

PacketCable™ Line Control Signaling System Architecture Technical Report

PKT-TR-ARCH-LCS-V01-010730

Notice

This PacketCable Technical Report 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.

© Copyright 2001 Cable Television Laboratories, Inc.

All rights reserved.

Document Status Sheet

Document Control Number: PKT-TR-ARCH-LCS-V01-010730

Document Title: PacketCable™ Line Control Signaling System
Architecture Technical Report

Revision History: V01-010730: Release

Date: July 30, 2001

Contents

1	INTRODUCTION	1
1.1	Purpose	1
1.2	Scope.....	1
1.2.1	PacketCable Reference Architecture	2
1.2.2	PacketCable LCS Reference Architecture	3
2	REFERENCES	4
3	ABBREVIATIONS AND ACRONYMS.....	5
4	SYSTEM ARCHITECTURE.....	7
4.1	Line Control Signaling System Architecture.....	7
4.2	IPDT / LDS Interworking: the GR-303 Interface.....	8
4.3	Migration to Full VoIP	9
5	SYSTEM COMPONENTS	10
5.1	Local Digital Switch	10
5.2	IP Digital Terminal.....	11
5.3	Cable Modem Termination System.....	12
5.4	Embedded MTA.....	12
5.5	Operations Support Systems.....	13
5.6	Managed IP Backbone	13
6	SYSTEM INTERFACES	14
6.1	Physical and Data Link Layer Interfaces	14
6.2	Call Signaling Interfaces	14
6.2.1	Line Control Signaling (LCS) Framework.....	16
6.2.2	PSTN Signaling Framework	24
6.3	Media Streams.....	25
6.4	MTA Device Provisioning	26
6.5	Event Messages Interfaces	26
6.5.1	Event Message Framework	26
6.6	Quality-of-Service	28
6.6.1	Dynamic Quality of Service Overview	28
6.6.2	Layer-Two vs. Layer-Four MTA QoS Signaling.....	32
6.6.3	Dynamic Quality of Service Implementation	32
6.7	Audio Servers.....	37
6.8	Security.....	37

APPENDIX A CALL FLOWS43

A.1 Common Call Flows.....44

A.1.1 Originate Call.....44

A.1.2 Originate Call, Terminating Party Available46

A.1.3 Originate Call, Called Party Busy.....47

A.1.4 Originate Call, Glare Condition48

A.1.5 Originate Call, Insufficient Resources50

A.1.6 Receive Call53

A.1.7 MTA Disconnect Call55

A.1.8 Rogue MTA Disconnect Call.....57

A.1.9 PSTN Disconnect Call59

A.1.10 E911 Maintain Call62

A.1.11 E911 Disconnect Call64

A.1.12 Process Call Waiting66

A.1.13 Process 3-Way Call68

A.1.14 Visual Message Waiting Indication70

A.1.15 Telemetry Transport73

A.1.16 Audit Endpoint75

A.1.17 Audit Connection76

A.2 Common Call Flow Macros77

A.2.1 Create Access Network Connection77

A.3 NCS with RTP Named Telephony Events Macros.....79

A.3.1 Notify Off-Hook (RTP)79

A.3.2 Notify On-Hook (RTP)80

A.3.3 Notify Hook Flash (RTP).....81

A.3.4 Ring MTA (RTP).....82

A.3.5 Open Loop Short (RTP).....83

A.3.6 Open Loop Long (RTP)84

A.4 NCS Translation Signaling Macros.....85

A.4.1 Notify Off-Hook (NCS Translation).....85

A.4.2 Notify On-Hook (NCS Translation).....86

A.4.3 Notify Hook Flash (NCS Translation)87

A.4.4 Ring MTA (NCS Translation)88

A.4.5 Open Loop Short (NCS Translation)90

A.4.6 Open Loop Long (NCS Translation)91

APPENDIX B IPDT PROVISIONING AND MANAGEMENT92

B.1 Objects and Attributes in the IDT’s MIB92

B.2 Notifications and Actions associated with MIB Objects:94

APPENDIX C ACKNOWLEDGEMENTS96

Figures

Figure 1. PacketCable™ Network Component Reference Model.....	3
Figure 2. LCS System Reference Model.....	4
Figure 3. HFC Access Network Architecture with GR-303 Interworking	7
Figure 4. GR-303 Interface	8
Figure 5. PacketCable™ Full VoIP System Architecture.....	9
Figure 6. Call Signaling Interfaces	15
Figure 7. RTP Media Stream Flows in a PacketCable Network	25
Figure 8. RTP Packet Format	26
Figure 9. Representative Event Messages Architecture.....	27
Figure 10. Event Message Interfaces	27
Figure 11. PacketCable QoS Signaling Interfaces	29
Figure 12. LCS Security Interface.....	38
Figure 13. Originate Call, to Terminating Party State.....	45
Figure 14. Originate Call, Terminating Party Available.....	46
Figure 15. Originate Call, Called Party Busy.....	47
Figure 16. Originate Call, Glare Condition	49
Figure 17. Originate Call, Insufficient Resources	52
Figure 18. Receive Call	54
Figure 19. MTA Disconnect Call	56
Figure 20. Rogue MTA Disconnect Call.....	58
Figure 21. PSTN Disconnect Call	61
Figure 22. E911 Maintain Call.....	63
Figure 23. E911 Disconnect Call	65
Figure 24. Process Call Waiting.....	67
Figure 25. Process 3-Way Call	69
Figure 26. On-Hook Data Transmission.....	72
Figure 27. Telemetry Transport	74
Figure 28. Audit Endpoint	75
Figure 29. Audit Connection	76
Figure 30. Create Access Network Connection	78
Figure 31. Notify Off-Hook (RTP).....	79
Figure 32. Notify On-Hook.....	80
Figure 33. Notify Hook Flash	81
Figure 34. Ring MTA (RTP).....	82
Figure 35. Open Loop Short (RTP).....	83
Figure 36. Open Loop Long (RTP)	84
Figure 37. Notify Off-Hook (NCS Translation).....	85
Figure 38. Notify On-Hook (NCS Translation).....	86
Figure 39. Notify Hook Flash (NCS Translation)	87
Figure 40. Ring MTA (NCS Translation)	89
Figure 41. Open Loop Short (NCS)	90
Figure 42. Open Loop Long (NCS).....	91

Tables

Table 1. Physical Interfaces.....	14
Table 2. Call Signaling Interfaces	16
Table 3. GR-303 to RFC 2833 ABCD Event Mappings.....	19
Table 4. Example RFC 2833 Use – Hook-flash Reporting	20
Table 5. Example RFC 2833 Use – Hook-flash Reporting	21
Table 6. Event Message Interfaces	28
Table 7. QoS Interfaces for Standalone and Embedded MTAs.....	30
Table 8. QoS Interfaces.....	31
Table 9. Security Interfaces	40

1 INTRODUCTION

1.1 Purpose

This Technical Report (TR) describes the architecture and protocols for an Internet Protocol (IP)-based cable telephony service, referred to as the Line Control Signaling (LCS) architecture. The intent of this document is to provide a technical description of this architecture, and where appropriate, identify the portions of the PacketCable™ specifications that apply to this architecture and their use.

The objective of this architecture is to allow a Public Switched Telephone Network (PSTN) Class 5 Local Digital Switch (LDS) to perform as much call processing as possible, while providing IP-based transport in the local access cable network. This is accomplished through a Internet Protocol Digital Terminal (IPDT) using a GR-303 interface between PacketCable and the PSTN LDS. While the PacketCable architecture supports end-to-end IP Telephony using Call Management Servers (CMSs) for call control and interdomain signaling; Media Gateway Controllers, Media Gateways, and Signaling Gateways for access to the PSTN; Audio Servers and Record Keeping Servers (RKSSs) for network announcements and event management/billing information; the LCS architecture uses the IPDT for access to the PSTN and Class 5 LDSs for call control, announcements, and billing.

The LCS architecture is an interim step toward the PacketCable end-to-end IP architecture. Cable Operators may use the LCS architecture to get a deployable IP-based cable telephony service into the field before the complete PacketCable solution is available. While the PacketCable architecture has the potential to support multiple applications with telephony being the initial application, LCS is only intended to support the telephony application including primary line telephony. The LCS architecture must be able to migrate to the full PacketCable architecture.

LCS, as described in this Technical Report, uses existing PacketCable 1.0 and 1.1 specifications to support telephony service. Some modifications have been made to the specifications to support the architecture through Engineering Change Notices (ECNs).

1.2 Scope

To facilitate rapid time-to-market deployment of LCS products and services from PacketCable vendors and Cable Operators respectively, this TR outlines a limited scope of work. This TR includes a description of the Telcordia™ GR-303 Architecture, a description of the relevant system components and a description of the GR-303 system interfaces as they relate to PacketCable Specifications and protocols.

It is the intent of the LCS implementation to affect the PacketCable specifications as little as possible, not affecting existing implementations while at the same time broadening the capabilities of PacketCable PSTN connectivity via a GR-303 interface. Therefore, the number of required Engineering Change Notices (ECNs) applied to PacketCable Specifications for implementation has been kept to a minimum. This was accomplished through a review of the full suite of PacketCable Specifications (1.0, 1.1, and 1.2) and any impact on the associated functional components of the architecture, e.g. the MTA. Specifications including, but not limited to, Network Based Call Signaling (NCS), Codecs, DQoS, Event Messaging, Security, Audio Server, and Provisioning were examined for affect from the GR-303 interface. Note that these ECNs are also valid for non-LCS implementations.

This TR will consider two implementations of PacketCable on the GR-303 interface:

1. An implementation of the GR-303 interface using only the NCS protocol modified to allow for support of missing analog line signals. This implementation acknowledges certain delay issues covered in more detail in Section 6.
2. An implementation of the GR-303 interface through a modification of the NCS protocol, which allows for implementation of selected portions of the IETF RFC 2833.

The Dynamic Quality of Service (DQoS) Specification has been modified to permit LCS control of subscriber calls. One of the main points covered by these changes was to clarify that Gate Coordination is

indeed optional for DQoS implementations, since LCS will make use of DQoS Gate Control but not Gate Coordination. Another main point was to make Event Message generation not required since accounting and billing are handled outside of the PacketCable network for LCS voice calls. This change affected DQoS in that Event Message information is passed between the CMS and CMTS during DQoS signaling exchanges, and a way to signal the CMTS to not generate Event Messages for a given call was needed. These changes and a few minor specification clarifications were applied to DQoS through DQoS ECNs detailed in Section 6.6.

The Event Messages (EM) Specification has been modified to make Event Message generation not required for calls involving the LSC system.

This TR considers the G.711 codec as its primary codec with all other identified PacketCable codecs to be considered for further study.

Except for the NCS, DQoS, and Event Messages Specifications changes noted, and unless changed through the established ECN process, all other Specifications are unaffected and can be implemented in existing fashion in the LCS architecture. The Cable Operators will be responsible for specific implementations and may find this TR and the Appendix Call Flows a useful guide.

The GR-303 reference document is: **Telcordia, GR-303-CORE Issue 02, December 1998**; “Integrated Digital Loop Carrier System Generic Requirements, Objectives and Interface”. This reference includes implementations of party lines, coin phones, ground start loops, and two different types of signaling: CSC, which is entirely out-of-band, and TMC, which is a hybrid of in-band and out-of-band signaling. The PacketCable architecture supports loop start signaling only and will use the TMC version of signaling as described in GR-303-CORE.

Finally, the system components within a GR-303 implementation are positioned to migrate from the switched IP architecture to PacketCable’s full VoIP architecture by transforming the IPDT into the Media Gateway component of Figure 1. As an unbundled network element within the PacketCable Architecture, the Media Gateway will deploy the TGCP interface between the gateway and the CMS.

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 flow,” “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.

1.2.1 PacketCable Reference Architecture

PacketCable is a set of protocols developed to deliver enhanced communications services using packetized data transmission technology to a consumer’s home or business over the cable network. The “PacketCable Architecture Framework” (PKT-TR-ARCH-V01-991201) is the reference Technical Report for understanding PacketCable Interface Specifications, Technical reports, and other PacketCable documents.

The reference architecture for the PacketCable™ Network is shown in Figure 1 below, with a relevant emphasis on the PSTN Gateway.

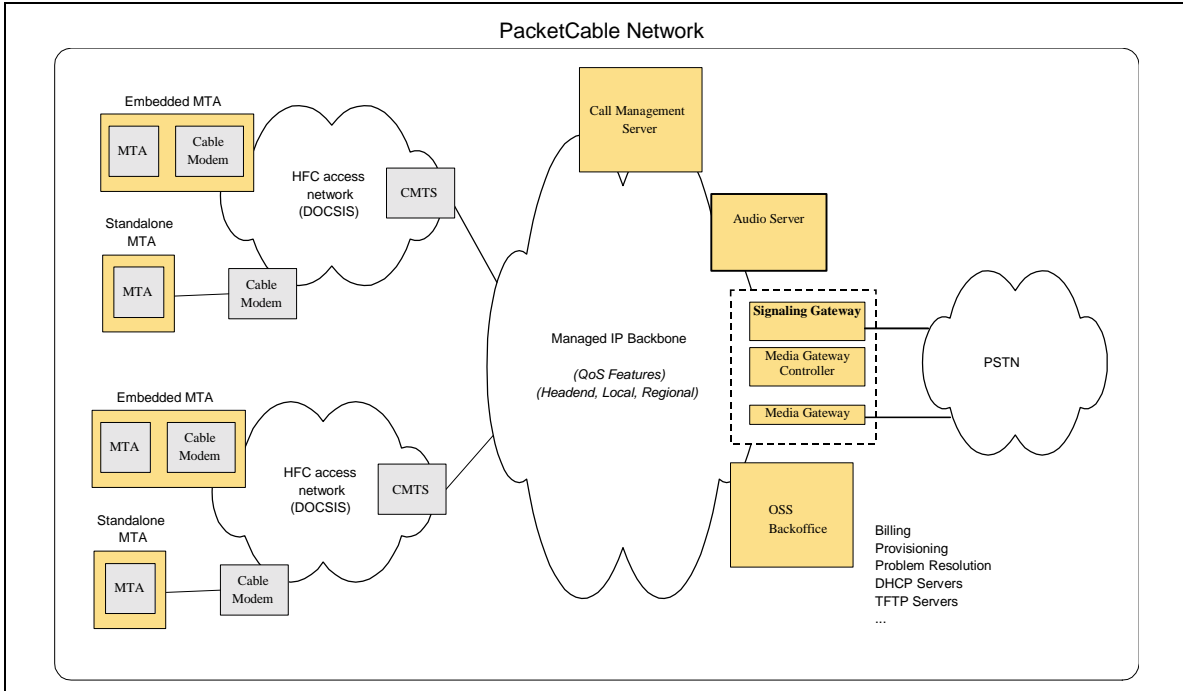


Figure 1. PacketCable™ Network Component Reference Model

1.2.2 PacketCable LCS Reference Architecture

In the Line Control Signaling (LCS) System Architecture, an IPDT gateway provides interworking between the PacketCable network and the PSTN through a local digital switch (LDS) located within the PSTN. The communications interface between the IPDT gateway and the LDS uses the Telcordia GR-303 standard and the Call Management Server/Media Gateway Controller function for the gateway is integrated into the IPDT.

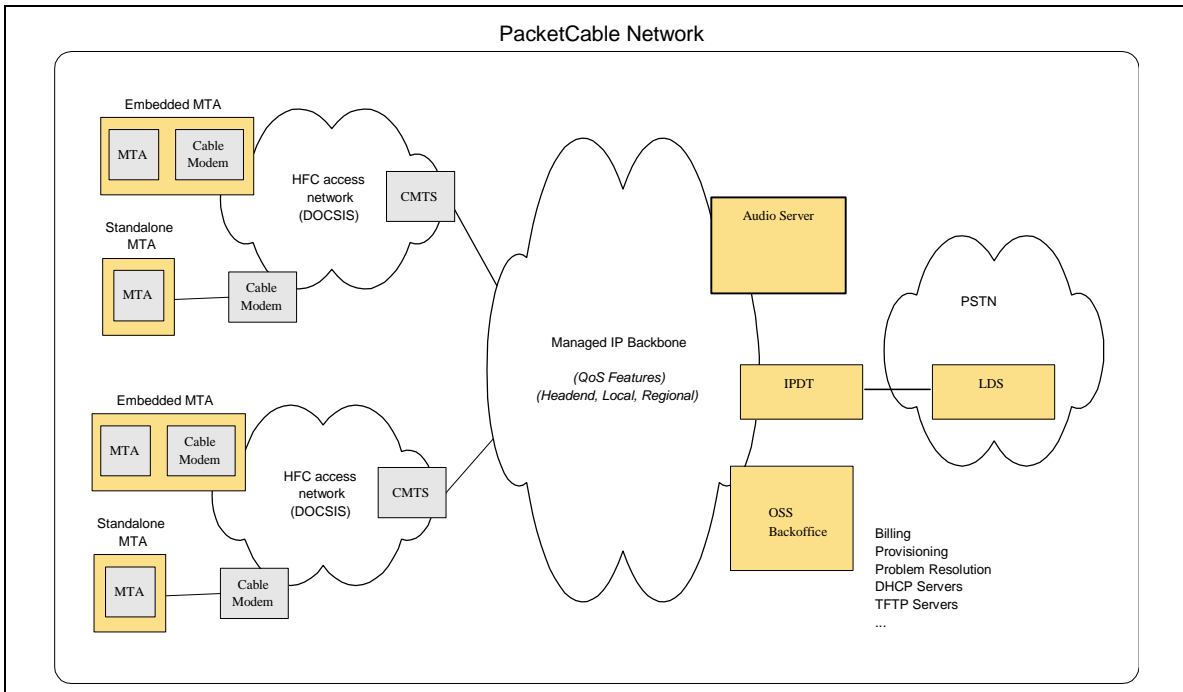


Figure 2. LCS System Reference Model

2 REFERENCES

- [1]. "PacketCable 1.0 Architecture Framework Technical Report," PKT-TR-ARCH-V01-991201, December 1, 1999, CableLabs. www.packetcable.com
- [2]. "PacketCable Network-Based Call Signaling Protocol Specification," PKT-SP-EC-MGCP-I03-010620, June 20, 2001, CableLabs. www.PacketCable.com
- [3]. "PacketCable PSTN Gateway Call Signaling Protocol Specification" PKT-SP-TGCP-I01-991201, December 1, 1999, CableLabs. www.PacketCable.com
- [4]. "PacketCable Event Messages Specification," PKT-SP-EM-I02-001128, November 28, 2000, CableLabs. www.PacketCable.com
- [5]. "PacketCable Audio/Video CODECS Specification," PKT-SP-CODEC-I02-010620, June 20, 2001, CableLabs. www.PacketCable.com
- [6]. "PacketCable Security Specification" PKT-SP-SEC-I03-010626, June 26, 2001, CableLabs. www.PacketCable.com
- [7]. "PacketCable Dynamic Quality of Service Specification," PKT-SP-DQOS-I02-000818, August 18, 2000, CableLabs. www.PacketCable.com
- [8]. "PacketCable MTA Device Provisioning Specification," PKT-SP-PROV-I02-010323, March, 23 2001, CableLabs. www.PacketCable.com
- [9]. "Integrated Digital Loop Carrier System Generic Requirements, Objectives and Interface," Telcordia, GR-303-CORE Issue 02, December 1998.
- [10]. Schulzrinne, H., Petrack, S., "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals", RFC 2833, May 2000. www.ietf.org

3 ABBREVIATIONS AND ACRONYMS

This document uses the following abbreviations and acronyms.

AS	Audio Server
CLASS	Custom Local Area Signaling Services
CG	Communications Gateway
CM	DOCSIS Cable Modem
CMS	Call Management Server. Controls the audio/video call connections. Also called a Call Agent in MGCP/SGCP terminology.
CMTS	Cable Modem Termination System, the device at a cable head-end which implements the DOCSIS RFI MAC protocol and connects to CMs over an HFC network.
COPS	Common Open Policy Server Protocol as defined by RFC 2748
CPE	Customer Premise Equipment
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System, see www.ietf.org or RFC 1034 and RFC 1035 for details.
DOCSIS	Data Over Cable System Interface Specification
DS0	Digital Signal Level 0
DS1	Digital Signal Level 1
DTF	Digital Transmission Facility
DTMF	Dual-tone Multi Frequency (tones)
EMS	Element Management System
E-MTA	Embedded MTA – a single node, which contains both an MTA and a cable modem MAC/PHY.
EOC	Embedded Operations Channel
ESF	Extended Super Frame
FQDN	Fully Qualified Domain Name, Refer to IETF RFC 821 for details
GR-303	Generic Requirements 303, a Telcordia specification for Integrated Digital Loop Carrier
H.248	A protocol for media gateway control being developed by ITU. See www.itu.ch .
HFC	Hybrid Fiber/Coax(ial [cable]), HFC system is a broadband bi-directional shared media transmission system using fiber trunks between the head-end and the fiber nodes, and coaxial distribution from the fiber nodes to the customer locations.
IANA	Internet Assigned Numbers Authority. See www.iana.org or www.ietf.org for details.
IDT	Integrated Digital Terminal
IEEE	International Electrical & Electronic Engineers. See www.ieee.org for detail.
IETF	Internet Engineering Task Force. See www.ietf.org for details.
IP	Internet Protocol
IPDT	Internet Protocol Digital Terminal
ITU-T	International Telecommunications Union – Telecommunications Services Sector
IVR	Interactive Voice Response System
LDS	Local Digital Switch

LSSGR	LATA Switching Service Generic Requirements
MEGACO	Media Gateway Control IETF working group. See www.ietf.org for details.
MG	Media Gateway
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol. Protocol follow on to SGCP.
MIB	Management Information Base
MSO	Multi-System Operator, a Cable company that operates many head-end locations in multiple cities or locales.
MTA	Multi-media Terminal Adapter – contains the interface to the physical telephony or video equipment, a network interface, CODECs, and all signaling and encapsulation functions required for VoIP transport, class features signaling, and QoS signaling.
NCS	Network Based Call Signaling
Orig. Call	Originating Call. A call initiated by the subscriber CPE
OSS	Operational Support Systems
POTS	Plain Old Telephone Service
PSTN	Public Switched Telephone Network.
RDT	Remote Digital Terminal
RKS	Record Keeping Server
RTP	Real Time Protocol, a protocol defined in RFC 1889 for encapsulating encoded voice and video streams.
SDP	Session Description Protocol.
SGCP	Simple Gateway Control Protocol. Earlier draft of MGCP.
SID	Service ID. A 14-bit number assigned by a CMTS to identify an upstream virtual circuit. Each SID separately requests and is granted the right to use upstream bandwidth.
S-MTA	Standalone MTA – a single node which contains an MTA and a non-DOCSIS MAC (e.g., Ethernet).
SNMP	Simple Network Management Protocol. See www.snmp.org and www.ietf.org for more details.
SS7	Signal System #7
T1	North American DS1, capacity of 1.555Mbps can contain up to 24 separate voice conversations
Term. Call	Terminating Call. A call the network initiates towards the subscriber CPE
TFTP	Trivial File Transfer Protocol
TGCP	Trunking Gateway Control Protocol
TMC	Timeslot Management Channel
TN	Telephone Number
TOD	Time of Day
TOS	Type of Service. An 8-bit field of every IP version 4 packet. In a Diffserv domain, the TOS byte is treated as the Diffserv Code Point, or DSCP
VoIP	Voice over IP

4 SYSTEM ARCHITECTURE

The LCS architecture is an extension of the PacketCable 1.0 architecture. This section outlines the basic architecture, the GR-303 interface, and the migration path from Line Control Signaling to Network-Based Call Signaling.

4.1 Line Control Signaling System Architecture

The LCS System Architecture comprises a DOCSIS™ 1.1 Hybrid Fiber Coax (HFC) access network interworking with PSTN local digital switches (LDS) through an Internet Protocol Digital Terminal (IPDT).

This system is illustrated by physical, data link, and transport layers between components in Figure 3. The HFC access network includes all of the system components required to support PacketCable Voice-over-IP (VoIP) telephony in the LCS application, and for future full VoIP system operations. The IPDT provides all the functions of a media gateway controller (MGC), signaling gateway (SG), media gateway (MG), and a partial call management server (CMS).

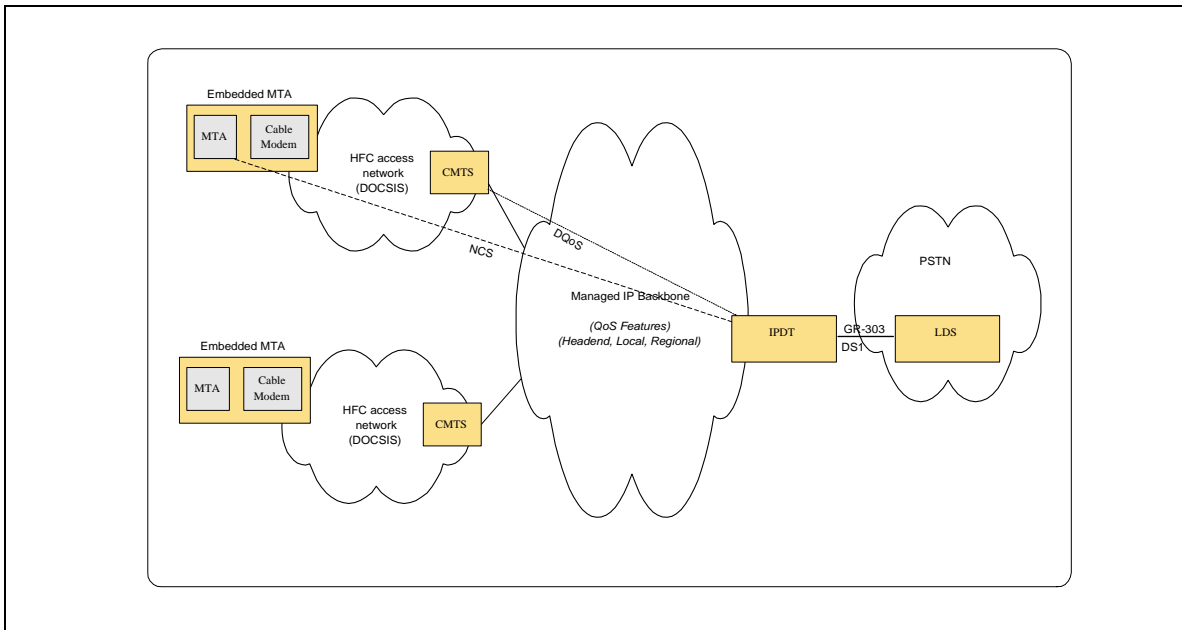


Figure 3. HFC Access Network Architecture with GR-303 Interworking

4.2 IPDT / LDS Interworking: the GR-303 Interface

The GR-303 interface defines an *interface group (IG)* that connects an *integrated digital terminal (IDT)* in a local digital switch with a *remote digital terminal (RDT)*. The IPDT is a specialized version of an RDT. The RDT allows the local loop termination to be located remotely from the central office switch, reducing the length and cost of the local loop, while increasing the geographic area served by the LDS. In addition, the interface allows dynamic allocation of available DS1 and subscriber loop bandwidth, thereby supporting concentration.

Figure 4 illustrates the interface.

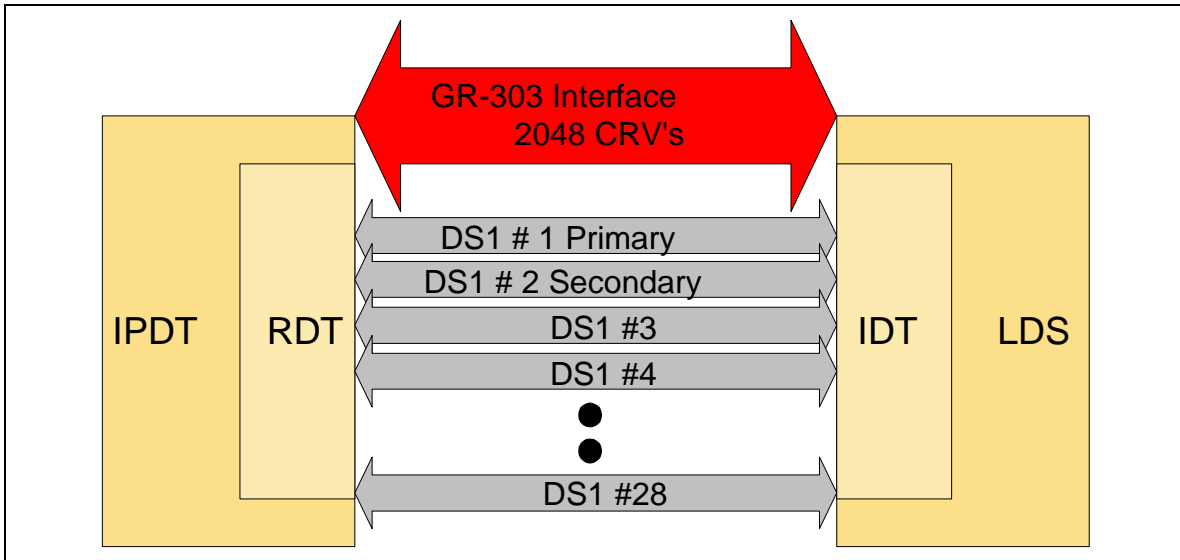


Figure 4. GR-303 Interface

The digital trunk facility (DTF) consists of one or more DS1 (T1) lines, possibly contained in an STS-1 or OC-3 transmission format.

A total of 4 DS0s in each GR-303 Interface are used for the TMC and EOC message channels. The information in these channels complies with ITU-T Q.921 standard at layer 2. The TMC uses a subset of ITU-T Q.931 standard at layer 3 to convey out-of-band signaling information. The *embedded operations channel (EOC)* occupies one primary and one backup DS0 to carry messages concerning Operations, Surveillance and Management between the IDT and the RDT. In particular, the IDT sends frequent scheduled audits to the RDT over the EOC.

The IG comprises between two and 28 DS1 (T1) lines. The primary *timeslot management channel (TMC)* occupies a DS0 on one DS1 and secondary TMC occupies a DS0 on a different DS1 for backup. Signaling messages for setup and teardown of calls are carried between the IDT and RDT in the TMC. A unique logical identifier for a particular subscriber provisioned on the interface is a *call reference value (CRV)*. Each IG supports up to 28 DS1s for a total of 668 (672 – 2 TMC and 2 EOC) channels available for calls. Each IG also supports 2048 CRVs, allowing for approximately 3:1 concentration of CRVs to available DS0s. Assignment of multiple IGs to the RDT allows for even greater concentration of bandwidth available on the access side of the RDT.

The EOC uses the ROSE, CMIS and CMISE specifications to convey administrative and operational status information between the IDT and RDT. In band signaling information is conveyed across the GR-303 interface by means of robbed-bit ABCD codes in the voice channel.

4.3 Migration to Full VoIP

The LCS architecture is an *interim* and *transitional* architecture. This architecture has been developed to enable early deployment of IP telephony capability in the HFC access network. This allows cable system operators a path of entry into the evolving IP telephony marketplace.

The architecture has been developed to take advantage of PacketCable member investment in equipment. The system components migrate naturally from the switched IP architecture to PacketCable’s full VoIP architecture by transforming the IPDT into a Media Gateway, as illustrated in Figure 5.

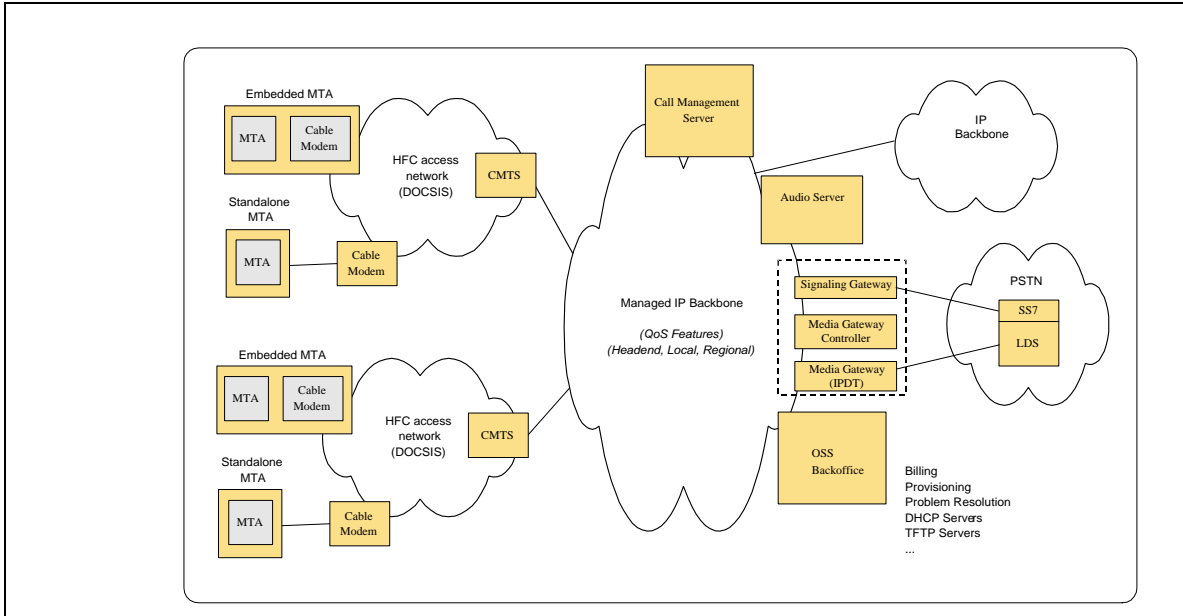


Figure 5. PacketCable™ Full VoIP System Architecture

5 SYSTEM COMPONENTS

This section will describe at a high level the capabilities of logical components of the Line Control Signaling Architecture required to support the GR-303 interface. Each component may physically be a single piece of equipment existing as a node on a network, a set of distributed network nodes, or it may be co-located or bundled with another components. PacketCable 1.0 architecture as described in PKT-TR-ARCH-V01-991201 is assumed as baseline architecture.

The major change to the components from the baseline architecture is the addition of one new PacketCable component: the IP Digital Terminal (IPDT). Other components are either not required by the LCS architecture, or have reduced capabilities since some functionality is no longer needed. Some additional provisioning is required to support GR-303.

In addition, the change from a PSTN interface and its intelligent network equipment (SSP, STP, SCP, etc.) to a Local Digital Switch (LDS) has a major impact on PacketCable goals and objectives. PacketCable 1.0 NCS assumed unfettered access to the PSTN network, while LCS restricts the cable system to working as a remote device to a local switch for telephony features. For this reason, the LDS is briefly described.

5.1 Local Digital Switch

The Local Digital Switch (LDS) is an external component to the LCS system that provides it with line side call features and interfaces to the PSTN. It is a Central Office (CO) equipped with an Integrated Digital Terminal (IDT) supporting subscriber lines remoted over digital trunk. These digital trunks, or 'digital loop carriers,' normally terminate in a Remote Digital Terminal (RDT) that concentrates the analog line traffic, converts analog line signaling into digital signals, and performs remote line maintenance. The line-side digital interface between the LDS's IDT and the RDT is defined by one of two standards: GR-303, and V5.2. This specification focuses on GR-303, although a similar architecture could be used to handle V5.2. This integrated digital line system has been long deployed to reduce operating, installation, and capital equipment costs while delivering an equivalent range of telecommunications services to a direct analog line interface. A large percentage of subscriber lines in North America today are connected to the central office over remote digital interfaces instead of direct analog lines.

The basic idea of LCS is to reuse the existing subscriber line related features capabilities of the CO by mimicking the IDT-RDT digital loop interface. To simulate this interface to the LDS, LCS will use some existing components of the PacketCable NCS architecture (E-MTA, CMTS, OSS), and an additional gateway component called the IP digital Terminal (IPDT). There are a number of consequences of this approach:

1. The Local Digital Switch combined with the PacketCable IPDT will replace a number of NCS components that provide line side (local) call services. This includes PacketCable 1.0 NCS components as defined in the architecture framework: CMS, MGC, SG, MG, RKS, and AS.
2. Some line and billing related OSS functions will be handled by the LDS system operations support system (OSS) instead of the PacketCable OSS and RKS.
3. The subscriber line features are determined by the LDS feature set, and not necessarily by PacketCable 1.0 requirements. At this time, most LDSs support all the PacketCable call features, and even many not specified in PacketCable 1.0; however LDS's do not typically support internet, web, or IP-based multi-media features, which must be considered beyond the scope of LCS. Since the LCS architecture is considered an evolution to NCS, it may be important for the cable operating company to restrict the subscriber line call features allowed in the LDS to those defined by PacketCable 1.0 to avoid backward compatibility issues in the future since many existing LDS features may prove difficult or impracticable to implement in the PacketCable Architecture
4. The LDS OSS has certain provisioning expectations and will manage some line and subscriber features although not the full scope of line related calling features. The LCS OSS also has its requirements on provisioning, many of which are subscriber related and call feature related. These two provisioning systems will have to be synchronized, if not integrated. There will also be some

additional requirements on LCS provisioning to handle the requirements of the IPDT gateway and digital loop trunk provisioning.

5. There is a testing ‘mismatch’ between the maintenance expectations of the LDS and PacketCable in the area of line testing. GR-303 was designed to remotely control and test analog lines over CO/IDT/RDT using digital loop carriers. The cable architecture controls and tests lines over CMTS/MTA using HFC networks. The LCS will not manage the testing of PacketCable components and lines; instead PacketCable testing procedures are used.

5.2 IP Digital Terminal

The IP Digital Terminal (IPDT) provides the signaling gateway and media gateway inter-working between the LDS and the PacketCable network. It interfaces to the LDS over digital trunks which carry signaling and voice traffic. It interfaces to other components of the PacketCable architecture over an IP network, which carries signaling and voice traffic. From the LDS side, it simulates a remote digital loop carrier. From the PacketCable network side, it replaces the subscriber line call related functions and voice to packet translation functions. Thus it will:

1. Convert outgoing PacketCable voice packets into digital circuit voice traffic and pass it to the LDS over digital trunks. Thus, the IPDT replaces some functions of the MG.
2. Convert incoming voice circuit traffic on digital trunks into PacketCable packet traffic.
3. Convert PacketCable call control signaling into digital line interface signaling (GR-303 TMC and ABCD robbed-bit) required by the LDS.
 - Originating call establishment is initiated through a mapping of NCS hang-down (off-hook) to a GR-303 SETUP message.
 - When RFC 2833 is not used during an established call, NCS hang-up, hang-down (off-hook) and hook-flash events must be mapped into the respective upstream ABCD signals. In the case of the hook-flash, by the time the IPDT has received this message the subscriber hook-flash has already completed and the subscriber is back off-hook. The IPDT must emulate the hook-flash toward the LDS by providing an ABCD on-hook signal of appropriate duration to be recognized as a hook-flash by the LDS (e.g. 500ms is a typical minimal emulation period for LDS flash recognition). This causes delays that are noticeable to the subscriber.
4. When RFC 2833 is used during an established call, upstream RFC 2833 loop open (i.e. on-hook) states are passed immediately from the IPDT to the LDS by being mapped directly into ABCD loop open signals. Upstream RFC 2833 loop closed (i.e.; off-hook) states are passed immediately from the IPDT to the LDS by being mapped directly into ABCD loop closed signals. Loop closed event packets replace the first three voice packets after the off-hook is detected, then RTP voice transmission is continued. No noticeable signaling delays are experienced.
5. Convert digital line interface signaling from the LDS (GR-303 TMC and ABCD robbed-bit) into PacketCable call control signaling.
 - Terminating call establishment is initiated through a mapping of GR-303 SETUP message to NCS connection control messages.
 - Call release is always initiated by the LDS through GR-303 TMC messages.
 - When RFC2833 is used during an established call, downstream ABCD ringing and loop current feed open states are passed immediately from the IPDT to the MTA by being mapped directly into RFC 2833 RTP event packets at the IPDT. FSK data is passed transparently in the voice stream. No noticeable signaling delays are experienced.
 - When RFC2833 is not used during an established call, downstream ABCD ringing, loop current feed open and FSK data must be processed by the IPDT and turned into functional NCS messages. For example, when ringing state and caller ID FSK are received it is necessary for the IPDT to determine ringing cadence from the downstream ABCD signaling and collect FSK

caller ID data from the downstream DS0 media stream. The IPDT then must pass this information to the MTA in an NCS message. This causes a delay of one full ringing cycle in subscriber alerting and caller ID delivery.

6. Manage its end of the digital trunks, including alarms, initialization, and maintenance.
7. Manage its own provisioning information as needed to support the inter-working (mapping between LCS and LDS numbering, trunk identities, etc.).

The IPDT itself generally does not perform call processing, call feature related line provisioning, or billing functions- these are handled by the LDS. When RFC 2833 is not in use there are exceptions, however, since some call processing must be done when line state signaling is not mapped directly through the IP network between the IPDT and MTA.

5.3 Cable Modem Termination System

The Cable Modem Termination System (CMTS) is the component that terminates one or more DOCSIS 1.1-based Hybrid Fiber/Coax (HFC) access links and provides connectivity to one or more wide area networks (typically IP or ATM). It is located at the cable television system head-end or distribution hub.

On the HFC network side, it supports several layers of interface protocols to manage the population of MTAs as defined by DOCSIS. On the PacketCable network side, it supports the transport of RTP voice packets with appropriate priority and quality-of-service constraints.

For LCS, its functions are identical to the PacketCable 1.0 functions.

5.4 Embedded MTA

An embedded MTA (E-MTA) is a single hardware device that incorporates a DOCSIS 1.1 cable modem as well as a PacketCable MTA component. Following this, the E-MTA has two logical parts, which is physically combined into one device: a Cable Modem (CM), and a Multi-media Terminal Adapter (MTA). Every home using PacketCable 1.0 services has at least one such device. On the subscriber side, it supports one or more phone lines, and optionally includes a local 10BaseT port for high-speed data access. On the network side it support the PacketCable 1.0/DOCSIS 1.1/HFC network requirements.

The cable modem is a modulator that provides data transmission over the cable network using the DOCSIS 1.1 protocol. In PacketCable, the CM plays a key role in handling the media stream and provides services such as classification of traffic into service flows, rate shaping, and prioritized queuing.

The MTA is a PacketCable 1.0 client device that contains a subscriber-side interface to the subscriber's CPE (e.g., telephone) and a network-side voice and signaling interface to elements in the network. It basically handles the two-way translation of voice to IP packets, and POTS telephone to IP based signaling conversion. It provides codecs and all signaling and encapsulation functions required for media transport and call signaling. PacketCable 1.0 MTAs are required to support the Network Call Signaling (NCS) protocol.

Compared to PacketCable 1.0 NCS, the LCS E-MTA has the following changes:

1. While the E-MTA is still required to support the NCS signaling, there is an optional additional requirement for a subset of the IETF's RFC 2833 RTP based signaling. This requires additional functionality in an LCS ready E-MTA beyond that explicitly called out in PacketCable 1.0.
2. Only G.711 codec translation and compression is supported.
3. The E-MTA still interfaces on the physical and DOCSIS 1.1 level with the CMTS, but has two architecture changes (which are transparent to the E-MTA):
 - the telephone signaling interface is with IPDT instead of the MGC/CMS
 - the voice to packet translations are handled by the IPDT instead of the MG

Otherwise, the LCS version of the E-MTA follows PacketCable 1.0 specifications.

5.5 Operations Support Systems

The Operations Support System components are typically used as part of the network's "back office" to manage, administer, and provision the PacketCable systems. They provide fault management, performance management, security management, accounting management, and configuration management for all devices in the PacketCable architecture. The OSS components are mostly unchanged from PacketCable 1.0. The main impact of LCS on the OSS components are:

1. Additional management is required for the IPDT, including:
 - GR-303 trunk provisioning and management
 - GR-303 trunk numbering mapping to NCS trunk numbering mapping
 - Subscriber data required for NCS to GR-303 translation
2. Less management is required for call related features
3. The Record Keeping Server (RKS) will not be used to keep billing (CDR) data, as the LDS will keep its own records

5.6 Managed IP Backbone

The Managed IP Backbone is an element external to the system that provides the communications mechanism between the CMTS and the IPDT. It is unchanged from PacketCable 1.0.

6 SYSTEM INTERFACES

Protocol specifications have been or are currently being defined for most component interfaces within the PacketCable architecture. An overview of these component interfaces is provided in the PacketCable 1.0 Architecture Framework Technical Report and the PacketCable 1.2 Architecture Framework Technical Report. The individual PacketCable protocol specifications should be consulted for complete requirements of each component interface. (available at: www.PacketCable.com)

New interfaces pertaining to support of Line Control Signaling have been added to the PacketCable architecture. These will be detailed in the following section. Additionally, several interfaces defined for the PacketCable architecture do not exist in the PacketCable LCS system architecture. Since an IP Digital Terminal (IPDT) integrates part of the functionality of a Call Management Server (CMS) with that of a Signaling Gateway (SG), Media Gateway Controller (MGC) and Media Gateway (MG); none of the interfaces between these components is considered an open interface in the LSC system. Since a IPDT utilizes the GR-303 interface and a Local Digital Switch (LDS) to provide feature functionality, subscriber and PSTN connectivity, PacketCable's inter-domain signaling and audio server interfaces do not apply either.

The following sections overview the use of the existing and new PacketCable component interfaces within the LSC system. Frequent references are made to interface names introduced by the PacketCable 1.0 and 1.2 Architecture Framework Technical Reports. Please see those reports for additional details.

6.1 Physical and Data Link Layer Interfaces

The lower layer interfaces supported between the components used in the GR-303 sub-system of the PacketCable architecture are as shown in the following table:

Table 1. Physical Interfaces

COMPONENT INTERFACE	PHYSICAL INTERFACE
CMTS-HFC Network and HFC Network-MTA	DOCSIS 1.1
LDS-IPDT	DS1 ESF
CMTS-IPDT	IEEE 802.3
MTA-IPDT	No direct physical interface exists
MTA-CPE	Twisted pair/RJ-11 employing loop-start and DTMF tone signaling; dial pulse signaling is not supported
CMTS-Network Servers and IPDT-Network Servers	IEEE 802.3
CMTS-EMS and IPDT-EMS	IEEE 802.3

6.2 Call Signaling Interfaces

Call signaling requires multiple interfaces within the LSC system of the PacketCable architecture. These interfaces are identified in the diagram below. Each interface in the diagram is labeled, and further described in the subsequent table. Where interfaces are new to the PacketCable architecture for LCS descriptive notes have been added to the table.

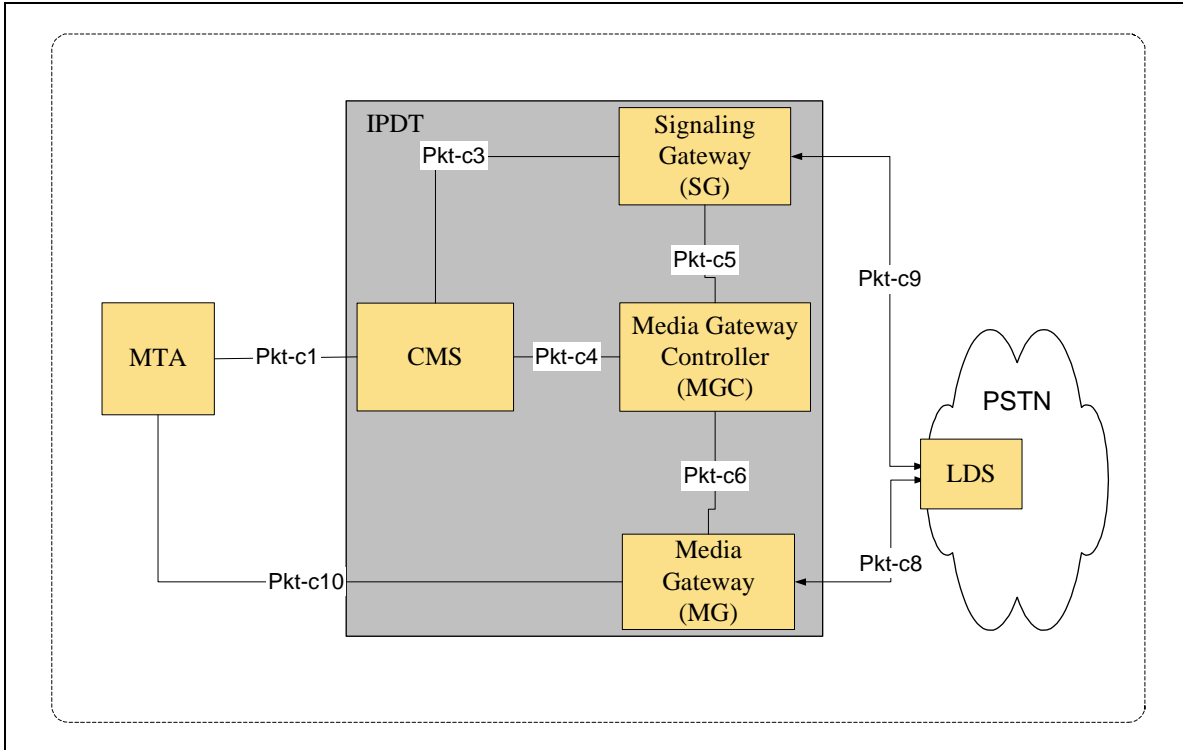


Figure 6. Call Signaling Interfaces

Table 2. Call Signaling Interfaces

Interface	PacketCable Functional Components	Description
Pkt-c1	MTA – CMS	Call signaling messages exchanged between the MTA and CMS using the NCS protocol Note: A subset of this existing PacketCable interface is used by the LSC system.
Pkt-c3	CMS – SG	Call signaling messages exchanged between CMS and SG using the ISTP protocol Note: Not an open interface in the LSC system; CMS and SG are integrated within a single physical IPDT
Pkt-c4	CMS – MGC	Call signaling messages exchanged between the CMS and MGC; the protocol for this interface is CMSS. Note: Not an open interface in the LSC system; CMS and MGC are integrated within a single physical IPDT entity
Pkt-c5	SG – MGC	Call signaling messages exchanged between the MGC and SG using ISTP. Note: Not an open interface in the LSC system; SG and MGC are integrated within a single physical IPDT entity
Pkt-c6	MGC – MG	Interface for media control of the media gateway and possibly in-band signaling using the TGCP protocol. Note: Not an open interface in the LSC system; MGC and MG are integrated within a single physical IPDT entity
Pkt-c8	MG – LDS	Bearer channel connectivity from the Media Gateway to the PSTN supporting in-band robbed-bit ABCD signaling Note: Robbed-bit ABCD signaling is added to this existing PacketCable interface by the inclusion of the LSC system into the PacketCable architecture.
Pkt-c9	SG – LDS	GR-303 common channel signaling messages to/from the PSTN (i.e.- the TMC signaling link) Note: GR-303 out-of-band TMC signaling introduces this new signaling interface on the PSTN side of the PacketCable architecture.
Pkt-c10	MTA – MG	Line state signaling messages exchanged between the MTA and MG using RFC 2833 ABCD events Note: RFC 2833 ABCD event signaling adds new functionality into the PacketCable architecture. Future applications may make use of other RFC 2833 features such as DTMF event relaying.

6.2.1 Line Control Signaling (LCS) Framework

The PacketCable NCS architecture places call state and feature implementation in a centralized component, the CMS, and places device control intelligence in the MTA. The MTA passes device events to the CMS, and responds to commands issued from the CMS. The CMS is responsible for setting up and tearing down calls, providing advanced services [CLASS and custom calling features], performing call authorization, and generating billing event records, etc. In the PacketCable LSC system, complete CMS functionality is achieved by integration of a partial CMS and the SG, MGC and MG functions into a single entity (the IPDT), and through IPDT cooperation with a PSTN LDS using the GR-303 interface. SS7 connectivity is

handled completely by the LDS, while the IPDT is responsible for extending concentrated LDS line access over an IP network to cable telephone subscribers served by a CMTS and an MTA.

The LSC system introduced in this document inter-works GR-303's Timeslot Management Channel (TMC) and in-band ABCD robbed-bit line state signaling with PacketCable's NCS protocol. Only a small subset of the full NCS protocol is required for this inter-working (Pkt-c1). However, use of only the NCS protocol for this inter-working leads to several signaling delays that are perceptible to end users. Most notable among these are the delay of one full ringing cycle in subscriber ringing and caller ID delivery, and the hook-flash recognition delay (at the LDS this delay is minimally 500ms in typical scenarios). To solve these delay problems, IETF RFC 2833 support is added (Pkt-c10). RFC 2833 extends the in-band robbed-bit ABCD signaling states to and from the PSTN DS0 bearer interface over an IP network by using RTP event payloads. A small subset of RFC 2833, the ABCD trunk events, is needed in the PacketCable LCS GR-303 sub-system. The combination of NCS and RFC 2833 signaling on the access network side effectively removes all significant signaling inter-working delays, providing subscribers with PSTN-equivalent voice and feature quality.

6.2.1.1 NCS Protocol Usage

The NCS protocol was developed for use in a distributed switch environment. Switch functions such as signaling, routing and prioritization as well as billing and record keeping are distributed across the HFC/DOCSIS and Ethernet/IP networks in various network elements such as the MTA and CMTS, the DNS and RKS. In this Distributed Switching environment, the MTA assumes many switch-like signaling functions. In particular, the MTA has the ability to interpret and act upon signals created by the end-user. It can, for example, distinguish between a disconnect and a hook flash.

This TR describes a use for the NCS protocol in a Centralized Switch environment. Here the function of the HFC/DOCSIS and Ethernet/IP networks is to provide access to the PSTN LDS. The PSTN performs Switch functions such as routing, billing and record keeping. The MTA, HFC/DOCSIS and Ethernet/IP networks work together to perform some of the functions of a Remote Digital Terminal to the PSTN switch. They must translate analog signals created by end-user telephones into digital signals and digital voice, then route the signals and voice to the PSTN switch.

Adapting the NCS protocol to the Centralized Switch environment involves several choices. First, the MTA must lose some of its switch-like signaling functions. Interpretation of signals by the MTA introduces unnecessary and unacceptable delays in call processing. Second, the encoding and routing of signaling information need to be tailored to the application. Two different models for encoding and routing signaling information are described below. Both use existing NCS signaling messages to establish originating and terminating calls. Once a call is established, however, the two models differ.

The first model uses NCS messages alone. The second uses NCS messages together with a subset of RTP signaling messages specified in RFC 2833. The first method may be vulnerable to signaling-packet loss, network delays and network jitter in heavily loaded networks. It may also result in longer signaling delays since the NCS messages will typically have a lower priority than RTP messages. The second method is more robust and faster in heavily loaded networks due to the higher priority of RTP messages and redundancy of line state signaling information achieved through repetition of this information at the voice RTP packetization rate.

Both methods described below require changes to the existing NCS specification. The first method requires that the NCS specification add a command which permits the IPDT to signal a short (150-350ms) and a long (800-1000ms) duration loop current feed open state (i.e., Open Switch Interval) to the MTA. The short duration signal is used for the LDS-based suppressed ringing feature which is necessary to support applications such as home security system monitoring and remote meter reading. The long duration signal is used during normal call release procedures to cause fax and answering machines to hang up before the LDS delivers a series of tones and announcements commonly referred to as line permanent sequence. The ECN that adds these signals to NCS is mgcp-n-01089.

The second method requires that NCS' use of SDP (Session Description Protocol) be augmented to provide for the signaling of RFC2833's RTP Named Telephony Events. This work is illustrated in ECN mgcp-n-01058v2.

6.2.1.1.1 Use of NCS I03 Protocol

When RFC 2833 is not implemented, direct application of the NCS Protocol is necessary in the NCS/GR-303 inter-working function of the IPDT. When used in this manner, the MTA is instructed to send notifications of hook state changes and is immediately placed in Send-Receive mode for the duration of the call. The MTA does not generate local dial tone or collect digits, as the CLASS 5 switch provides these functions over the LCS interface. In many cases there is one obvious and straightforward way to do NCS/GR-303 inter-working as shown in the call flows of Appendix B. There are, however, situations where the interpretation of state changes as events (and conversely) is less clear.

Consider, for example, power ringing, directed from the CLASS 5 switch toward the CPE. In a pure NCS network, ring cadences are passed as NCS signaling messages from the CMS. However in the mixed environment of GR-303 and the NCS protocol, there is no such initial message and the inter-working function could proceed in one of two ways. The first possibility is to observe the robbed-bit ring on/ring off pattern from the switch and then deduce and deliver an NCS message containing the ring cadence. This results in a delay of one full ringing cycle before the message can be sent. Alternatively, the inter-working function could pass through the ring on and ring off transitions as individual NCS messages as the ring state changes are detected. In this case there may be some additional delay, and loss of an NCS message will result in playing out the wrong ringing cadence. It is also possible that the loss of caller ID capability for a given call would be experienced. Additionally, for distinctive ringing cadences this latter method will generate a lot of extra NCS message traffic and is very likely to produce inaccurate cadences. Implementations not using RFC 2833 must make a choice between one of the aforementioned methods.

As another example, consider subscriber hook-flash detection and reporting. An MTA instructed to detect hook-flash by an IPDT will time the subscriber on-hook period, looking for an on-hook duration that lies within the hook-flash detection interval (i.e., LSSGR requirements allow 300ms to 1100ms flash duration). At completion of the hook-flash, when the subscriber is again in an off-hook state, the MTA will format and send an NCS message to the IPDT. The IPDT then must emulate the hook-flash toward the LDS by applying a period of loop open (on-hook) in the ABCD signaling for this subscriber. Typical LDS hook-flash detection requires approximately a 500ms on-hook indication, which implies at least that much additional delay in hook-flash reporting. It is noted here that another NCS-only alternative exists, but that this alternative would not reduce this delay. Using that alternative the IPDT would ask the MTA to ignore hook-flash and would instead rely on NCS hang-up and hang-down (off-hook) events to indicate hook-flash. Since the NCS hang-up event is specified as using the timing for flash response enabled, the LSSGR indicates that hang-up cannot be reported until at least 1100ms of on-hook has been detected. Consequently, this latter method introduces at least an additional 1100ms of delay in hook-flash reporting and would require a hook-flash duration of greater than 1100ms, which is longer than allowed by LSSGR requirements (LSSGR specifies a hook-flash duration between 300 and 1100ms).

From an operational viewpoint, all NCS-only solutions are sensitive to packet loss and to a lesser extent, jitter in the network. Although NCS has a recovery mechanism for packet loss, the time interval for recovery is very long relative to the timing needed in ringing or hook flash.

The call flows of Appendix II show the implementation choices made and raise the issues discovered. While the base NCS protocol functionally provides the necessary signaling for all call scenarios, there may be instances where the unaltered protocol may be unsuitable due to operational impairments. The chief concerns are in the areas of networks with frequent packet loss and events with fine delay sensitivity, as illustrated above. The degree to which these are concerns are implementation issues for the service operator.

6.2.1.2 RFC 2833 Protocol Usage

In order to solve the significant LCS/NCS inter-working signaling delays previously mentioned; RFC 2833 was introduced into the LCS architecture. RFC 2833 allows for line state carried in GR-303 ABCD robbed-bit signaling to be extended across the access network to and from the MTA. It also has the side effect of making the IPDT implementation simpler since the inter-working requirements now simply imply direct mapping functions between RFC 2833 events and ABCD signaling patterns.

RFC 2833 usage is established dynamically per call through NCS signaling and under the command of the IPDT. Examples of call signaling necessary to dynamically establish RFC 2833 use are shown in the Call Flows section of this document. NCS usage for signaling the RFC 2833 application is also described at the end of this section.

RFC 2833 signaling takes place only during an active two-way RTP voice connection between an IPDT's Media Gateway and an MTA endpoint (Pkt-c10). The following sections describe RFC 2833 signaling within that context.

6.2.1.2.1 ABCD Event Support

The PacketCable LSC system requires that a small subset of RFC 2833 be implemented. This subset is itself a subset of the allowed ABCD trunk events, which fall into the event numbering range of 144 – 159. Additionally, since support for loop-start line signaling is all that is required by PacketCable only the ABCD events pertinent to loop-start signaling will be described in the following sections. Other line signaling types supported by GR-303 (i.e., ground start, loop reverse battery, coin, multi-party, FXS and FXO) are considered out-of-scope for PacketCable GR-303 applications.

6.2.1.2.2 Normal Processing

The following table describes the GR-303 loop-start ABCD signaling events and their mapping to and from RFC 2833 named telephony events. Note that for ABCD signaling, “events” are really states.

Table 3. GR-303 to RFC 2833 ABCD Event Mappings

GR-303 ABCD EVENT	EVENT MEANING	MAPPING TO/FROM RFC 2833
UPSTREAM GR-303 ABCD EVENTS		
Loop closure	Line is off-hook	Off-hook is implied by presence of RTP voice packets (<u>no RTP event is needed for off-hook</u>)
Loop open	Line is on-hook	On-hook RTP packets replace null RTP voice packets at voice packet rate
DOWNSTREAM GR-303 ABCD EVENTS		
Ringling	Apply ring voltage to line	Ringling RTP packets replace null RTP voice packets at voice packet rate
Normal battery	Apply normal battery to line (this is the default state for provisioned MTA lines)	Normal battery is implied by presence of RTP voice packets (<u>no RTP event is needed for normal battery</u>)
Reverse battery	Apply reverse battery to line (not used in PacketCable application; used for reporting of far end answer for applications like hotel/motel PABX)	Not used (<u>no RTP event is needed</u>)
Loop current feed open LCFO	Apply open loop to line (used in forward disconnect, VMWI and telemetry data/suppressed ringing feature scenarios)	LCFO RTP packets replace null RTP voice packets at voice packet rate

The PacketCable implementation of RFC 2833 ABCD events takes advantage of the mutual exclusivity of several ABCD states with the presence of RTP voice. For example, the ringing state is mutually exclusive with voice transmission in the downstream direction, so RFC 2833 ringing events can be sent in place of

RTP voice packets for as long as the ringing condition persists. The normal battery state is implied by the presence of RTP voice packets in the downstream direction, making the use of RFC 2833's normal battery ABCD event unnecessary. In the upstream direction, the on-hook state is mutually exclusive with voice after an initial off-hook during an NCS call connection. Consequently, RFC 2833 ABCD on-hook events can be sent in place of RTP voice packets for as long as the on-hook condition persists. It is noted here that the initial state of a connection is that voice RTP is being sent upstream, even if the line is on-hook. This is done to support the possibility of upstream on-hook data transmission features (per Telcordia GR-30-CORE). As a result, three RFC 2833 off-hook packets are required to be sent on every on-hook to off-hook transition during an active NCS call connection.

There are several reasons for the RFC 2833 optimizations. First, it is desirable to not impose additional bandwidth usage requirements on an Unsolicited Grant Service DOCSIS service flow being used for voice traffic. RFC 2833 signaling sent simultaneously with RTP voice packets would cause a larger grant to be needed. Second, unnecessary replacement of voice packets with signaling packets is also undesirable, and this approach minimizes such replacement. A positive side effect of these optimizations is that state information is sent with a great deal of redundancy, minimizing the chances of bursty packet loss causing undesirable results.

The following shows an example of a subscriber hook-flash and the resulting RFC 2833 signaling between the MTA and IPDT.

Table 4. Example RFC 2833 Use – Hook-flash Reporting

Time (milliseconds)	Line state detected at MTA	RTP packet sent to IPDT	NCS message traffic between IPDT and MTA
0	off-hook	voice	
10	off-hook	voice	
20	off-hook	voice	
30	off-hook	voice	
40	on-hook	on-hook	
50	on-hook	on-hook	
60	on-hook	on-hook	
70	on-hook	on-hook	
80	on-hook	on-hook	
90	on-hook	on-hook	
On-hook continues for 800 ms (flash can be from 300-1100ms per GR-506-CORE)			
830	on-hook	on-hook	
840	off-hook	on-hook	End of event bit set in this packet
850	off-hook	off-hook	Optional NPTY(O:hf) sent to IPDT
860	off-hook	off-hook	
870	off-hook	off-hook	
880	off-hook	voice	
890	off-hook	voice	
900	off-hook	voice	
910	off-hook	voice	

6.2.1.2.3 Error Processing

Rules for error processing are needed to deal with prolonged periods of packet loss. This poses certain requirements on IPDT and MTA implementations.

6.2.1.2.3.1 MTA Processing for Lost Packets

Lost RFC 2833 ABCD ringing event packets must not extend ringing beyond a few packet inter-arrival periods. The MTA should not extend ringing beyond three packetization intervals when packet loss is detected during the reception of ringing events.

Lost RFC 2833 ABCD loop current feed open event packets must not extend open interval beyond a few packet inter-arrival periods. The MTA should not extend open loop beyond three packetization intervals when packet loss is detected during the reception of loop current feed open events.

6.2.1.2.3.2 IPDT Processing for Lost Packets

Lost RFC 2833 ABCD loop open event packets should extend the on-hook interval as long as packet loss persists. The IPDT should extend on-hook until valid voice RTP is received upstream when packet loss is detected during the reception of loop open events.

During NCS call connections which start with the MTA endpoint on-hook (i.e., incoming calls to the endpoint from the IPDT), loss of the initial RFC 2833 ABCD loop closed event packets can be detected at the IPDT when the associated NCS NTFY off-hook is received. Lost RFC 2833 ABCD loop closed event packets can be assumed by the presence of RTP voice at the IPDT subsequent to the first on-hook to off-hook transition.

6.2.1.2.4 RFC 2833 Packet Format and Fields Usage

The following table describes the RTP packet format and its usage to carry RFC 2833 named telephony events. The first 12 bytes are RTP header, while the last four bytes are the RFC 2833 portion.

Table 5. Example RFC 2833 Use – Hook-flash Reporting

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
V P X			CC			M		PT		Sequence Number																					
Timestamp																															
Synchronization Source																															
Event Number								E R		Volume								Duration													
Padding																															

- V - Version of RTP (=2)
- P - Padding (=0)
- X - Extension (=0)
- C - CSRC count (=0)
- M - Marker bit (=1 for beginning of new event; =0 otherwise)
- P - Payload type; dynamically established by call signaling protocol (NCS)
- Seq Number - Monotonically increasing number
- Timestamp - Gives timestamp of when event began; for PacketCable this would be the timestamp of the regularly scheduled voice packet that was replaced by the first event packet
- Synch Source - RTP synchronization source identifier (see RFC 1889); same as for RTP voice
- Event Number - RFC 2833 event number (144-159 for ABCD events; ABCD pattern 0000 maps to 144, ABCD pattern 1111 maps to 159, etc.); the events used by PacketCable are the following:
 - 144 – ring
 - 149 – loop open (on-hook)
 - 159 – loop current feed open
- E - End of event bit (=1 for end of event; =0 otherwise)
- R - Reserved (must be set to zero)
- Volume - Set to zero for ABCD events
- Duration - Gives total event duration (in timestamp units) relative to timestamp
- Padding - Pad the packet to match the size of the corresponding RTP voice packet that is being replaced. The padding value is undefined.

When a new event occurs that needs reporting via an RFC 2833 event (i.e., ring, loop open, loop closed, or loop current feed open), the first RFC 2833 event that is sent should have the M-bit set. Subsequent packets reporting the same event should have the same timestamp as that in the first reported event, and should have the Duration field incremented relative to the original timestamp. When the last packet reporting an event has been sent there must be a packet sent with the E-bit set prior to reporting the next state.

It is important to note there are two potential problems with the processing of the Duration field and E-bit with respect to their use for ABCD event reporting.

1. Duration field wrap-around

- It is possible to have an ABCD event last longer than eight seconds
- Eight seconds is maximum duration value assuming recommended timestamp time base of 8000hz
- At this time RFC 2833 does not specify what to do in this case; PacketCable suggests the following options:
- Allow duration wrap-around (i.e., treat the field as an unsigned integer)
- Specify use of duration equal to zero meaning ignore this field

The former method is considered more general. The IETF AVT Working Group has been alerted and is presently working on a solution to this problem. Current thinking is to go with the wrap-around approach.

2. End of event marker usage

- Setting of end of event marker causes unnecessary overwriting of one voice packet (I.e.-since next event must be detected before end of event can be sent)
- Since ABCD events imply states and are themselves mutually exclusive with each other, providing end of event before beginning of next event adds no value
- Options:
 - Use end of event marker and live with it
 - Do not use end of event marker for ABCD events (propose special use to IETF for inclusion in RFC 2833)

The former method is specified under RFC 2833, while the latter is preferable for this application. The IETF AVT Working Group has been alerted of the counterproductive nature of this bit when used with the ABCD trunk events and will be discussing the possibility of allowing the E-bit to be optional. It is recommended to use the E-bit until given an exception in the RFC.

To support transparent use of DOCSIS 1.1 Payload Header Suppression, RFC 2833 event packets should be padded to match the length of the corresponding RTP voice packet that would have otherwise been sent up or downstream. The padding value is undefined.

6.2.1.2.5 NCS Enhanced with RTP Named Telephony Events

The GR-303 Switched IP Telephony System architecture relies on general PacketCable NCS support for RFC2833 RTP Named Telephony Events. This support is reflected in several areas of NCS protocol usage:

- Line Supervision Media Package. The MTA and IPDT MUST support the Line Supervision Media Package, which defines the RTP events that must be supported for PacketCable GR-303 systems.
- LocalConnectionOptions. When connections are created between the IPDT and the MTA, the LocalConnectionOptions of the CreateConnection (CRCX) message includes the Line Supervision Media Package.

- AuditEndpoint. When auditing endpoint capabilities, the CMS can examine the audit response for Line Supervision Media Package support by the endpoint.
- AuditConnection. When auditing connections, the CMS can determine whether RTP events for GR-303 line supervision are being used.
- SDP Usage. The SDP for local and remote session descriptors are used to specify the support for RTP Named Telephony events in the media stream.

Additional detail and examples of usage are provided in the following subsections.

6.2.1.2.5.1 Line Control Signaling Package

This package provides support for transport of line supervision signals in the media stream using RFC2833 event packets in PacketCable GR-303 switched IP systems. The required signals are a subset of the RFC 2833 ABCD codes in the event range 144-159. The media format local connection option is used to describe the events that are detected by the MTA and signaled to the MG, and also to describe the events that are relayed by the MG to the MTA from the GR-303.

6.2.1.2.5.2 Local Connection Options

The Line Supervision Media Package local connection option uses the MGCP extension mechanism as described in <draft-andreasen-mgcp-rfc2705bis-00.txt>. Example:

```
L:p:10,a:PCMU;telephone-event fmtpl: "telephone-event 144,149,159"
```

6.2.1.2.5.3 Audit Endpoint

The NCS AuditEndpoint may be used by the CMS integrated into the IPDT to determine whether an MTA supports the Line Supervision Media Package. The package name "LCS" is returned in the audit endpoint response by MTAs that support the package. Example:

```
A: a:G729A, p:30-90, e:on, s:on, v:L;S;LCS,
m:sendonly;recvonly;sendrecv;inactive,
dq-gi,sc-st, sc-rtp: 00/51;03
```

In addition, the MTA MAY include the response for the generic media format local connection option, as defined for the Media Format Package. This response consists of the keyword fmlpl followed by a colon and a quoted string describing the telephone events supported. This mechanism is available to describe other media formats and payload types, as well. When used for telephony events, it must conform to RFC 2833. Example (using a proposed NCS Package):

```
A: a:G729A;telephone-event fmlpl:"telephone-event 144,149,159",
p:30-90, e:on, s:on, v:L;S;FM,
m:sendonly;recvonly;sendrecv;inactive,
dq-gi,sc-st, sc-rtp: 00/51;03
```

6.2.1.2.5.4 Audit Connection

The NCS AuditConnection message may be used by the CMS integrated into the IPDT to determine whether line supervision media events are in use. This is reflected in the local connection options returned in the audit response. Example:

```
L: p:10, a:PCMU;LCS/telephone-event
```

6.2.1.2.5.5 SDP Usage

The NCS Create Connection and Modify Connection messages may be used by the CMS to request that line supervision media events be used on a connection. The MTA reflects this in the session description that it returns in response to these commands, as well. Example:

```
v=0
o=- 4723891 7428910 IN IP4
128.96.63.25
s=-
c=IN IP4 128.96.63.25
t= 0 0
m=audio 1296 RTP/AVP 96 97
a=rtpmap:96 G726-32/8000
a=rtpmap:97 telephone-event/8000
a=fmtp:97 0-15,144,149,159
```

6.2.2 PSTN Signaling Framework

PSTN signaling interfaces are summarized in Table 2 (Pkt-c8 and Pkt-c9). These interfaces provide PacketCable subscribers with access to all PSTN-based voice and data services and with connectivity to PSTN and other PacketCable subscribers. All features are provided transparently via the PSTN so the addition of new voice services has no impact on the elements of the PacketCable architecture. Additionally, since every call made by an IPDT-based subscriber is served directly by an LDS, complex services like CALEA and E911 are available without impact to PacketCable components.

The PacketCable PSTN signaling framework consists of a Call Management Server that interacts with the PSTN via a PSTN gateway, which is further subdivided into three functional components: the Media Gateway (MG), Media Gateway Controller (MGC) and Signaling Gateway (SG). In the LCS Architecture, the IPDT contains the functionality of a thin CMS, as well as all of the functional components of the decomposed PSTN gateway. The components of the decomposed PSTN gateway will be described with respect to their usage in an IPDT.

The IPDT Media Gateway provides bearer connectivity and in-band robbed-bit ABCD signaling to and from the PSTN over the Pkt-c8 interface. Additionally, it relays ABCD signaling states between the PSTN robbed-bit signaling interface and the RFC 2833 ABCD event RTP packets used to signal line state information on the access side (Pkt-c10).

The IPDT Media Gateway Controller implements all the Media Gateway connection state and intelligence and controls the operation of the Media Gateway. This includes creation, modification and deletion of connections upon the instruction of the IPDT CMS.

The IPDT Signaling Gateway provides out-of-band TMC and EOC signaling interface connectivity with the PSTN. The Signaling Gateway terminates the lower layers of these signaling channels and relays application layer protocol information to and from the IPDT CMS.

The IPDT CMS provides overall call and connection control for the IPDT. It coordinates connection establishment and release through inter-working GR-303 TMC signaling messages with the connection and device control messages of the NCS protocol, and through interfacing with the Media Gateway Controller to indirectly control the Media Gateway portion of the IPDT.

The GR-303 sub-system also supports GR-303 Embedded Operations Channel (EOC) communications between the IPDT and LDS. The EOC provides provisioning, maintenance and fault management functionality for the IPDT. EOC operations do not require direct signaling inter-working with any PacketCable protocols. The CMS portion of the IPDT is responsible for managing the application-level EOC interface with the LDS.

6.3 Media Streams

The IETF standard RTP (RFC 1899 - Real-Time Transport Protocol) is used to transport all media streams in the PacketCable network. PacketCable utilizes the RTP profile for audio and video streams as defined in RFC 1990.

The primary media flow paths in the PacketCable LSC system are shown in Figure 7 and are further described in the text that follows.

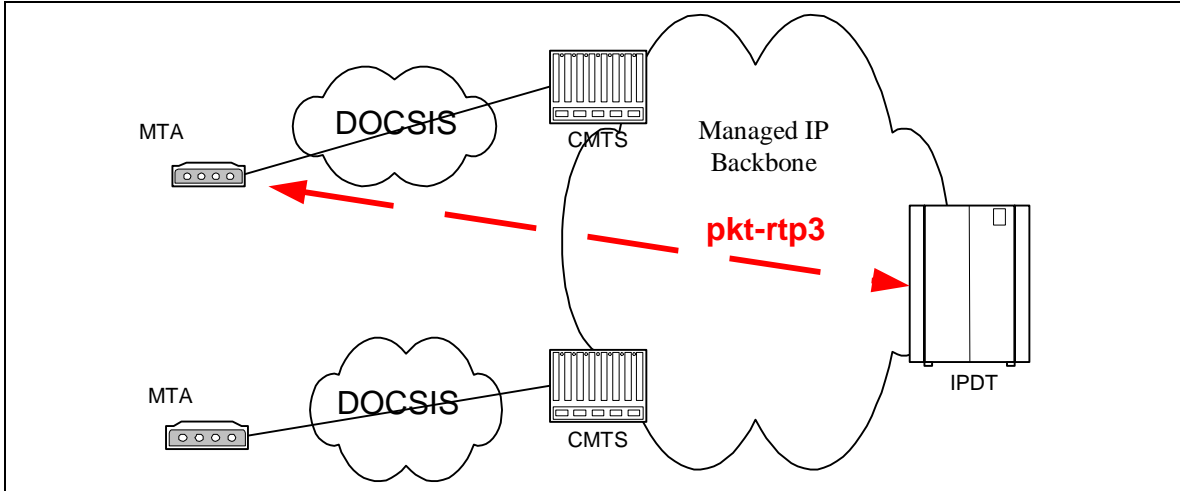


Figure 7. RTP Media Stream Flows in a PacketCable Network

pkt-rtp3: Media flow between the MG and the MTA. Includes, for example, tones, announcements, and PSTN media flow sent to the MTA from the Media Gateway. Also includes upstream media flow between the MTA and MG.

RTP encodes a single channel of multimedia information in a single direction. The standard calls for a 12-byte header with each packet. An 8-bit RTP Payload Type (PT) is defined to indicate which encoding algorithm is used. Most of the standard audio and video algorithms are assigned to particular PT values in the range 0 through 95. The range 96 through 127 is reserved for “dynamic” RTP payload types. The range 128 through 255 is reserved for private administration.

The packet format for RTP data transmitted over IP over Ethernet is depicted in Figure 8 below.

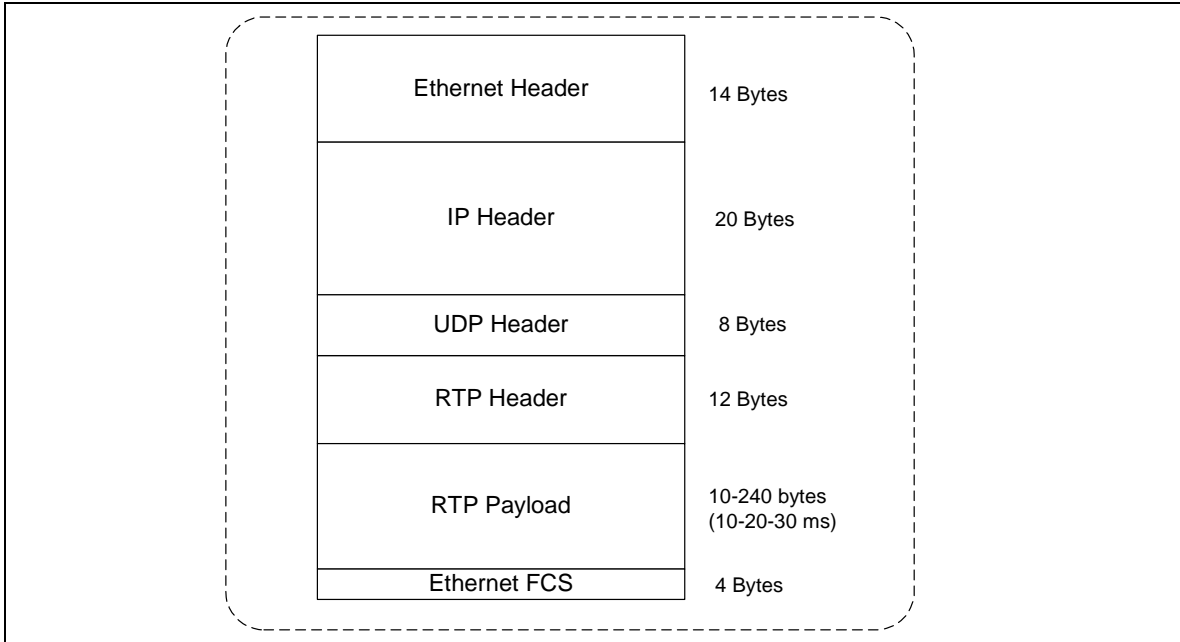


Figure 8. RTP Packet Format

The length of the RTP Payload as well as the frequency with which packets are transmitted depends on the algorithm as defined by the Payload Type field.

RTP sessions are established dynamically by the endpoints involved, so there is no “well-known” UDP port number. The Session Description Protocol (SDP) was developed by the IETF to communicate the particular IP address and UDP port an RTP session is using.

The packet header overhead of Ethernet, IP, UDP, and RTP is significant when compared to a typical RTP Payload size, which can be as small as 10 bytes for packetized voice. The DOCSIS 1.1 specification addresses this issue with a Payload Header Suppression feature for abbreviating common headers. To support transparent use of DOCSIS 1.1 Payload Header Suppression, RFC 2833 event packets should be padded to match the length of the corresponding RTP voice packet that would have otherwise been sent up or downstream. The padding value is undefined.

6.4 MTA Device Provisioning

MTA device provisioning is not impacted by the LCS Architecture. A description of MTA device provisioning can be found in the PacketCable MTA Device Provisioning Specification, PKT-SP-PROV-I02-010323.

6.5 Event Messages Interfaces

6.5.1 Event Message Framework

A PacketCable Event Message is a data record containing information about network usage and activities. A single Event Message may contain a complete set of data regarding usage or it may only contain part of the total usage information. When correlated by the Record Keeping System (RKS), information contained in multiple Event Messages provides a complete record of the service. This complete record of the service is often referred to as a Call Detail Record (CDR). Event Messages or CDRs may be sent to one or more back office applications such as a billing system, fraud detection system, or pre-paid services processor.

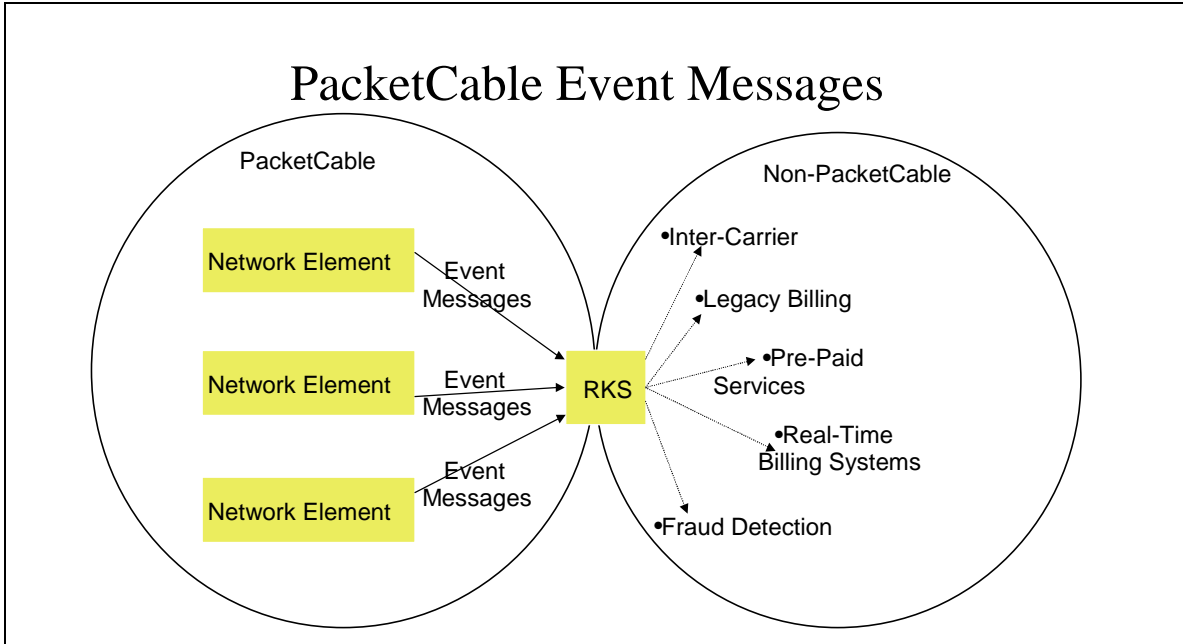


Figure 9. Representative Event Messages Architecture

PacketCable Event Message generation for subscriber billing is not required for services provided by the LCS system (ECN em-n-01085v2). In an LCS system, all CDRs for subscriber billing are generated by the LDS. The IPDT functional component of an LCS system supports minimal CMS and MGC functionality and does not contain sufficient information to generate PacketCable Event Messages, therefore the IPDT does not generate PacketCable Event Messages for services provided by the LCS system. Although the IPDT communicates with the CMTS, the IPDT does not send the Event-Generation-Info object in the GATE-SET message (ECN dqos-n-01072), therefore the CMTS is unable to generate Event Messages for services provided by the LCS system.

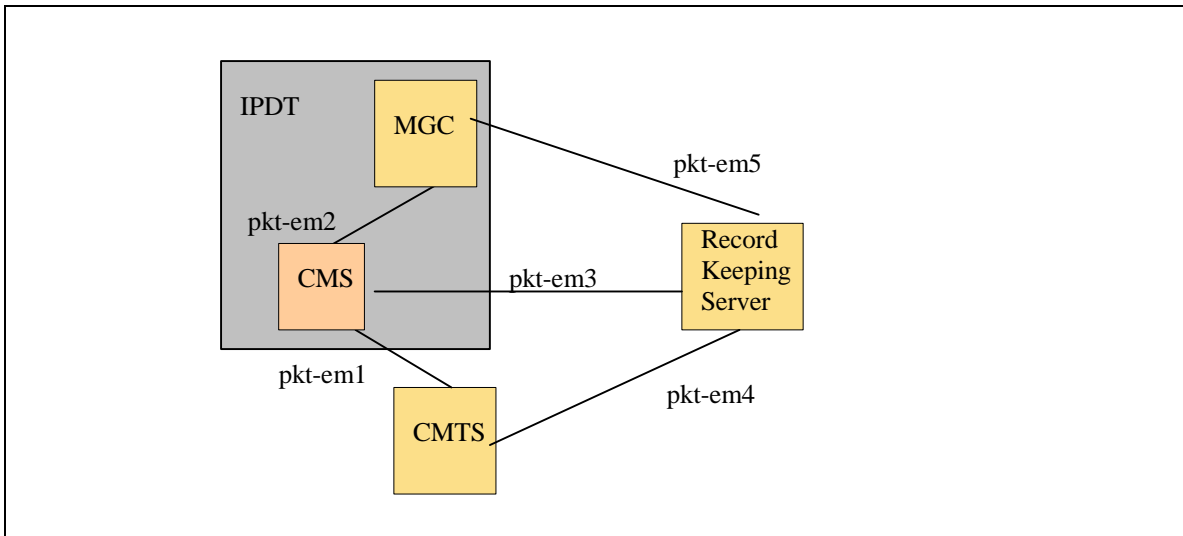


Figure 10. Event Message Interfaces

It should be noted here that the LSC system makes optional the previously mandatory Event Message interfaces between a CMS and an RKS, between an MGC and an RKS, and between a CMTS and an RKS.

An ECR has been generated for the DQoS specification noting these semantics (i.e., that the absence of Event-Generation-Info implies that the CMTS should not communicate information relating to a Gate with the RKS (ECN dqos-n-01072)). Since Event Messaging does not apply to LCS, the Event Messaging spec has also been annotated (ECN em-n-01085v2).

The following table describes the Event Message interfaces shown in the preceding diagram.

Table 6. Event Message Interfaces

Interface	PacketCable Functional Component	Description
pkt-em1	CMS-CMTS	DQoS Gate-Set message carrying Billing Correlation ID and other data required for CMTS to send Event Messages to an RKS. Note: This existing PacketCable interface is not used by the LSC system. DQoS signaling between and IPDT CMS and the CMTS will not include Event-Generation-Info objects.
Pkt-em2	CMS-MGC	Vendor-proprietary interface carrying Billing Correlation ID and other data required billing data. Either the CMS or MGC may originate a call and therefore need to create the Billing Correlation ID and send this data to the other. Note: Not an open interface in the LSC system; CMS and MGC are integrated within a single physical IPDT entity
Pkt-em3	CMS-RKS	RADIUS protocol carrying PacketCable Event Messages. Note: This existing PacketCable interface is not used by the LSC system.
Pkt-em4	CMTS-RKS	RADIUS protocol carrying PacketCable Event Messages. Note: This existing PacketCable interface is not used by the LSC system. DQoS signaling between and IPDT CMS and the CMTS will not include Event-Generation-Info objects. This omission will indicate to the CMTS to not send Event Messages to the RKS for calls so signaled.
Pkt-em5	MGC-RKS	RADIUS protocol carrying PacketCable Event Messages. Note: This existing PacketCable interface is not used by the LSC system.

6.6 Quality-of-Service

6.6.1 Dynamic Quality of Service Overview

PacketCable Dynamic QoS (D-QoS) utilizes the call signaling information at the time that the call is made to dynamically authorize resources for the call. Dynamic QoS prevents various theft of service attack types by integrating QoS messaging with other protocols and network elements. The network elements that are necessary for Dynamic QoS control in the LSC system are shown in Figure 11.

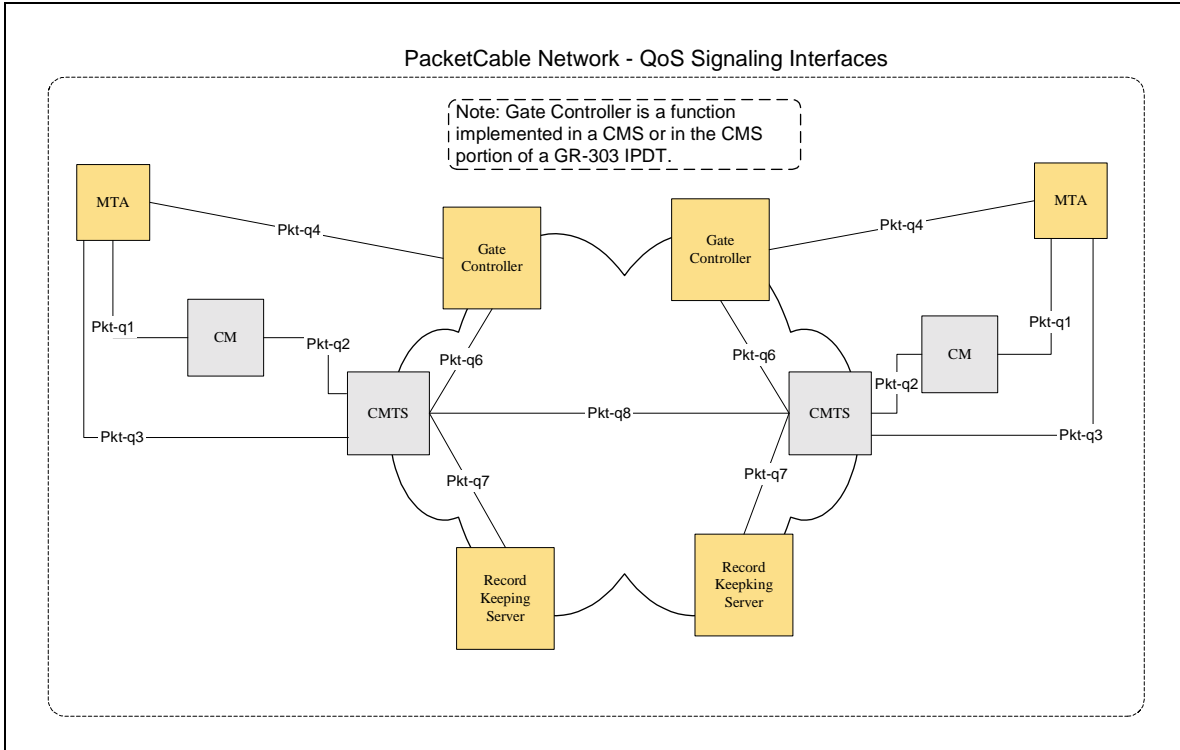


Figure 11. PacketCable QoS Signaling Interfaces

The function within the CMTS that performs traffic classification and enforces QoS policy on media streams is called a Gate. The Gate Controller (GC) element manages Gates for PacketCable media streams. The following key information is included in signaling between the GC and the CMTS:

Maximum Allowed QoS Envelope – The maximum allowed QoS envelope enumerates the maximum QoS resource (e.g., “2 grants of 160 bytes per 10ms”) the MTA is allowed to admit for a given media stream bearer flow. If the MTA requests a value greater than the parameters contained within the envelope the request will be denied.

Identity of the media stream endpoints – The GC/CMS authorizes the parties that are involved in a media stream bearer flow. Using this information the CMTS can police the data stream to ensure that the data stream is destined and originated from the parties that are authorized.

Billing Information – The GC/CMS creates opaque billing information that the CMTS does not have to decode. The information might be as simple as billing identity or the nature of the call. The CMTS forwards this billing information to the RKS as the call is activated or terminated. In the GR-303 subsystem all billing information is collected at the LDS. Consequently there is no need for the IPDT’s GC/CMS to create opaque billing information that the CMTS would need to forward to the RKS.

The role of each of the PacketCable components implementing D-QoS is as follows:

Call Management Server/Gate Controller – The CMS/GC is responsible for QoS authorization.

In the LCS application, the IPDT’s CMS/GC will authorize QoS on a per-call basis when the superordinate LDS first authorizes the call attempt. Once a call attempt has been authorized by the LDS, the IPDT CMS/GC will establish a Gate for the call with the CMTS and will signal DQoS parameters (e.g., Gate ID) via NCS to the MTA involved in the call.

At the end of the call, the LDS will initiate call disconnect. As a result the IPDT will control the release of resources by signaling to the MTA (i.e., via NCS), and the MTA will respond by signaling to the CMTS (i.e.-via DOCSIS 1.1 MAC or RSVP+). The CMTS will then ensure that the access network resources are properly released through interaction with the CM associated with the MTA involved in the call.

The IPDT must verify at the completion of all calls that the CMTS has properly released network resources.

CMTS – Using information supplied by the GC/CMS, the CMTS performs admission control on the QoS requests and at the same time polices the data stream to make sure that the data stream is originated and sent to authorized-media stream parties. The CMTS interacts with CM, MTA and IPDT. The responsibilities of CMTS with respect to each of these elements are:

CMTS to CM – The CMTS is responsible of setting up and tearing down service flows in such a way that the service level agreement it made with the MTA is met. Inasmuch as the CMTS does not trust the CM it polices the traffic from the CM such that the CM works in the way CMTS requested.

CMTS to MTA – The MTA makes dynamic requests for modification of QoS traffic parameters. When the CMTS receives the request it makes an authorization check to find out whether the requested characteristics are within the authorized QoS envelope and also whether the media stream endpoints are authorized. Then it provisions the QoS attributes for the RFI link on the CMTS and activates the appropriate QoS traffic parameters via signaling with the CM. When all the provisioning and authorization checks succeed the CMTS sends a success message to the GC/CMS indicating that MTA and CMTS are engaged in a Service Level Agreement.

CMTS to IPDT – The CMTS responds to requests from an IPDT to authorize voice call bandwidth for an MTA endpoint. It does so by creating a Gate for the call and by returning the Gate ID to the IPDT for subsequent use in signaling with the MTA (i.e., via NCS). The CMTS also responds to requests by the IPDT for Gate information and will delete Gates at the request of the IPDT during certain abnormal situations.

Cable Modem (CM) – Even though the CM is an untrusted entity the CM is responsible for the correct operation of the QoS link between itself and the CMTS. The CMTS makes sure that the CM cannot abuse the RFI link, but it is the responsibility of the CM to utilize the RFI link to provide services that are defined by the DOCSIS 1.1 specification.

MTA – The MTA is the entity to which the Service Level Agreement is provided by the access network. The MTA is responsible for the proper use of the QoS link. If it exceeds the traffic authorized by the SLA then the MTA will not receive the QoS characteristics that it requested.

For the LCS application, the MTA uses single stage QoS bandwidth allocation – when an originating or terminating call is processed the QoS resources are reserved and committed in a single step.

The following table identifies the component interfaces and how each interface is used in the Dynamic QoS Specification (DQoS). Two alternatives are shown for this specification: first a general interface that is applicable to either embedded or standalone MTAs; and second, an optional interface that is available only to embedded MTAs.

Table 7. QoS Interfaces for Standalone and Embedded MTAs

Interface	PacketCable Functional Components	DQoS Embedded/ Standalone MTA	D-QoS Embedded MTA (optional)
Pkt-q1	MTA – CM	N/A	E-MTA, MAC Control Service Interface
Pkt-q2	CM – CMTS (DOCSIS)	DOCSIS, CMTS-initiated	DOCSIS, CM-initiated
Pkt-q3	MTA – CMTS	RSVP+	N/A
Pkt-q4	MTA – GC/CMS	NCS	NCS
Pkt-q6	GC – CMTS	Gate Management	Gate Management
Pkt-q7	CMTS – RKS	Billing	Billing
Pkt-q8	CMTS – CMTS	Gate Management	Gate Management

The function of each QoS interface is described in Table 8.

Table 8. QoS Interfaces

Interface	Functional Components	Description
Pkt-q1	MTA – CM	<p>This interface is only defined for the embedded MTA. The interface decomposes into three sub-interfaces:</p> <p><i>Control</i>: used to manage DOCSIS service-flows and their associated QoS traffic parameters and classification rules.</p> <p><i>Synchronization</i>: used to synchronize packet and scheduling for minimization of latency and jitter.</p> <p><i>Transport</i>: used to process packets in the media stream and perform appropriate per-packet QoS processing.</p> <p>The MTA/CM interface is conceptually defined in Appendix E of the DOCSIS RFI specification. For standalone MTAs no instance of this interface is defined.</p>
Pkt-q2	CM – CMTS	<p>This is the DOCSIS QoS interface (control, scheduling, and transport). Control functions can be initiated from either the CM or the CMTS. However the CMTS is the final policy arbiter and granter of resources by performing admission control for the DOCSIS network.</p> <p>This interface is defined in the DOCSIS RFI specification.</p>
Pkt-q3	MTA – CMTS	<p>The interface is used to request bandwidth and QoS in terms of delay using standard RSVP and extensions specified in the DQoS specification. As a result of message exchanges between the MTA and CMTS, service flows are activated using the CMTS-originated signaling on interface Pkt-q2.</p>
Pkt-q4	MTA – CMS/GC	<p>Many parameters are signaled across this interface such as media stream, IP addresses, port numbers, and the selection of codec and packetization characteristics. NCS also provides specific parameters for signaling DQoS protocol information.</p> <p>Note: A subset of this existing PacketCable interface is used by the LSC system.</p>
Pkt-q6	CMS/GC – CMTS	<p>This interface is used to manage the dynamic Gates for media stream sessions. This interface enables the PacketCable network to request and authorize QoS.</p> <p>Note: A subset of this existing PacketCable interface is used by the LSC system.</p>
Pkt-q7	CMTS – RKS	<p>This interface is used by the CMTS to signal to the RKS all changes in session authorization and usage.</p> <p>Note: This interface is not used in the LSC system since billing is handled completely by the LDS.</p>
Pkt-q8	CMTS – CMTS	<p>This interface is used for coordination of resources between the CMTS of the local MTA and the CMTS of the remote MTA. The CMTS is responsible for the allocation and policing of local QoS resources. When a call traverses the PSTN (i.e.-there is no remote CMTS and remote MTA involved in a full IP call) the DQoS specification requires that a CMS must act as a CMTS proxy if it intends to fulfill the Pkt-q8 requirements of DQoS.</p> <p>Note: Analysis of the use of this interface within the LSC system has revealed that there is no benefit to supporting CMTS proxying from an IPDT. Therefore, this interface will not be implemented from an IPDT.</p>

6.6.2 Layer-Two vs. Layer-Four MTA QoS Signaling

QoS signaling from the MTA can be performed either at layer two (DOCSIS) or layer four (RSVP). Layer-two signaling is accessible to CM and CMTS devices that exist at the RF boundary of the DOCSIS access network. Layer-four signaling is required for devices that are one or more hops removed from the RF boundary of the DOCSIS access network.

Layer-two QoS signaling is initiated by an embedded MTA. The MTA utilizes the implicit interface for controlling the DOCSIS MAC service flows as suggested by Appendix E of the DOCSIS 1.1 RFI specification.

Layer-four QoS signaling is initiated by either an embedded MTA or standalone MTA. Enhanced RSVP is used for this signaling and is intercepted by the CMTS. The CMTS utilizes layer-two QoS signaling to communicate QoS signaling changes to the CM.

6.6.3 Dynamic Quality of Service Implementation

A minimal sub-set of the PacketCable Dynamic Quality of Service (DQoS) specification will need to be implemented in order to mitigate theft and denial of service within the LSC system. Capabilities already existing in DQoS will be used, with only very minor changes being required. Several DQoS spec clarifications and error corrections have been recommended for support of IPDT applications. (Reference ECNs: dqos-n-01069, dqos-n-01070, dqos-n-01071, dqos-n-01072, dqos-n-01073, and dqos-n-01082)

6.6.3.1 Dynamic Quality of Service Functional Requirements

6.6.3.1.1 Gate Establishment

IPDTs must establish DQoS Gates for every voice call by initiating DQoS *GATE-SET* message exchanges with the CMTS. The IPDT then must signal the GateID returned in the *GATE-SET-ACK* by the CMTS via NCS connection commands (e.g., CRCX) to the MTA associated with that call.

MTAs must use the GateID signaled in NCS connection commands received from the IPDT in subsequent resource reservation and committal message exchanges (e.g., DOCSIS E-MTA MAC DSA-REQ/RSP/ACK) with the CMTS.

Gate establishment makes use of existing DQoS, DOCSIS 1.1 E-MTA MAC and NCS messages and procedures.

6.6.3.1.2 Packet Classification and Filtering

The CMTS must perform packet classification and filtering based upon the signaled Gate Spec (i.e.-from IPDT-initiated *GATE-SET* messages) and through using the parameters received via DOCSIS E-MTA MAC messaging from the MTA. The Gate Spec received from an IPDT will be missing MTA upstream UDP source port and downstream UDP receive port information. The CMTS must obtain these ports via E-MTA MAC messaging and use them to populate packet classifiers for the respective upstream and downstream flows.

Packet classification and filtering is per existing DQoS and DOCSIS 1.1 messages and procedures.

6.6.3.1.3 Gate Closure

IPDTs will expect the CMTS to close Gates based upon signaling from the MTA (DOCSIS E-MTA MAC DSD-REQ) or based upon service flow inactivity timeouts. The IPDT must verify that Gates have been closed in all call release scenarios and must delete Gates that have not been properly closed by the CMTS (i.e., by using the *GATE-INFO* and *GATE-DELETE* message exchanges). The IPDT will always validate the identity of Gates that it deletes to minimize the possibility of deleting valid Gates.

Gate closure makes use of existing DQoS messages and procedures.

6.6.3.1.4 Miscellaneous

IPDTs must protect against deleting valid Gates. They will do this by comparing Gate information received in *GATE-INFO-ACK* messages against what is expected to be received for the given GateID.

Note that *GATE-INFO-ACK* messages should only be returned in two situations, both of which are abnormal cases. The first case is where the CMTS has not already closed the Gate, which would be the case when an MTA fails to initiate resource release when signaled to do so via a *DLCX* (e.g., via a *DSD-REQ* to the CMTS).

The second case is where the CMTS already closed the Gate associated with the IPDT's GateID but has already recycled the GateID for use in another Gate. In either case, the IPDT must be able to validate Gate information prior to sending a *GATE-DELETE* to force delete the Gate.

IPDTs must try to protect against denial of service attacks brought on by a misbehaving MTA requesting many more simultaneous UGS sessions than it is entitled to. This is done systematically by not allowing unprovisioned endpoints to initiate voice calls and by limiting provisioned endpoints to a single active voice call at a time.

Additional protection may be obtained through use of the Activity-Count object available in *GATE-ALLOC* and *GATE-SET* message exchanges. However, since it is possible that different endpoints on a given MTA could be drawing service from different CMSs (e.g., IPDT and/or full IP CMS) it is not practical for a given CMS to know the exact number of Gates to allow for an MTA. The Activity-Count object could be used to provide a practical upper bound on the number of open Gates. The IPDT may send an Activity-Count with a practical upper limit (e.g., four, eight, or sixteen Gates) or may choose to respond to received Activity-Counts that exceed this practical upper limit (i.e., by force deleting of the most recently allocated Gate).

6.6.3.2 Dynamic Quality of Service Interface Usage

6.6.3.2.1 DQoS Gate Control Messaging

LCS IPDTs must implement the COPS-based Gate Control interface of the DQoS specification. The required messages and parameters are enumerated below.

6.6.3.2.2 Required Gate Control Messages

The IPDT must implement DQoS messages needed for COPS association initialization and keep-alive, as well as messages necessary for setting and deleting Gates and querying for Gate information. The messages required for use by the IPDT are as follows:

COPS Association Initialization and Keep-Alive:

- COPS CLIENT-OPEN (CMTS to IPDT)
- COPS CLIENT-ACCEPT (IPDT to CMTS)
- COPS REQUEST (CMTS to IPDT)
- COPS KEEP-ALIVE (IPDT to CMTS and CMTS to IPDT)

DQoS Gate Control:

- DQoS GATE-SET (IPDT to CMTS)
- DQoS GATE-SET-ACK (CMTS to IPDT)
- DQoS GATE-SET-ERR (CMTS to IPDT)
- DQoS GATE-INFO (IPDT to CMTS)
- DQoS GATE-INFO-ACK (CMTS to IPDT)
- DQoS GATE-INFO-ERR (CMTS to IPDT)
- DQoS GATE-DELETE (IPDT to CMTS)
- DQoS GATE-DELETE-ACK (CMTS to IPDT)
- DQoS GATE-DELETE-ERR (CMTS to IPDT)

6.6.3.2.3 *Optional Gate Control Messages*

The IPDT may implement the following DQoS messages:

DQoS Gate Control:

DQoS GATE-ALLOC (IPDT to CMTS)

DQoS GATE-ALLOC-ACK (CMTS to IPDT)

DQoS GATE-ALLOC-ERR (CMTS to IPDT)

6.6.3.2.4 *Gate Control Object Usage*

The IPDT must implement the COPS and DQoS objects needed to support the COPS association initialization and keep-alive messages and also the DQoS Gate Control messages specified above. The necessary standard COPS objects must be implemented as specified in the COPS standard. Additional COPS objects specified by DQoS for Gate Control must be implemented by the IPDT. These are enumerated below:

Additional COPS Objects for Gate Control

Transaction-ID

Use is per DQoS specification

Subscriber-ID

Use is per DQoS specification

Gate-ID

Use is per DQoS specification

Activity-Count

When used in a GATE-ALLOC or GATE-SET message, this object specifies the number of Gates that can be simultaneously allocated to the indicated Subscriber-ID. When returned in a GATE-ALLOC-ACK or a GATE-SET-ACK it indicates the number of Gates assigned to a single subscriber. Activity-Count is useful for preventing denial of service attacks in that the number of simultaneous open Gates allowed at an MTA can be controlled.

Gate-Spec (two will exist per Gate)

Direction

Use is per DQoS specification

Protocol-ID

Use is per DQoS specification

Flags

Use is per DQoS specification

Session Class

0x01 = Normal priority VoIP session should be used by IPDTs

Source IP Address

Use is per DQoS specification

Destination IP Address

Use is per DQoS specification

Source Port

When source is an MTA this should be set to zero (wild-carded) since this information is not available to an IPDT at the time of sending GATE-SET

Destination Port

When the destination is an MTA this should be set to zero (wild-carded) since this information is not available to an IPDT at the time of sending GATE-SET

DS Field

Use is per DQoS specification

Timer-T1 value

Use is per DQoS specification

Timer-T2 value

Should be set to zero by the IPDT since T2 timing should not be done with single step reserve and commit

Token Bucket Rate [r]

Use is per DQoS specification

Token Bucket Size [b]

Use is per DQoS specification

Peak Data Rate [p]

Use is per DQoS specification

Minimum Policed Unit [m]

Use is per DQoS specification

Maximum Packet Size [M]

Use is per DQoS specification

Rate [R]

Use is per DQoS specification

Slack Term [S]

Use is per DQoS specification

Remote-Gate-Info

This optional object will not be used for IPDT applications. Its omission implies to the CMTS that no Gate Coordination will be done for this Gate, meaning no Gate-Open or Gate-Close message exchanges should be attempted by the CMTS and the CMTS should not expect to receive such messages from the Gate Controller (i.e., the IPDT).

Event-Generation-Info

This optional object will not be used for IPDT applications. Omission of this object will indicate that no event generation should be done for this Gate. Note: Since all accounting and billing are done by the LDS in an IPDT environment, interaction with a Record Keeping Server (RKS) must not be done for IPDT-controlled calls.

Media-Connection-Event-Info

This optional object will not be used for IPDT application. Its omission implies to the CMTS that Call-Answer and Call-Disconnect event messages are not to be used.

PacketCable-Error

Use is per DQoS specification

Electronic-Surveillance-Parameters

This optional object will not be used for IPDT applications. Note: Since all electronic surveillance of voice calls is handled by the LDS in an IPDT environment, interaction with an Electronic Surveillance Delivery Function must not be done for IPDT-controlled calls.

Session-Description-Parameters

This optional object will not be used for IPDT applications. Note: Since all accounting and billing are done by the LDS in an IPDT environment, interaction with a RKS must not be done for IPDT-controlled calls.

Gate-Coordination-Port

This optional object will not be used for IPDT applications. Its omission, along with omission of the Remote-Gate-Info object implies to the CMTS that no Gate Coordination will be done for this Gate.

6.6.3.2.5 DQoS Gate Coordination Messaging

LCS IPDTs will not use the RADIUS-based Gate Coordination interface of the specification. It was determined that the functionality existing in the COPS-based Gate Control interface is sufficient for LCS IPDTs to prevent all theft and denial of service scenarios that are foreseen.

This recommendation does not change the DQoS requirement on CMTS components, as they still must implement the Gate Coordination interface as part of the full IP PacketCable architecture. However, the CMTS will not be requested to initiate Gate Coordination message exchanges by an IPDT so this part of the DQoS state machine will not be exercised when used with the LCS application.

6.6.3.3 IPDT Implementation Requirements

The following lists QoS implementation requirements for the IPDT component of the LSC system.

1. The IPDT **MUST** implement the DQoS Gate Control interface and shall support the messages and objects specified in the DQoS Gate Control Messaging section of this report.
2. The IPDT **MAY** implement the DQoS Gate Coordination interface. If the IPDT chooses not to support DQoS Gate Coordination then the IPDT **MUST** omit the Remote-Gate-Info and Gate-Coordination-Port objects from GATE-SET messages that it sends to the CMTS. Omission of these objects should prevent the CMTS from performing Gate Coordination for the associated Gate.
3. The IPDT **MUST** signal the CMTS that Event Messages should not be generated for IPDT-initiated voice calls. It **MUST** do so by omitting the Event-Generation-Info object from the GATE-SET message to indicate that Event Messaging is not to be used for this Gate.
4. The IPDT **MUST** establish DQoS Gates for every voice call by initiating DQoS GATE-SET message exchanges with the CMTS, and by signaling the GateID returned by the CMTS in subsequent NCS connection control messaging exchanged with the associated MTA.
5. The contents of IPDT-generated Gate-Specs **MUST** contain the MTA IP address and IPDT IP address in the appropriate positions for the upstream and downstream Gates. It **MUST** also contain the IPDT receive UDP port as the destination port of the upstream Gate, and it **MUST** contain the IPDT transmit UDP Port in the source port of the downstream Gate. The IPDT **MUST** set the UDP port of the MTA to zero in the appropriate positions for the upstream and downstream Gates.
6. The IPDT **MUST** ensure that DQoS Gates it established have been properly closed by the CMTS for every voice call. The IPDT **MUST** initiate DQoS GATE-INFO message exchanges with the CMTS, and **MUST** delete any Gates that have been left open. The DQoS GATE-DELETE message **MUST** be used for these abnormal cases.
7. The IPDT **MUST** ensure that it minimizes the possibility of deleting any Gates that are properly open. This could occur if a CMTS has recycled a closed GateID too quickly. The IPDT **MUST** validate information it receives in a GATE-INFO-ACK for Gates that it expected to be closed to prevent improper deletions.
8. The IPDT **SHOULD** make use of the Activity-Count object to place a protective limit on the number of simultaneously active Gates an MTA can have. If this capability is used the IPDT **SHOULD** consider that it may not actually have knowledge of the true number of Gates that can be open simultaneously by an MTA (e.g., in cases where an MTA's endpoints are not homed on the same IPDT).
9. The IPDT **SHOULD** monitor VoIP activity and **SHOULD** initiate on-hook ABCD transmission toward the switch whenever it detects an inactivity period that is comparable to CMTS Timeout for Active QoS Parameters.

6.6.3.4 PacketCable Specification Impacts

The QoS philosophy of the IPDT application suggests that the following clarifications and changes be made to the DQoS specification.

1. Gate State Transition Diagram
 - a) GATE-DELETE/Send GATE-DELETE-ACK/Send Release is missing from the Committed -> End transition. In fact, transitions for GATE-DELETE in all state is missing and have been added to the State Transition Diagram. It has also been corrected per ECN dqos-n-01071
 - b) An Authorized -> Committed transition needs to be added to state that a RESERVE and COMMIT with no Gate Coordination necessary (i.e., due to missing optional Remote-Gate-Info object in prior GATE-SET message) is a valid transition. This has always been possible, but needs to be elaborated upon in the specification (this is a clarification to the current DQoS specification; reference ECN dqos-n-01070).
2. Commentary should be added to the DQoS specification indicating that CMTS implementations must use caution in recycling GateIDs so that GateID glare does not become an issue. Reuse of a given GateID soon after Gate closure by the CMTS is not desirable. (This is a revision to the current DQOS specification; reference ECN dqos-n-01073)
3. Text clarifying that the Remote-Gate-Info object is an optional DQoS COPS object needs to be added to supplement the message structures shown in the Gate Control Protocol Message Formats section (5.3 in draft I03) that show this. Explanations should be added that indicate that no Gate Coordination at all is to be done when no Remote-Gate-Info object is received in the GATE-SET message. (This is a clarification to the current DQOS specification; reference ECN dqos-n-01069.)

IPDT applications will not use Event Messages or RKS components so will not signal an RKS for record keeping purposes. (This is a clarification to the current DQOS specification; ECN dqos-n-01072). The Event Messages specification has been updated (ECN em-n-01085v2) to indicate that Event Message generation to an RKS is not required in all environments. Specifically, environments where billing and accounting are handled by a separate system should not have RKS requirements. Many MUST requirements should become conditional because of this.

6.7 Audio Servers

Announcements are typically needed for calls that do not complete. Additionally, they may be used to provide enhanced information services to the caller (e.g., calling card, n11 services, etc.). The signaling interfaces defined to support PacketCable Announcement Services are not needed by the LCS system. All announcement functionality is provided by the LDS for PacketCable LCS subscribers. The use of Audio Servers in an LCS architecture is reserved for further study.

6.8 Security

The Line Control Signaling (LCS) architecture is based on PacketCable 1.0 Security Specification version I03. The LCS architecture makes use of existing interfaces and security mechanisms already defined in the PacketCable Security Specification. The LCS architecture additionally defines two new interfaces which are outlined in Figure 12 and Table 9 below. To obtain an in-depth understanding of the security mechanisms the PacketCable Security Specification must be consulted using this section as a guide.

The I03 version of the Security Specification has been released but it will be replaced by an I04 version of the specification in Q3 2001. The I04 version supercedes the I03 version and contains textual changes that affect the wording of the specification. It also contains changes that are reflected in Table 9 but are incorrect in the I03 version of the Security Specification. Specifically, the descriptions for pkt-s0 and pkt-s1 have been corrected in the Security Specification I04 version as well as this Technical Report.

Certain functional elements in the full IP PacketCable 1.0 solution are removed in an LCS solution due to the Local Digital Switch (LDS) providing the same functionality. When a functional element is removed

then the protocol interfaces between the removed functional element and other elements are also removed. Hence, the need to secure those protocol interfaces is removed.

The LCS billing function is performed in the LDS. The Record Keeping Server (RKS) and its interfaces to the CMS and CMTS are removed from the LCS architecture.

The CALEA (Electronic Surveillance) function is performed in the LDS. The Delivery Function (DF) and its interfaces to the MG, CMS, and CMTS are removed from the LCS architecture.

Each of PacketCable’s protocol interfaces is subject to threats that could pose security risks to both the subscriber and service provider. The PacketCable architecture addresses these threats by specifying, for each defined protocol interface, the underlying security mechanisms (such as IPSec) that provide the protocol interface with the security services it requires, e.g., authentication, integrity, confidentiality.

For example, the media stream path may traverse a large number of potentially unknown Internet service and backbone service providers’ wires. As a result, the media stream may be vulnerable to malicious eavesdropping, resulting in a loss of communications privacy. PacketCable core security services include a mechanism for providing end-to-end encryption of RTP media streams, thus substantially reducing the threat to privacy.

The security services available through PacketCable’s core service layer are authentication, access control, integrity, confidentiality and non-repudiation. A PacketCable protocol interface may employ zero, one or more of these services to address its particular security requirements.

PacketCable security addresses the security requirements of each constituent protocol interface by:

1. identifying the threat model specific to each constituent protocol interface
2. identifying the security services (authentication, authorization, confidentiality, integrity, and non-repudiation) required to address the identified threats.
3. specifying the particular security mechanism providing the required security services.

The security mechanisms include both the security protocol (e.g., IPSec, RTP-layer security, and SNMPv3 security) and the supporting key management protocol (e.g., IKE, PKINIT/Kerberos) where applicable.

The following diagram provides a summary of all the PacketCable protocol interfaces and the mechanisms used to secure those interfaces.

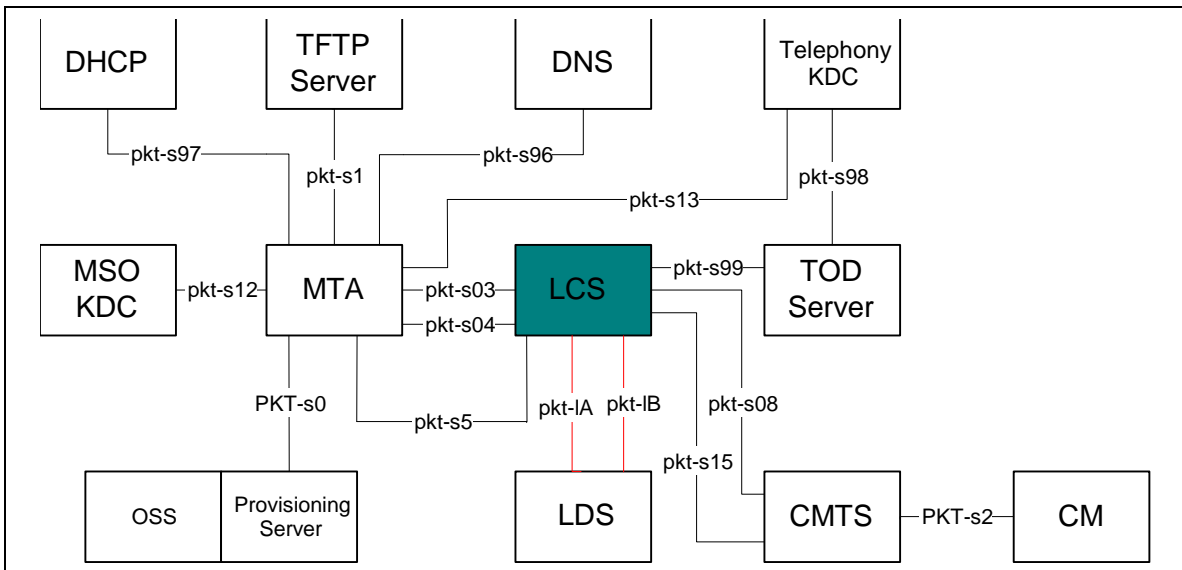


Figure 12. LCS Security Interface

Table 9 describes each of the interfaces shown in the above diagram.

The interface labeling in Table 9 is the same as those from the I03 Security Specification except where noted here. These interfaces are represented by solid black lines in the diagram above. Certain interfaces present in the Security Specification are not present in the LCS architecture. Therefore, they are not included here (i.e., pkt-s6, pkt-s7, etc.) Certain interfaces are described in the Security Specification but are not included in the corresponding Security Specification interface table. These interfaces are included in this table and are labeled in the 9x range (i.e., pkt-s99, pkt-s98, etc.). They are represented by solid black lines in the above diagram. New interfaces added by the LCS architecture are labeled in the form pkt-lx where x is a letter of the alphabet (i.e., pkt-lA, pkt-lB, etc.). These interfaces are represented by dashed red lines in the diagram above.

The description in Table 9 is the same as those in the Security Specification except where noted.

The LCS is comprised of the Gate Controller (GC), Call Management Server (CMS), Media Gateway Controller (MGC), Signaling Gateway (SG), and Media Gateway (MG) functional components.

Table 9. Security Interfaces

Interface	PacketCable Functional Components	Description
pkt-s0	MTA –PS/OSS	SNMPv3: Immediately after the DHCP sequence, the MTA performs Kerberos-based key management with the Provisioning Server to establish SNMPv3 keys. All SNMP messages are authenticated with privacy being optional.
pkt-s1	MTA – TFTP	TFTP: MTA Configuration file download. When the Provisioning Server sends an SNMP Set command to the MTA, it includes both the configuration name and the hash of the file. Later, when the MTA downloads the file, it authenticates the configuration file using the hash value. The configuration file may be optionally encrypted.
pkt-s2	CM – CMTS	DOCSIS 1.1: Secured with BPI+ using BPI Key Management. BPI+ privacy layer on the HFC link.
pkt-s3	MTA – MG	RTP: End-to-end media packets between an MTA and MG. Note that the MTA to MTA interface is not part of the LCS solution. RTP packets are encrypted directly with the chosen cipher. Message integrity is optionally provided by an HMAC (Hashed Message Authentication Code). Keys are randomly generated and exchanged by the two endpoints inside the signaling messages via the CMS.
pkt-s4	MTA – MG	RTCP: RTCP control protocol for RTP. Message integrity and encrypted by selected cipher. The RTCP keys are derived using the same secret negotiated during the RTP key management. No additional key management messages are needed or utilized.
Pkt-s5	MTA – CMS	NCS: Message integrity and privacy via IPsec. Key management is with Kerberos with PKINIT (public key initial authentication) extension.
Pkt-s8	CMS – CMTS	COPS: Gate Control messages between the GC and the CMTS, used to download QoS authorization to the CMTS. Message integrity and privacy provided with IPSEC. Key management is IKE with preshared keys (IKE-).
pkt-s12	MTA – MSO KDC	PKINIT: An AS-REQ message is sent to the KDC as before, except public key cryptography is used in the initial authentication step. The KDC verifies the certificate and issues a ticket granting ticket (TGT). The KDC authenticates the message using its public key signature.
Pkt-s13	MTA – Tel KDC	PKINIT: See pkt-s12 above.
Pkt-s15	CMS – CMTS	Gate Coordination - DQoS: message integrity is provided with an application-layer (Radius) authenticator. Keys are distributed by local CMS over COPS. This protocol interface is optional and not recommended for use in the LCS architecture.
Pkt-s96	MTA - DNS	DNS: Used by the MTA to obtain IP addresses for KDC and TFTP servers. Cryptographic methods are not specified on this interface for all PacketCable architectures. Securing this interface is at the discretion of the system operator.
Pkt-s97	MTA – DHCP	DHCP: used by the various network elements to obtain an IP address. Cryptographic methods are not specified on this

Interface	PacketCable Functional Components	Description
		interface for all PacketCable architectures. Securing this interface is at the discretion of the system operator.
Pkt-s98	Telephony KDC – TOD Server	Network Time Protocol (NTP): used by the Telephony KDC to obtain time from the Time Of Day (TOD) server. This protocol and interface is being specified in the I05 version of the security specification. The method to security this protocol has not been determined yet.
Pkt-s99	CMS – TOD Server	Network Time Protocol (NTP): used by the CMS to obtain time from the Time Of Day (TOD) server. This protocol and interface is being specified in the I05 version of the security specification. The method to security this protocol has not been determined yet.
Pkt-IA	MG – LDS	Analog: Used to send voice between the LDS and the LCS. New interface specified by the LCS. Cryptographic methods are not specified on this interface to secure it. Securing this interface is at the discretion of the system operator.
Pkt-IB	SG – LDS	GR-303: Used to send signaling and control information between the LDS and LCS. New interface specified by the LCS. Cryptographic methods are not specified on this interface to secure it. Securing this interface is at the discretion of the system operator.
pkt-s0	MTA – Provisioning App	SNMPv3: Immediately after the DHCP sequence, the MTA performs Kerberos-based key management with the Provisioning Server to establish SNMPv3 keys. All SNMP messages are authenticated with privacy being optional.
pkt-s1	MTA – TFTP	TFTP: MTA Configuration file download. When the Provisioning Server sends an SNMP Set command to the MTA, it includes both the configuration name and the hash of the file. Later, when the MTA downloads the file, it authenticates the configuration file using the hash value. The configuration file may be optionally encrypted.
pkt-s2	CM – CMTS	DOCSIS 1.1: Secured with BPI+ using BPI Key Management. BPI+ privacy layer on the HFC link.
pkt-s3	MTA – MG	RTP: End-to-end media packets between an MTA and MG. RTP packets are encrypted directly with the chosen cipher. Message integrity is optionally provided by an HMAC (Hashed Message Authentication Code). Keys are randomly generated and exchanged by the two endpoints inside the signaling messages via the CMS.
pkt-s4	MTA – MG	RTCP: End-to-end media control packets between an MTA and a MG. RTCP packets are encrypted directly with the chosen cipher. Message integrity is optionally provided by an HMAC (Hashed Message Authentication Code). Keys are randomly generated and exchanged by the two endpoints inside the signaling messages via CMS. Specified in the I03 specification. The Architecture team is studying whether this interface is required or optional.
Pkt-s5	MTA – CMS	NCS: Message integrity and privacy via IPSec. Key management is with Kerberos with PKINIT (public key initial authentication)

Interface	PacketCable Functional Components	Description
		extension.
Pkt-s6	GC – CMTS	COPS: Gate Control messages between the GC and the CMTS, used to download QoS authorization to the CMTS. Message integrity and privacy provided with IPSEC. Key management is IKE with preshared keys.
Pkt-s07	CMS – CMTS	Gate Coordination - DQoS: message integrity is provided with an application-layer (Radius) authenticator. Keys are distributed by local CMS over COPS. This protocol interface is optional and not recommended for use in the LCS architecture.
pkt-s08	MTA – MSO KDC	PKINIT: An AS-REQ message is sent to the KDC as before, except public key cryptography is used in the initial authentication step. The KDC verifies the certificate and issues a ticket granting ticket (TGT). The KDC authenticates the message using its public key signature.
Pkt-s09	MTA – Tel KDC	PKINIT: See pkt-s08 above.
Pkt-s10	CMS – TOD Server	Network Time Protocol (NTP): used by the CMS to obtain time from the Time Of Day (TOD) server. This protocol and interface is being specified in the I03 version of the security specification. The method to security this protocol has not been determined yet.
Pkt-s11	Telephony KDC – TOD Server	Network Time Protocol (NTP): used by the Telephony KDC to obtain time from the Time Of Day (TOD) server. This protocol and interface is being specified in the I03 version of the security specification. The method to security this protocol has not been determined yet.
Pkt-s12	MTA – DHCP	DHCP: used by the various network elements to obtain an IP address. Cryptographic methods are not specified on this interface to secure it. Securing this interface is at the discretion of the system operator.
Pkt-s13	MTA - DNS	DNS: Used by the MTA to obtain IP addresses for KDC and TFTP servers. While DNSSEC is being defined in the IETF cryptographic methods are not specified by PacketCable on this interface. Securing this interface is at the discretion of the system operator.
Pkt-s14	MG – LDS	Analog: Cryptographic methods are not specified on this interface to secure it. Securing this interface is at the discretion of the system operator.
Pkt-s15	SG – LDS	GR-303: Cryptographic methods are not specified on this interface to secure it. Securing this interface is at the discretion of the system operator.

Appendix A CALL FLOWS

The call flows presented in this section illustrate how the system processes calls including call origination from either end, call termination from either end, and for several representative CLASS features. The call flows include the following elements: a customer phone and CPE (MTA), CMTS, IPDT, and a Class 5 LDS.

The IPDT translates NCS directly to signaling on the GR-303's timeslot management channel (TMC). The IPDT also receives and generates GR-303 ABCD robbed-bit signals. The MTA communicates using NCS signaling to the Call Agent in the IPDT. The IPDT continually translates between the GR-303 bit-stream and the RTP packet stream, which carry bearer audio traffic, tones, digits, and other in-band signaling communications.

All call signaling messages are processed by the LDS over a GR-303 connection with the IPDT. From the perspective of the MTA and IPDT, all calls appear to be off-net calls (even calls that originate and terminate at the MTA and share the same CMTS). Thus the call flows comprise one standard call flow with variants for whether call origination is at the local MTA or behind the LDS.

CLASS features from the LDS to the MTA can be broadly classified as either on-hook transmissions or off-hook transmissions. On-hook transmissions include Caller ID and require special treatment in the IPDT and MTA. These call flows are treated in detail. Transmissions for off-hook call flows require no special treatment by the IPDT or the MTA. Call waiting and 3-way calling services also are shown, but call flows associated with other CLASS services and with special services are referred to the standard call flow since their operations simply involve standard call setups and terminations with the LDS.

Each of the call flows also illustrates the basic operations associated with PacketCable™ DQOS for the establishing of authorization gates at the CMTS and subsequent verification of gate information when the MTA manages its DOCSIS bandwidth.

A.1 Common Call Flows

A.1.1 Originate Call

This sequence describes successful initiation of a call by the subscriber.

The subscriber must be able to pick up the handset, hear dial tone, direct dial any domestic or international phone number, special numbers and operator services. The subscriber hears ringback tone as the remote phone is ringing. When the remote phone is taken off hook, a conversation can be held between the originating and terminating phones.

1: off hook()

The MTA has been instructed to detect and report off-hook and on-hook events after receiving a response to an RSIP (after reset) or DLCX (after tear down of a previous connection).

The subscriber picks up handset to initiate a call.

2: NTFY(o:hd)

The MTA in the CPE detects the user off- hook and notifies the Call Agent in the IPDT that the telephony endpoint is off hook. The MTA enters lockstep notification state on the endpoint while awaiting a response from the IPDT.

```
NTFY 2 aaln/1@rgwl.mso.net MGCP 1.0 NCS 1.0
X: 22331234
O: hd
```

3: 200 OK()

The Call Agent acknowledges the event notification, and SHOULD piggyback a notification request to instruct the MTA to ignore hook flash.

4: RQNT(R:hf(I))

The Call Agent sends a notification request instructing the MTA to ignore hook flash (NOTE: ignoring hook-flash is only requested when RFC 2833 is in use). This request SHOULD be piggybacked with the preceding acknowledgement. The MTA passes hook state changes through RTP event packets to the IPDT.

```
RQNT 276 aaln/1@rgwl.mso.net MGCP 1.0 NCS 1.0
X: 22331235
R: hf(I)
```

5: 200 OK()

The MTA acknowledges the RQNT.

6: Lookup(Line@FQDN, CRV)

The IPDT uses the analog access line number and FQDN of the telephony endpoint to look up the corresponding CRV. This lookup is not a DNS. Rather it is an internal database lookup. DNS lookups should normally not be done in realtime during call signaling operations. The IPDT must be provisioned with the relationships between telephony endpoint FQDN and CRV.

7: SETUP(crv)

The IPDT sends a SETUP message to the LDS over the GR-303 TMC interface.

8: CONNECT(crv, DS0)

The LDS assigns a DS0 for the call, and sends a CONNECT message over the GR-303 TMC interface to the IPDT to create a connection with the DS0 time slot assignment. At this point, a media path is established between the LDS and the IPDT on the assigned DS0.

9: Voice Path

The voice path between the IPDT and LDS in the IPDT-to-LDS direction is established.

10: ABCD Code(loop closed)

The IPDT relays the off hook status to the LDS in the assigned DS0 channel.

11: Create Access Network Connection

The CMS creates a voice path connection between the MTA and the IPDT, completing the path between the MTA and the LDS. See A.2.1 *Create Access Network Connection*.

12: CONNECT ACK()

The IPDT signals the LDS that the connection setup is complete by sending a CONNECT ACK message over the TMC channel.

13: ABCD Code(normal battery)

The LDS signals the normal battery condition to the IPDT.

14: In-band Tone (dial tone)

The LDS generates audio dial tone which is presented to the caller through the audio stream.

15: In-band Tone (dialed digits)

The caller proceeds to dial digits which are transmitted in the audio stream and collected by the LDS. At this point, the LDS is determining the state of the terminating party.

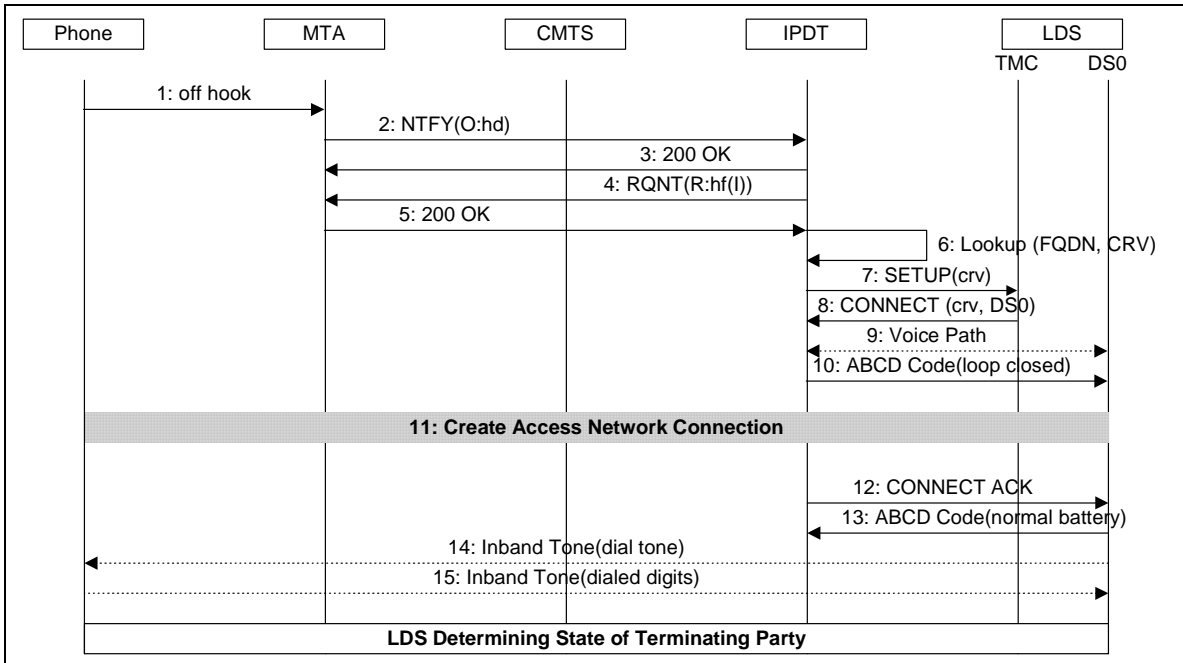


Figure 13. Originate Call, to Terminating Party State

A.1.2 Originate Call, Terminating Party Available

1: Originate Call

The on-net subscriber originates a call, A.1.1 *Originate Call*.

2: LDS Determines State of Terminating Party is Available

The LDS determines, through interactions with the PSTN, that the terminating party is available.

3: In-band Tone (ringback)

When the dial digits have been entered, the LDS initiates call procedures through the PSTN, eventually generating ringback tone in the audio stream to the caller when the called party's telephone starts ringing.

4: ABCD Code (normal battery)

5: Voice Channel Established (talk)

Two-way conversation proceeds.

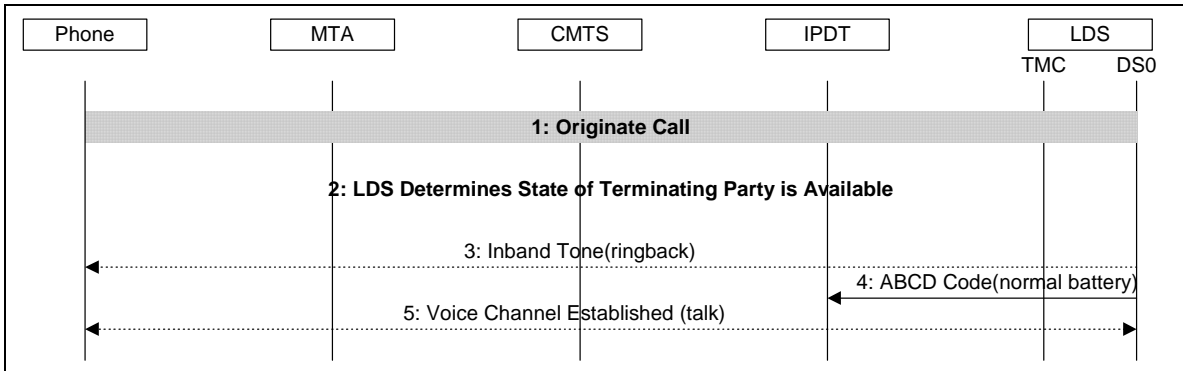


Figure 14. Originate Call, Terminating Party Available

A.1.3 Originate Call, Called Party Busy

This sequence describes initiation of a call by the subscriber when the called party is busy.

1: Originate Call

The on-net subscriber originates a call, A.1.1 *Originate Call*.

2: LDS Determines State of Terminating Party is Available

The LDS determines, through interactions with the PSTN, that the terminating party is busy.

3: In-band Tone (busy tone)

When the dialed digits have been entered, the LDS initiates call procedures through the PSTN, eventually generating busy tone in the audio stream to the caller when the called party's telephone is determined to be already off-hook.

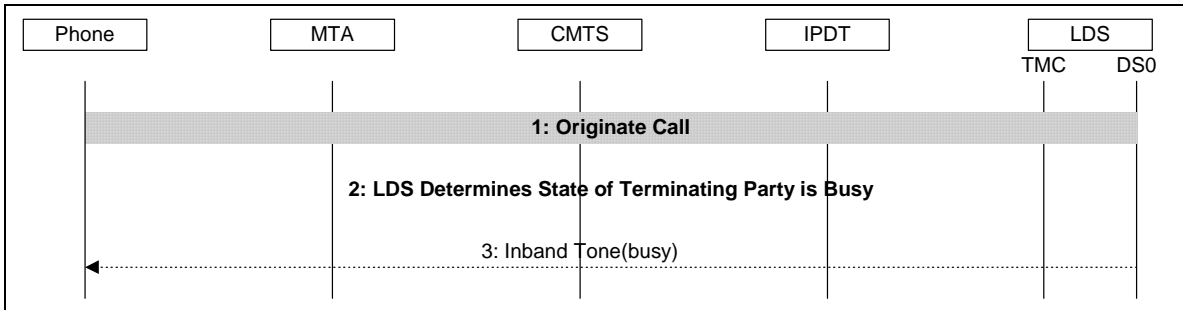


Figure 15. Originate Call, Called Party Busy

A.1.4 Originate Call, Glare Condition

This sequence describes initiation of a call by the subscriber at approximately the same time that the LDS attempts to have the subscriber receive a call from the PSTN, a condition commonly referred to as glare. The call coming from the PSTN takes precedence over the subscriber's action.

1: off hook()

Subscriber picks up handset to initiate a call.

2: NTFY(o:hd)

The MTA in the CPE detects the user off-hook and notifies the Call Agent in the IPDT that the telephony endpoint is off hook. The MTA enters lockstep notification state on the endpoint while awaiting a response from the IPDT.

```
NTFY 2 aaln/1@rgw1.mso.net MGCP 1.0 NCS 1.0
X: 22331234
O: hd
```

3: 200 OK()

The Call Agent acknowledges the event notification, and SHOULD piggyback a notification request to instruct the MTA to ignore hook flash.

4: RQNT(R:hf(I))

The Call Agent sends a notification request instructing the MTA to ignore hook flash (Note: Ignoring hook-flash is only requested when RFC 2833 is in use). This request SHOULD be piggybacked with the preceding acknowledgement. The MTA passes hook state changes through RTP event packets to the IPDT.

```
RQNT 276 aaln/1@rgw1.mso.net MGCP 1.0 NCS 1.0
X: 22331235
R: hf(I)
```

5: 200 OK()

The MTA acknowledges the RQNT.

6: FQDN Lookup(Line@FQDN,CRV)

The IPDT uses the analog access line and FQDN of the telephony endpoint to look up the corresponding CRV. This lookup is not a DNS. Rather it is an internal database lookup. DNS lookups should normally not be done in realtime during call signaling operations. The IPDT must be provisioned with the relationships between telephony endpoint FQDN and CRV.

7: SETUP(CRV,DS0)

LDS attempts to setup connection for call received from PSTN.

8: SETUP(CRV)

The IPDT sends a SETUP message to the LDS over the GR-303 TMC interface.

9: Create Access Network Connection

The CMS creates a voice path connection between the MTA and the IPDT, completing the path between the MTA and the LDS. See A.2.1 *Create Access Network Connection*.

10: CONNECT ()

The IPDT signals the LDS that the connection setup is complete by sending a CONNECT message over the TMC channel.

11: ABCD Code(normal battery)

The LDS signals the normal battery condition to the IPDT.

12: Voice Channel Established (talk)

The caller converses with calling party.

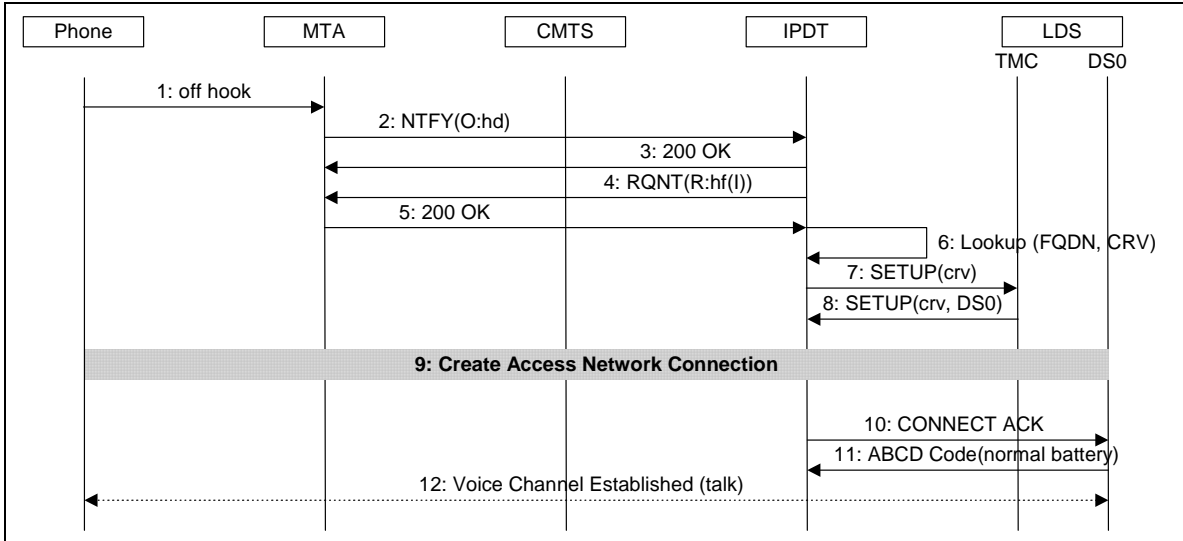


Figure 16. Originate Call, Glare Condition

A.1.5 Originate Call, Insufficient Resources

This sequence describes events that occur when there are insufficient HFC access network resources to establish a call. The subscriber is alerted to the condition by a congestion tone. The tone must be generated locally by the MTA because no audio channel has been established between the LDS and the MTA.

This situation differs from insufficient resources being available on the PSTN side of the call. When there are insufficient resources available to the LDS, it may generate congestion tone, or it may connect the subscriber to an announcement server. But in both cases the audio channel already has been established.

1: off hook()

Subscriber picks up handset to initiate a call.

2: NTFY(o:hd)

The MTA in the CPE detects the user off-hook and notifies the Call Agent in the IPDT that the telephony endpoint is off hook. The MTA enters lockstep notification state on the endpoint while awaiting a response from the IPDT.

```
NTFY 2 aaln/1@rgwl.mso.net MGCP 1.0 NCS 1.0
X: 22331234
O: hd
```

3: 200 OK()

The Call Agent in the IPDT acknowledges reception of the off hook notification.

4: RQNT(R:hf(I))

The Call Agent sends a notification request instructing the MTA to ignore hook flash (NOTE: ignoring hook-flash is only requested when RFC 2833 is in use). This request SHOULD be piggybacked with the preceding acknowledgement. The MTA passes hook state changes through RTP event packets to the IPDT.

```
RQNT 276 aaln/1@rgwl.mso.net MGCP 1.0 NCS 1.0
X: 22331235
R: hf(I)
```

5: 200 OK()

The MTA acknowledges the RQNT.

6: Lookup(Line@FQDN, CRV)

The IPDT uses the analog access line and FQDN of the telephony endpoint to look up the corresponding CRV. This lookup is not a DNS. Rather it is an internal database lookup. DNS lookups should normally not be done in realtime during call signaling operations. The IPDT must be provisioned with the relationships between telephony endpoint FQDN and CRV.

7: SETUP(crv)

The IPDT sends a SETUP message to the LDS over the GR-303 TMC interface.

8: CONNECT(crv, DS0)

The LDS selects a DS0 for the call, and sends a CONNECT message over the GR-303 TMC interface to the IPDT to create a connection with the DS0 time slot assignment.

9: GATE-SET()

The CMS in the IPDT establishes a gate at the CMTS defining the authorized bandwidth available to the MTA for connections. The operation is requested without gate coordination.

10: GATE-SET-ACK(GateID)

The CMTS acknowledges the gate set operation, returning the ID of the allocated gate.

11: CRCX(sendrecv,SDP,gateID)

With the time slot assignment made, the Call Agent in the IPDT sends a create connection request to the MTA in the CPE to establish a two-way media path between the IPDT and the MTA. The connection request specifies that the connection should be created send/receive and active. The request also includes the remote session description providing the RTP address at the IPDT to which audio and event packets are sent for this connection. When

DQoS is active, the CRCX includes the Gate ID set for the connection. The following example shows RFC 2833 usage being signaled. This would not be the case for the NCS-only translation.

```
CRCX 277 aaln/1@rgwl.mso.net MGCP 1.0 NCS 1.0
C: 01000997
L: p:10, a:PCMU;telephone-event fmp:"telephone-event
144,149,159", dq-gi:gateIDx
M: sendrecv
X: 22331236
R: hf(I)
S:
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0 96
a=rtpmap:0 PCMU/8000
a=rtpmap:96 telephone-event/8000
a=fmp:96 144,149,159
```

12: Process LocalConnection Options()

The MTA in the CPE parses and processes all of the parameters and options provided by the Call Agent in the CRCX request. This allows the MTA to determine how to request upstream bandwidth.

13: 100 PENDING()

When the MTA determines that it has the resources to create the requested connection, it sends a provisional acknowledgement to the Call Agent in the IPDT.

```
100 277 PENDING
I: 123E
v=0
o=- 87652 948357 IN IP4 128.2.3.4
s=-
c=IN IP4 128.2.3.4
t= 0 0
m=audio 8765 RTP/AVP 0 96
a=rtpmap:0 PCMU/8000
a=rtpmap:96 telephone-event/8000
a=fmp:96 144,159
```

14: DSA-REQ(admitted+active)

The MTA attempts to activate an unsolicited grant service (UGS) service flow or add a grant to an existing UGS service flow by sending a DSA-REQ or DSC-REQ to the CMTS. When DQoS is required, the request includes the Gate ID received in the CRCX request from the Call Agent.

15: DSA-RSP()

The CMTS verifies the request against the gate parameters, if necessary, but is unable to allocate the requested bandwidth on the HFC access network. The DSP-RSP that is sent back to the MTA reflects that the bandwidth request was denied.

16: DSA-ACK ()

The MTA acknowledges the DSA-RSP or DSC-RSP.

17: 250 NAK()

The MTA rejects the connection request.

18: RELEASE COMPLETE(Temporary Failure)

The IPDT terminates the connection attempt by sending a release message to the LDS.

19: GATE-DELETE()

Because the connection was not created successfully, there will be no subsequent connection deletion and subsequent deallocation of the gate by the CMTS. In this instance the Call Agent in the IPDT must explicitly delete the gate.

20: GATE-DELETE-ACK()

The CMTS closes and discards the gate, and acknowledges the gate deletion.

21: RQNT(reorder tone)

The Call Agent in the IPDT may instruct the MTA to play reorder tone to alert the subscriber that the call cannot be completed. The Call Agent also may elect not to apply this tone, but to continue to attempt to allocate bandwidth for the call.

22: reorder tone

The subscriber hears the alert signal.

23: on hook()

The subscriber places the handset back on hook and may retry the call at a later time.

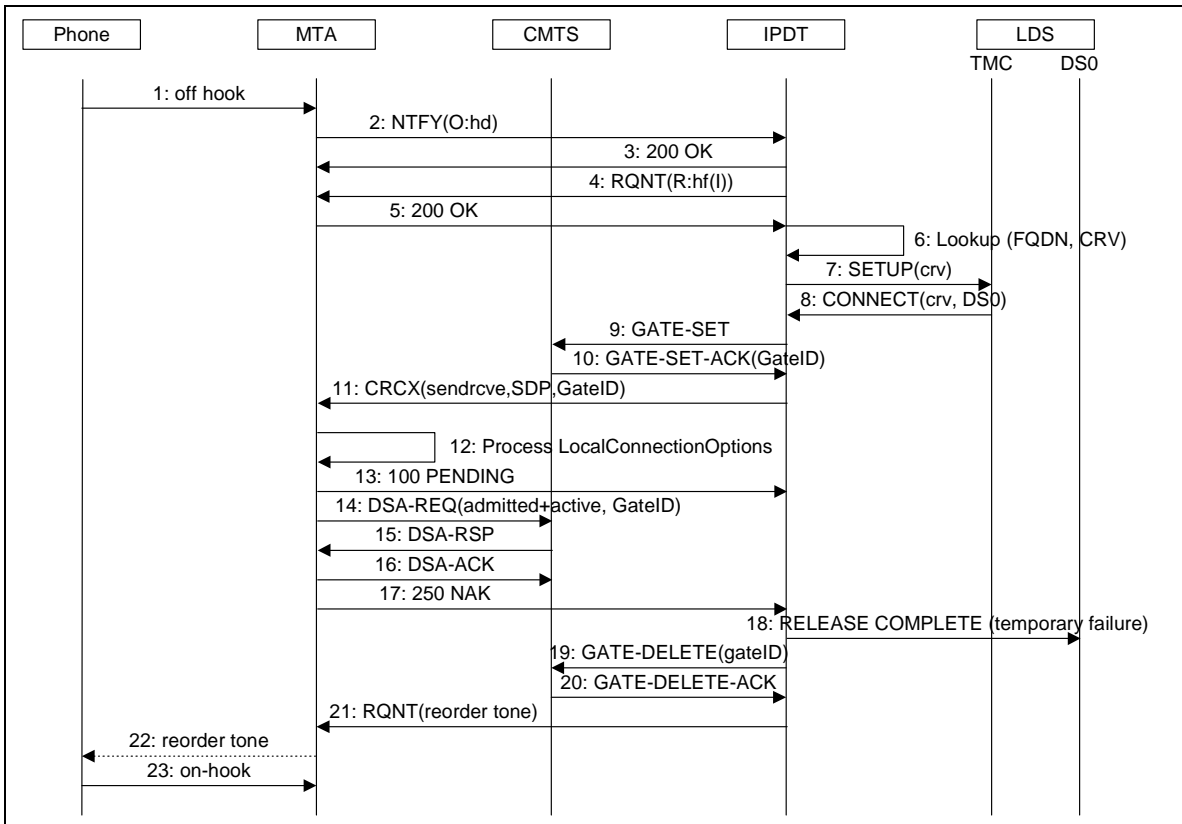


Figure 17. Originate Call, Insufficient Resources

A.1.6 Receive Call

This sequence describes receiving an incoming call from the PSTN.

1: SETUP(crv, DS0)

The MTA has been instructed to detect and notify off-hook events following an RSIP after reset or DLCX after tear down of a previous call.

The LDS attempts to setup connection for call received from PSTN. The SETUP message sent to the IPDT over the GR-303 TMC interface includes the call reference value and assigned DS0 for the connection.

2: CRV Lookup(CRV,Line@FQDN)

The IPDT uses the CRV supplied by the LDS in the SETUP message to find the analog access line and FQDN for the subscriber's telephone line.

3: Create Access Network Connection

The CMS creates a voice path connection between the MTA and the IPDT, completing the path between the MTA and the LDS. See A.2.1 *Create Access Network Connection*.

4: CONNECT ACK()

The IPDT signals the LDS that the connection setup is complete by sending a CONNECT ACK message over the TMC channel.

5: ABCD Code(normal battery)

The LDS signals the normal battery condition to the IPDT.

6: Ring MTA()

The LDS signals ring cadence through ABCD codes. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

7: off hook()

The subscriber picks up the telephone handset.

8: Notify Off-Hook()

The MTA signals the IPDT that the subscriber's handset is off-hook. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

9: RQNT(R:hf(I))

The Call Agent instructs the MTA to look for on hook (NOTE: ignoring hook-flash is only requested when RFC 2833 is in use). This request may be piggybacked with the preceding acknowledgement.

10: 200 OK()

The MTA acknowledges the request.

11: Voice Channel Established (talk)

The caller converses with calling party.

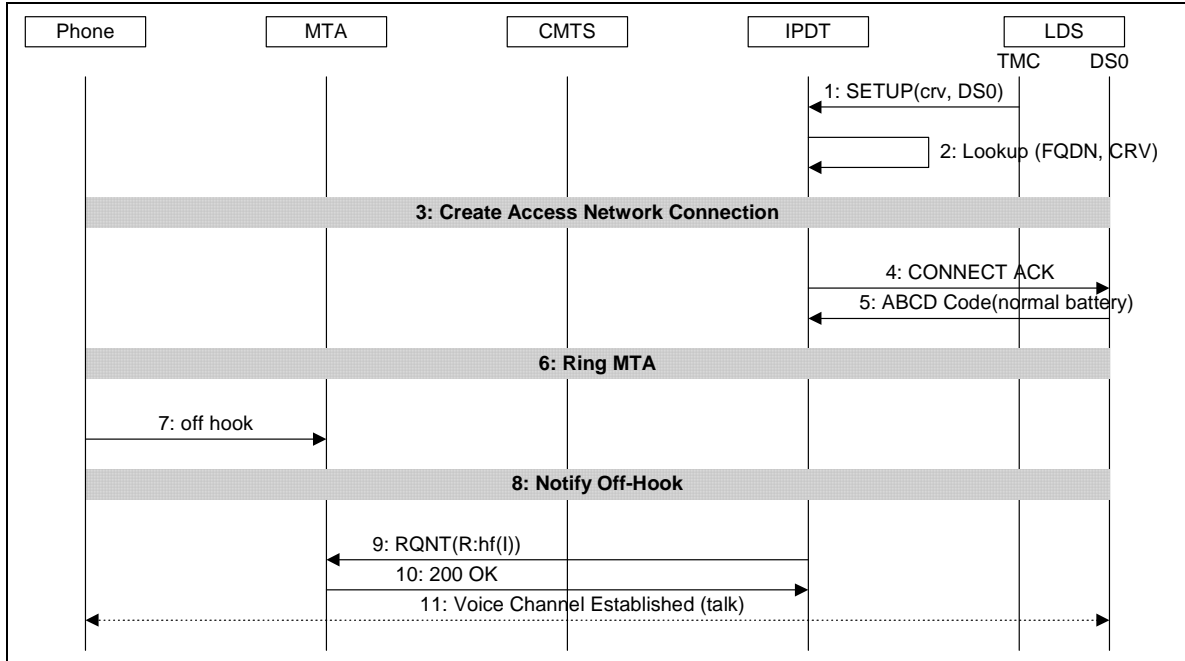


Figure 18. Receive Call

A.1.7 MTA Disconnect Call

This sequence describes operations that occur when the subscriber terminates a call.

1: Existing Call in Progress

The subscriber participates in an established call.

2: on hook()

Subscriber terminates call by placing his handset on hook.

3: Notify On-hook()

This notifies the IPDT that the subscriber's handset is on-hook. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

4: RQNT(R:hd)

The Call Agent sends the MTA a notification request for the MTA to start monitoring for off hook. This request SHOULD be piggybacked with the preceding acknowledgement.

5: 200 OK()

The MTA acknowledges the request.

6: DISCONNECT(crv)

The LDS commands the IPDT to disconnect the time slot. At this point, the media path between the IPDT and the LDS on the previously assigned DS0 is removed.

7: DLCX()

The Call Agent in the IPDT sends a delete connection command to the MTA.

```
DLCX 297 aaln/1@rgwl.mso.net MGCP 1.0 NCS 1.0
C: 01000997
I: 123F
X: 1112433
R:
```

8: DSD-REQ()

The MTA sends a DSD-REQ or DSC-REQ to the CMTS to release the bandwidth allocated for the connection.

9: DSD-RSP()

The CMTS releases the bandwidth allocated for the call, and sends a response to the MTA.

10: DSD-ACK()

The MTA acknowledges release of bandwidth.

11: 250 OK()

The MTA acknowledges the request and transmits the call statistics.

```
250 297 ok
P: PS=1245, OS=62345, PR=0, OR=0, PL=0, JI=0, LA=48
```

12: GATE-INFO(gateID)

Although the gate is closed and discarded by the CMTS during processing of the MTA's DSD-REQ or DSC-REQ that relinquishes the connection's bandwidth, the IPDT queries the CMTS for information about the gate. In the event that a rogue MTA has acknowledged the DLCX request without actually relinquishing the bandwidth.

13: GATE-INFO-ERR()

The CMTS acknowledges the request, but returns an error indicating that the gate does not exist. This is the response expected by the IPDT.

14: RELEASE(crv)

The IPDT sends a GR-303 TMC message to complete the release of the connection.

15: RELEASE COMPLETE(crvc)

The LDS acknowledges the release.

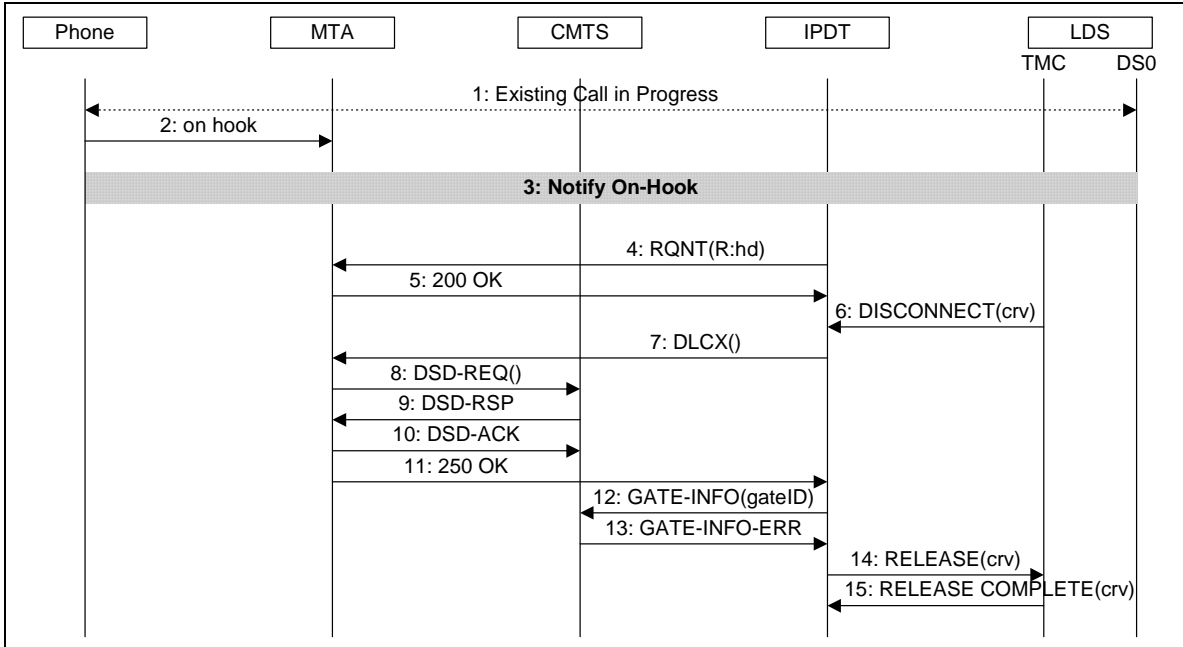


Figure 19. MTA Disconnect Call

A.1.8 Rogue MTA Disconnect Call

This sequence describes operations that occur when a subscriber using a modified MTA attempts to steal service by terminating a call with the PSTN, but maintaining the HFC access network bandwidth.

1: Existing Call in Progress

The subscriber participates in an established call.

2: on hook()

Subscriber terminates call by placing his handset on hook.

3: Notify On-Hook()

This notifies the IPDT that the subscriber's handset is on-hook. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

4: RQNT(R:hd)

The Call Agent sends the MTA a notification request for the MTA to start monitoring for off hook. This request SHOULD be piggybacked with the preceding acknowledgement.

5: 200 OK()

The MTA acknowledges the request.

6: DISCONNECT(crv)

The LDS commands the IPDT to disconnect the time slot. At this point, the media path between the IPDT and the LDS on the previously assigned DS0 is removed.

7: DLCX()

The Call Agent in the IPDT sends a delete connection command to the MTA in the CPE. When DQoS is active, this request includes the gate information.

```
DLCX 297 aaln/1@rgw1.mso.net MGCP 1.0 NCS 1.0
C: 01000997
I: 123F
X: 1112433
R:
```

8: 250 OK()

The rogue MTA acknowledges the request and transmits the call statistics, without actually relinquishing the allocated bandwidth.

```
250 297 ok
P: PS=1245, OS=62345, PR=0, OR=0, PL=0, JI=0, LA=48
```

9: GATE-INFO(gateID)

Although the gate is supposed to be closed and discarded by the CMTS during processing of the MTA's DSD-REQ or DSC-REQ that relinquishes the connection's bandwidth, the IPDT queries the CMTS for information about the gate. In this case, the rogue MTA has acknowledged the DLCX request without actually relinquishing the bandwidth.

10: GATE-INFO-ACK

Because the gate still is present, the CMTS provides the IPDT with the requested information.

11: GATE-DELETE()

To prevent theft of service, the IPDT commands the CMTS to delete the gate.

12: GATE-DELETE-ACK()

The CMTS acknowledges the request.

13: RELEASE(crv)

The IPDT sends a GR-303 TMC message to complete the release of the connection.

14: RELEASE COMPLETE(crv)

The LDS acknowledges the release.

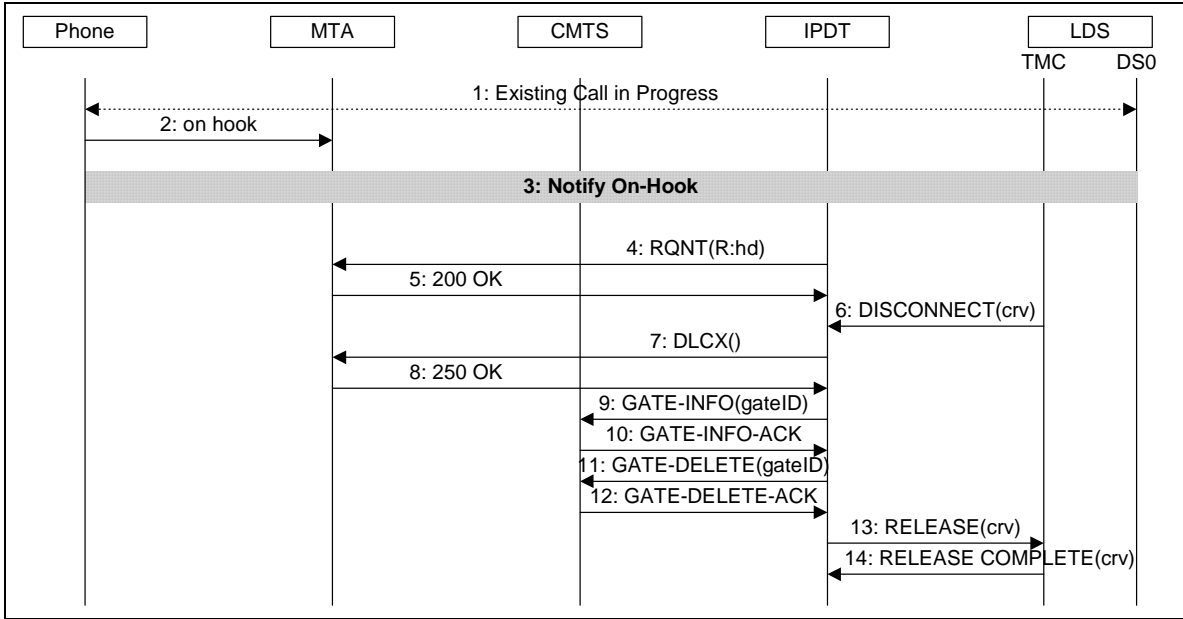


Figure 20. Rogue MTA Disconnect Call

A.1.9 PSTN Disconnect Call

This sequence describes operations that occur when the far-end party connected to the PSTN terminates a call.

1: Existing Conversation in Progress

The subscriber participates in an established call. The far end, PSTN connected party hangs up.

2: inband tone (dial tone)

The LDS generates dial tone inband.

3: on hook()

The subscriber may hang up promptly. The sequence then follows the normal sequence of events that occur when the MTA disconnects the call.

If the subscriber does not hang up promptly, the sequence continues as follows.

4: Timeout()

The LDS times out the release of the line.

5: Open Interval Long()

The LDS signals the open loop condition to the IPDT. This persists from 800 milliseconds to 1 second.

6: in-band tone (off hook warning)

The LDS generates an off-hook warning tone (a.k.a. "howler") to alert the subscriber to place the handset back on-hook.

7: on hook()

The subscriber may place the handset back on hook because of the off-hook warning tone prompt. The sequence then follows the normal sequence of events that occur when the MTA disconnects the call.

8: Timeout()

The LDS times out the off-hook warning tone.

9: DISCONNECT()

The LDS requests the IPDT to disconnect the line.

10: DLCX()

The Call Agent in the IPDT sends a delete connection command to the MTA.

```
DLCX 297 aaln/1@rgw1.mso.net MGCP 1.0 NCS 1.0
C: 01000997
I: 123F
X: 1112433
R:
```

11: DSD-REQ()

The MTA sends a DSD-REQ or DSC-REQ to the CMTS to release the bandwidth allocated for the connection.

12: DSD-RSP()

The CMTS releases the bandwidth allocated for the call, and sends a response to the MTA.

13: DSD-ACK()

The MTA acknowledges release of bandwidth.

14: 250 OK()

The MTA acknowledges the request and transmits the call statistics.

```
250 297 ok
P: PS=1245, OS=62345, PR=0, OR=0, PL=0, JI=0, LA=48
```

15: GATE-INFO()

Although the gate is closed and discarded by the CMTS during processing of the MTA's DSD-REQ or DSC-REQ that relinquishes the connection's bandwidth, the IPDT queries the CMTS to ensure that the gate has been deleted.

16: GATE-INFO-ERR()

The CMTS acknowledges the request, but returns an error indicating that the gate does not exist. This is the response expected by the IPDT.

17: RELEASE(crv)

The IPDT sends a GR-303 TMC message to complete the release of the connection.

18: RELEASE COMPLETE(crv)

The LDS acknowledges the release.

19: on hook()

Eventually, the subscriber will place the handset back on hook, but there is no connection with the Call Agent.

20: NTFY(on hook)

The MTA in the CPE notifies the Call Agent at the IPDT that the endpoint is on hook.

21: 200 OK()

The Call Agent acknowledges the notification.

22: RQNT(R:hd)

The Call Agent in the IPDT requests the MTA to notify next occurrence of off hook.

23: 200 OK()

The MTA acknowledges the notification request.

24: INFORMATION(on hook)

The IPDT notifies the LDS that the line now is on hook.

25: INFORMATION(on hook)

The LDS acknowledges the status.

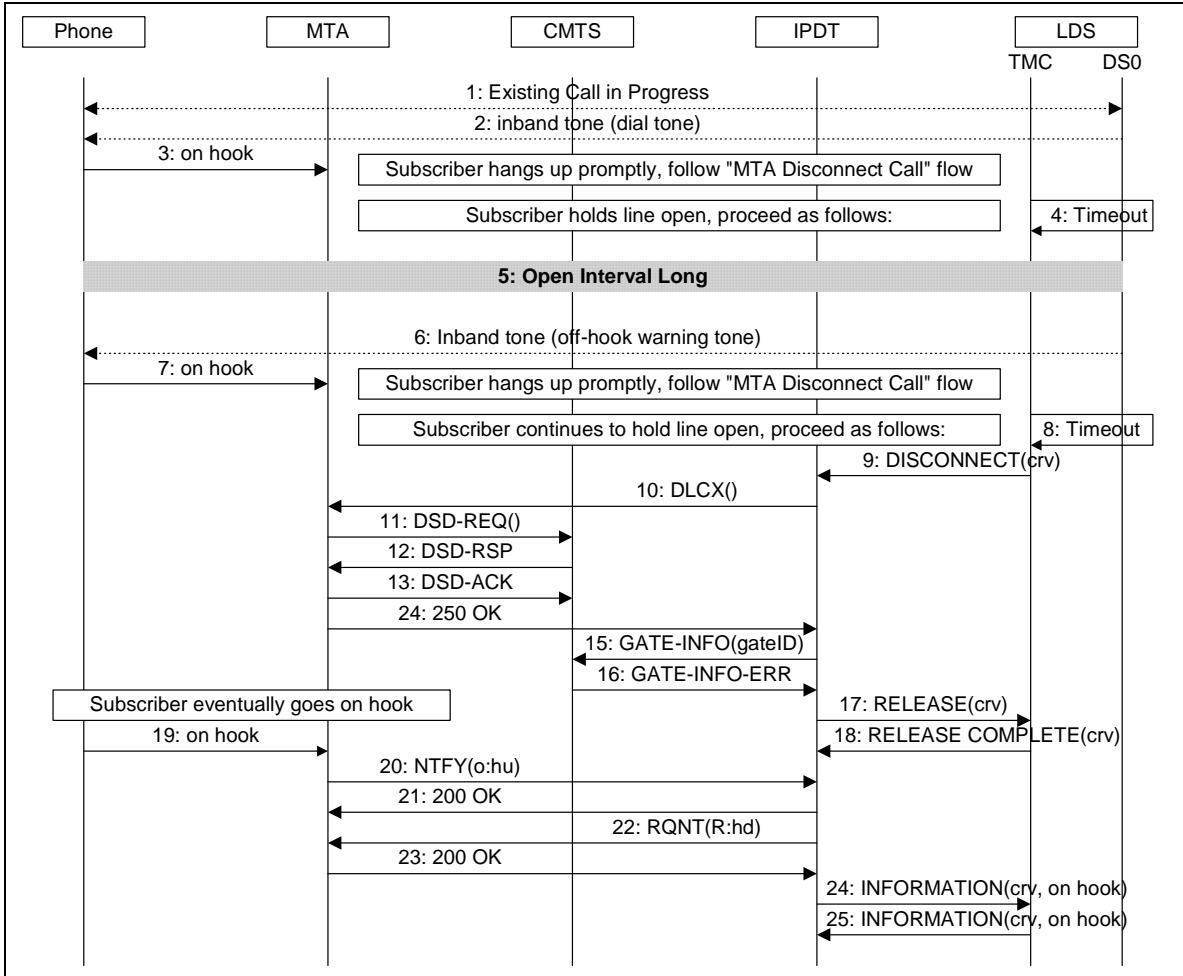


Figure 21. PSTN Disconnect Call

A.1.10 E911 Maintain Call

The switched IP telephony system (GR-303) accommodates emergency calls (E911) by keeping control of the E911 call within the LDS and PSTN. The E911 operator has priority in the handling of the call. This sequence describes the operations that occur when a subscriber hangs up during an E911 call but the E911 operator attempts to keep the subscriber's call active.

1: Existing Conversation in Progress

The subscriber participates in an established call.

2: on hook()

Subscriber terminates call by placing his handset on-hook.

3: Notify On-Hook()

The MTA notifies the IPDT that the subscriber's handset is on-hook. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

4: Alert Operator()

The LDS notifies the E911 operator that the subscriber's handset is on-hook. The operator requests call back.

5: RQNT(R:hd)

The Call Agent sends the MTA a notification request for the MTA to start monitoring for off-hook. This request SHOULD be piggybacked with the preceding acknowledgement.

6: 200 OK()

The MTA acknowledges the request.

7: Ring MTA()

The LDS signals ring cadence through ABCD codes. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

8: off-hook

The subscriber picks up the handset again.

9: Notify Off-Hook()

The MTA signals the IPDT that the subscriber's handset is off-hook. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

10: talk

The subscriber and operator resume their conversation.

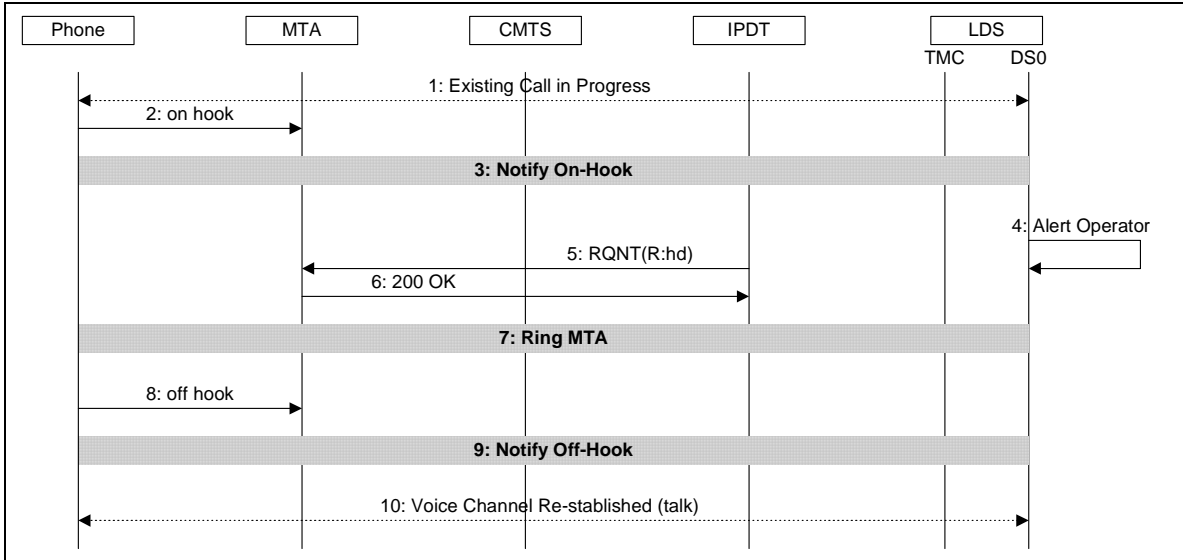


Figure 22. E911 Maintain Call

A.1.11 E911 Disconnect Call

The switched IP telephony system (GR-303) accommodates emergency calls (E911) by keeping control of the E911 call within the LDS and PSTN. The E911 operator has priority in the handling of the call. This sequence describes the operations that occur when the E911 operator elects to terminate an E911 call.

1: Existing Conversation in Progress

The subscriber participates in an established call.

2: on hook()

Subscriber terminates call by placing his handset on-hook.

3: Notify On-Hook()

The MTA notifies the IPDT that the subscriber's handset is on-hook. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

4: Alert Operator()

The LDS notifies the E911 operator that the subscriber's handset is on-hook. The operator disconnects the call.

5: RQNT(R:hd)

The Call Agent sends the MTA a notification request for the MTA to start monitoring for off-hook. This request SHOULD be piggybacked with the preceding acknowledgement.

6: 200 OK()

The MTA acknowledges the request.

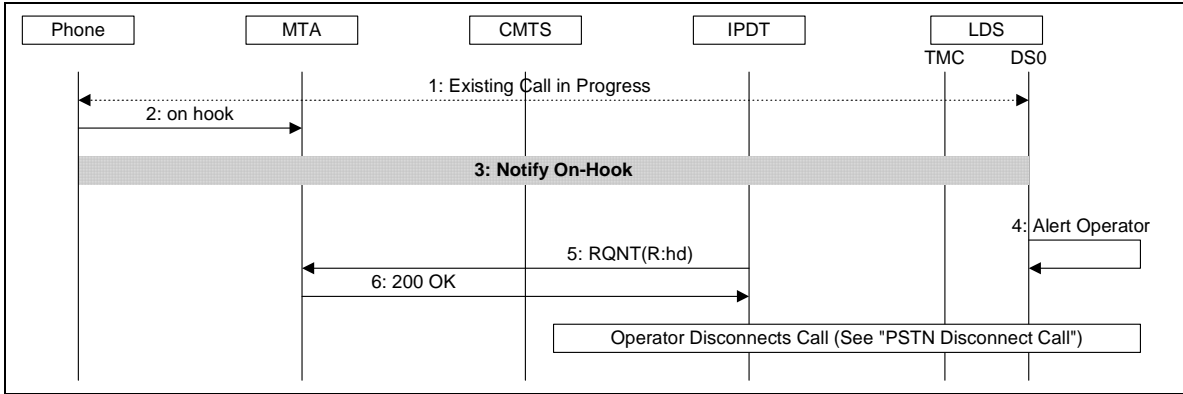


Figure 23. E911 Disconnect Call

A.1.12 Process Call Waiting

This sequence describes operations performed when the subscriber, who has subscribed to the Call Waiting feature, already is engaged in one call when a third party attempts to call. This sequence assumes that the Call Agent at the IPDT instructs the MTA at the CPE to ignore hook-flash (NOTE: ignoring hook-flash is only requested when RFC 2833 is in use). This allows the LDS to perform its normal hook flash timing and detection.

1: Existing Call in Progress

The subscriber is carrying on an existing conversation with another party.

2: Inband Tone (call waiting)

A third party attempts to call the subscriber. The LDS plays a call waiting tone that is audible to the subscriber while the calling party is hearing ringback, generated by the LDS to which the calling party is connected.

3: Hook Flash()

The subscriber performs a hook flash to switch from the current call to the incoming call. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

4: Switch()

The LDS switches the remote connections, placing the original call on hold and cutting through the new calling part. The existing DS0 on the DTF can be used to carry the conversation.

5: Talk

The subscriber carries on conversation with new calling party. The subscriber can switch between the calls by repetitively operating hook flash. The hook state transitions are sent to the LDS as described previously. The LDS switches between the calling parties.

6: Remote Disconnect()

Eventually, one or the other of the calling parties hangs up.

7: Comfort Noise

The LS disconnects the remote party. It maintains the connection to the subscriber and plays comfort noise or silence.

8: Hook Flash()

The subscriber operates hook flash to return to remaining calling party.

9: Switch()

The LDS switches back to the remaining call.

10: Talk

The subscriber resumes the remaining conversation.

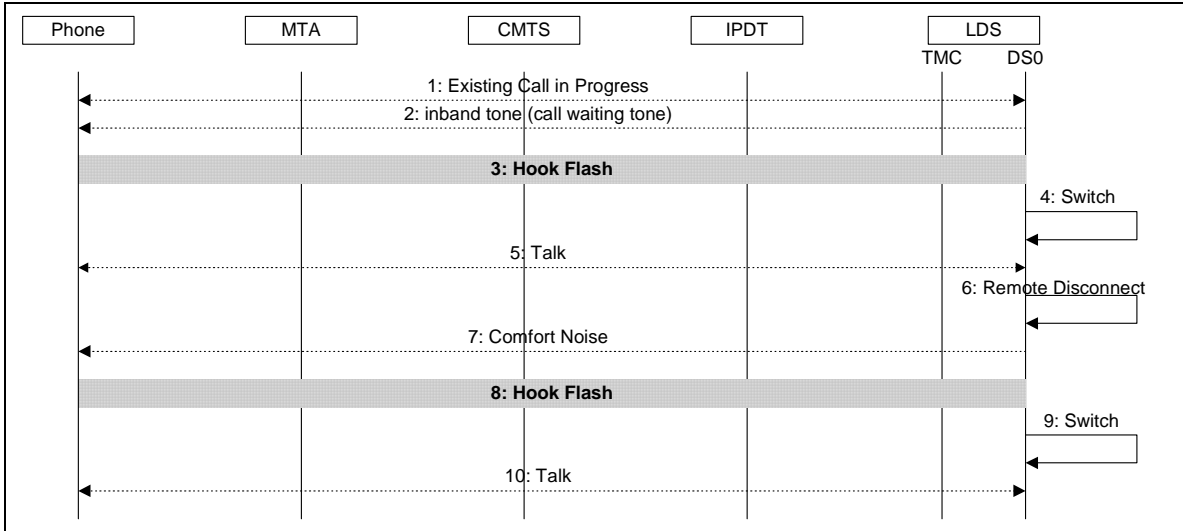


Figure 24. Process Call Waiting

A.1.13 Process 3-Way Call

The sequence describes the operations that occur when the subscriber uses a three-way calling feature. With an existing call in progress, the caller can operate hook flash to obtain a dial tone and initiate a new call. After the second call is established, the caller can operate hook flash again to conference both remote parties together.

1: Existing Call In Progress

The subscriber is carrying on an existing conversation with another party.

2: Hook Flash()

The subscriber performs a hook flash to place the existing call on hold and obtain a dial tone. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

3: inband tone (dial tone)

The LDS generates a dial tone heard by the subscriber.

4: inband tone (dialed digits)

The caller proceeds to dial digits which are transmitted in the audio stream and collected by the LDS.

5: inband tone (ringback)

When the called party can accept the call, the LDS generates in-band ringback tone heard by the caller.

6: Talk

The called party answers, and the caller and new called party can converse.

7: Hook Flash()

The subscriber performs a hook flash to conference together the active call and the call previously placed on hold.

8: Bridge()

The LDS bridges the parties together.

9: Talk

All three parties can converse.

10: Remote Disconnect()

One of the remote parties hangs up.

11: Silence

The LDS plays silence on the connection to prompt caller action.

12: Hook Flash()

The subscriber performs a hook flash to switch to the remaining call.

13: Switch()

The LDS reconnects the remaining remote party.

14: Talk

The subscriber resumes conversation with remaining remote party.



Figure 25. Process 3-Way Call

A.1.14 Visual Message Waiting Indication

This sequence describes operations that occur when data is transmitted from telephone network elements to the subscriber's equipment while the telephone line is on-hook.

The subscriber may subscribe to services that provide visual notifications for new messages, such as voice mail, fax/electronic mail and bulletin boards. When the subscriber's telephone is on-hook, notifications for these services may be sent. In the HFC access network, a connection is established to allow the notification messages to be transmitted from the LDS to the subscriber's equipment. The subscriber also may elect to have an abbreviated ring generated to provide an audible indication of these features if the message notification reached the LDS while the phone was off-hook.

1: Message Waiting

The feature server notifies the switch that there is a message waiting for the subscriber.

2: SETUP(crv,DS0)

The LDS requests the IPDT to setup a connection for the on-hook data transmission.

3: GATE-SET()

The CMS in the IPDT establishes a gate at the CMTS defining the authorized bandwidth available to the MTA for connections. The operation is requested without gate coordination.

4: GATE-SET-ACK(GateID)

The CMTS acknowledges the gate set operation, returning the ID of the allocated gate.

5: CRCX(sendrcv, gateID)

The Call Agent requests the MTA at the CPE to create a connection. The connection request specifies that the connection should be created send/receive and active. When DQoS is active, the CRCX includes the Gate ID set for the connection. The request also includes the remote session description providing the RTP address at the IPDT to which audio and event packets are sent for this connection.

6: Process LocalConnectionOptions()

The MTA in the CPE parses and processes all of the parameters and options provided by the Call Agent in the CRCX request. This allows the MTA to determine how to request upstream bandwidth.

7: 100 PENDING()

Because the MTA must allocate bandwidth for the connection, it sends a provisional response to the Call Agent.

8: DSA-REQ(gateID)

The MTA asks the CMTS for bandwidth for the connection.

9: DSA-RSP()

The CMTS verifies that the MTA is authorized, allocates bandwidth and sends a response.

10: DSA-ACK()

The MTA acknowledges the bandwidth allocation.

11: 200 OK(SDP)

The MTA acknowledges the connection request to the Call Agent.

12: ABCD Code(on hook)

The IPDT seizes the line and passes the current hook state to the LDS.

13: CONNECT ACK()

The IPDT signals the LDS that the connection setup is complete by sending a CONNECT ACK message over the TMC channel.

14: Open Interval Short

The LDS sends an open loop request to the IPDT through the ABCD signaling, preceded and followed by normal battery condition. The open loop alerts the subscriber's equipment, and is

maintained for 150-350 milliseconds. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

NOTE: The LDS sends EITHER the splash ring OR the open loop, but not both.

15: Message Waiting Modulation()

The LDS generates the message waiting notification as an in-band audio FSK spill. The FSK spill must occur in an interval between 300 milliseconds and 500 milliseconds after completion of the open loop by the MTA. The method used by the IPDT to relay the message waiting indication to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

16: RTP Audio Payload()

The IPDT translates the message waiting modulation to RTP audio payload packets, as with any other audio signal.

17: PCM()

The MTA plays out the RTP audio payload to the subscriber's equipment.

18: DISCONNECT()

The LDS requests the IPDT to delete the connection.

19: DLCX()

The Call Agent at the IPDT requests the MTA to delete the connection.

20: 100 PENDING()

Because the MTA must release DOCSIS bandwidth, it sends a provisional response.

21: DSD-REQ()

The MTA requests the CMTS to release the bandwidth allocated for the connection.

22: DSD-RSP()

The CMTS releases the bandwidth and sends a response.

23: DSD-ACK()

The MTA acknowledges release of the bandwidth.

24: 200 OK()

The MTA acknowledges the connection deletion to the Call Agent.

25: GATE-INFO()

Although the gate is closed and discarded by the CMTS during processing of the MTA's DSD-REQ or DSC-REQ that relinquishes the connection's bandwidth, the IPDT queries the CMTS to ensure that the gate has been deleted.

26: GATE-INFO-ERR()

The CMTS acknowledges the request, but returns an error indicating that the gate does not exist. This is the response expected by the IPDT.

27: RELEASE(crv)

The IPDT sends a TMC release message to notify the LDS that the connection has been deleted.

28: RELEASE COMPLETE(crv)

The LDS acknowledges the release.

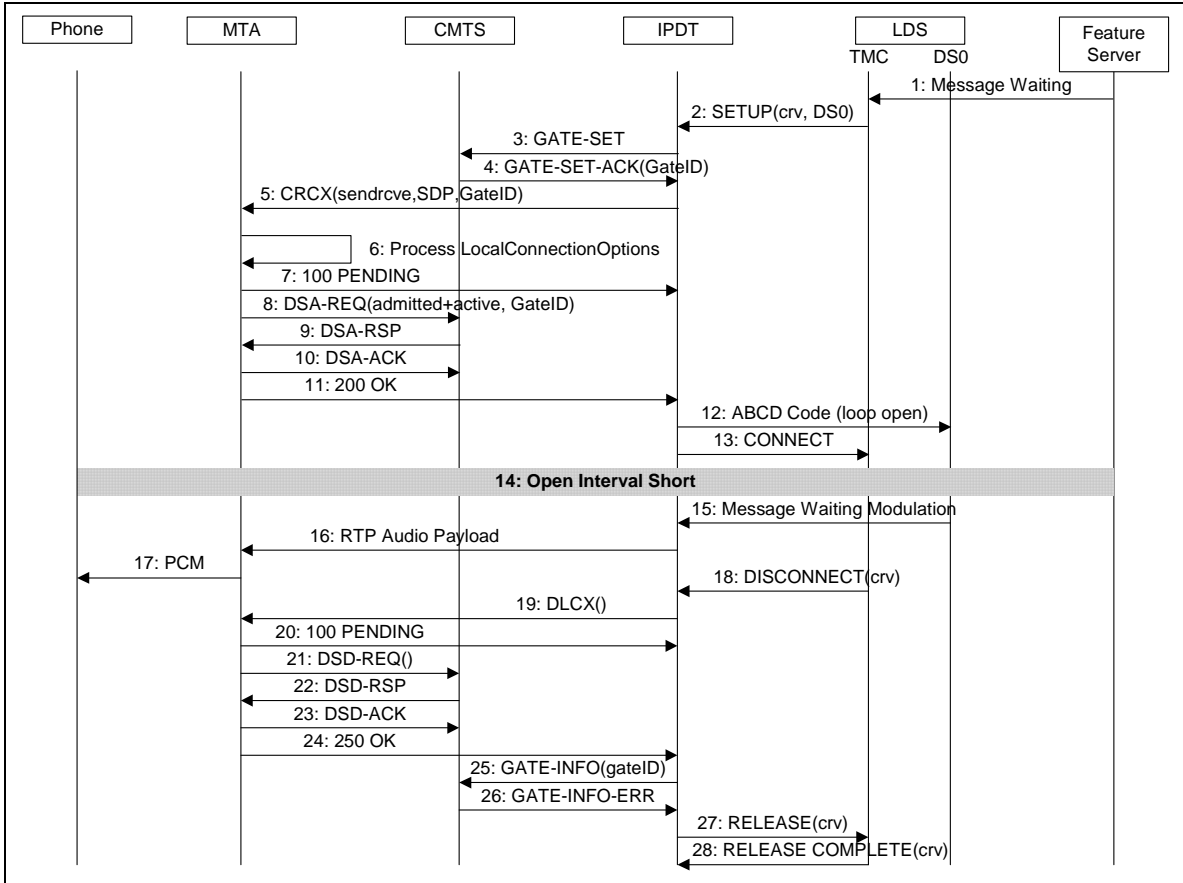


Figure 26. On-Hook Data Transmission

A.1.15 Telemetry Transport

This sequence describes the operations performed when a telemetry data system needs to interact with subscriber equipment to collect data.

1: Data Call

The telemetry system establishes a call through the PSTN to the LDS, targeted to subscriber equipment located on the HFC access network.

2: SETUP(crv,DS0)

The LDS initiates the alerting sequence by sending a SETUP containing the assigned call reference value and DS0 identifier on the GR-303 TMC channel to the IPDT. At this point a media path is established between the IPDT and LDS on the assigned DS0.

3: GATE-SET()

The CMS in the IPDT establishes a gate at the CMTS defining the authorized bandwidth available to the MTA for connections. The operation is requested without gate coordination.

4: GATE-SET-ACK(GateID)

The CMTS acknowledges the gate set operation, returning the ID of the allocated gate.

5: CRCX(sendrecv,gateID)

The Call Agent requests the MTA at the CPE to create a connection. The connection request specifies that the connection should be created send/receive and active. When DQoS is active, the CRCX includes the Gate ID set for the connection. The request also includes the remote session description providing the RTP address at the IPDT to which audio and event packets are sent for this connection.

6: Process LocalConnection Options()

The MTA in the CPE parses and processes all of the parameters and options provided by the Call Agent in the CRCX request. This allows the MTA to determine how to request upstream bandwidth.

7: 100 PENDING()

Because the MTA must allocate bandwidth for the connection, it sends a provisional response to the Call Agent.

8: DSA-REQ(gateID)

The MTA asks the CMTS for bandwidth for the connection.

9: DSA-RSP()

The CMTS verifies that the MTA is authorized, allocates bandwidth and sends a response.

10: DSA-ACK()

The MTA acknowledges the bandwidth allocation.

11: 200 OK(SDP)

The MTA acknowledges the connection request to the Call Agent.

12: ABCD Code(on hook)

The IPDT seizes the line and passes the current hook state to the LDS.

13: CONNECT(crv)

The IPDT signals the LDS that the connection setup is complete by sending a CONNECT ACK message over the TMC channel.

14: Voice Path

At this point the voice path is established between the telemetry data system and the MTA at the CPE.

15: Open Interval Short()

The LDS applies an open signal interval (OSI) of between 150 and 350 milliseconds to the line within the ABCD bits in the channel, preceded and followed by normal battery condition. This alerts the subscriber equipment to prepare for a telemetry operation. The method used by the IPDT to relay the line status to the MTA depends on use of NCS Translation signaling or NCS augmented with RTP named telephony events.

NOTE: The LDS sends EITHER splash ring OR the open loop, but not both.

16: off hook()

The subscriber equipment responds to the OSI by going off hook.

17: Notify Off-Hook()

The MTA immediately forwards the hook state change to the IPDT in RTP event packets in the audio stream.

18: RQNT(R:hf(i))

The IPDT acknowledges the event notification and instructs the MTA to ignore hook flash (NOTE: ignoring hook-flash is only requested when RFC 2833 is in use). This request SHOULD be piggybacked with the preceding acknowledgement.

19: 200 OK()

The MTA acknowledges the request.

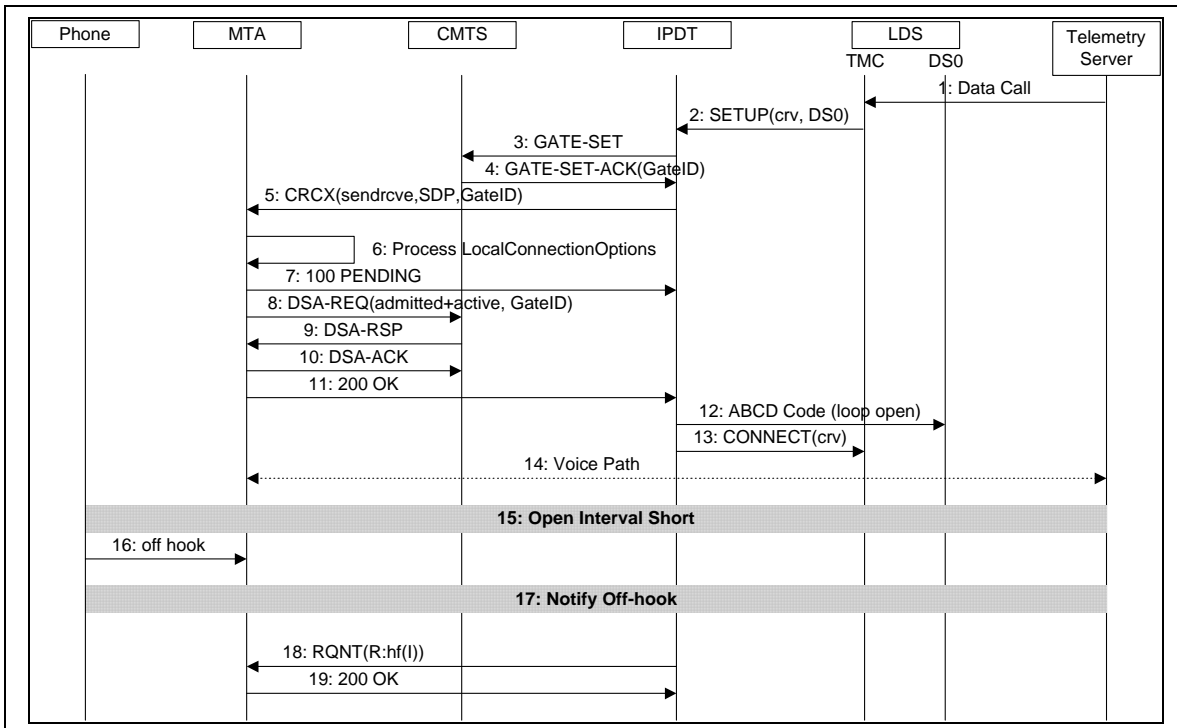


Figure 27. Telemetry Transport

A.1.16 Audit Endpoint

This sequence describes the operations performed by the Call Agent to determine the capabilities of the endpoints supported by the MTA. The responses from the MTA indicate support for the GR-303 NCS package.

1: RSIP(*,restart)

After completing its initialization, the MTA sends the Call Agent in the IPDT a restart in progress (RSIP) message indicating restart of all of its endpoints.

2: AUEP(*)

The Call Agent at the IPDT determines the number of endpoints present at the MTA by sending a wild-carded audit endpoint (AUEP) command:

```
AUEP 1231 *@rgwl.mso.net MGCP 1.0 NCS 1.0
```

3: 200 OK(endpoints)

The MTA responds, passing back a list of endpoint names.

```
200 1231 OK
aaln/1@rgwl.mso.net
aaln/2@rgwl.mso.net
aaln/3@rgwl.mso.net
aaln/4@rgwl.mso.net
```

4: AUEP(aaln/1)

The Call Agent now audits each endpoint for its capabilities. In a GR-303 system, the IPDT needs to determine whether the GR-303 package is supported.

```
AUEP 1232 aaln/1@rgwl.mso.net
F: A
```

5: 200 OK()

The MTA responds with the list of the endpoint's capabilities.

```
200 1232 OK
A: a:PCMU;telephone-event fmp:"telephone-event 144,149,159",
p:30-90, e:on, s:on, v:L;S;LCS,
m:sendonly;recvonly;sendrecv;inactive,
DQ-GI,SC-ST, SC-RTP: 00/51;03
```

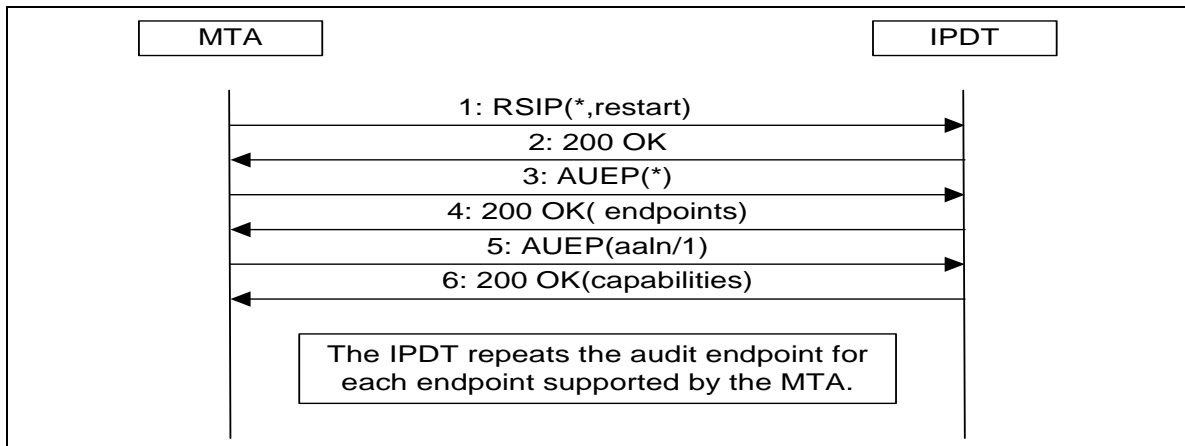


Figure 28. Audit Endpoint

A.1.17 Audit Connection

This sequence describes the operations performed by the Call Agent to determine the characteristics of active connections on an endpoint. The responses from the MTA indicate presence of RTP event packets on the connection.

1: AUEP()

The Call Agent at the IPDT audits a specific endpoint to obtain the list of connection identifiers for connections present on the endpoint, if any.

```
AUEP 1411 aaln/1@rgwl.mso.net
F: I
```

2: 200 OK()

The MTA responds with the list of connection identifiers currently active. In this example, just one connection is listed.

```
200 1411 OK
I: 00010004
```

3: AUCX()

The Call Agent at the IPDT audits the connection to determine the current characteristics of the call.

```
AUCX 1412 aaln/1@rgwl.mso.net, 00010004, L, LC
```

4: 200 OK()

The MTA responds with the requested information.

```
200 1412 OK
L: p:10, a:PCMU;telephone-event fntp:"telephone-event
144,149,159"
v=0
o=- 4723891 7428910 IN IP4
128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=audio 3456 RTP/AVP 0 96
a=rtpmap:0 G711/8000
a=rtpmap:96 telephone-event/8000
a=fntp:96 144,159
```

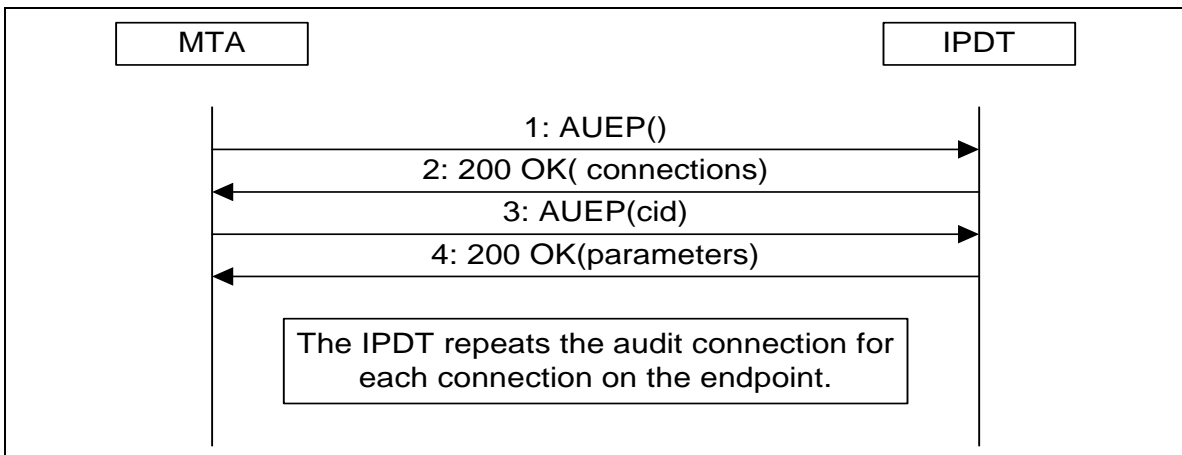


Figure 29. Audit Connection

A.2 Common Call Flow Macros

This section contains call flow segments common to several of the call flows presented in the preceding section. Rather than repeat these comment segments in each call flow, references are made to the flows in this section.

A.2.1 Create Access Network Connection

This call flow describes the common sequence of operations performed when a connection is created on the HFC access network.

1: GATE-SET()

The CMS in the IPDT establishes a gate at the CMTS defining the authorized bandwidth available to the MTA for connections. The operation is requested without gate coordination.

2: GATE-SET-ACK(GateID)

The CMTS acknowledges the gate set operation, returning the ID of the allocated gate.

3: CRCX(sendrcve,SDP,gateID)

With the time slot assignment made, the Call Agent in the IPDT sends a create connection request to the MTA in the CPE to establish a two-way media path between the IPDT and the MTA. The connection request specifies that the connection should be created send/receive and active. The request also includes the remote session description providing the RTP address at the IPDT to which audio and event packets are sent for this connection. When DQoS is active, the CRCX includes the Gate ID set for the connection. The following example shows RFC 2833 usage being signaled. This would not be the case for the NCS-only translation.

```
CRCX 277 aaln/1@rgw1.mso.net MGCP 1.0 NCS 1.0
C: 01000997
L: p:10, a:PCMU;telephone-event fntp:"telephone-event
144,149,159", dq-gi:gateIDx
M: sendrcv
X: 22331236
R: hf(I)
S:
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0 96
a=rtpmap:0 PCMU/8000
a=rtpmap:96 telephone-event/8000
a=fntp:96 144,149,159
```

4: Process LocalConnectionOptions()

The MTA in the CPE parses and processes all of the parameters and options provided by the Call Agent in the CRCX request. This allows the MTA to determine how to request upstream bandwidth.

5: 100 PENDING()

When the MTA determines that it has the resources to create the requested connection, it sends a provisional acknowledgement to the Call Agent in the IPDT. The following example shows RFC 2833 usage being signaled. This would not be the case for the NCS-only translations.

```
100 277 PENDING
I: 123E
v=0
o=- 87652 948357 IN IP4 128.2.3.4
```

```

s=-
c=IN IP4 128.2.3.4
t= 0 0
m=audio 8765 RTP/AVP 0 96
    
```

6: DSA-REQ(admitted+active,gateID)

The MTA attempts to activate an unsolicited grant service (UGS) service flow or add a grant to an existing UGS service flow by sending a DSA-REQ or DSC-REQ to the CMTS. When DQoS is required, the request includes the Gate ID received in the CRCX request from the Call Agent.

7: DSA-RSP()

The CMTS verifies the request against the gate parameters, if necessary, allocates the requested bandwidth, and acknowledges the request with a DSA-RSP or DSC-RSP, as appropriate.

8: DSA-ACK()

The MTA acknowledges the DSA-RSP or DSC-RSP.

9: 200 OK(SDP)

After bandwidth is allocated, the MTA acknowledges the CRCX, passing its session description back to the Call Agent in the IPDT. At this point, a two-way media path exists between the user phone and the LDS. The following example shows RFC 2833 usage being signaled. This would not be the case for the NCS-only translation.

```

200 277 OK
I: 123E
v=0
o=- 87652 948357 IN IP4 128.2.3.4
s=-
c=IN IP4 128.2.3.4
t= 0 0
m=audio 8765 RTP/AVP 0 96
a=rtpmap:0 PCMU/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 144,159
    
```

10: Voice Path

The media path between the subscriber’s telephone and the IPDT is established.

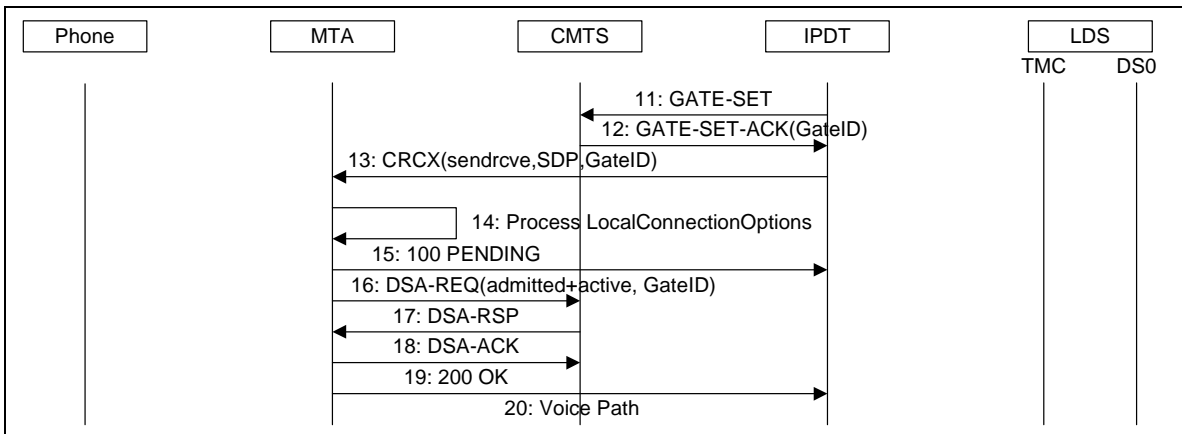


Figure 30. Create Access Network Connection

A.3 NCS with RTP Named Telephony Events Macros

A.3.1 Notify Off-Hook (RTP)

This call flow describes the sequence of operations performed by the MTA to notify off-hook transitions when RTP named telephony events are supported by the IPDT and MTA and an RTP stream already has been established between the IPDT and the MTA.

1: off hook()

The subscriber picks up a handset.

2: RTP Event(loop closed)

The MTA immediately notifies the IPDT of the off-hook by transmitting normal voice packets, reflecting the loop closed status.

3: ABCD Code(loop closed)

The IPDT immediately relays the loop status to the LDS using robbed-bit signaling.

4: NTFY(off hook)

When the off-hook detection time defined for the 'hd' event has expired, the MTA sends an NCS event notification to the IPDT.

5: 200 OK()

The IPDT acknowledges the event notification.

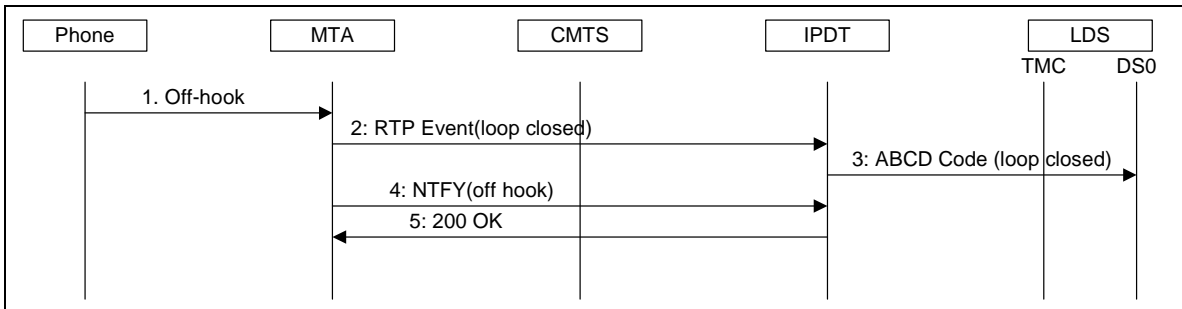


Figure 31. Notify Off-Hook (RTP)

A.3.2 Notify On-Hook (RTP)

This call flow describes the sequence of operations performed by the MTA to notify on-hook transitions when RTP named telephony events are supported by the IPDT and MTA and an RTP stream already has been established between the IPDT and the MTA.

1: on hook()

The subscriber hangs up a handset.

2: RTP Event(loop open)

The MTA immediately notifies the IPDT of the on-hook with an in-band RTP named telephony event reflecting the loop open status.

3: ABCD Code(loop open)

The IPDT immediately relays the loop status to the LDS using robbed-bit signaling.

4: NTFY(on hook)

When the on-hook time has expired, the MTA sends an NCS event notification to the IPDT.

5: 200 OK()

The IPDT acknowledges the event notification.

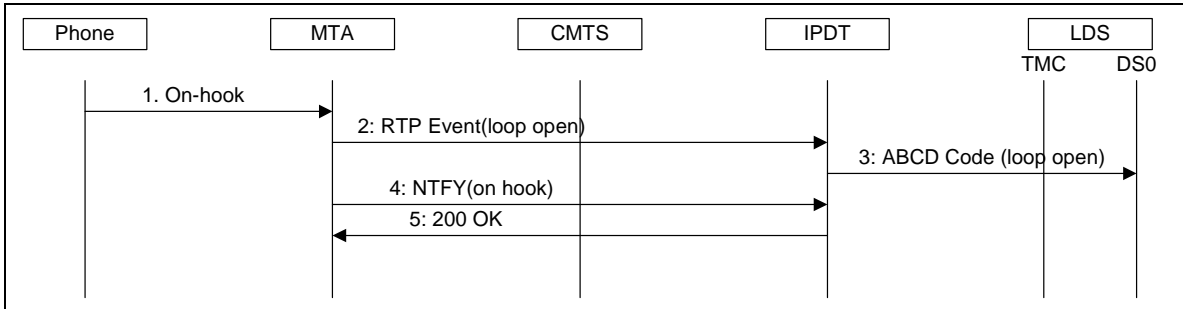


Figure 32. Notify On-Hook

A.3.3 Notify Hook Flash (RTP)

This call flow describes the sequence of operations performed by the system to accomplish a hook flash operation.

1: on hook()

The subscriber performs a hook flash to obtain dial tone and initiate a new call. This is detected as separate on-hook and off-hook state transitions.

2: RTP Event(on hook)

The MTA has been instructed previously by the Call Agent to ignore hook flash, so it starts its on-hook timer, but immediately sends an in-band RTP event packet showing the hook state change.

3: ABCD Code(loop open)

The IPDT relays the hook state with an ABCD code.

4: off hook()

The subscriber completes the hook flash.

5: RTP Event(off hook)

The MTA terminates its on-hook notification timer without sending a notification to the Call Agent, but relays the hook state change immediately to the IPDT with an in-band RTP event.

6: ABCD Code (loop closed)

The IPDT relays the hook state change to the LDS, which detects the valid hook flash event.

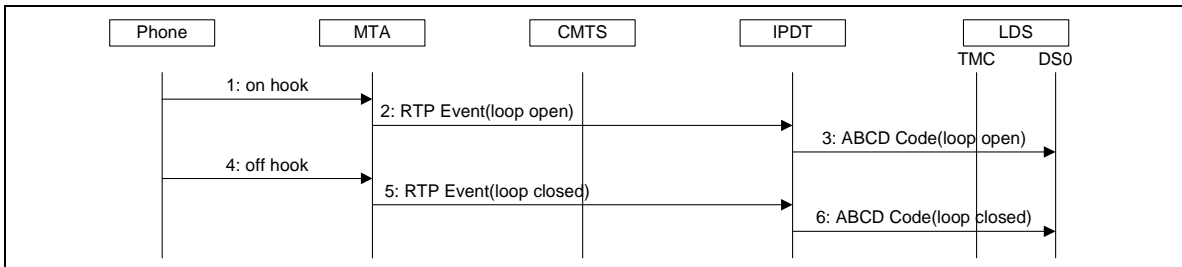


Figure 33. Notify Hook Flash

A.3.4 Ring MTA (RTP)

This call flow describes the sequence of operations performed by the system to apply power ringing at the MTA, when the IPDT and MTA support RTP named telephony events. The IPDT simply passes the ring and normal battery pattern generated by the LDS on to the MTA as RTP event packets.

1: ABCD Code(ring)

The LDS generates an interval of power ringing, reflected in the inband robbed-bit signaling.

2: RTP Event(ring)

The IPDT passes the ring status to the MTA in-band in RTP event packets. Because power ringing is mutually exclusive of voice, these packets are substituted for normal audio as long as the LDS sustains the ring ABCD code.

3: ring

The MTA generates power ring voltage while receiving ring event packets.

4: ABCD Code(normal battery)

At the end of a ring cycle, the LDS restores normal battery status in the robbed-bit signaling. The IPDT simply stops sending ring event packets and resumes sending normal audio packets. In response, the MTA stops generating power ringing and resumes playing out audio.

5: Caller ID

If caller ID information is available, the LDS transmits the caller ID as an FSK spill during the interval between the first and second ring. Because all timing relationships are maintained by the event packets for power ringing, the FSK spill passes in-band from the LDS to the MTA.

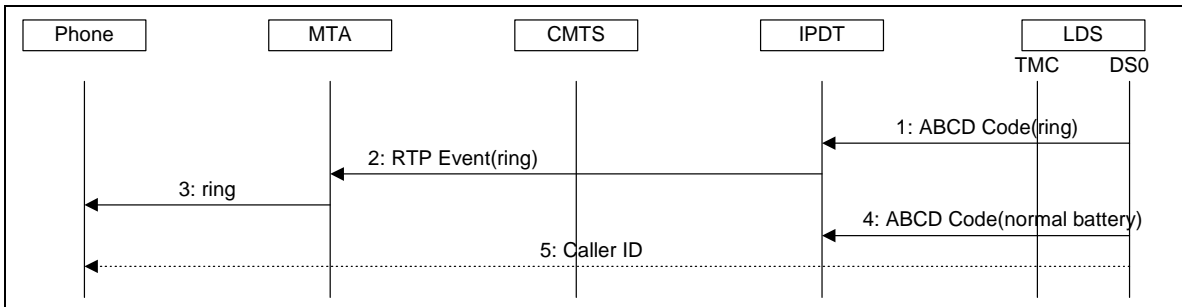


Figure 34. Ring MTA (RTP)

A.3.5 Open Loop Short (RTP)

This call flow describes the sequence of operations performed by the system to relay an open loop short interval from the LDS to the MTA when RTP named telephony events are supported by the IPDT and MTA.

1: ABCD Code(loop open)

The LDS starts a loop open interval through robbed-bit signaling.

2: RTP Event(loop open)

The IPDT relays the line condition to the MTA using RTP event packets.

3: open loop()

The MTA sets an open loop condition on the subscriber's line.

4: ABCD Code(normal battery)

The LDS terminates the open interval by setting normal battery condition.

5: RTP Event(normal battery)

The IPDT relays the line condition to the MTA using RTP event packets.

6: normal battery()

The MTA sets normal battery condition on the subscriber's line.

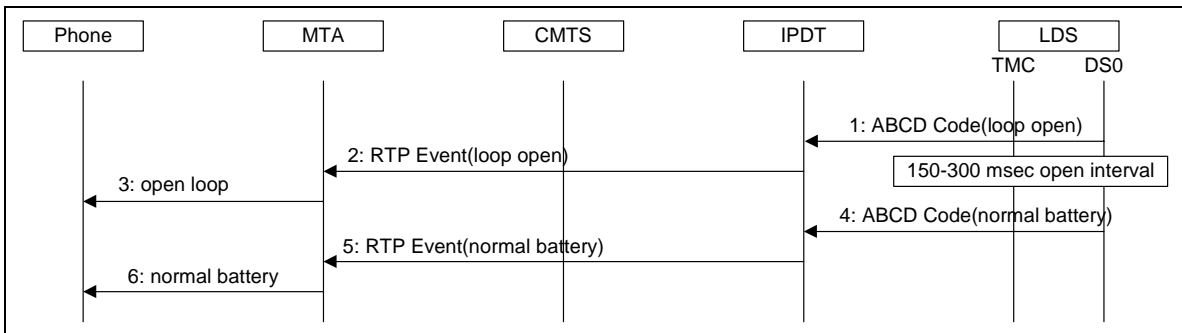


Figure 35. Open Loop Short (RTP)

A.3.6 Open Loop Long (RTP)

This call flow describes the sequence of operations performed by the system to relay an open loop short interval from the LDS to the MTA when RTP named telephony events are supported by the IPDT and MTA.

1: ABCD Code(loop open)

The LDS starts a loop open interval through robbed-bit signaling.

2: RTP Event(loop open)

The IPDT relays the line condition to the MTA using RTP event packets.

3: open loop()

The MTA sets an open loop condition on the subscriber's line.

4: ABCD Code(normal battery)

The LDS terminates the open interval by setting normal battery condition.

5: RTP Event(normal battery)

The IPDT relays the line condition to the MTA using RTP event packets.

6: normal battery()

The MTA sets normal battery condition on the subscriber's line.

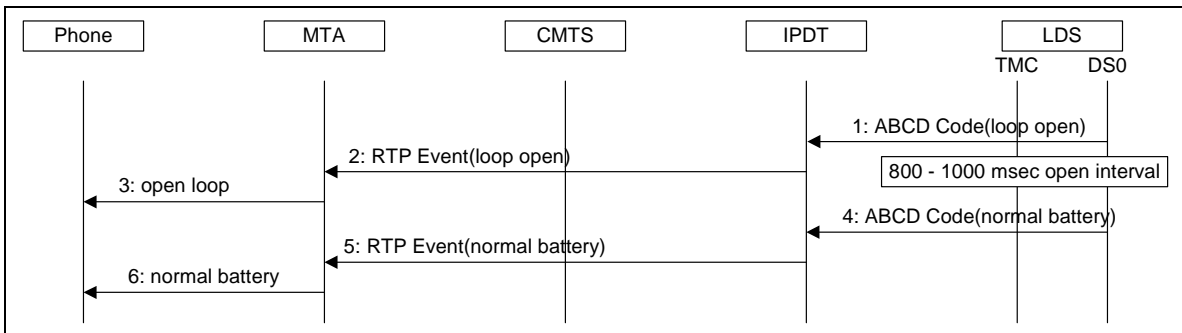


Figure 36. Open Loop Long (RTP)

A.4 NCS Translation Signaling Macros

A.4.1 Notify Off-Hook (NCS Translation)

1: off hook()

The subscriber picks up the telephone handset.

2: NTFY(off hook)

The MTA in the CPE notifies the IPTD of an off hook event.

```
NTFY 2 aaln/1@rgwl.mso.net MGCP 1.0 NCS 1.0
X: 22331236
O:hd
```

3: ABCD Code(loop closed)

The IPDT transmits ABCD code signal indicating to the LDS that the line has been answered.

4: 200 OK()

The IPDT acknowledges the event notification.

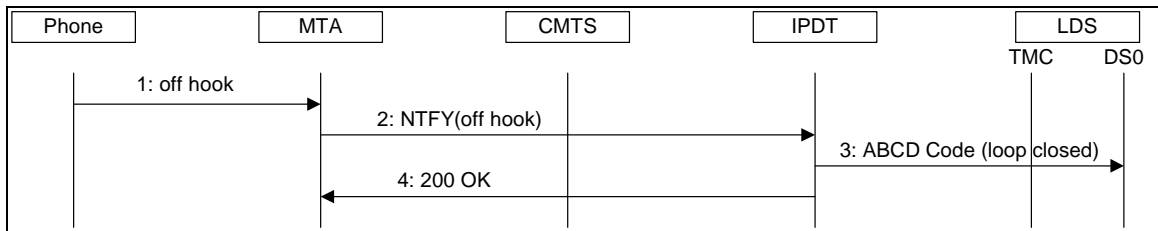


Figure 37. Notify Off-Hook (NCS Translation)

A.4.2 Notify On-Hook (NCS Translation)

1: on hook()

Subscriber terminates call by placing his handset on hook.

2: NTFY(on hook)

The MTA in the CPE notifies the IPTD of an on hook event.

```
NTFY 3 aaln/1@rgwl.mso.net MGCP 1.0 NCS 1.0
X: 22331236
O:hu
```

3: ABCD Code (loop open)

The IPDT relays the on-hook status to the LDS.

4: 200 OK()

The Call Agent acknowledges the event notification.

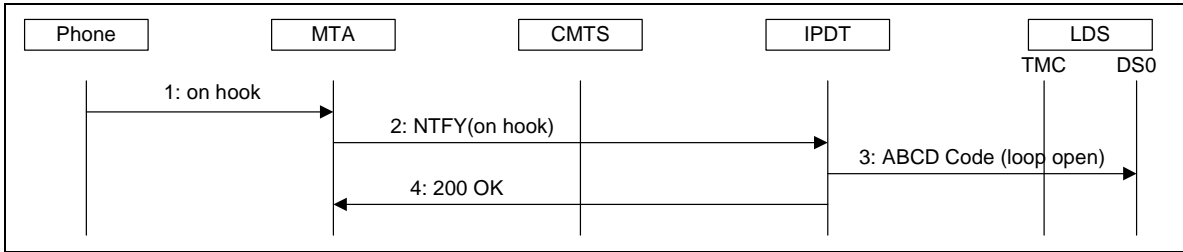


Figure 38. Notify On-Hook (NCS Translation)

A.4.3 Notify Hook Flash (NCS Translation)

When a subscriber initiates a hook flash, this is not interpreted by the MTA or the IPDT (although the MTA may ignore very rapid transitions) and it is the responsibility of the LDS to observe whether the sequence of on hook/off hook constitutes a valid hook flash. The sequence of events is exactly the macros of an on hook followed by an off hook, as above.

1a,b: hook flash()

The subscriber performs a hook flash as two events, first going on-hook.

2: NTFY(hook flash)

The MTA in the CPE notifies the IPTD of an on hook event.

```
NTFY 3 aaln/1@rgw1.mso.net MGCP 1.0 NCS 1.0
X: 22331236
O:hf
```

3: ABCD Code (loop open)

The IPDT relays the on-hook status to the LDS.

4: ABCD Code (loop closed)

The IPDT relays the off-hook status to the LDS.

5: 200 OK

The Call Agent acknowledges the event notification.

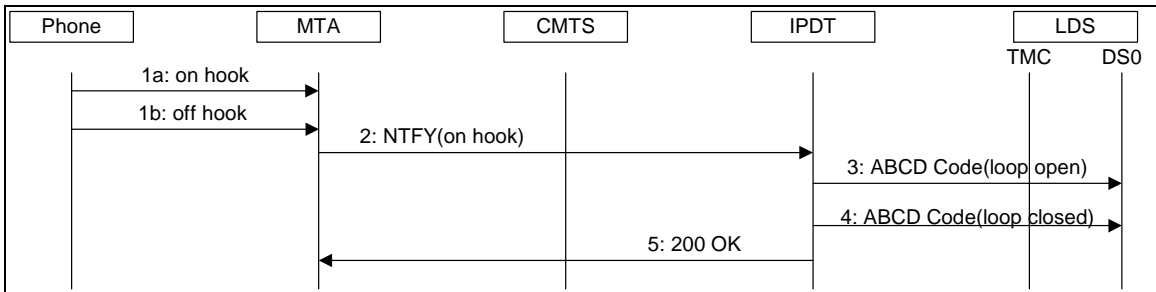


Figure 39. Notify Hook Flash (NCS Translation)

A.4.4 Ring MTA (NCS Translation)

1: ABCD Code(ring)

The LDS applies ringing on the media DS0 toward the IPDT using robbed-bit ABCD signaling.

2: RQNT(ring r=s7)

The IPDT sends a signal request indicating distinctive ringing pattern r7. This pattern is used in the Packet Cable LCS implementation to indicate a constant ring of up to 2 seconds. A ringing event of less than 2 seconds will be indicated with a subsequent message containing a signal request with no ringing pattern thus terminating the ring.

```
RQNT 278 aaln/1@rgw1.mso.net MGCP 1.0 NCS 1.0
X: 22331236
R:hd
r=s7
```

3: Ring

The MTA generates power ring voltage.

4: ABCD Code(normal battery)

The LDS restores normal battery status in the robbed-bit signaling.

5: RQNT(ring-off r=-)

The IPDT sends a signal request with empty ringing pattern.

```
RQNT 279 aaln/1@rgw1.mso.net MGCP 1.0 NCS 1.0
X: 22331236
R:hd
r=-
```

6: Ring off

The MTA ceases to generate power ring voltage

7-12: Continue

The IPDT process iterates for each ring on/ring off interval in step with the ABCD ring codes transitions generated by the LDS, and continues until the subscriber takes the handset off-hook or the initiating party disconnects, prompting the LDS to stop generating the ring pattern.

To reproduce the ringing cadence with adequate fidelity requires that NCS packets are not lost and are transferred with a jitter below a threshold based on satisfactory user perception.

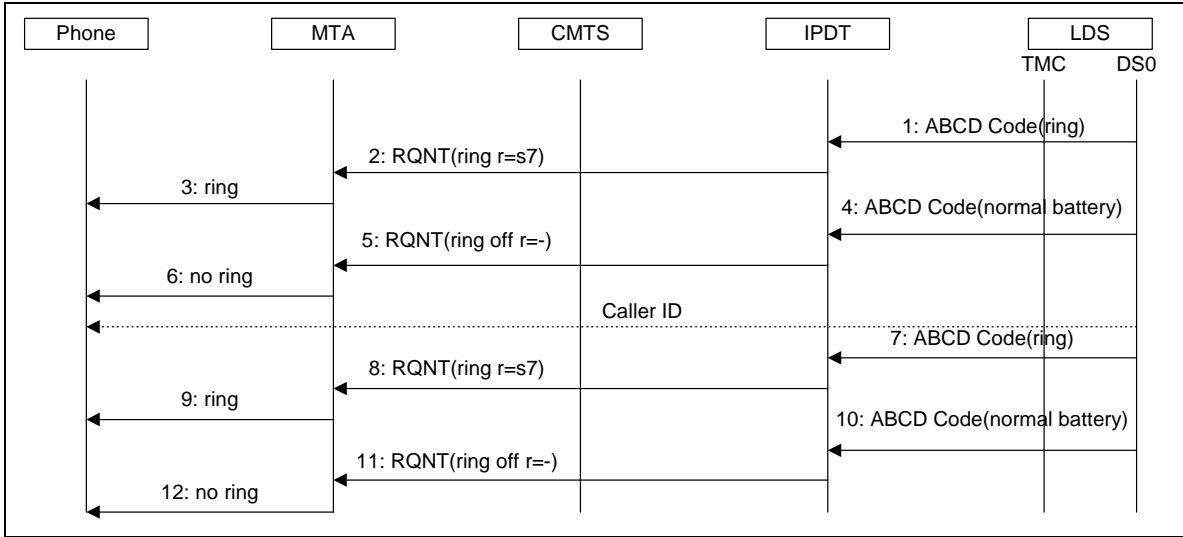


Figure 40. Ring MTA (NCS Translation)

A.4.5 Open Loop Short (NCS Translation)

This call flow describes the sequence of operations performed by the system to relay an open loop short interval from the LDS to the MTA when NCS translated telephony events are supported by the IPDT and MTA.

1: ABCD Code(loop open)

The LDS starts a loop open interval through robbed-bit signaling.

2: NCS signaling for OSI

The IPDT relays the line condition to the MTA using NCS signaling for OSI (Ref: mgcp-n-01089).

3: open loop()

The MTA sets an open loop condition on the subscriber's line.

4: ABCD Code (normal battery)

The LDS terminates the open interval by setting normal battery condition.

5: NCS signaling to turn off OSI

The LDS terminates the open interval by setting normal battery condition which is translated by the IPDT into a NCS RQNT with an empty signal request

6: normal battery()

The MTA sets normal battery condition on the subscriber's line.

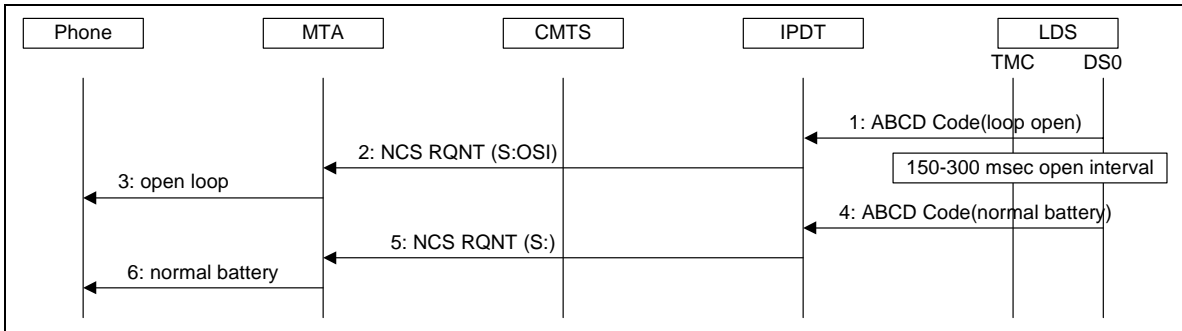


Figure 41. Open Loop Short (NCS)

A.4.6 Open Loop Long (NCS Translation)

This call flow describes the sequence of operations performed by the system to relay an open loop long interval from the LDS to the MTA when NCS translated telephony events are supported by the IPDT and MTA.

1: ABCD Code(loop open)

The LDS starts a loop open interval through robbed-bit signaling.

2: NCS Signaling for OSI

The IPDT relays the line condition to the MTA using NCS signaling for OSI.

3: open loop ()

The MTA sets an open loop condition on the subscriber's line.

4: ABCD Code(normal battery)

The LDS terminates the open interval by setting normal battery condition.

5: NCS signaling to turn off OSI

The LDS terminates the open interval by setting normal battery condition which is translated by the IPDT into a NCS RQNT with an empty signal request

6: normal battery()

The MTA sets normal battery condition on the subscriber's line.

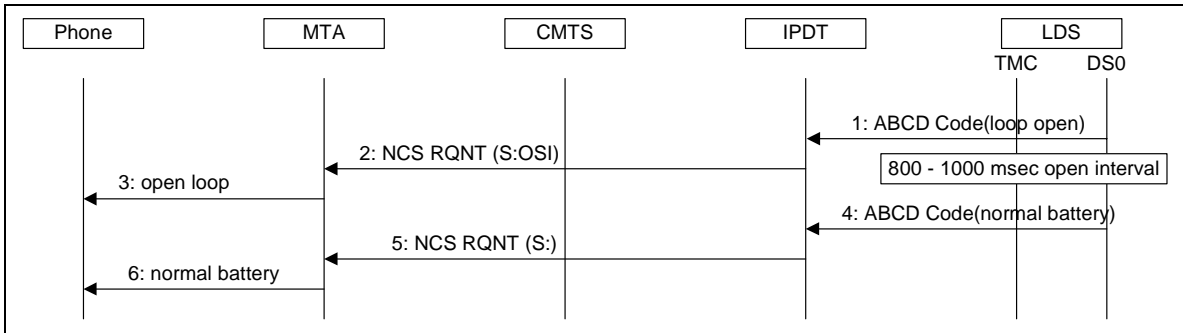


Figure 42. Open Loop Long (NCS)

Appendix B IPDT Provisioning and Management

The Telcordia Specification, GR-303 – IMD, presents generic requirements for communication over the EOC using CMIS and ASN.1. It describes a large set of a Managed Objects and their associated Attributes, Notifications and Actions. These Objects are used by the IDT and the RDT to implement various system functions such as protection switches and line state management. The GR-303-IMD does not mandate the use of all of these objects, and only some of the attributes, notifications and actions associated with them are mandatory.

The following table provides a subset of the objects specified in the GR-303-IMD which must be supported by the IPDT. Individual implementations may use additional objects. The IPDT will need to translate switch actions directed at these objects into actions and objects which are under the control of Packet Cable Network Elements. Additionally, some of these objects may need to be provisioned by means of Element Management System(s) associated with the IPDT.

Note that some objects identified in the GR-303-IMD are designed specifically to permit a switch to perform tests on individual subscriber metallic loops. The IPDT will not permit the switch to perform such tests, however, and none of these objects are included in the list below. Testing the Ethernet/IP and DOCSIS/HFC networks will be left to the network management systems of these networks.

B.1 Objects and Attributes in the IDT's MIB

Object	Mandatory Attributes
Alarm Count List	Alarm Count List ID Alarm Count Info
Equipment	Equipment ID Primary Service State Secondary Service State Alarm Severity Assignment List
IDLC Call Processing Profile	IDLC CP Profile ID T308 T303 T305 T396 T397
IDLC Data Link Profile	IDLC DL Profile ID SAPI Maximum I Frames N200 T200 T203
IDLC Terminal	Terminal ID Signaling Method DS1VT Count (Optional)
Network Element	Network Element ID Primary Service State

Object	Mandatory Attributes
	Service State System Title Alarm Severity Assignment List System Clock
Protection Group	Protection Group ID Protection Type Revertive Activate Lockout Number of Protecting
DS0 Channel Termination	DS0 channel Term ID Primary Service State Secondary Service State Robbed Bit Signal
DS1 Framed Path Termination	DS1 Framed Path Term ID Primary Service State Secondary Service State DS1 Frame Format
IDLC Data Link Termination	IDLC Data Link Term ID Primary Service State Secondary Service State Data Link Type Protection Group Pointer
Analog Line Termination	Primary Service State Secondary Service State Generic Signal Function Code Alarm Severity Assignment List Line Circuit Address Event Report Control Pointers Call Reference Value
DS1 Line Termination	DS1 Line Term ID Primary Service State Secondary Service State DS1 Line Code Alarm Severity Assignment List Line Circuit Address

B.2 Notifications and Actions associated with MIB Objects:

Each object in the GR303-IMD specified MIB is associated with certain Notifications and Actions. The distributed RDT may be expected to notify a Network Management entity (perhaps one which is part of the switch) of certain changes in the RDT's data base and may be further expected to alter its data base in response to commands issued by the Network Management entity. Below is a summary of the actions and notifications associated with each object identified in B.1. These are provided for reference purposes only and can be found in greater detail in the GR303-IMD itself. The purpose for providing this information in this appendix is to focus the implementer's attention on the fact that requests from external management agents, must be handled in conformance with the GR303 specification and that failure to observe these requests may cause unexpected behaviors in both the RDT itself and in the External Management entities. (An X indicates that there is no action associated with the object.)

OBJECT	NOTIFICATION	ACTION
Alarm Count List	Object Creation Reporting	X
	Object Deletion Reporting	
	Event Reporting	
Cross Connection	Object Creation Reporting	Remove
	Object Deletion Reporting	Restore
	Attribute Change Reporting	
Equipment	Object Creation Reporting	Remove
	Object Deletion Reporting	Restore
	Attribute Change Reporting	Diagnose
	Event Reporting	Exercise
		Activate External Entity
	Deactivate External Entity	
	Boot Processor	
IDLC Call Processing Profile	Object Creation Reporting	X
	Object Deletion Reporting	
	Attribute Change Reporting	
IDLC Data Link Profile	Object Creation Reporting	X
	Object Deletion Reporting	
	Attribute Change Reporting	
	Event Reporting	
IDLC Terminal	Object Creation Reporting	X
	Object Deletion Reporting	
	Attribute Change Reporting	
Network Element	Attribute Change Reporting	Remove
	Event Reporting	Restore
		Activate External Entity
		Deactivate External Entity
		Boot Processor
	Restart Processor	
Protection Group	Object Creation Reporting	Protection Switch
	Object Deletion Reporting	Exercise Protection Switch
	Attribute Change Reporting	Protection Release
	Automatic Protection Switch Reporting	Lockout of Protection
		Release Lockout of Protection
	Protection Bridge	
Protection Group Unit	Object Creation Reporting	X
	Object Deletion Reporting	
	Attribute Change Reporting	
Termination Point	Object Creation Reporting	X
	Object Deletion Reporting	

OBJECT	NOTIFICATION	ACTION
	Attribute Change Reporting	
Cross Connection	Object Creation Reporting	X
	Object Deletion Reporting	
	Attribute Change Reporting	
DS0 Channel Termination	Object Creation Reporting	Remove
	Object Deletion Reporting	Restore
	Attribute Change Reporting	
Framed Path Termination	Object Creation Reporting	Remove
	Object Deletion Reporting	Restore
	Attribute Change Reporting	
DS1 Framed Path Termination	Object Creation Reporting	Remove
	Object Deletion Reporting	Restore
	Attribute Change Reporting	
	Event Reporting	
IDLC Data Link Termination	Object Creation Reporting	Remove
	Object Deletion Reporting	Restore
	Attribute Change Reporting	
	Event Reporting	
Line Termination	Object Creation Reporting	Remove
	Object Deletion Reporting	Restore
	Attribute Change Reporting	
Analog Line Termination	Object Creation Reporting	Remove
	Object Deletion Reporting	Restore
	Attribute Change Reporting	Connect Test Response Circuit
	Event Reporting	Release Test Response Circuit

APPENDIX C Acknowledgements

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 Technical Report:

Neil Olsen	ADC
Doug Nortz	AT&T Broadband
Bob Lukas	Broadcom
Rick Kelly	CableLabs
David McIntosh	CableLabs
Matt Osman	CableLabs
Glenn Russell	CableLabs
Maria Stachelek	CableLabs
Noam Dimant	ComMATCH
Masood Parvaresh	General Bandwidth
John Sirney	General Bandwidth
Rex Coldren	Lucent/AGCS
Brian Hagar	Lucent/AGCS
John Short	Lucent/AGCS
Gerry Van Daele	Lucent/AGCS
Bob Stein	Motorola
Dave Flanagan	Motorola
Joe Mierwa	Nortel Networks Cable Solutions
Bercak Beser	Pacific Broadband
Carol Davids	Tellabs
Roy Spitzer	Telogy/TI
Marty Borden	Tollbridge Technologies
Bruce McLeod	Tollbridge Technologies
Madeline Ng	Tollbridge Technologies