present in the received INVITE, the CMS_T MUST add one to the response in accordance with the procedures in section 8.4.12.

If there is a History-Info header present in the received INVITE, then the CMS_T MUST check the data in the History-Info header against the forward-to address. If this check reveals a forwarding loop, the CMS_T MUST respond with 408-Request-Timeout.

If the check does not reveal a loop, the CMS_T MUST add itself to the header in conformance with the procedures outlined in section 8.4.12.2 and include the entire updated header in the 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.
History- Info:	MUST be present, as described in 8.4.12.
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.

ACK (CMS_O -> CMS_T) Header:	Requirements On CMS_T For Message Checking
	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 0.

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.
History- Info:	MUST be present as described in 8.4.12.
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.

The CMS_O MUST copy the contents of the History-Info header in the 3xx response to a History-Info header in the new INVITE.

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
History-Info	As defined in 8.4.12.
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.

INVITE (CMS_O -> CMS_F) Header:	Additional Requirements for Message Generation
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:	

Error (CMS_T -> CMS_O) Header:	Requirements On CMS_O for Message Checking
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 Emergency Call Procedures

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.

CMS_O MUST indicate in the SIP signaling that this is an emergency call by including a Priority header in the INVITE containing the value "emergency".

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 7.9.

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 active in an emergency call. When a PSAP originates an emergency callback call as defined in Section 8.4.2.3 the terminating CMS MUST recognize that the call is an emergency call based on the presence of a Priority header containing the value "emergency" in the received INVITE request.

8.4.2.1 Network Hold Support

Network Hold is an optional capability that provides the PSAP operator with the ability to control when the emergency call is released. If Network Hold is supported, then when the originating party hangs up during an emergency call, the call is not released. Instead, the call is maintained, and Network Hold is applied by placing the media path on hold and sending a Network Hold "disconnect" signal toward the PSAP operator. When the originating party subsequently goes back off-hook, Network Hold is removed by restoring 2-way media communication and sending a Network Hold "resume" signal toward the PSAP operator.

If Network Hold is supported and configured, then the originating CMS and the MGC MUST support the procedures defined here. Otherwise, the following procedures do not apply.

The CMS_O processing of an originating disconnect request (i.e., when originating party goes on-hook) received during an emergency call differs from standard basic call processing of originating disconnect only after the emergency call is answered. If the originator of the emergency call goes on-hook before CMS_O receives a 200-OK (answer) response to the INVITE request, then CMS_O MUST process the disconnect request in accordance with the normal session release procedures defined in RFC 3261 [6] (i.e., send CANCEL request, etc).

However, if the originator of the emergency call goes on-hook after CMS_O receives a 200-OK response to INVITE, then CMS_O MUST NOT initiate normal session release procedures. Instead, CMS_O MUST initiate a Network Hold timer, and apply Network Hold by following the call-hold procedures defined in Section 8.4.4.1. In addition to the call-hold procedures defined in section 8.4.4.1, CMS_O MUST populate the re-INVITE(hold) or UPDATE(hold) request with a Priority header containing the value "emergency" to indicate to the MGC that special emergency Network Hold disconnect procedures apply (the MGC procedures for support of emergency Network Hold disconnect are defined in section 8.4.2.4.3). The message flow to support Network Hold disconnect procedures is shown in Figure 11.

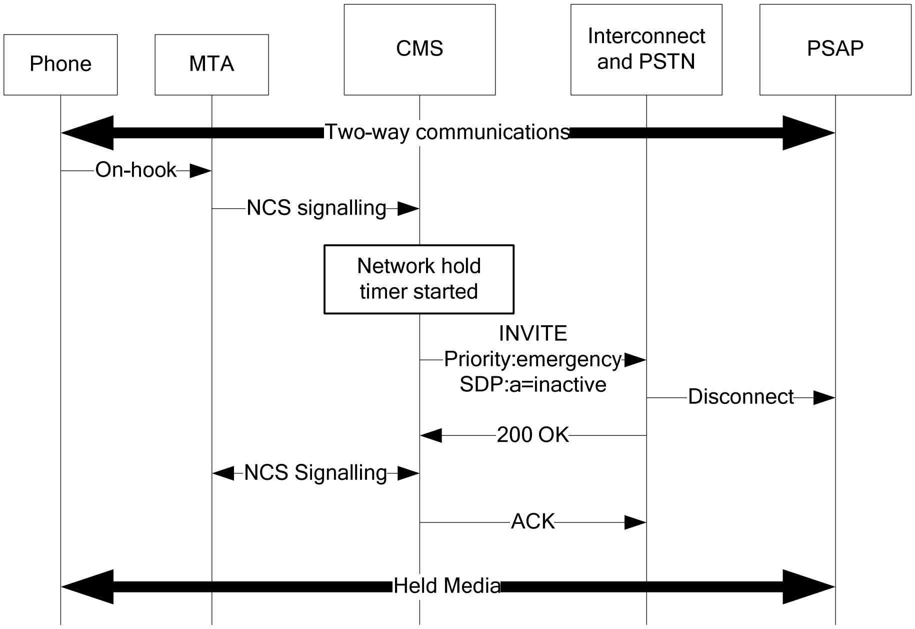


Figure 11. Emergency Call; Network Hold Disconnect procedures

If the originator of the emergency call goes back off-hook while Network Hold is in effect, then CMS_O MUST cancel the Network Hold timer, and remove Network Hold following the call-hold resume procedures defined in section 8.4.2.2. In addition to the call-hold resume procedures defined in Section 8.4.2.2, CMS_O MUST populate the re-INVITE(resume) or UPDATE(resume) request with a Priority header containing the value "emergency" to indicate to the MGC that special emergency Network Hold resume procedures apply (the MGC procedures for support of emergency Network Hold resume are defined in section 8.4.2.4.3). The message flow to support Network Hold resume procedures is shown in Figure 12.

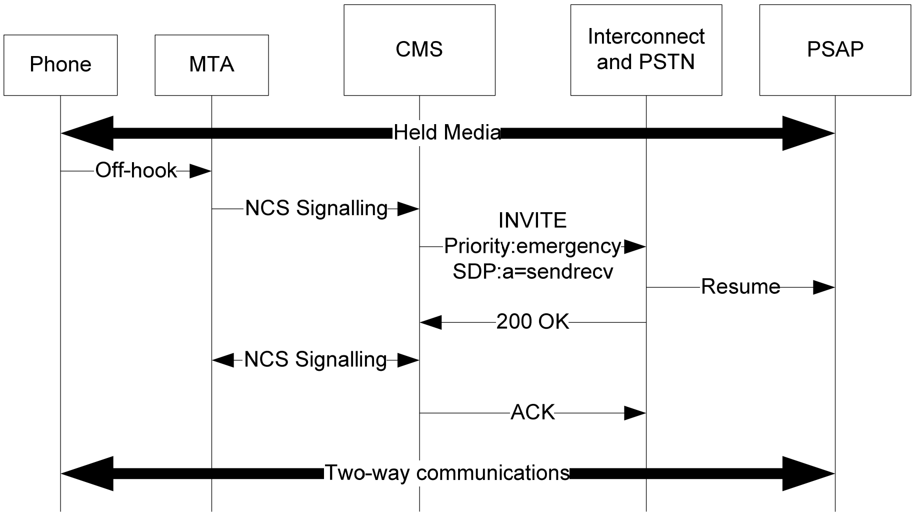


Figure 12. Emergency Call; Network Hold Resume procedures

When the CMS_O network hold timer expires, the CMS_O MUST initiate session release procedures by sending a BYE request as defined in RFC 3261 [6].

The value of the network hold timer should be provisioned on CMS_O. A value of '0' means that the network does not support network hold although this is not recommended unless it is known that all PSAPs reachable from this CMS do not support network hold.

8.4.2.2 PSAP Operator Ringback

Operator Ringback is a feature that allows the PSAP to alert the originating party in an established emergency call. The type of alerting depends on whether Network Hold is applied or not (i.e., whether the originating party is on-hook or off-hook). If Network Hold is not applied (caller is off-hook), the caller can be alerted with a special in-band audible alerting (usually ROH tone). If Network Hold is applied (caller is on-hook) then the caller is alerted with standard power ringing. While the caller is receiving the operator ringback alerting signal (e.g., power ringing or ROH tone), the PSAP receives ringback tone.

8.4.2.2.1 PSAP Operator Ringback while Network Hold not Applied (Caller Off-Hook)

When the PSAP indicates that the caller should be alerted and Network Hold is not in effect, the inband audible alerting signal can be applied in one of several ways. For example, the PSAP could instruct the MG endpoint to send the audible in-band alerting signal toward the originating party via the established bearer path. This mechanism requires no session signaling, and therefore is transparent to and requires no support of CMS_O or the MGC.

Alternatively, on receiving an Operator Ringback request from the PSAP, the MGC could reconfigure the session bearer path to include a media server which provides the audible in-band alerting signal (e.g., ROH) toward the originating party. In this case, the media server provides the audible ringback tone to the PSAP operator. The session is reconfigured using a standard re-INVITE or UPDATE request, and therefore places no new requirements on CMS_O.

8.4.2.2.2 PSAP Operator Ringback while Network Hold Applied (Caller On-Hook)

When the PSAP indicates that the caller should be alerted while Network Hold is applied, the MGC MUST send a re-INVITE request containing an Alert-Info header specifying the type of alerting to be applied, and a Priority header containing the value "emergency". On receiving the re-INVITE, CMS_O MUST process the request in accordance with normal terminating call procedures; i.e., signal the originating MTA to apply physical alerting as indicated in the Alert-Info header, and send a 180-Ringing response to the MGC.

When the alerting emergency call originator goes back off-hook, Network Hold is removed and normal 2-way communication is resumed. However, the signaling to remove Network Hold in this case differs from that described in section 8.4.2.1. In this case, on receiving an indication that the emergency call originator has gone back off-hook, CMS_O MUST send a 200-OK response to re-INVITE and cancel the Network Hold timer. On receiving the 200-OK response to re-INVITE, the MGC MUST perform the Network Hold resume procedures defined in section 8.4.2.4.3. The message flow to support PSAP Operator Ringback while Network Hold is applied is shown in Figure 13.

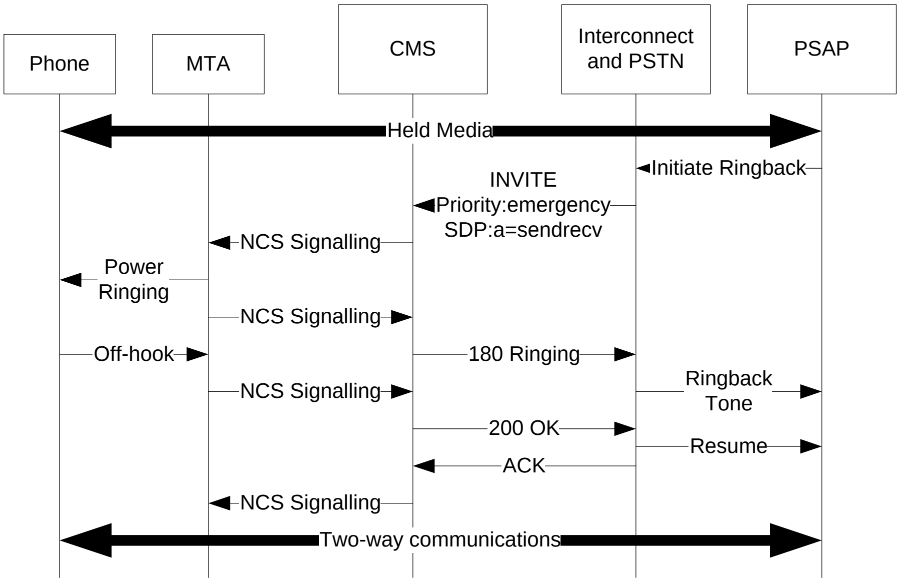


Figure 13. Emergency Call; PSAP Operator Ringback While on Hold

Note that while the operator ringback signal is being applied, the network provides audible ringback tone toward the PSAP operator until the emergency call originator goes back off-hook. If the MGC receives a disconnect indication from the PSAP operator during alerting, it MUST initiate session release procedures as defined in RFC 3261 [6]. If the Network Hold timer expires during alerting, CMS_O MUST initiate session release procedures as defined in RFC 3261 [6].

8.4.2.3 PSAP CallBack (PSAP Originated Emergency Call)

After an emergency call has ended (due to a caller-initiated disconnect before 200 OK, PSAP forced disconnect, or Network Hold timer timeout), the PSAP operator may wish to re-establish a connection with the emergency caller. In this case, the PSAP initiates a new call towards the original emergency calling party based on information received in the previous call. The callback call arrives as a new terminating call at the CMS that controls target party. The terminating INVITE contains a Priority header containing the value "emergency", thus enabling the CMS to identify this as an operator-initiated emergency callback call. Once the call is answered and in a stable 2-way talk state, it receives the same treatment as if the MTA had originated the emergency call.

On receiving an INVITE with a Priority header containing "emergency", CMS_T MUST mark the call as an emergency call. Once the call is answered and in a stable 2-way talk state, CMS_T MUST support the emergency call Network Hold Procedures defined for CMS_O in Section 8.4.2.1, and the PSAP Ringback procedures defined in Section 8.4.2.2. The message flow for the PSAP callback call is shown in Figure 14.

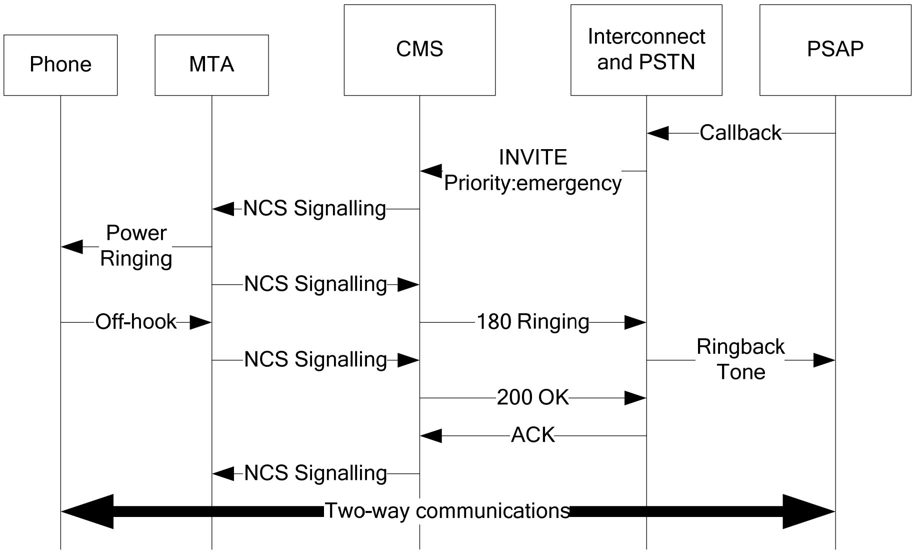


Figure 14. Emergency Call; PSAP Operator Callback

8.4.2.4 MGC Procedures

8.4.2.4.1 Invite processing

The MGC procedures to support emergency calls differ depending on whether the trunk interface to the PSTN Selective Router is via SS7 ISUP or Multi-Frequency (MF) Trunks.

SS7 ISUP Trunks:

Both shared and non-shared ISUP trunks may be used to the selective router. When shared facilities are used the MGC MUST limit those facilities with the use of Simulated Facility Groups (SFGs) as documented in Telcordia GR 2956 [50].

On receiving an INVITE request containing a Priority header with a value "emergency", the MGC MUST build the resulting ISUP IAM message in accordance with Telcordia GR 2956 [50] including the following mappings:

- The called party number MUST be set to the emergency services number (for example, 911) for that location.
- The P-Asserted-Identity header MUST be mapped to the ISUP calling party number.
- The P-Asserted-Identity header MUST be mapped to the ISUP Charge Number parameter according to Telcordia GR 2956 [50] requirement R-5.
- The PIDF-LO MUST be mapped to Generic Digits Parameter with a Type of Digits set to "Calling Geodetic Location" (CGN).
- The nature of address parameter MUST be set to 'National Number'
- The ISUP Calling Party Category MUST be set to "emergency service call"

If the MGC is configured to support Network Hold, then it MUST send the ISUP IAM with hold_possible SAP. The MGC will then receive an indication in the ACM on whether hold should be supported in the hold_possible SAP.

Multi-Frequency (MF) Trunks:

The call MUST be processed as described in Telcordia GR 529 [48]. The called party number MUST be changed to the emergency services number (for example, 911) for that location. The P-Asserted-Identity header MUST be mapped to the calling party number.

8.4.2.4.2 Network Hold processing

The MGC processing of a Network Hold event depends on whether the trunk interface to the PSTN Selective Router is via SS7 ISUP or Multi-Frequency (MF) Trunks.

SS7 ISUP Trunks:

The MGC MUST process Network Hold disconnect requests in accordance with Telcordia GR 1277 [49] (Operator SS7) including the following mappings:

- On receiving a Network Hold disconnect request from CMS_O as defined in section 8.4.2.1, the MGC MUST instruct the SG to send an ISUP FACILITY message with disconnect_request SAP.
- On receiving a Network Hold resume request from CMS_O as defined in section 8.4.2.1, the MGC MUST instruct the SG to send an ISUP FACILITY message with reconnect_request SAP.

Multi-Frequency (MF) Trunks:

The MGC MUST process the Network Hold disconnect request in accordance with Telcordia GR 529 [48] including the following mappings:

- On receiving a Network Hold disconnect request from CMS_O as defined in section 8.4.2.1, the MGC MUST instruct the MG to send an on-hook signal
- On receiving a Network Hold resume request from CMS_O as defined in section 8.4.2.1, the MGC MUST instruct the MG to send an off-hook signal

8.4.2.4.3 PSAP Ringback processing

The MGC processing of a PSAP Ringback event depends on whether the trunk interface to the PSTN Selective Router is via SS7 ISUP or Multi-Frequency (MF) Trunks.

SS7 ISUP Trunks:

The MGC MUST process the operator ringback request in accordance with Telcordia GR 1277 [49] (Operator SS7).

On receiving an indication from the SG that an ISUP FACILITY message with the ringback_request SAP was received,

- if Network Hold is in effect, then the MGC MUST send a re-INVITE to CMS_O to initiate the ringback function at the MTA. The MGC MUST include an alert-info header in the re-INVITE.
- if Network Hold is not in effect, then the MGC procedures to apply alerting tone at the MTA (e.g., ROH) are not specified (e.g., the MGC could reconfigure the network path to a media server that provides the tone).

Multi-Frequency (MF) Trunks:

The MGC MUST process the operator ringback request in accordance with Telcordia GR 529 [48].

On receiving a Ringback request from the MG,

- if Network Hold is in effect, then the MGC MUST send a re-INVITE to CMS_O to initiate the ringback function at the MTA. The MGC MUST include an alert-info header in the re-INVITE.
- if Network Hold is not in effect, then the MGC procedures to apply alerting tone at the MTA (e.g., ROH) are not specified (e.g., the MGC could reconfigure the network path to a media server that provides the tone).

8.4.2.4.4 PSAP Callback (PSAP Originated Emergency Call)

The MGC MUST indicate in the SIP signaling that the PSAP callback call as an emergency call by including a Priority header in the INVITE containing the value "emergency". The method of determining that the call is a PSAP callback call is as follows:

SS7 ISUP Trunks:

- if an incoming call is received over a trunk group dedicated to emergency calls, then by default the MGC can assume it is an emergency callback call.
- if the ISUP Calling Party Category for an incoming call is set to "emergency service call" or "high priority emergency service call", then the MGC can assume it is a emergency callback call.

Multi-Frequency (MF) Trunks:

PacketCable does not define an MF Trunk interface for support of PSAP emergency callback calls.

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

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.

When the REFER is generated outside of an established dialog the CMS_I MUST include a Target-Dialog header in the REFER which matches the call-leg identification (From Tag, To tag and Call-ID) of an established call-leg between CMS_I and CMS_O. The Request-URI of the REFER MUST be the value of the most recently received Contact header from CMS_O.

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. If CMSI has discovered a Contact header for CMS_T and that Contact header is a GRUU then the URI from this Contact MUST be used by CMS_I to populate the Refer-To header. Otherwise 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. In either case, 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.
Target- Dialog:	MUST be present only if the REFER is sent outside of an established dialog. If included MUST be as described in 7.14.
Content-Length:	MUST be present, and MUST indicate a zero-length body.
	An empty line (CRLF) MUST be present.

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.

If the REFER is received outside of an existing dialog at a Bridge Server performing conferencing and the included Target-Dialog header identifies an existing call-leg then 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 call-leg identified by the Target-Dialog header. 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.

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

200-OK (CMS_I -> CMS_O) Header:	Requirements On CMS_I for Message Generation
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: re-INVITE/UPDATE(hold)

To place a call on hold, a re-INVITE(hold) or 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 re-INVITE or UPDATE request, with an additional "a=sendonly" or "a=inactive" 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 (with the exception of the request type, the re-INVITE message encoding is identical):

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:	

UPDATE(Hold) **(CMS$_I$ -> CMS-Agent$_R$) Header:**	**Requirements On CMS$_I$ for Message Generation** **Requirements On CMS$_R$ for Message Checking**
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=	MUST be present.
o=	
s=	
c=	
b=	
t=	
a=	a= line MUST be present and MUST indicate "sendonly" or "inactive".
m=	

If the re-INVITE(hold) or 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=	MUST be present.
s=	
c=	
b=	
t=	
a=	a= line MUST be included and encoded as defined in RFC 4566 [3]
m=	

After sending the re-INVITE(hold) or 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, a re-INVITE(resume) or an UPDATE(resume) is sent. A re-INVITE/UPDATE(resume) is a re-INVITE/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 re-INVITE or UPDATE 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)	**Requirements On CMS$_I$ for Message Generation**
(CMS$_i$ -> CMS$_r$) Header:	**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 re-INVITE/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 a re-INVITE(resume) or 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_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=	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 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 8.4.4.4.1. An INVITE(Codec-change) MAY be used instead, as described in 8.4.4.4.2.

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 8.4.4.4.2.

8.4.4.4.1 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 8.4.4.4.2; 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.

8.4.4.4.2 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

INVITE(codec-change) (CMS_I->CMS_R) Header:	Requirements on CMS_I for message generation Requirements on CMS_R for message checking
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=	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.

183-Session-Progress **(CMS$_R$ -> CMS$_I$) Header:**	**Requirements On CMS$_R$ for Message Generation** **Requirements On CMS$_I$ for Message Checking**
v= o= s= c= b= t= a= m=	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 0.

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:	

200-OK (CMS_R -> CMS_I) Header:	Requirements On CMS_R for Message Generation Requirements On CMS_I for Message Checking
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:	

BYE (CMS$_I$ -> CMS$_R$) Header:	**Requirements on CMS$_I$ For Message Generation** **Requirements on CMSR For Message Checking**
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 15.

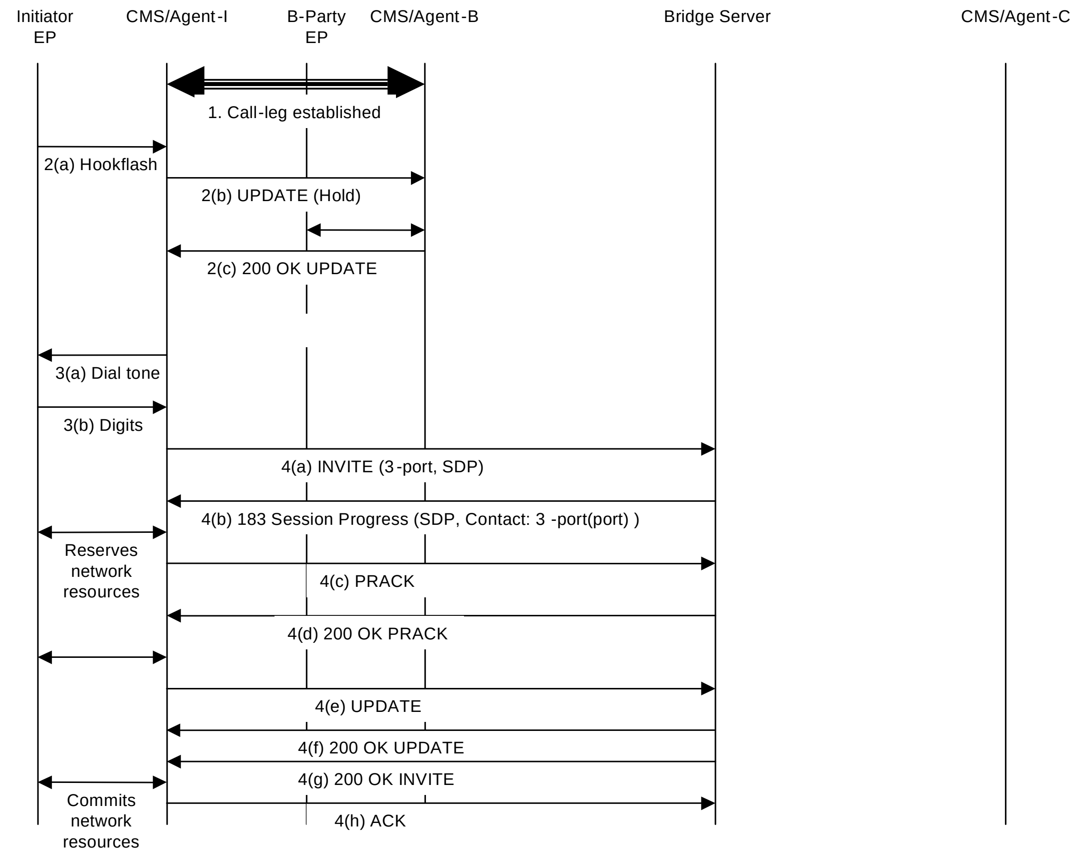


Figure 15. End of transferred call

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 16 and is representative of any failure case that prevents the initial connection to the conference bridge.

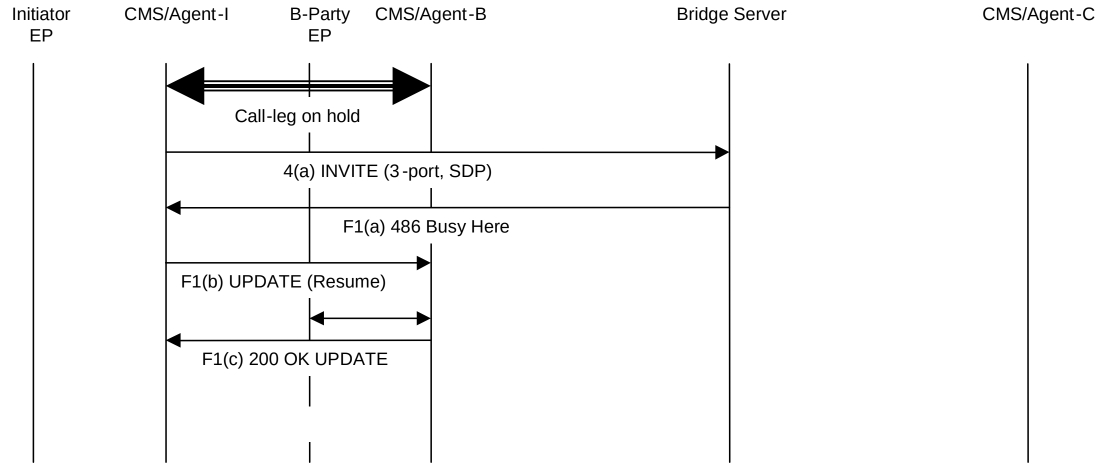


Figure 16. Failure ca– F1 – no free conference circuits

5. Once the Bridge Server has accepted the call, CMSI issues REFER either within the existing dialog or outside of the dialog but containing a Target-Dialog header identifying the existing dialog to the Bridge Server, requesting that it establish a call to Party C. The Bridge Server sends a NOTIFY to CMS_I and establishes the new call on the same conference bridge as the first call. During alerting, it plays ringing audio tone through the bridge to the Initiator. When Party C answers (which could be at any one of a number of points in the following sequence of steps), the Bridge Server sends a final NOTIFY to CMSI. This step is shown in Figure 17.

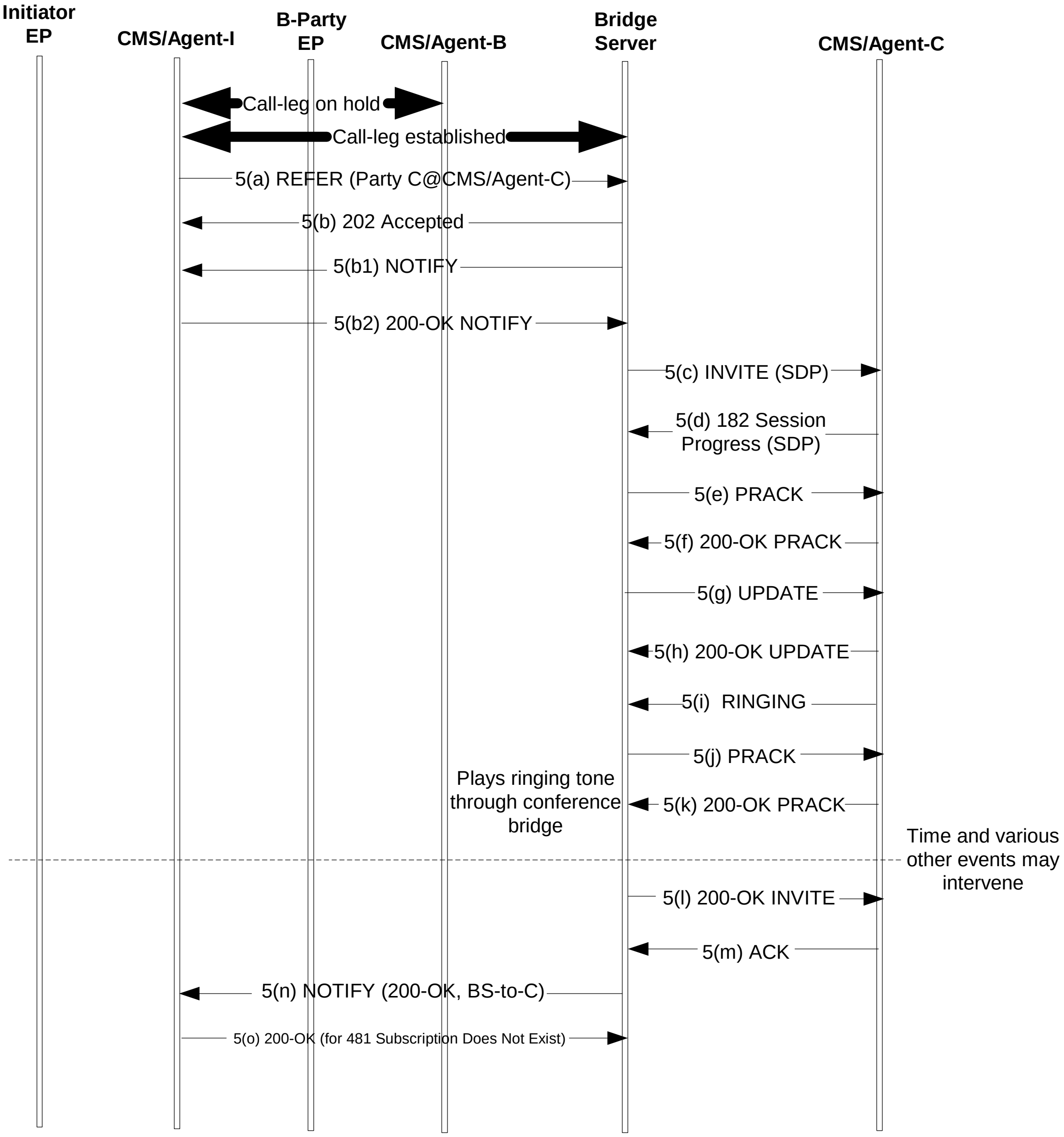


Figure 17. Establishing the leg from Bridge Server to Party C

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 18. 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.

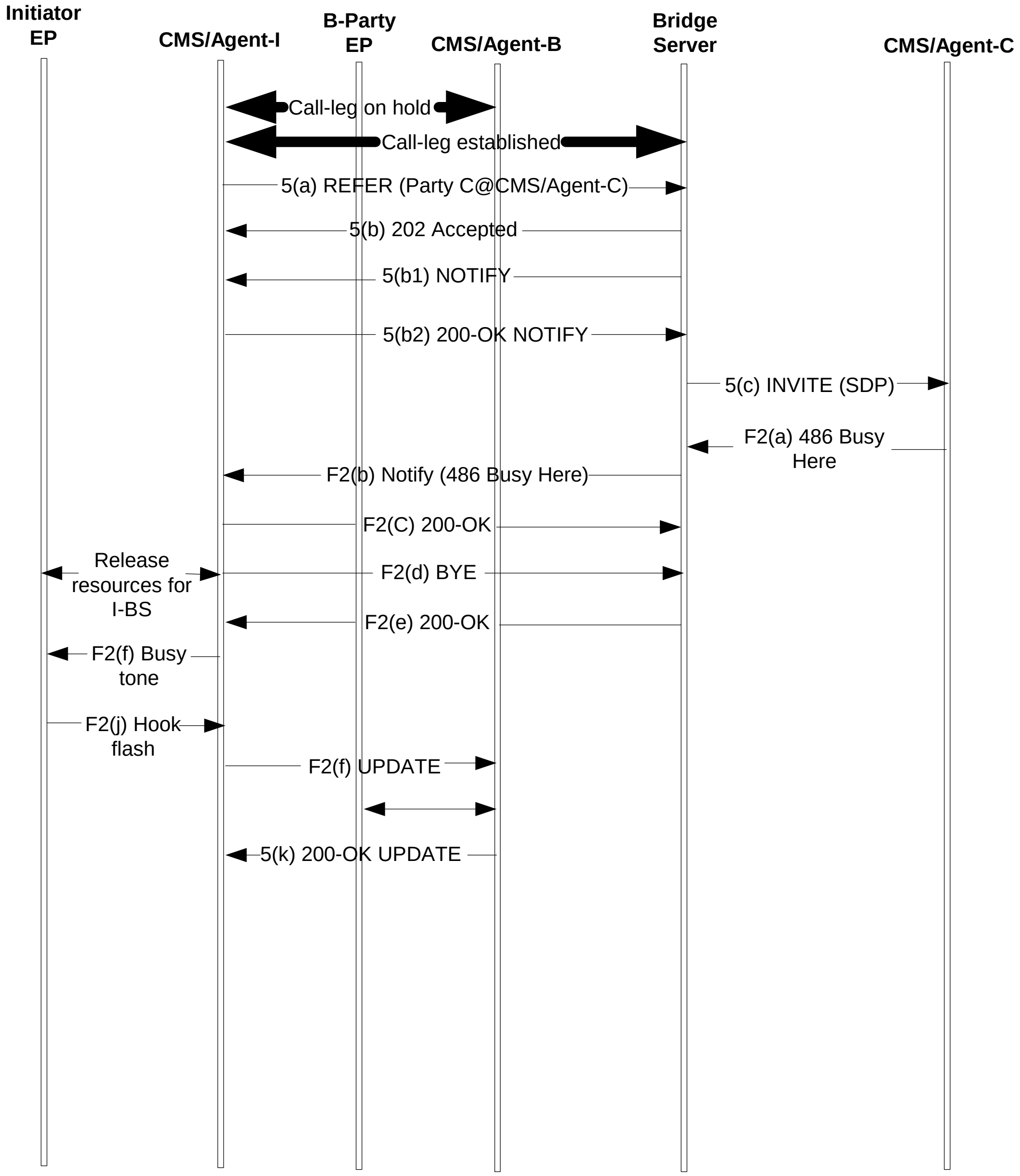


Figure 18. Failure ca– F2 – Party C busy

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 19.

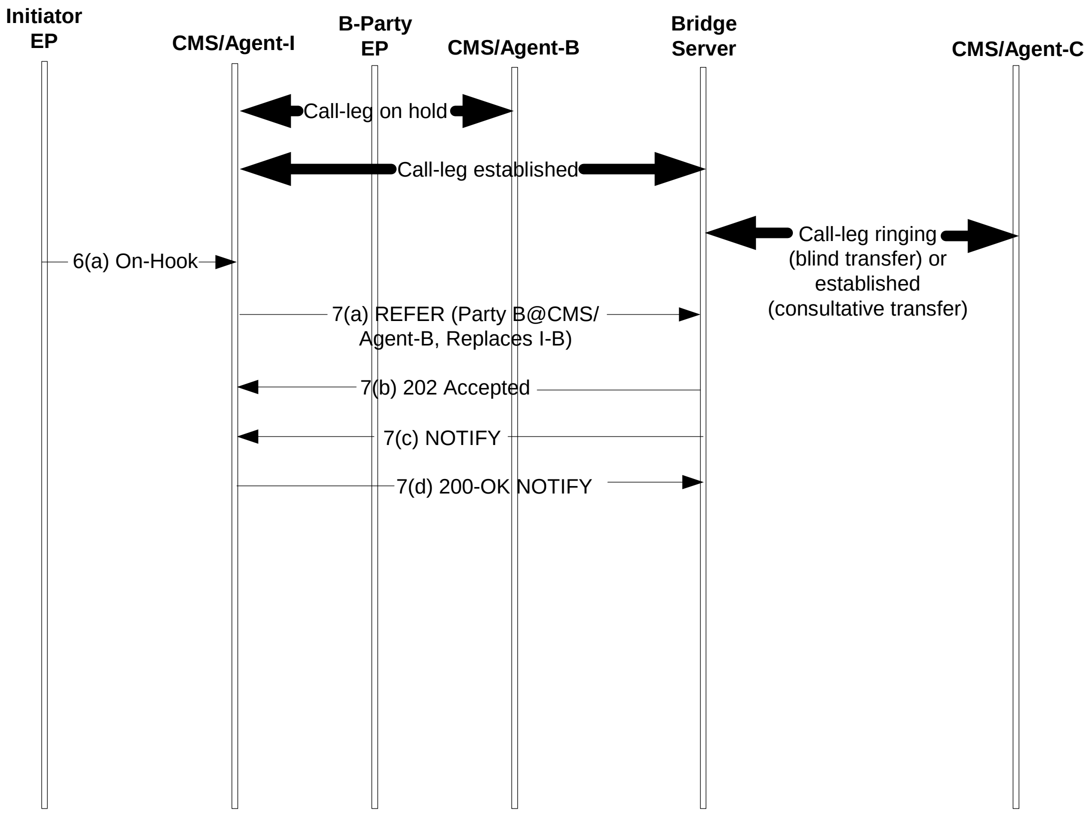


Figure 19. On-hook initiates transfer action

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 20.

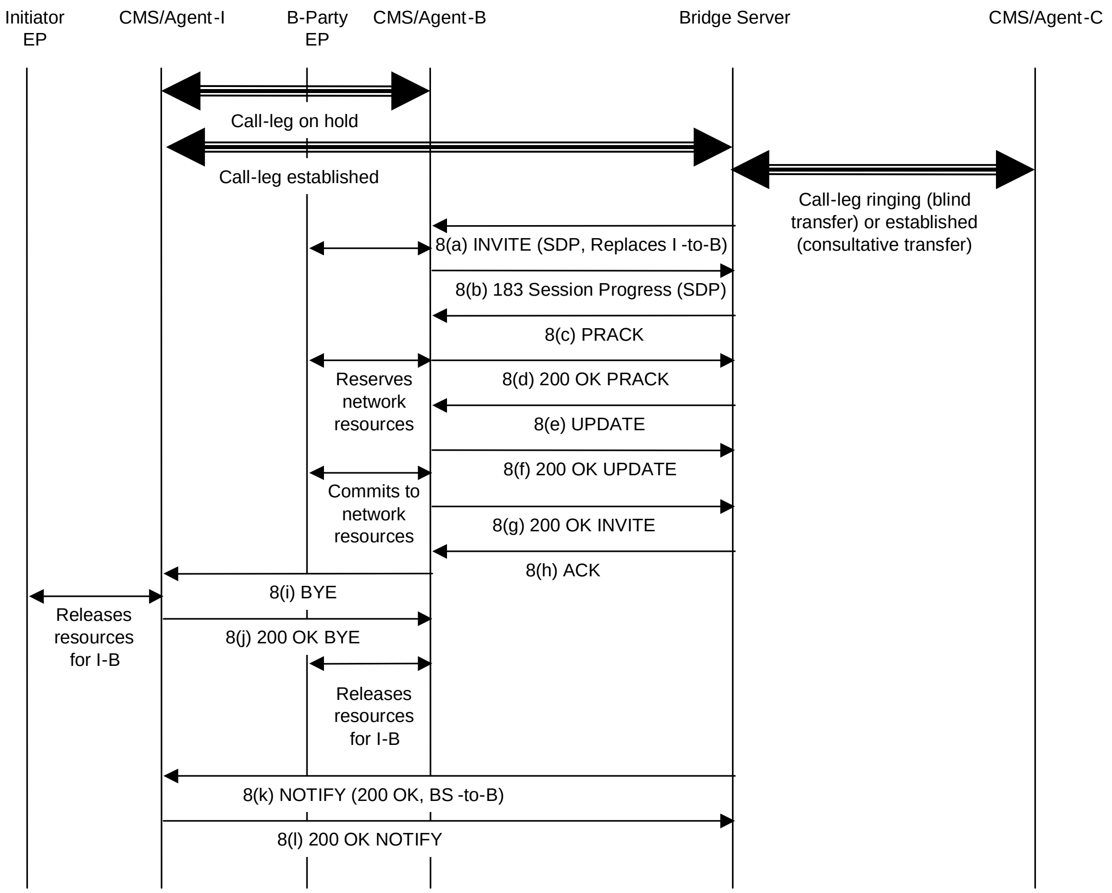

Initiator EP
CMS/Agent-I
B-Party EP
CMS/Agent-B
Bridge Server
CMS/Agent-C
Call-leg on hold
Call-leg established
Call-leg ringing (blind transfer) or established (consultative transfer)
8(a) INVITE (SDP, Replaces I -to-B)
8(b) 183 Session Progress (SDP)
8(c) PRACK
Reserves network resources
8(d) 200 OK PRACK
8(e) UPDATE
8(f) 200 OK UPDATE
Commits to network resources
8(g) 200 OK INVITE
8(h) ACK
8(i) BYE
Releases resources for I-B
8(j) 200 OK BYE
Releases resources for I-B
8(k) NOTIFY (200 OK, BS -to-B)
8(l) 200 OK NOTIFY

Figure 20. 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 20 have succeeded.

8. 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 bridge16. This step is shown in Figure 21.

[16] 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.

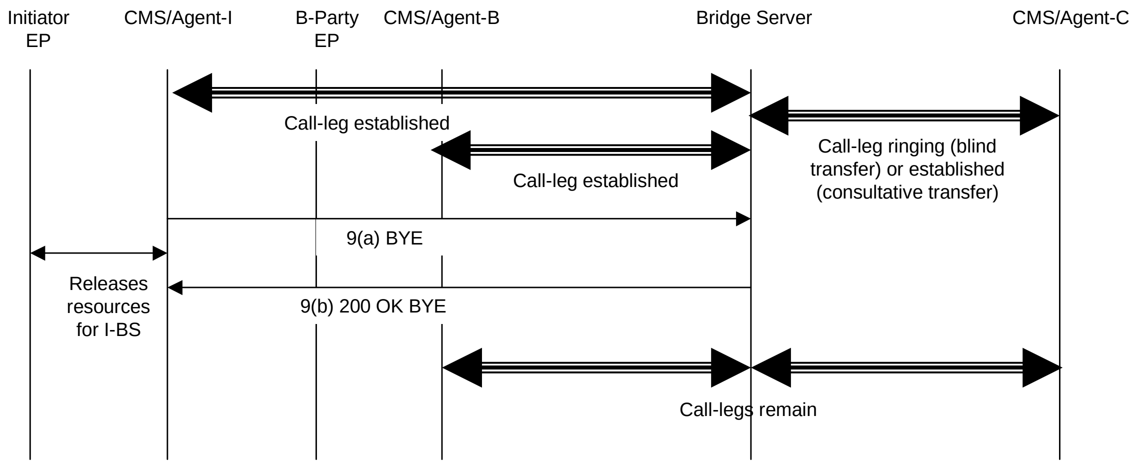


Figure 21. Transfer completed, CMSI ends involvement in call

9. 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 22.

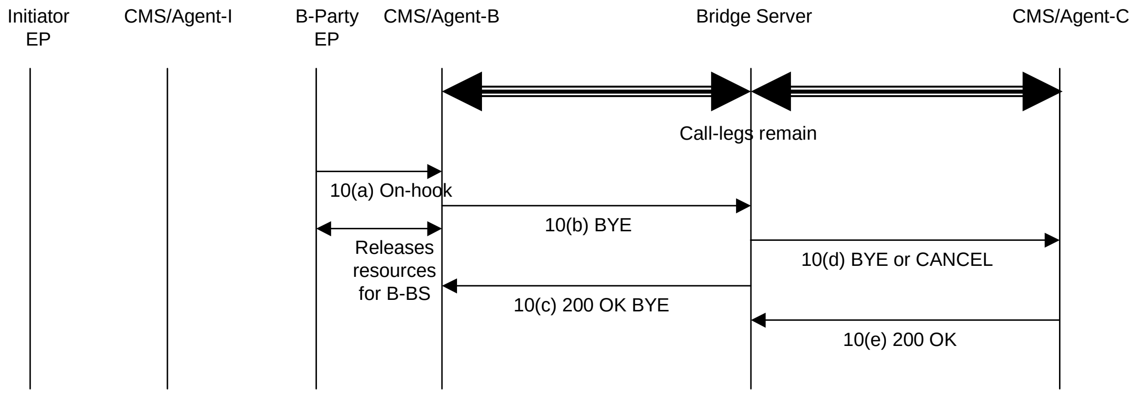


Figure 22. End of transferred call

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 15 to Figure 18 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 19**.**
8. The actions of the Bridge Server and CMS_B in response to the REFER are identical to step 8 Figure 20 of the call transfer procedure. The one exception to this is that when CMS_I receives the NOTIFY (message 8(k) in Figure 20), it does nothing further until the Initiator goes on-hook or the Bridge Server terminates the call-leg.

8.4.8 Automatic Recall and Callback

In support of Automatic Recall and Callback, CMSS supports the extensions defined in Sections 7.16 and 6.20.9.

8.4.8.1 CMS$_O$ Sends INVITE to Target

When a user invokes an AR or AC call, CMS$_O$ MUST follow the procedures given for a basic call, as given in Section 8.4.1, with the following exceptions. CMS$_O$ MUST populate the request URI of the INVITE request based on the stored last dialed address for Auto Callback, or the stored address for the last call received for Auto Recall. CMS$_O$ MUST add the Call-Info header with a purpose of "answer_if_not_busy".

CMS$_O$ MUST encode the INVITE request as follows:

INVITE (CMS$_O$ -> CMS$_T$) Header:	Additional Requirements For Message Generation
INVITE URI SIP/2.0	URI based on stored last party called or last calling party.
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
Privacy:	As described in 8.4.1.1.
P-DCS-Billing-Info:	As defined in 7.7.
Max-Forwards	As defined in 6.13 and 6.20.22.
From:	As defined in 6.13 and 6.20.20.
To:	As defined in 6.20.39.
Call-ID:	As defined in 6.13 and 6.20.8.
Cseq:	As defined in 6.20.16.
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.
Call-Info	MUST be present and MUST contain "purpose=answer_if_not_busy".
	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 follow the requirements as discussed in sections 8.4.1.1 through 8.4.1.5.

If CMS_O receives a 200-OK response to INVITE, then it MUST follow the basic call procedures defined in section 8.4.1.8.1. If CMS_O receives a 486-Busy-Here or 600-Busy-Everywhere response to the INVITE, then it MUST follow the AC/AR procedures defined in sections 8.4.8.2 and 8.4.8.3. If CMS_O receives any other final 4xx, 5xx, or 6xx response, then it MUST follow the error response procedures defined in section 8.4.1.8.3.

8.4.8.2 CMS_O Sends SUBSCRIBE to Target

On receiving a 486-Busy-Here or 600-Busy-Everywhere response to an AC/AR INVITE request (defined in section 8.4.8.1), CMS_O MUST initiate a timer that limits the overall duration of the AC/AR campon attempt. CMS_O MUST then establish a subscription to the dialog event package of the target endpoint, by sending a SUBSCRIBE request to CMS_T, formatted as follows:

SUBSCRIBE (CMS_O -> CMS_T) Header:	Additional Requirements For Message Generation
SUBSCRIBE URI SIP/2.0	MUST contain the URI returned in the Contact header of the INVITE response, if the contact address is a GRUU. Otherwise, MUST contain the URI of the stored last called or calling party.
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 "dialog" as defined in 7.16.
Expires:	MUST contain the overall (or remaining) AC/AR duration timer value, as specified in 7.5.
Accept:	MUST contain "application/dialog-info+xml" as defined in 7.5 and 7.16.
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 receiving the SUBSCRIBE request, CMS_T MUST verify that the message is encoded as specified above. CMS_T MUST identify the target subscriber based on the request URI of the SUBSCRIBE request. If the request URI contains a GRUU, then CMS_T MUST derive the identity of the target subscriber from the GRUU. Otherwise (e.g., the request URI contains a non-GRUU SIP URI with "user=phone", or a Tel URI) CMS_T MUST use the request URI as the identity of the target subscriber. If the SUBSCRIBE message does not meet the above requirements or it is not for a valid target subscriber, then CMS_T MUST consider the request to be in error and return an appropriate 4xx, 5xx, or 6xx error code. Otherwise, CMS_T MUST return a 200-OK response and continue processing as described below:

200-OK (CMS_T -> CMS_O) Header:	**Requirements On CMS_T for Message Generation** **Requirements On CMS_O 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.

8.4.8.3 CMS_T Sends NOTIFY to CMS_O

Upon receiving a valid SUBSCRIBE request, CMS_T MUST send an immediate NOTIFY with the dialog state of the target subscriber. Whenever the dialog state of the target subscriber changes, CMS_T MUST send an updated NOTIFY with the new dialog state. The NOTIFY request is described in Sections 7.5 and 7.16. CMS_T MUST encode the NOTIFY request as follows:

NOTIFY (CMS_T -> CMS_O) Header:	**Requirements On CMS_T for Message Generation** **Requirements On CMS_O for Message Checking**
NOTIFY URL SIP/2.0	As described in 7.5.
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 "dialog" as defined in 7.16.
Subscription-State:	As described in 7.5.
Content-Type:	MUST contain "application/dialog-info+xml" as defined in 7.5 and 7.16.
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 7.16.

Upon receiving the NOTIFY request, CMS_O MUST verify that the message is encoded as specified above. If the NOTIFY message does not meet the above requirements, CMS_O MUST consider the request to be in error and return an appropriate 4xx, 5xx, 6xx error code. Otherwise, CMS$_O$ MUST respond with a 200-OK and continue processing as described below:

200-OK (CMS$_O$ -> CMS$_T$) Header:	Requirement for CMS$_O$ for Message Generation Requirement for CMS$_T$ 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.

Upon receiving a NOTIFY message of "target is idle", CMS$_O$ MUST first cancel the dialog-event subscription by first sending a SUBSCRIBE message with an Expires header containing the value "0". Once the subscription is cancelled, CMS$_O$ MUST send a new INVITE request to the target endpoint as described in Section 8.4.8.1. If CMS$_O$ receives a 486-Busy-Here or 600-Busy-Everywhere response to the INVITE, then it MUST automatically re-subscribe to the dialog event package for the remainder of the overall AC/AR timer as specified in section 8.4.8.2. (Note, a "busy" response could be returned in this case as a result of a race condition, where the target endpoint sends a NOTIFY of "target is idle", and then becomes busy in a new call before the subsequent INVITE is received).

If the overall AC/AR duration timer expires before the target becomes idle, then CMS$_O$ MUST abandon the AC/AR attempt, and cancel the subscription by sending a SUSCRIBE request to CMS$_T$ with an Expires header containing the value "0".

8.4.8.4 INVITE from PacketCable to PSTN

If MGC$_T$ receives an INVITE from CMS$_O$ with a Call-Info header declaring "purpose=answer_if_not_busy", then MGC$_T$ MUST take the following actions:

- Send a TCAP Initial Query message to the PSTN network, requesting the busy/idle status of the target, according to the Originating SPCS procedures defined in GR 227 [47].
- Upon receiving an initial TCAP response indicating that the target is idle according to the originating SPCS procedures defined in [47], immediately send an SS7 ISUP Initial Address Message to the PSTN target, placing a call to the target phone. Normal call processing ensues according to Section 8.4.1 of this document.
- Upon receiving an initial TCAP response indicating that the target is busy according to the originating SPCS procedures defined in [47], send a 486-Busy-Here response to the INVITE to CMS$_O$. The Contact header in the 486-Busy-Here response MUST contain a GRUU that identifies the active TCAP transaction within MGC$_T$. The GRUU MUST remain valid for the duration of the TCAP transaction. The CMS$_O$ procedures for processing the 486-Busy-Here response are described in section 8.4.8.2.
- Upon receiving a SUBSCRIBE message from CMS$_O$ with a request URI containing a GRUU that identifies the active TCAP transaction, send a SIP NOTIFY message to CMS$_O$ indicating that the current busy/idle status of the PSTN target endpoint. If the endpoint is busy, CMS$_O$ maintains the subscription waiting for the target to become idle, as described in section 8.4.8.3.

- After MGC_T has responded to the initial INVITE with a 486-Busy-Here message, and then upon receiving a TCAP response indicating that the target is now idle according to the Originating SPCS procedures defined in [47], MGC_T MUST send a SIP NOTIFY request to CMS_O for the active dialog-event subscription, indicating that the target is idle as specified in section 8.4.8.3. On receiving a notification that the target is now idle, CMS_O continues with the AC/AR procedures as defined in 8.4.8.3.
- If MGC_T receives a SUBSCRIBE request with an Expires header containing the value "0" from CMS_O, then it MUST terminate the subscription as specified in section 7.5, and cancel the TCAP transaction as specified in [47].

The PSTN could cancel the TCAP request with a Cancel Terminating Scanning message. If MGC_T receives a TCAP Cancel Terminating Scanning message, then it MUST terminate the subscription to the dialog event package by sending a NOTIFY to CMS_O indicating that the subscription is to be terminated as specified in 7.5.

In processing the dialog event package subscription from CMS_O, MGC_T MUST comply with [46]. In processing the TCAP Intersystem AR Signaling to scan the PSTN target's busy/idle status, MGC_T MUST comply with procedures in [47] for the originating SPCS.

8.4.8.5 Initial Query Request from PSTN to PacketCable

When the MGC receives a TCAP Initial Query message according to the terminating SPCS procedures specified in GR 227 [47], and functioning as CMS_O as specified in section 8.4.8.2 and 8.4.8.3, it MUST send:

- A SUBSCRIBE request to the target's dialog event package as specified in section 8.4.8.2. In this case, the request URI of the SUBSCRIBE request contains the URI representing the target subscriber identified in the received TCAP query.
- A TCAP Cancel Terminating Scanning message to the PSTN to cancel the AR/AC request, if the above initial SUBSCRIBE request is rejected.
- A TCAP message to the PSTN according to the terminating SPCS procedures specified in [47], indicating that the target is still busy, upon receiving a NOTIFY response that the target is still busy.
- A TCAP message to the PSTN according to the terminating SPCS procedures specified in [47], indicating that the target is now idle, upon receiving a NOTIFY response that the target is now idle.
- A SUBSCRIBE request to CMS_T with an Expires header containing the value "0" to terminate the subscription to the target endpoint's Dialog Event Package if it receives a TCAP Cancel Terminating Scanning message from the PSTN to cancel the AR/AC request due to timer expiration. In this case, the MGC MUST process the final NOTIFY request received from CMS_T for the terminating subscription as specified in [47].

In processing the dialog event package subscription toward the target endpoint, the MGC MUST comply with [46]. In processing the TCAP Intersystem AR/AC Signaling to scan the target endpoint's busy/idle status, the MGC MUST comply with [47]'s procedures for the terminating SPCS.

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.

8.4.10 SDP Interworking

CMSS compliant implementations need to support rules and procedures to allow for interworking between E-MTAs, MGs and UEs. This is due to the protocol differences between NCS/TGCP and SIP, and the use of different SDP attributes. Specifically:

- NCS/TGCP and SIP - NCS and TGCP are master/slave protocols that do not use the SDP Offer/Answer procedures defined in [37]. On the other hand, SIP does use Offer/Answer.
- SDP attributes - PacketCable 1.5 defines SDP attributes that are not used in PacketCable 2.0 and vice versa.
- Other protocol differences such as the use of connection information involving IP address 0.0.0.0, the use of media description including port 0, and the use of multiple media descriptions.

CMSS compliant implementations, as components that translate between NCS/TGCP and SIP, are therefore responsible for ensuring that sessions can still be established even with these differences. Following are specific interworking requirements.

While there are several bandwidth modifier values, PacketCable E-MTAs are required to send and receive only the bandwidth modifier with a value of "AS". In order to interoperate with elements that will use additional bandwidth modifier values, CMSS compliant implementations MUST convey only the bandwidth modifier (b=) of "AS" in SDP passed from CMSS to NCS/TGCP, unless provisioned to the contrary.There are some SDP interworking issues that are solved by configuration of CMSS compliant implementations.

If sending media security parameters in LCO attributes, CMSS compliant implementations must be configured to include NULL ciphersuites. When interworking with endpoints that are not E-MTAs, the RemoteConnectionDescriptor will be returned without a ciphersuites list, and the E-MTA will assume NULL ciphersuites. This allows a NULL intersection of ciphersuites to enable E-MTAs to interwork with non-E-MTAs.

CMSS compliant implementations MUST follow the procedures in [37]. CMSS compliant implementations MUST ensure interworking between the SIP interface, which does use offer/answer, and the NCS/TGCP interface, which does not. The following provides requirements for general offer/answer model compliance, handling hold scenarios, and managing the use of PacketCable 1.5 specific attributes that apply to CMSS implementations. In this section, the CMS that sends an offer and receives an answer is referred to as CMS_O, while the CMS that receives an offer and sends an answer is referred to as CMS_T.

CMSS compliant implementations MUST support receiving offers with more than one media descriptor. If the received SDP offer contains more than one media (m=) descriptor, CMS_T MUST forward the first audio descriptor with a non-zero port if present to MTA_T. If none of the media descriptors indicates audio, CMS_T MUST forward the first image/t38 descriptor with a non-zero port if present to MTA_T. Otherwise, CMS_T MUST reject the offer with the appropriate SIP response. When forwarding the modified SDP to MTA_T, CMS_T MUST remove all other media descriptions from the received SDP before forwarding to MTA_T. Additionally, CMS_T MUST store enough information about the received SDP to be able to construct a valid answer.

CMSS compliant implementations MUST support sending answers with more then one media descriptor. The SDP answer provided by CMS_T MUST contain the same number of media descriptors as the offer. CMS_T builds its answer based on the received LocalConnectionDescriptor (LCD) returned by the terminating endpoint and the information stored for the received SDP offer. If the LCD provided by the terminating endpoint does not contain the same number of media descriptors as was contained in the corresponding SDP offer, CMS_T MUST add the necessary corresponding media descriptors to the received LCD before sending the answer to CMS_O. For each added media descriptor, CMS_T MUST set the respective port number to zero and include at least one payload type.

Attribute descriptors associated with added media descriptors are not required. CMS_T MUST store enough information about the resulting SDP answer to be able to construct a valid SDP offer should a mid-dialog update be necessary. RFC 3262, [7], requires that an answer be included in the first reliable response after an offer is received. In the case of an offer being received in an INVITE and PRACK is being used, the answer supplied by CMST MUST be included in the 18x message. This means that CMS_T MUST wait until an LCD is received from the terminating endpoint before sending a 18x message with its SDP answer.

Several stimuli exist which can result in the CMS sending a new SDP offer within an existing dialog (e.g., transition of the endpoint to and from T.38 when the procedures of the FXR package are used, or placing a call on hold). When generating a new SDP offer within an existing dialog the following rules apply:

- The CMS SHOULD increment the version in the origin field of the o- line by one from the version in the previous offer;
- The SDP offer generated by the CMS MUST contain at least the same number of media descriptors as the previous SDP offer or answer provided by the CMS. For each added media descriptor which was not provided by the endpoint, the CMS MUST set the respective port number to zero and include at least one payload type. Attribute lines associated with added media descriptors are not required.

When the CMSS compliant implementation receives a new offer within an existing dialog, and this SDP is included in an RCD in an MDCX to an NCS or TGCP endpoint, and an LCD is not received in the 200 response, the CMSS compliant implementation MUST provide answer SDP on behalf of the NCS/TGCP endpoint. When sending the answer SDP on the CMSS interface, the following rules apply: If a stream is offered as sendonly, the CMSS compliant implementation MUST mark the corresponding stream as recvonly or inactive in the answer. If an offered media stream is listed as inactive, the CMSS compliant implementation MUST mark the corresponding stream as inactive in the answer. If an offered media stream is recvonly, the CMSS compliant implementation MUST mark the corresponding stream as sendonly or inactive in the answer.

8.4.10.1 Offer/Answer Requirements for Call Hold

Call hold can be invoked by either the local or remote endpoint. When invoked at the local endpoint, the CMS can choose whether it wants to update the session and inform the remote endpoint of the hold invocation. When invoked at the remote endpoint, the local CMS needs to be prepared to receive a re-INVITE with a number of different hold indicators. (Note, an existing session can also be updated using the SIP UPDATE method, so the procedures described here apply to both re-INVITE and UPDATE.)

When an MTA has had its connection placed in inactive mode or recvonly mode due to invoking call hold, the CMS MAY send a re-INVITE to the remote endpoint as described in section 8.4 of [37] and include an a=recvonly or a=inactive line as appropriate. If interworking with pre-RFC 3261 endpoints is desired, the CMSS compliant implementation MAY also include c=IN IP4 0.0.0.0 in the offer. The CMSS compliant implementation MUST be prepared to receive the corresponding answer. Upon receipt of the corresponding answer, the CMSS compliant implementation MAY choose not to convey the received answer SDP to the MTA.

Receipt of a re-INVITE that contains an SDP offer may signify a number of events. In order to determine whether the re-INVITE is related to a call hold event invoked by the remote endpoint, the CMSS compliant implementation MUST support the following hold indicators:

1. Media description (m=) with RTP port 0
2. Connection information (c=) with IP address 0.0.0.0
3. Stream Mode SDP Attributes (“a=inactive”, “a=sendonly”, “a=recvonly” and “a=sendrecv”)

Parsing SDP for these indicators will be limited to: the first session description; connection information (c=) and attribute lines (a=) at the session description level; and the connection information and attribute lines associated with the first audio media description. The standard precedence rules apply with respect to the media and session description levels. For requirements in the following sections pertaining to call hold, references to the stream mode

SDP attribute refer to the attribute with respect to the first audio media description regardless of whether a stream mode SDP attribute is specified at all (default “a=sendrecv”), or specified at the session-level, or specified at the media-level, or specified at both the media-level and session-level.

The cases addressed below assume that a SIP re-INVITE with an SDP offer to enable call hold is received for an existing connection that has an NCS/TGCP ConnectionMode of “sendrecv”. For completeness, the CMSS compliant implementation needs to address additional cases, such as the case where the existing connection has a NCS/TGCP ConnectionMode other than “sendrecv”, and the case where a similar SDP offer is received and no connection exists.

If a SIP re-INVITE offer SDP includes RTP port 0 in the m=audio line, the following requirements apply: Upon receipt of the SIP re-INVITE, the CMSS compliant implementation MUST send an NCS/TGCP ModifyConnection command with a ConnectionMode of "inactive". Unless provisioned to pass SIP re-INVITEs containing RTP port 0, the CMSS compliant implementation MUST NOT include a SIP re-INVITE offer SDP involving port 0 in the NCS/TGCP ModifyConnection command. When sending answer SDP, the CMSS compliant implementation MAY set the RTP port in the m=audio line to 0.

If a SIP re-INVITE offer SDP includes IP address 0.0.0.0, a non-zero RTP port in the m=audio line, and optionally the "a=sendonly" SDP attribute, the following requirements apply: The “a=recvonly” and “a=sendrecv” SDP attributes would be considered invalid in conjunction with an IP address of 0.0.0.0 and MUST be treated as “a=inactive” and “a=sendonly”, respectively. Upon receipt of the SIP Invite, the CMSS compliant implementation MUST send an NCS/TGCP ModifyConnection command with a ConnectionMode of “recvonly”. Unless provisioned to pass SIP re-INVITEs containing IP address 0.0.0.0, the CMSS compliant implementation MUST NOT include a SIP re-INVITE offer SDP involving IP address 0.0.0.0 in the NCS/TGCP ModifyConnection command. The CMSS compliant implementation MUST include an “a=recvonly” SDP attribute in the answer.

If a SIP re-INVITE offer SDP includes an IP address other than 0.0.0.0, a non-zero RTP port in the m=audio line and the “a=sendonly” SDP attribute, the following requirements apply: Upon receipt of the SIP Invite, the CMSS compliant implementation MUST send an NCS/TGCP ModifyConnection command with a ConnectionMode of “recvonly”. The CMSS compliant implementation MUST include an “a=recvonly” SDP attribute in the answer.

If a SIP re-INVITE offer SDP includes a non-zero RTP port in the m=audio line and the “a=inactive” SDP attribute, the following requirements apply: Upon receipt of the SIP Invite, the CMSS compliant implementation MUST send an NCS/TGCP ModifyConnection command with a ConnectionMode of “inactive”. The CMSS compliant implementation MUST include an “a=inactive” SDP attribute in the answer.

The cases addressed below assume that the NCS/TGCP ConnectionMode is “inactive” or “recvonly” at the time the SIP re-INVITE to disable call hold is received. In disabling call hold, the CMSS compliant implementation needs to take into consideration the NCS/TGCP ConnectionMode at the time call hold was enabled and other possible feature interactions.

If a received SIP re-INVITE offer SDP does not include stream mode SDP attributes (i.e., "a=sendonly", "a=recvonly", "a=inactive", and "a=sendrecv"), the CMSS compliant implementation MUST assume "a=sendrecv".

If a received SIP re-INVITE offer SDP includes an IP address other than 0.0.0.0, a non-zero RTP port in the m=audio line and the “a=sendrecv” or no stream mode SDP attribute, the following requirements apply: The CMSS compliant implementation MUST convey a SIP re-INVITE offer SDP using a NCS/TGCP ModifyConnection command with a ConnectionMode of “sendrecv”. Note that the SIP “a=recvonly” SDP attribute does not affect the receive characteristics of the NCS/TGCP endpoint. The CMSS compliant implementation could have an “awareness” regarding why the “receive” characteristics of the NCS/TGCP endpoint should be disabled in this situation. For example, with a previous ConnectionMode of “sendrecv” and a SIP “a=recvonly” SDP attribute, the CMSS compliant implementation could opt to modify the connection to “sendonly”. The CMSS compliant implementation could have a similar awareness with respect to “send” characteristics.

8.4.10.2 ptime and mptime SDP Interworking

PacketCable 1.5 MTAs and Media Gateways are mandated to send an a=mptime line in all LCD. However, this attribute line is not defined in ITU or IETF and is specific to PacketCable 1.0 and 1.5. Therefore, PacketCable 2.0 devices and non-PacketCable devices will not recognize this line. Further, PacketCable 1.5 MTAs and Media Gateways are not required to include the a=ptime line as defined in [3] or [64]. If this line is not included by a PacketCable 1.5 endpoint, the non-PacketCable 1.5 recipient of the SDP should assume that the default ptime for the codecs in the m=audio line is preferred. The default packetization times are defined in sections 4.2 and 4.5 of [65]. For most codecs, the default packetization is 20ms, but some codecs such as G.723 have different default packetization rates.

Therefore, the CMSS compliant implementation needs to be prepared for the situation where one of more of the following is occurring:

1. PacketCable 1.5 endpoint does not send a=ptime attribute line
2. Non-PacketCable 1.5 endpoint does not honor an a=ptime: line sent by a PacketCable 1.5 MTA or MG or CMSS interface
3. Non-PacketCable 1.5 endpoint interprets the absence of a=ptime attribute line as meaning the default packetization rate for the codec in use (generally 20ms)
4. Non-PacketCable 1.5 endpoint sends an a=ptime line
5. Non-PacketCable 1.5 endpoint does not send an a=ptime line

CMSS compliant implementations MUST add an a=ptime attribute if the PacketCable 1.5 MTA sends an mptime attribute with one value or multiple equal values (e.g., mptime: 20 20 20) and a ptime attribute is not present in the offer. In this case the value contained within the added a=ptime attribute MUST be equal to the value provided in the a=mptime attribute. CMSS compliant implementations MAY send the a=ptime attribute if the PacketCable 1.5 MTA sends an mptime attribute with multiple non-equal values (e.g., mptime: 10 20 30) if the a=ptime attribute is not present. If a CMSS compliant implementation sends the a=ptime attribute based on the presence of the mptime attribute, it MUST send the a=ptime attribute with a value present in the mptime attribute which is applicable to all listed codecs. There may be cases where a non-PacketCable 1.5 client answers with an SDP containing packetization rates that were not offered. CMSS compliant implementations SHOULD NOT specify a p: or mp: LocalConnectionOption that conflicts with the contents of the RCD.

8.4.11 MGC Caller ID Procedures

8.4.11.1 MGC Processing - Egress

The following closely follows [62].

When an MGC receives a SIP request destined for the PSTN, it translates the call signaling from SIP to SS7. The SS7 interface is defined in [63]. The ISUP call setup messages for circuit switch calls indicate presentation statuses of calling identity items in data fields known as presentation sub fields. A presentation subfield exists for each calling identity item.

Before constructing the SS7 messages, the MGC first determines the presentation status as follows:

- If a Privacy header is present in the SIP request, and contains a privacy value of "id," then the presentation status is "anonymous"; else
- If no P-Asserted-Identity headers are present in the SIP request then the presentation status is "unavailable"; else
- The presentation status is "public."

The MGC determines the value to be included in the presentation subfield of the CPN and GN parameters of the IAM. The format for the CPN parameter field is included in Appendix A.1 of [62].The format for the GN parameter field is included in Appendix A.2 of [62].

If the presentation status is "public," the MGC MUST encode the value "00" in the presentation subfield of the CPN parameter of the IAM to indicate that presentation of the caller's calling number is allowed.

If the presentation status is "anonymous," the MGC MUST encode the value "01" in the presentation subfield of the CPN parameter of the IAM to indicate that presentation of the caller's calling number is restricted.

If the presentation status is "unavailable" then no CPN parameter is included in the IAM.

Irrespective of the presentation status, the MGC MUST encode the value "11" meaning 'network provided' in the screening indicator subfield of the CPN parameter of the IAM if a CPN parameter is included.

The MGC MUST populate the Address Signals within a CPN parameter, if included, according to the following rules:

1. If a TEL URI is present in the P-Asserted-Identity header and contains a global-number (as defined in [28]) (if the first character is a "+") then the global-number-digits component shall be extracted and all visual-separator characters as well as the leading "+" character removed. Then all the remaining characters from the global-number-digits component shall be used to populate the address signals. The MGC SHOULD populate the value "001" meaning 'ISDN Telephony Numbering Plan (E.164)' in the numbering plan indicator subfield of the CPN parameter and the value "000 0100" meaning 'International Number' in the nature of address subfield of the CPN parameter. The MGC MAY perform a country code match, if so the country code is removed from the address signals and the value "000 0011" meaning 'National Number' populated in the nature of address subfield.
2. If a TEL URI is present in the P-Asserted-Identity header and contains a local-number (as defined in [28]) (if the first character is NOT a "+"), then the local-number-digits component shall be extracted from the URI and all visual-separator characters removed. Then all the remaining characters from the local-number-digits component shall used to populate the address signals. Population of the numbering plan indicator and nature of address subfields in this case is done via local rules on the MGC for the handling of non-global numbers.
3. If no TEL URI is present in the P-Asserted-Identity header, but there is a SIP or SIPS URI with a user part beginning with "+," the user part is used to determine the number field in accordance with the rules in step 1.

If a display name is available and is not a null string ("") the MGC MUST encode a GN parameter in the IAM to convey the presentation subfield information for the calling name. To prevent unnecessary TCAP queries when a display name is unavailable, the MGC SHOULD provide the GN only when the presentation status is "anonymous".

If the presentation status as determined according to Table 1 is "public" the MGC MUST populate the GN parameter as follows:

- "001" is in the "Type of Name" subfield in the GN parameter to indicate "calling name."
- "0" is in the "Availability" subfield in the GN parameter to indicate "name available or name availability unknown."
- "00" in the "Presentation" subfield in the GN parameter to indicate "presentation allowed."
- "Character" subfield are sent in the GN parameter. This is non-conformant to Telcordia [62] however provision of the name information in the SS7 messaging will reduce unnecessary TCAP queries.

If the presentation status is "anonymous" the MGC MUST populate the GN parameter as follows:

- "001" in the "Type of Name" subfield in the GN parameter to indicate "calling name."

- "0" in the "Availability" subfield in the GN parameter to indicate "name available or name availability unknown."
- "01" in the "Presentation" subfield in the GN parameter to indicate "presentation restricted."
- the "Character" subfields are not sent in the GN parameter.

These requirements are summarized in Table 1.

Table 1 - Egress PSTN Gateway Encoding Requirements

Pre-Conditions		**CPN Field Population**			**GN Field Population**	
Presentation Status	**Display Name Available**	**Presentation Indicator**	**Screening Indicator**	**Address Signals**	**Presentation Indicator**	**IA5 Characters**
Public	Yes	00 - Allowed	11 – Network Provided	As per the rules above.	00 - Allowed	Populated
Public	No	00 - Allowed	11 – Network Provided	As per the rules above	No GN Encoded	No GN encoded
Anonymous	Yes	01 - Restricted	11 – Network Provided	As per the rules above	01 - Restricted	No characters encoded
Anonymous	No	01 - Restricted	11 – Network Provided	As per the rules above	01 – Restricted	No characters encoded.

8.4.11.2 MGC Processing - Ingress

When a MGC receives an incoming ISUP IAM, it translates the call from Common Channel Signaling SS7 ISUP to SIP [63]. The ISUP call setup messages for circuit switch calls indicate presentation statuses of calling identity items in data fields known as presentation sub fields. A presentation subfield exists for each calling identity item.

Before constructing the SIP INVITE, the MGC first determines the presentation status as follows:

- If either of the CPN or GN parameters indicates that presentation is restricted then the presentation status is "anonymous."
- If the CPN screening indicator is any value other than "01 – User Provided, Screening Passed" or "11 – Network Provided" then the presentation status is "anonymous"
- If both the CPN and GN parameters indicate that presentation is allowed then the presentation state is "public"
- If a CNAM database query is required as determined by the GN population, the presentation status is based upon the results of the query.

A query to a CNAM database should only be launched by the ingress gateway if the presentation subfield is set to a value of '10 – Blocking Toggle'.

The ingress MGC requirements are summarized in Table 2.

Table 2 - Ingress PSTN Gateway Encoding Requirements

CPN Field Population	GN Field Population	MGC Action	Action Result	Determined Presentation Status	P-Asserted-Identity		Privacy
Presentation Indicator	Presentation Indicator				URI	Display	
00 - Allowed	00 - Allowed	Build INVITE		Public	From CPN	From GN if present	none
00 – Allowed	10 – Blocking Toggle	TCAP CNAM query	PPS = Anonymous	Public	From CPN	From TCAP CNAM query	none
00 - Allowed	10 – Blocking Toggle	TCAP CNAM query	PPS = Public	Anonymous	From CPN	From TCAP CNAM query	id
00 - Allowed	01 - Restricted	Build INVITE		Anonymous	From CPN	From GN if present	id
10 - Restricted	Any	Build INVITE		Anonymous	From CPN	From GN if present	id
Not Present	Any	Build INVITE		Unavailable	unknown@ unknown.invalid	""	none

The MGC determines the population of the P-Asserted-Identity and Privacy headers in the SIP INVITE based upon the presentation status determination.

If the presentation status is "public" then the MGC MUST populate the P-Asserted-Identity and Privacy headers as follows:

- A Privacy header with a value of "none".
- A P-Asserted-Identity header with a tel URI, or a SIP(S) URI with a user=phone parameter, constructed from the CPN received. If the nature of address in the received CPN is 'International Number' then the URI will contain a global-number as defined in [28], that is a '+' followed by the address signals from the CPN. For other received values of nature of address the URI is built according to local rules on the MGC that enable the qualification of the number via the use of phone-context or other mechanisms.
- If the presentation status is "public" and if a GN is received containing IA5 characters then a display name is included in the P-Asserted-Identity.
- If the presentation status is "public" and if the CPN is unavailable, the MGC populates the P-Asserted-Identity header with a URI of "sip:unknown@unknown.invalid", and a display name of "".

If the presentation status is "anonymous" then the MGC MUST populate the P-Asserted-Identity and Privacy headers as follows:

- A Privacy header with a value of "id".
- A P-Asserted-Identity header with a URI constructed from the CPN received.
- A display name in the P-Asserted-Identity with a value determined as specified in Table 2 above.

8.4.12 Procedures for Request History Information

RFC 4244 defines a standard mechanism for capturing the history information associated with a Session Initiation Protocol (SIP) request. This capability enables many enhanced services by providing the information as to how and why a call arrives at a specific application or user. RFC 4244 also defines a new SIP header, History-Info, for capturing the history information in requests. CMSS compliant implementations MUST support the History-Info header as described in [41].

8.4.12.1 Preventing Forwarding Loops and Limiting the Number of Forwarding Attempts

For each of the Call Forwarding features, the PacketCable network provides a mechanism to prevent forwarding loops. A call forwarding loop is defined to be the scenario that occurs when a targeted subscriber for a call forwards the call to another destination; if the forwarded-to destination also has call forwarding configured, the call can forward back (directly or indirectly) to the original targeted subscriber. When a loop is detected, the network node that performs the detection performs a configurable action, the default being call rejection. A CMSS compliant implementation MUST detect call forwarding loops.

The CMSS compliant implementation MUST support a configurable limit on the number of times an individual call may be subject to forwarding. If the number of forwarding attempts for a single call exceeds this limit, the CMSS compliant implementation MUST perform a configurable action, the default being call rejection.

The CMSS compliant implementation SHOULD support loop prevention and forwarding limit detection via the mechanisms described in this section. However, this mechanism alone may not be sufficient to detect loops when calls are forwarded to networks not supporting these mechanisms (for example, the PSTN or a network not supporting the required SIP headers). Therefore a CMSS compliant implementation MAY support additional loop prevention and forwarding limit detection methods as long as the requirements of forwarding limit restriction and loop detection are met.

If the CMSS compliant implementation supports the prevention of forwarding loops via analysis of the History-Info header present in the INVITE then it MUST compare the forward-to address with the set of targeted-to URI (hi-targeted-to-uri) entries from the History-Info header. If there is a match then a loop has occurred. If no History-Info header is present then it is not possible to perform loop detection via this mechanism.

If the CMSS compliant implementation determines that a loop has occurred (regardless of the loop detection method used), the CMSS compliant implementation MUST handle the call based upon a configurable action. The CMSS compliant implementation MUST support the default loop detection action of call rejection. The CMSS compliant implementation MUST send a final response appropriate to the type of call forwarding being performed if the call is to be rejected. If the forwarding action is CFBL and the call is to be rejected, the CMSS compliant implementation MUST respond with a 486-User-Busy message. If the forwarding action is CFDA and the call is to be rejected, the CMSS compliant implementation MUST respond with a 408-Request-Timeout message. If the forwarding action is CFV or SCF and the call is to be rejected, the CMSS compliant implementation MUST respond with a 480-Temporarily-Unavailable message.

If the CMSS compliant implementation supports the prevention of forwarding loops by enforcing a maximum number of forwarding attempts, then it MUST calculate the number of forwarding attempts by counting the number of entries in the History-Info header that have a nested Reason header which include a protocol-cause parameter and a reason-text parameter populated as defined in Section 8.4.12.2 If no History-Info header is present then it is not possible to determine the number of forwarding attempts via this mechanism.

If the number of forwarding attempts exceeds the configured limit then the CMSS compliant implementation MUST handle the call based on a configurable action. The CMSS compliant implementation MUST support the default action of call rejection. The CMSS compliant implementation MUST send a final response appropriate to the type of call forwarding performed if the call is to be rejected. If the forwarding action is Call Forward Busy Line and the call is to be rejected, the CMSS compliant implementation MUST respond with a 486-User-Busy message. If the forwarding action is Call Forward Don't Answer and the call is to be rejected, the CMSS compliant implementation MUST respond with a 408-Request-Timeout message. If the forwarding action is Call Forward Variable or Selective Call Forward and the call is to be rejected, the CMSS compliant implementation MUST respond with a 480-Temporarily-Unavailable message.

8.4.12.2 Setting of the Call Forwarding parameters

After checking the number of forwards, a number of fields of the INVITE request are set by the CMSS compliant implementation as defined in the following subsections.

8.4.12.2.1 First Forwarded INVITE

The absence of a History-Info header in the INVITE or the presence of a History-Info with no entries containing a nested Reason header with a protocol-cause parameter means that this is the first instance of a forwarding action.

When this is the first forward the INVITE has undergone, the following requirements apply. The CMSS compliant implementation MUST set the Request URI to the public user identity where the INVITE is to be forwarded. The CMSS compliant implementation MUST NOT change the contents of the P-Asserted-Identity header in the INVITE.

The CMSS compliant implementation MUST set the History-Info header (redirection information, redirecting number, original called number). Two History-Info entries MUST be generated as described below.

The first added entry in the History-Info header by the CMSS compliant implementation MUST include as the hi-targeted-to-uri the URI of the called party that was addressed with this INVITE. If the called party's presentation status is set to anonymous, the CMSS compliant implementation MUST escape the privacy header "history" within the hi-targeted-to-uri; a Reason is not added. If no History-Info header was previously present in the INVITE then the CMSS compliant implementation MUST set the Index to index=1. If this is an additional entry to an already present History-Info header then the CMSS compliant implementation MUST set the index according to the rules in [41].

The second added entry in the History-Info header by the CMSS compliant implementation MUST include as the hi-targeted-to-uri the address to where the INVITE is forwarded. If no History-Info header was present prior to this procedure then the CMSS compliant implementation MUST set the index to index=1.1. If this is an additional entry to an already present History-Info header then the CMSS compliant implementation MUST set the index according to the rules in [41].

When adding a second entry, the CMSS compliant implementation MUST also include a nested Reason header (redirecting reason and redirecting indicator) escaped in the History-Info header, this is populated according to the forwarding conditions.

[42] defines the mapping between the forwarding conditions and the coding of the protocol-cause parameter in the Reason header. The CMSS compliant implementation MUST populate the Reason header with a protocol-cause value of "486" and a reason-text value of "CFBL" when the forwarding condition is CFBL. The CMSS compliant implementation MUST populate the Reason header with a protocol-cause value of "408" and a reason-text value of "CFDA" when the forwarding condition is CFDA. The CMSS compliant implementation MUST populate the Reason header with a protocol-cause value of "302" and a reason-text value of "CFV/SCF" when the forwarding condition is CFV or SCF.

Finally, the CMSS compliant implementation MUST set the To header as per the following rules. If the forwarding party does not request privacy to be applied (that is, their presentation status is set to public) then the CMSS compliant implementation MUST NOT change the To header. In the case where the forwarding party requests privacy (that is, their presentation status is set to anonymous) the CMSS compliant implementation MUST change the To header to be the URI to where the INVITE is forwarded.

8.4.12.2.2 Second or Subsequent Forwarded INVITE

When this is the second or subsequent forwarding of the INVITE, the CMSS compliant implementation MUST add a new entry to the History-Info header according to the rules defined in [41]. If the history entry representing the forwarding party already contains the correct privacy value for the forwarding party (in an escaped privacy header) then the CMSS compliant implementation MUST NOT modify the History-Info header. Otherwise if the forwarding party requests privacy (that is, their presentation status is set to anonymous) the CMSS compliant implementation MUST ensure the privacy header "history" is escaped within the hi-targeted-to-uri.

The entry MUST contain as the hi-targeted-to-uri the address to where the INVITE is forwarded. The CMSS compliant implementation MUST populate the Reason header (redirecting reason and redirecting indicator) escaped in this history-info header according to the forwarding conditions and notification subscription option.

The CMSS compliant implementation MUST code the protocol-cause parameter in the Reason header based on forwarding conditions as defined in [42]. The CMSS compliant implementation MUST populate the Reason header with a protocol-cause value of "486" and a reason-text value of "CFBL" if the forwarding condition is CFBL. The CMSS compliant implementation MUST populate the Reason header with a protocol-cause value of "408" and a reason-text value of "CFDA" if the forwarding condition is CFDA. The CMSS compliant implementation MUST populate the Reason header with a protocol-cause value of "302" and a reason-text value of "CFV/SCF" if the forwarding condition is CFV or SCF. The CMSS compliant implementation MUST set the Request URI to the public user identity where the INVITE is to be forwarded.

The CMSS compliant implementation MUST NOT change the contents of the P-Asserted-Identity header in the INVITE.

The CMSS compliant implementation MUST NOT change the To header field (original destination number) if the forwarding party does not request privacy (that is, their presentation status is set to public). The CMSS compliant implementation MUST change the To header field to be the URI where the INVITE is forwarded if the forwarding party requests privacy (that is, their presentation status is set to anonymous).

Figure 23 illustrates a multiple forwarding scenario, and Table 3 describes how the headers are populated. In this scenario, Alice calls Bob and is forwarded to Charlie, and is then forwarded to Ed and is in turn forwarded to Bob.

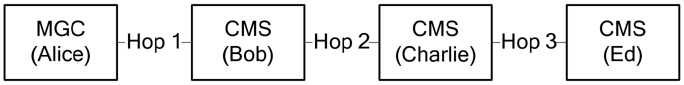


Figure 23. Multiple forwarding scenario

Table 3 - Parameters and Header Fields that are Modified In a CMSS Compliant Implementation

Element	**MGC (Alice)**	**CMS (Bob)**
H-I Index Added	1	1.1,1.1.1
Entry Added	Bob@domain	Bob@domain, Charlie@domain?Reason=SIP;protocol-cause=302;reason-text="CFV/SCF"
H-I Header	Bob@domain; index=1	Bob@domain; index=1, Bob@domain; index=1.1, <Charlie@domain?Reason=SIP;protocol-cause=302;reason-text="CFV/SCF">; index=1.1.1
Element	CMS (Charlie)	CMS (Ed)
H-I Index Added	1.1.1.1	No entry is added since loop detection at Ed's CMS detects a Call Forwarding loop. The forwarding target Bob@domain is already present in the set of URIs

Element	MGC (Alice)	CMS (Bob)
		contained in the received H-I header.
Entry Added	Ed@domain?Reason=SIP;protocol-cause=302;reason-text="CFV/SCF"	None as loop detection detects CF loop
H-I Header	Bob@domain; index=1, Bob@domain; index=1.1, <Charlie@domain?Reason=SIP;protocol-cause=302;reason-text="CFV/SCF">; index=1.1.1, <Ed@domain?Reason=SIP;protocol-cause=302;reason-text="CFV/SCF">; index=1.1.1.1	None as loop detection detects CF loop

8.4.12.3 MGC Inter-working

To prevent forwarding loops occurring between the PacketCable network and the PSTN, forwarding information needs to be maintained by the MGC when mapping from SIP to ISUP and vice versa. Given that all of the applicable ISUP parameters required are optional rather than mandatory in an ISUP IAM or ACM message, the procedures detailed here are only effective if support for the necessary parameters is enabled on the ISUP facility selected by the MGC.

8.4.12.3.1 MGC Processing – Egress

When a MGC receives a SIP request destined for the PSTN, it translates the call from SIP to Common Channel Signaling SS7 ISUP, as specified in [63]. The ISUP IAM that is mapped from a received SIP INVITE can optionally carry call redirection information.

If the received SIP INVITE contains a History-Info header indicating that the call has been forwarded (header entries populated following the rules defined in Section 8.4.12.2), then the MGC MUST populate the Original Called Number, Redirecting Number and Redirection Information parameters of the ISUP IAM based on information determined from the History-Info header.

If the selected ISUP facility supports the Original Called Number (OCN) parameter then the Original Called Number (OCN) field MUST be populated according to the following rules:

- Upon detection of the first entry in the History-Info header which has a nested Reason header containing both a protocol-cause parameter and a reason-text parameter populated as defined in Section 8.4.12.2, the entry at the index value immediately prior to that is the candidate entry for population of the OCN parameter. If this entry contains a TEL URI then the MGC determines whether this is a global-number or local-number as defined by [28]. If the first character is a "+" then it is a global-number, otherwise it is a local-number. If the MGC determines that it is a global-number, then it MUST extract the global-number-digits component and remove all visual-separator characters as well as the leading "+" character. Then the MGC MUST use all the remaining characters from the global-number-digits component to populate the address signals. The MGC MUST populate the value "001" meaning 'ISDN Telephony Numbering Plan (E.164)' in the numbering plan indicator subfield of the OCN parameter. If the MGC has the digit analysis capabilities to recognize and then format the number as a nationally significant number then it MAY remove any identified country code from the characters used to populate the address signals and populate the value "000 0011" meaning 'National Number' in the nature of address subfield of the OCN parameter. If no such digit analysis capabilities exist or are not applied then the MGC MUST populate the value "000 0100" meaning 'International Number' in the nature of address subfield of the OCN parameter.

- The History-Info header entry found at the index value immediately prior to the first entry which has a nested Reason header containing both a protocol-cause parameter and a reason-text parameter populated as defined in Section 8.4.12.2 is the candidate entry for population of the OCN parameter. If this entry contains a TEL URI then the MGC determines whether this is a global-number or local-number as defined in [28]. If the first character is NOT a "+", then the number is determined to be a local-number. If the MGC determined that this is a local number, then it MUST extract the local-number-digits component from the URI and remove all visual-separator characters. Then the MGC MUST use all the remaining characters from the local-number-digits component to populate the address signals. The MGC MUST populate the numbering plan indicator and nature of address subfields according to local rules on the MGC for the handling of non-global numbers.
- The History-Info header entry found at the index value immediately prior to the first entry which has a nested Reason header containing both a protocol-cause parameter and a reason-text parameter populated as defined in Section 8.4.12.2 is the candidate entry for the population of the OCN parameter. If this entry does not contain a TEL URI, but there is a SIP URI with a user part beginning with "+," the MGC MUST use the user part to determine the number field in accordance with the rules defined for case when a TEL URI with a global-number is present.

The Original Called Number Screening Indicator field MUST be set to "11" meaning 'network provided'

The Original Called Number Presentation Indicator is set based on whether a nested Privacy header is present. If a nested Privacy header is present then the MGC MUST set the Presentation Indicator to "01" meaning that the presentation is restricted. If a nested Privacy header is not present then the MGC MUST set the Presentation Indicator to "00" meaning that presentation is allowed.

If the selected ISUP facility supports the Redirection Information parameter then the fields within this parameter are populated as follows:

The MGC MUST set the Redirection Counter to the number of entries in the History-Info header which contain a nested Reason header that has a Protocol-cause parameter.

- The Original Redirecting Reason is mapped from the Protocol-cause and Reason-text parameters found in the nested Reason header at the first index entry where such an encoding exists (typically index value index=1.1). If the Protocol-cause is "486" and Reason-text is "CFBL" then the MGC MUST use a value of "0001" meaning 'user busy'. If the Protocol-cause is "408" and Reason-text is "CFDA" then the MGC MUST use a value of "0010" meaning 'no reply'. If the Protocol-cause is "302" and Reason-text is "CFV/SCF" then the MGC MUST use a value of "0011" meaning 'unconditional'. Other combinations are not indicative of call forwarding and so the MGC MUST NOT populate the ISUP parameters.
- The Redirecting Reason is mapped from the Protocol-cause and Reason-text parameters found in the nested Reason header at the final indexed entry in the History-Info header where such an encoding exists. If the Protocol-cause is "486" and Reason-text is "CFBL" then the MGC MUST use a value of "0001" meaning 'user busy'. If the Protocol-cause is "408" and Reason-text is "CFDA" then the MGC MUST use a value of "0010" meaning 'no reply'. f the Protocol-cause is "302" and Reason-text is "CFV/SCF" then the MGC MUST use a value of "0011" meaning 'unconditional'. Other combinations are not indicative of call forwarding and so the MGC MUST NOT populate the ISUP parameters.
- Of the set of entries in the History-Info header that contain both Protocol-cause and Reason-text parameters populated as defined in Section 8.4.12.2 in a nested Reason header entry, the MGC MUST populate the Redirecting Number parameter from the penultimate History-Info header entry in this set. The penultimate entry is used since the final entry that contains both Protocol-cause and Reason-text parameters is the target of the forwarding attempt (the called party for this leg); the entry preceding that is the previous target that itself forwarded and is therefore the Redirecting Number.

If the set of entries in the History-Info header that contains both Protocol-cause and Reason-text parameters populated as defined in Section 8.4.12.2 in a nested Reason header entry has only one item, a single forwarding has occurred. In this case the Redirecting Number parameter MUST be populated with the same information as the Original Called Number parameter; that is the information from index value index=1.

If the selected ISUP facility supports the Redirecting Number parameter then the fields within this parameter are populated as follows:

- The Redirecting Number parameter is populated using the penultimate entry in the History-Info header which contains a Protocol-cause parameter in a nested Reason header, unless only a single forwarding has taken place then the History-Info entry used is that with an index value of index=1. The MGC MUST populate the Address Information parameter of the Redirecting Number parameter using this History-Info entry according to the rules defined for the Address Information portion of the Original Called Number.
- The MGC MUST set the Screening Indicator to "11" meaning 'network provided';
- The Presentation Indicator is set based on the presence or absence of a nested Privacy header. If a nested Privacy header is present then the MGC MUST set the Presentation Indicator to "01" meaning that the presentation is restricted. If the Privacy header is not present then the MGC MUST set the Presentation Indicator to "00", meaning that presentation is allowed.

8.4.12.3.2 MGC Processing – Ingress

When a MGC receives an ISUP IAM, it translates the call from Common Channel Signaling SS7 ISUP to SIP, as specified in [63]. The IAM can optionally carry call redirection information.

If the received ISUP IAM contains the Original Called Number, Redirection Information and Redirecting Number parameters then the MGC MUST populate a History-Info header within the SIP INVITE. The History-Info needs to reflect as accurately as possible the call forwarding information associated with this call so that accurate determinations of loops which go between PacketCable and PSTN subscribers can be determined as well as the ability to accurately limit the number of forwarding attempts that take place.

The MGC MUST construct the History-Info header with at least two history entries according to the following rules:

- The first entry MUST include a hi-targeted-to-uri built from the contents of the Original Called Number parameter. This is a tel URI or a SIP URI with a user=phone parameter. If the nature of address in the received OCN is 'International Number' then the URI will contain a global-number as defined in [28], that is a '+' followed by the address information from the OCN. For other received values of nature of address the URI is built according to local rules on the MGC that enable the qualification of the number via the use of phone-context or other mechanisms. If the user wishes privacy (for example, the Presentation Indicator in the OCN is set to the value "01") the MGC MUST escape the privacy header "history" within the hi-targeted-to-uri. The MGC MUST NOT add a Reason. The MGC MUST set the Index of this first entry to index=1.
- A second entry MUST be added with the hi-targeted-to-uri built from the contents of the Redirecting Number parameter. This is a tel URI, or a SIP URI with a user=phone parameter. If the nature of address in the received Redirecting Number is 'International Number' then the URI will contain a global-number as defined in [28], that is a '+' followed by the address information from the Redirecting Number. For other received values of nature of address the URI is built according to local rules on the MGC that enable the qualification of the number via the use of phone-context or other mechanisms. If the user wishes privacy (for example, the Presentation Indicator in the Redirecting Number is set to the value "01") the MGC MUST escape the privacy header "history" within the hi-targeted-to-uri. The MGC MUST set the Index of this second entry to index=1.1.
- A nested Reason header is added to the second History-Info header with both Protocol-cause and Reason-text parameters populated based on mapping contents of the Redirection Information parameter's Redirecting Reason field. If Redirecting reason has a value of "0001" meaning 'user busy', the MGC MUST use a Protocol-cause of "486" and Reason-text of "CFBL". If Redirecting reason has a value of "0010" meaning 'no reply', the MGC MUST use a Protocol-cause of "408" and Reason-text of "CFDA". If Redirecting reason has a value of "0011" meaning 'unconditional', the MGC MUST use a Protocol-cause of "302" and Reason-text of "CFV/SCF". Otherwise the MGC MUST use a Protocol-cause of "302" and Reason-text of "CFV/SCF".

- If the Redirection Counter within the Redirection Information has a value of '1' then population of the History-Info is complete. If the Redirection Counter within the Redirection Information has a value greater than '1' then the MGC MUST add duplicate versions of the second History-Info entry (as defined in the bullets above); and MUST be added to the History-Info header until the number of entries is equal to the Redirection Counter. That is for a Redirection Counter of 'n' the final entry in the History-Info header has an index value of index=1.n.

8.4.13 Operator Services

Operator Services (Busy Line verification and Emergency Interrupt) are initiated from the MGC on behalf of a PSTN gateway connecting to special MF trunks groups from the OSPS system. The SIP messages SUBSCRIBE(Dialog Event) and INVITE(BLV) are initiated by the MGC after performing a basic call to the line being verified. The SUBSCRIBE is for the Dialog Event package described in 7.16. The subsequent INVITE results from the NOTIFY sent by the CMS. The INVITE includes a Join header for the first dialog reported in the NOTIFY.

CMS MUST be prepared to receive a SUBSCRIBE for the Dialog Event package at any time, and INVITE(BLV) following that. If not received from another CMS by security procedures specified in [26], the messages SHOULD be rejected.

The MGC sends a SUBSCRIBE for the dialog event package with an expires value of zero to the CMS with a URI of the targeted public ID to verify. The P-Asserted Identity in the SUBSCRIBE message MUST be the reserved operator ID as defined below.

SUBSCRIBE(Dialog Event) Headers:	Requirements On CMS for Message Checking
SUBSCRIBE URI SIP/2.0	MUST be present. MUST contain the URI of the target busy line.
P-Asserted-Identity	MUST be present. MUST be set to BLV Operator Identity.
Expires	Must be present. MUST be set to 0.
--all other headers, including SDP---	MUST be as specified for SUBSCRIBE.

The CMS MUST respond to a SUBSCRIBE to the dialog event package from the MGC with a 200-OK followed by a NOTIFY message with a list of established dialogs for the line being verified. The CMS SHOULD list the active dialog first. The headers required for this NOTIFY are defined below.

NOTIFY(Dialog Event) Headers:	Requirements On MGC for Message Checking
NOTIFY URI SIP/2.0	MUST be present. MUST contain the contact address 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 "dialog" as defined in 7.16.
Subscription-State:	As described in 7.5.

NOTIFY(Dialog Event) Headers:	Requirements On MGC for Message Checking
Content-Type:	MUST contain "application/dialog-info+xml" as defined in 7.5 and 7.16.
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 7.16

Upon receiving the NOTIFY identifying the active dialogs, the MGC MUST send an INVITE to the CMS with a Join header containing the first dialog ID in the NOTIFY. The MGC MUST encode the INVITE as follows:

INVITE(BLV) (MGC->CMS) Header:	Requirements On CMS for Message Checking
INVITE URI SIP/2.0	MUST be present.
	MUST contain the URI of the target busy line. MUST be present, if a dialog was returned in the NOTIFY from CMS.
Join	
P-Asserted-Identity	MUST be present. MUST be set to BLV Operator Identity.
--all other headers, including SDP---	MUST be as specified for INVITE.

On receiving the INVITE(BLV) with Join header, the CMS MUST NOT alert the user. Instead, the CMS MUST respond with a 183-Session-Progress to the INVITE(BLV) with Join header and complete the call as described in Sections 8.4.1.2.1, 8.4.1.4, and 8.4.1.7.

In order to emulate the PSTN busy-line-verify procedures, the MGC can include a mode attribute of "a=sendrecv" in the SDP offer of the INVITE(BLV). On receiving an INVITE(BLV) containing an SDP offer with "mode=sendrecv", the CMS SHOULD configure the OSPS connection and the active call connection into a conference mode, so that the OSPS operator can hear a mix of the send and receive audio on the active connection, and the target user can hear the operator (if the operator decides to barge-in).

The CMS MAY reject an INVITE with Join header if there are insufficient resources to perform audio mixing. The answer SDP sent by the CMS MAY contain "a=sendonly" to preserve existing behavior when the line being verified is known to have an active TDD, FAX or Modem call.

If the telephone line is in a state inconsistent with the contents of the Join header in the INVITE, the CMS MUST reject the INVITE with 481-Call-Transaction-Does-Not-Exist. The CMS SHOULD reject the INVITE with 488-Not-Acceptable for all other failures. When the MGC receives a 481-Call-Transaction-Does-Not-Exist or 488-Not-Acceptable response to the INVITE with Join header it MUST play reorder tone to the OSPS.

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:

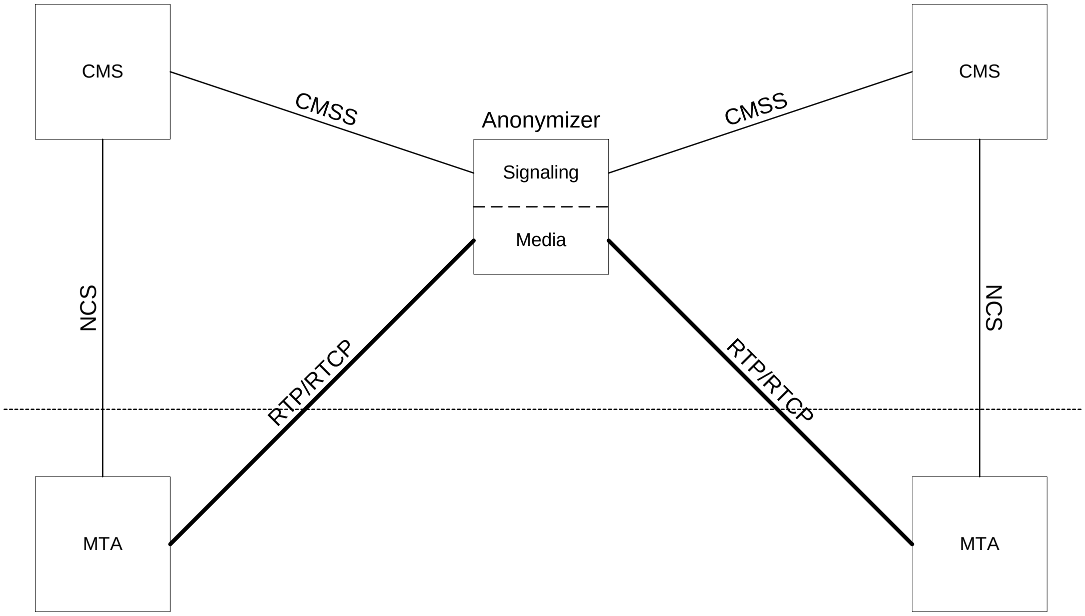


Figure 24. Application Layer Anonymizer

The anonymizer provides three Privacy functions:

Signaling content Privacy
Signaling IP address Privacy
Media IP address Privacy

all of which are illustrated in Figure 25.

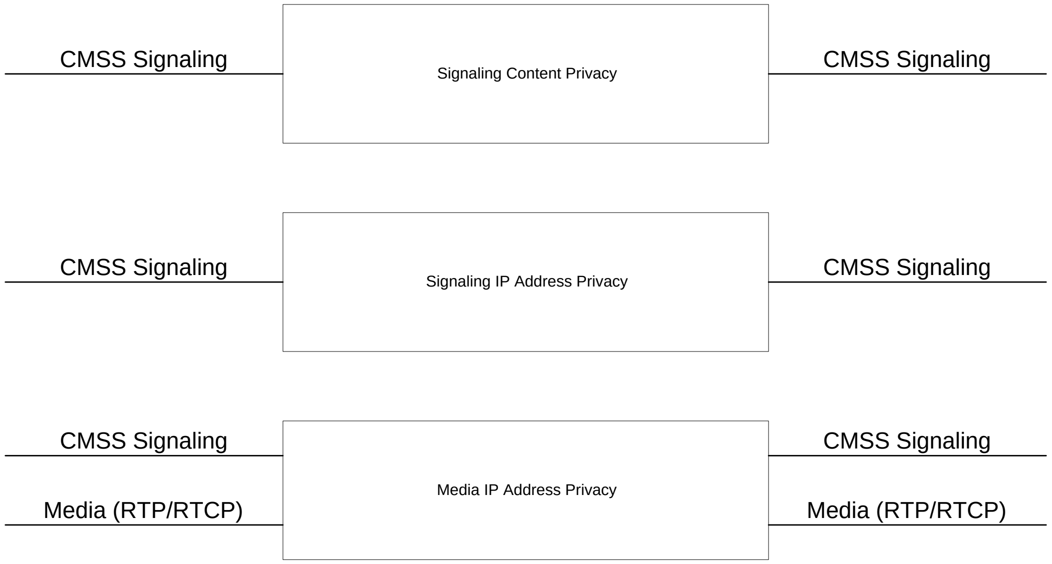


Figure 25. Anonymizer Functions.

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 26 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.

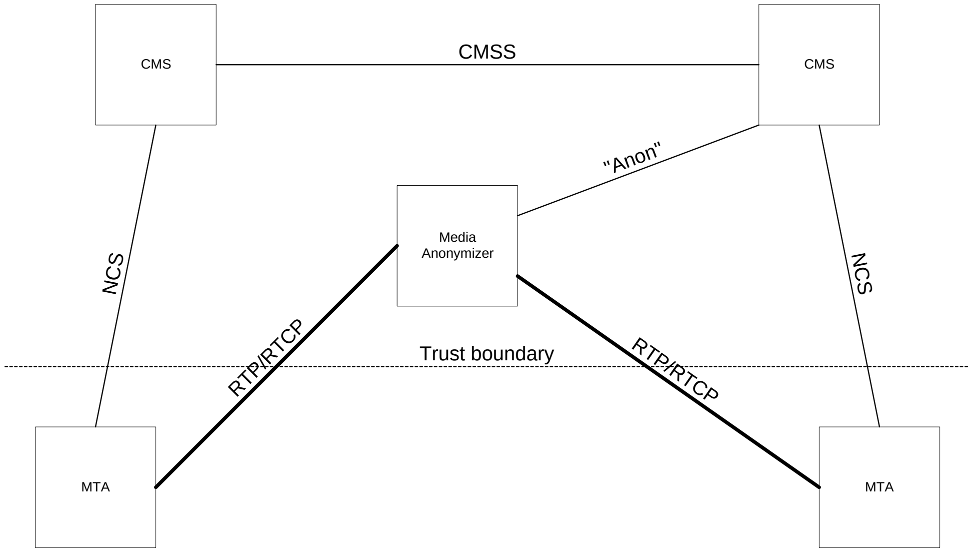


Figure 26. Privacy Issues

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 tables below.

Timer Label	Approximate Duration	Timer Description
Session timers (T3) at originating CMSs and CMSs		
T-setup	5 to 6 minutes	Timer between receiving a provisional response to an INVITE and receiving a 200-OK final response. If T-setup expires before receiving the ring or final response, the CMS sends a CANCEL and aborts the call attempt.
Session timers (T3) at terminating CMSs and CMSs		
T-ringing	3 to 4 minutes	Timer between receiving an INVITE request and connect. If T-ringing expires before connect, the CMS sends a 480-Temporarily-Unavailable response and releases the reserved resources, or invokes features such as call forwarding no-answer.

Appendix 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

Header	Send	Recv	Reference and Notes
Content-Encoding	O	M	See Section 6.20.12
Content-Language	O	M	See Section 6.20.13
Content-Length	M	M	See Section 6.20.14
Content-Type	M	M	See Section 6.20.15 The values "application/sdp", "message/sipfrag", and "application/simple-message-summary" MUST be supported.
CSeq	M	M	See Section 6.20.16
Date	O	O	See Section 6.20.17
Error-Info	O	O	See Section 6.20.18
Expires	M	M	See Section 6.20.19 and Section 7.5
From	M	M	See Section 6.20.20
In-Reply-To	O	O	See Section 6.20.21
Max-Forwards	S	M	See Section 6.20.22
Min-Expires	O	O	See Section 6.20.23
MIME-Version	O	M	See Section 6.20.24
Organization	O	O	See Section 6.20.25
Priority	O	O	See Section 6.20.26
Proxy-Authenticate	O	O	See Section 6.20.27
Proxy-Authorization	O	O	See Section 6.20.28
Proxy-Require	M	M	See Section 6.20.29 The option tag "Privacy" MUST be supported in accordance with Section 7.9
Record-Route	M	M	See Section 6.20.30
Reply-To	O	O	See Section 6.20.31
Require	M	M	See Section 6.20.32. The option tags "precondition", "replaces", and "100rel" MUST be supported. Furthermore, the option tag "P-DCS" MAY be sent and MUST be supported if received as described in Section 7.7.4.
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

Header	Send	Recv	Reference and Notes
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.
History-Info	M	M	See Section 8.4.12.
Target-Dialog	O	M	See Section 7.14.
Join	M	M	See Section 7.15

Appendix C ENUM Client Requirements

C.1 Introduction

The PacketCable ENUM Client is responsible for taking as an input an E.164 phone number (with country code) (e.g., +1-301-555-1212), an optional service selector, an optional database selector and an optional count of desired URIs and returning one or more URIs. The service selector is used by the ENUM client for filtering NAPTR responses as described in Section C.2.1. The database selector is used by the ENUM client to identify the portion of the DNS Tree where the ENUM data resides, its use is described in Section C.1.3.

C.1.1 General

This section describes the ENUM requirements necessary for usage by CMS, MGC and other PacketCable component implementations. ENUM is defined in RFC 3761 [30] and unless otherwise prescribed below, is required for the implementation of a PacketCable ENUM client [hereinafter "Client"].

The processing of the final output of an ENUM lookup up (e.g., a SIP or TEL URI) is outside the scope of this document.

Clients MUST be in accordance with the following ENUM related RFCs: RFC 3761 [30], 3764 [31], 4415 [32], and 3402 [33] except as noted in the following sections.

C.1.2 DNS Resolver Requirements

The Client, MUST implement a "stub" resolver as defined in RFC1034 [34]. The Client SHOULD implement a 'full' resolver per RFC1034 [34]. If the Client implements a full resolver, it MUST be configurable to act as a stub resolver - e.g., all ENUM queries are forwarded to an external resolver.

Since the data used for ENUM can reside in a DNS system that is partially or totally independent of an operator's DNS system used for other applications (e.g., email, web, etc) the ENUM client may need to be configured accordingly. As such, the ENUM client resolver MUST be capable of being configured to forward ENUM queries to an external DNS Server which may differ from the DNS Server used by other applications or processes on the system on which the ENUM Client resides.

C.1.3 Database Selector

The default database selector is the RFC3761 [30] default database - "e164.arpa". For the purposes of this specification, step 4 is modified to read as follows: "Append the database selector string (or if unspecified, the default string '.e164.arpa') to the end. Example: 8.4.1.0.6.4.9.7.0.2.4.4.e164.arpa or 2.1.2.1.5.5.5.1.0.3.1.enum.mso.net".

Valid database selectors MUST be specified using the recommended syntax from RFC1034 [34], i.e., labels consist only of letters, digits and hyphens. Clients MUST NOT send queries with database selectors which violate this syntax.

C.2 NAPTR

C.2.1 Service Field

A service selector matches any service field that begins with that string, e.g., "E2U" matches any service field beginning with "E2U". "" (blank) matches any service field.

The default selector for the service field is "E2U". Other possible values for the service field selector that can be requested of the client include "E2U+voice:tel" and "E2U+SIP". During ENUM processing, Clients MUST silently ignore returned NAPTR records where the selector does not match the service field.

Clients MUST be able to receive and process the "E2U+SIP" NAPTR service records as defined in [31].

Clients MUST be able to receive and process the "E2U+voice:tel" NAPTR service records as defined in [32].

Client processing of NAPTR records with other service types is outside the current version of this specification.

C.2.2 Case Insensitivity

For the purposes of selecting and matching records, the Client MUST match with the Service and Flags fields of the NAPTR records in a case-insensitive manner, e.g., "E2U+SIP", "E2u+sIP" and "e2u+sip" are all values which indicate an ENUM SIP NAPTR record.

The client MUST do a case insensitive match with the left hand side of the NAPTR Regular Expression.

Case insensitivity is defined only with respect to the 26 uppercase and 26 lower case alphabetic characters of the ASCII character set (e.g., A-Z and a-z) – "A" is equivalent to "a", etc. All other characters, regardless of character set, require an exact match.

C.2.3 Regular Expression Delimiters

Clients MUST be able to process the regular expression field of the NAPTR record for any valid delimiter as defined in [33]. While the preferred delimiter is the exclamation point (!), this specification does not impose requirements on the servers which serve the NAPTR records and clients should be prepared to process any valid delimiter they receive. The Client MUST ignore NAPTR records which contain malformed regular expressions. When ignoring malformed NAPTR records the Client MUST continue processing with the next available NAPTR record if any.

C.2.4 Non-Terminal NAPTR Records

At this time, PacketCable only supports NAPTR records with a flag field of "U". The flag indicates a "terminal" or final lookup record with the result being a URI. Clients MUST silently ignore any NAPTR record which contains a flag field with any value other than "U" (or its lower case equivalent "u").

C.2.5 Handling Multiple URIs

Differing from RFC3761, an ENUM client MAY return multiple URIs. Unless specified by the application, the Client MUST assume the number of desired URIs is 1. Consistent with RFC3402, when presented with multiple returned NAPTR records which match the service selector, the Client MUST follow this procedure:

1. Remove from consideration all non-terminal NAPTR records.
2. Sort the NAPTR records first by the "Order" field, and then by the "Preference" field within the "Order". The sort order for two records with identical Order and Preference fields is undefined and may be ordered any way the Client application desires.
3. If there are more than 10 NAPTR records remaining, remove the excess records at the bottom of the sorted list.
4. In sort order, attempt to match (see [33]) each NAPTR record. If the match is successful, and the returned string is a valid URI, add that URI to the result set along with the NAPTR's associated Order, Preference and Service fields.

5. Repeat step 4 until the end of the list is reached or until the number of desired URIs is produced. In no event shall the Client produce more than 5 URIs.
6. Return the result set of URIs along with their associated Service, Order, and Preference fields.

C.2.6 EDNS0 Support

Clients MUST implement EDNS0 as defined by [35]. Clients MUST include an EDNS0 OPT record in any UDP DNS request they send. The Client MUST set the OPT record to indicate a Sender UDP Payload Size of at least 4096 octets, which, in turn, implies that the Client MUST be able to receive and process a UDP payload of at least 4096 octets.

Appendix D 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 [53] and much of the text in 7.10 came from [60], which contained the following copyright notice:

Appendix E 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

The following ECNs have been incorporated into PKT-SP-CMSS1.5-I03-061228.

ECN	ECN Date	Summary
CMSS1.5-N-06.0337-3	9/11/06	Requirements on the CMS to support intercept with release 2 networks (CSCFs)
CMSS1.5-N-06.0342-3	9/18/06	ENUM Client Requirements
CMSS1.5-N-06.0355-2	9/18/06	Populate Contact header with GRUU
CMSS1.5-N-06.0356-3	9/18/06	Update draft references to latest draft/RFC
CMSS1.5-N-06.0357-4	9/18/06	Change preconditions from e2e to segmented
CMSS1.5-N-06.0359-4	9/18/06	Updates to the 1.5 CMSS Specification to Enable SDP Interworking
CMSS1.5-N-06.0360-2	9/18/06	Clarify CMSS specification to allow for interworking with IMS CSCFs
CMSS1.5-N-06.0365-3	9/25/06	MGC Caller ID ISUP Interworking
CMSS1.5-N-06.0366-3	9/25/06	Early Media Enhancements
CMSS1.5-N-06.0367-3	9/25/06	Addition of support for History-Info loop detection to CMSS
CMSS1.5-N-06.0368-4	9/25/06	Addition of Out of Dialog REFER Support to CMSS
CMSS1.5-N-06.0369-4	9/25/06	Operator Services
CMSS1.5-N-06.0370-1	9/18/06	Extension of CMSS to handle PC 2.0 generated response codes
CMSS1.5-N-06.0372-4	9/25/06	In Support of Dialog Event Package for RST Features
CMSS1.5-N-06.0373-4	9/25/06	Change emergency service procedures to align with PacketCable 2.0

The following ECNs have been incorporated into PKT-SP-CMSS1.5-I04-070412.

ECN	ECN Date	Summary
CMSS1.5-N-07.0400-3	3/12/07	Update and correct references in CMSS
CMSS1.5-N-07.0405-3	3/12/07	SIPS removal EC
CMSS1.5-N-07.0407-5	3/26/07	SIP-NCS/TGCP SDP interworking rules added to CMSS Offer/Answer rules

NOTE: PKT-SP-CMSS1.5-I04-070412 reposted May 23, 2007 with the following editorial-only corrections:

ECN	Date	Summary
N/A	5/23/07	Update to correct an error incurred in publication. Correction of heading numbers 8.4.11, 8.4.11.1 and 8.4.11.2 only as specified in I03.