



**Society of Cable
Telecommunications
Engineers**

**ENGINEERING COMMITTEE
Data Standards Subcommittee**

AMERICAN NATIONAL STANDARD

ANSI/SCTE 159-1 2010

**Multimedia Application and Service Part 1:
IPCablecom Multimedia**

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability and ultimately the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members, whether used domestically or internationally.

SCTE assumes no obligations or liability whatsoever to any party who may adopt the Standards. Such adopting party assumes all risks associated with adoption of these Standards, and accepts full responsibility for any damage and/or claims arising from the adoption of such Standards.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Patent holders who believe that they hold patents which are essential to the implementation of this standard have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE web site at <http://www.scte.org>.

All Rights Reserved

© Society of Cable Telecommunications Engineers, Inc. 2010
140 Philips Road
Exton, PA 19341

Note: DOCSIS® is a registered trademark of Cable Television Laboratories, Inc., and is used in this document with permission.

Contents

1	INTRODUCTION	1
1.1	PURPOSE.....	1
1.2	SCOPE.....	1
1.3	RELATION TO IPCABLECOM 1.X	1
1.4	SPECIFICATION LANGUAGE	3
2	REFERENCES	4
2.1	NORMATIVE REFERENCES	4
2.2	INFORMATIVE REFERENCES	4
2.3	REFERENCE ACQUISITION.....	5
3	TERMS AND DEFINITIONS	6
4	ABBREVIATIONS AND ACRONYMS.....	7
5	TECHNICAL OVERVIEW.....	9
5.1	QoS BACKGROUND	9
5.1.1	DOCSIS QoS Summary.....	9
5.1.2	IPCablecom 1.x QoS Summary.....	11
5.2	ARCHITECTURE	12
5.2.1	Client Types	12
5.2.2	IPCablecom Multimedia Devices	13
5.2.3	IPCablecom Multimedia Interfaces	17
5.2.4	State Information	20
6	AUTHORIZATION INTERFACE DESCRIPTION	21
6.1	GATES: THE FRAMEWORK FOR QoS CONTROL	21
6.1.1	Gate Identification (GateID).....	23
6.1.2	Application Manager Identification (AMID)	23
6.1.3	Subscriber Identification (SubscriberID).....	23
6.1.4	Gate Specification (GateSpec).....	23
6.1.5	Classifier.....	24
6.1.6	Traffic Profile	27
6.1.7	Event Generation Info.....	28
6.1.8	Time-Based Usage Limit.....	29
6.1.9	Volume-Based Usage Limit	29
6.1.10	Opaque Data.....	29
6.1.11	Gate Time Info	29
6.1.12	Gate Usage Info.....	29
6.1.13	User Identification (UserID).....	29
6.1.14	Shared Resource Identification (SharedResourceID).....	29
6.2	GATE TRANSITIONS	30
6.2.1	Authorized.....	32
6.2.2	Reserved.....	32
6.2.3	Committed.....	33
6.2.4	Committed-Recovery.....	34
6.3	COPS PROFILE FOR IPCABLECOM MULTIMEDIA.....	36
6.4	GATE CONTROL PROTOCOL MESSAGE FORMATS	38
6.4.1	COPS Common Message Format.....	38
6.4.2	Additional COPS Objects for Gate Control.....	39
6.4.3	Gate Control Messages.....	68
6.5	GATE CONTROL PROTOCOL OPERATION	72
6.5.1	Initialization Sequence.....	72
6.5.2	Operation Sequence.....	73

6.5.3	<i>Procedures for Validating Resource Envelopes</i>	<i>74</i>
6.5.4	<i>Procedures for Authorizing Resources through a Gate</i>	<i>77</i>
6.5.5	<i>Procedures for Querying a Gate.....</i>	<i>78</i>
6.5.6	<i>Procedures for Modifying a Gate</i>	<i>78</i>
6.5.7	<i>Procedures for Supporting Usage Limits.....</i>	<i>79</i>
6.5.8	<i>Procedures for Deleting a Gate</i>	<i>81</i>
6.5.9	<i>Procedure for Committing a Gate</i>	<i>82</i>
6.5.10	<i>Termination Sequence.....</i>	<i>82</i>
6.5.11	<i>Procedures for Multiple Grants Per Interval</i>	<i>82</i>
6.5.12	<i>Procedures for Identifying a PDP to a PEP</i>	<i>83</i>
6.5.13	<i>Procedures for Delivering Gate-Report-State Messages.....</i>	<i>84</i>
6.5.14	<i>Procedures for Gate Control Operations Initiated by a Policy Server.....</i>	<i>84</i>
6.5.15	<i>Procedures for State Synchronization.....</i>	<i>84</i>
6.5.16	<i>Procedures for Confirming PDP Receipt of Messages.....</i>	<i>87</i>
6.5.17	<i>Procedures for Indicating Updated or Lost Shared Resource.....</i>	<i>88</i>
7	EVENT MESSAGING INTERFACE DESCRIPTION	89
7.1	INTRODUCTION	89
7.2	RECORD KEEPING SERVER REQUIREMENTS	90
7.3	GENERAL IPCABLECOM MULTIMEDIA NETWORK ELEMENT REQUIREMENTS	91
7.3.1	<i>Element ID.....</i>	<i>91</i>
7.3.2	<i>Timing.....</i>	<i>91</i>
7.3.3	<i>Primary and Secondary RKS Considerations.....</i>	<i>91</i>
7.3.4	<i>Interaction with IPCablecom RKS.....</i>	<i>92</i>
7.4	EVENT MESSAGES FOR IPCABLECOM MULTIMEDIA.....	92
7.4.1	<i>Policy Events</i>	<i>92</i>
7.4.2	<i>QoS Events.....</i>	<i>95</i>
7.4.3	<i>Time_Change.....</i>	<i>97</i>
7.5	EVENT MESSAGING ATTRIBUTES FOR IPCABLECOM MULTIMEDIA	97
7.5.1	<i>EM_Header Attribute Structure.....</i>	<i>100</i>
7.5.2	<i>Billing Correlation ID (BCID) Field Structure</i>	<i>102</i>
7.5.3	<i>Status Field Structure</i>	<i>103</i>
7.5.4	<i>QoS Descriptor Attribute Structure</i>	<i>103</i>
7.6	RADIUS ACCOUNTING PROTOCOL	104
7.6.1	<i>Authentication and Confidentiality</i>	<i>105</i>
7.6.2	<i>Standard RADIUS Attributes.....</i>	<i>105</i>
7.6.3	<i>IPCablecom RADIUS Accounting-Request Packet Syntax</i>	<i>106</i>
8	SECURITY REQUIREMENTS.....	107
8.1	CMTS – CM QoS INTERFACE (PKT-MM-1)	107
8.2	POLICY SERVER – CMTS COPS INTERFACE (PKT-MM-2)	107
8.3	APPLICATION MANAGER – POLICY SERVER COPS INTERFACE (PKT-MM-3)	108
8.4	POLICY SERVER – RKS EVENT MESSAGE INTERFACE (PKT-MM-4)	108
8.5	CMTS – RKS EVENT MESSAGE INTERFACE (PKT-MM-5).....	108
9	MAPPING A FLOWSPEC TRAFFIC PROFILE TO DOCSIS	109
9.1	MAPPING FLOWSPECS TO DOCSIS SCHEDULING TYPES.....	109
9.2	MAPPING FLOWSPECS TO DOCSIS TRAFFIC PARAMETERS.....	110
9.3	DOCSIS UPSTREAM PARAMETERS	112
9.3.1	<i>Unsolicited Grant Scheduling (UGS)</i>	<i>112</i>
9.3.2	<i>Real-Time Polling Scheduling</i>	<i>113</i>
9.3.3	<i>Best Effort Scheduling</i>	<i>113</i>
9.3.4	<i>Upstream Packet Classification Encodings.....</i>	<i>114</i>
9.4	DOCSIS DOWNSTREAM PARAMETERS.....	115
9.4.1	<i>Downstream QoS Encodings for Guaranteed Service.....</i>	<i>115</i>
9.4.2	<i>Downstream QoS Encodings for Controlled Load Service.....</i>	<i>116</i>

9.4.3	<i>Downstream Packet Classification Encodings</i>	116
10	MESSAGE FLOWS	118
10.1	BASIC MESSAGE SEQUENCE	118
10.2	DETAILED MESSAGE SEQUENCE	120
APPENDIX A GUIDELINES FOR VERSION NUMBER ASSIGNMENT		145

List of Figures

FIGURE 1 - SESSION & RESOURCE CONTROL DOMAINS	13
FIGURE 2 - IPCABLECOM MULTIMEDIA ARCHITECTURAL FRAMEWORK	17
FIGURE 3 - GATE STATE TRANSITIONS	31
FIGURE 4 - QoS ADMISSION CONTROL LAYOUT	36
FIGURE 5 - COPS CONNECTION ESTABLISHMENT	73
FIGURE 6 - COPS KEEP-ALIVE EXCHANGE	73
FIGURE 7 - SAMPLE SIGNALING OF GATE-SET	77
FIGURE 8 - BASIC MESSAGE SEQUENCE	118
FIGURE 9 - DETAILED MESSAGE SEQUENCE	120

List of Tables

TABLE 1 - IPCABLECOM MULTIMEDIA INTERFACES.....	18
TABLE 2 - ERROR CODES.....	64
TABLE 3 - ENVELOPE COMPARISON RULES	75
TABLE 4 - UPSTREAM ENVELOPE COMPARISON	76
TABLE 5 - DOWNSTREAM ENVELOPE COMPARISON	76
TABLE 6 - IPCABLECOM MULTIMEDIA EM MESSAGE TYPES.....	89
TABLE 7 - IPCABLECOM 1.X TELEPHONY EM MESSAGE TYPES.....	90
TABLE 8 - POLICY_REQUEST EVENT MESSAGE	93
TABLE 9 - POLICY_DELETE EVENT MESSAGE	94
TABLE 10 - POLICY_UPDATE EVENT MESSAGE.....	94
TABLE 11 - QoS_RESERVE EVENT MESSAGE.....	95
TABLE 12 - QoS_COMMIT EVENT MESSAGE.....	96
TABLE 13 - QoS_RELEASE EVENT MESSAGE.....	97
TABLE 14 - TIME_CHANGE EVENT MESSAGE.....	97
TABLE 15 - IPCABLECOM ATTRIBUTES MAPPED TO IPCABLECOM MM EVENT MESSAGES.....	98
TABLE 16 - IPCABLECOM MM EVENT MESSAGE ATTRIBUTES.....	99
TABLE 17 - EM_HEADER ATTRIBUTE STRUCTURE	100
TABLE 18 - BCID FIELD DESCRIPTION	102
TABLE 19 - STATUS FIELD DESCRIPTION.....	103
TABLE 20 - QoS DESCRIPTOR DATA STRUCTURE	103
TABLE 21 - QoS STATUS BITMASK	104
TABLE 22 - RADIUS MESSAGE HEADER	105
TABLE 23 - MANDATORY RADIUS ATTRIBUTES.....	105
TABLE 24 - RADIUS ACCT_STATUS_TYPE	105
TABLE 25 - RADIUS VSA STRUCTURE FOR IPCABLECOM ATTRIBUTES	106
TABLE 26 - MULTIMEDIA SECURITY INTERFACES	107
TABLE 27 - MAPPING FLOW SPECS TYPES.....	109

1 INTRODUCTION

1.1 Purpose

The intent of this standard is to support the deployment of general Multimedia services by providing a technical definition of several IP-based signaling interfaces that leverage core QoS and policy management capabilities native to DOCSIS Versions 1.1 and greater. Throughout this specification, the term DOCSIS will be used to refer to any DOCSIS version greater than or equal to 1.1. For the purposes of this specification, Multimedia services may be defined as IP-based services (e.g., online gaming, video-conferencing, streaming media, etc.) requiring QoS-based network resources (as contrasted with services such as web browsing, e-mail, instant messaging and file-sharing that are commonly provided using best-effort flows). While telephony or voice-based services are not specifically excluded from this definition, the IPCom 1.x set of specifications provide coverage specific to this type of service delivery, and, therefore, those specifications should be consulted as appropriate.

IPCom Multimedia defines a service delivery framework that provides general-purpose QoS, event-based accounting, and security functionality founded upon the mechanisms defined in IPCom 1.x. However, due to the broader spectrum of applications and services addressed by this initiative, each of these functional areas has been revisited and generalized for the present purposes. Telephony-specific requirements and interfaces (e.g., call signaling, PSTN interconnection and electronic surveillance) are not part of IPCom Multimedia, while core functionality such as QoS resource management mechanisms, has been enhanced. Throughout this process, one of the primary objectives of this work has been to leverage and reuse as much of the existing body of IPCom 1.x investment, knowledge base, and technical functionality as possible.

Key features of the described Multimedia service delivery framework include:

- Simple, powerful access to DOCSIS QoS mechanisms supporting both time and volume-based network resource authorizations,
- Abstract, event-based network resource auditing and management mechanisms,
- A robust security infrastructure that provides integrity and appropriate levels of protection across all interfaces.

1.2 Scope

As outlined in the accompanying technical report [16], the current scope of this specification is limited to network-based QoS resource management and usage auditing capabilities. This approach was motivated by several criteria, including rapid time-to-market for QoS-enhanced Multimedia services (which may take the form of new applications or existing applications retrofitted per this specification), the absence of QoS signaling requirements on CPE devices, and security assurances provided by an absence of client-based QoS signaling.

In addition, the QoS signaling mechanisms outlined herein provide the foundation for client-based QoS signaling mechanisms which may be defined in the future. As such scenarios have already been considered and documented in the accompanying technical report, one of the secondary goals of this version of this document is to provide for smooth migration to a client-based model as market and technology advances dictate. It should also be noted that the scope of the QoS framework defined in this specification is limited to the access network and that the choice of specific backbone QoS strategies is left as an administrative decision to be addressed by each service provider as it deems best.

Requirements defined within this document will address multi-vendor interface compatibility and interoperability. Specific performance and fault-tolerance strategies are left as vendor differentiators.

1.3 Relation to IPCom 1.x

Since its inception the IPCom initiative has been positioned as an IP-based Multimedia service deployment infrastructure, exploiting and enhancing the underlying QoS capabilities of the DOCSIS access network. Based on market demand and technology readiness, VoIP was chosen as the first IP-based service to leverage these unique broadband access network capabilities, as the IPCom 1.x suite of specifications simultaneously defines both a general QoS-based service delivery framework and a number of telephony-specific functional elements and mechanisms.

Specifically, the IPCablecom 1.x VoIP architecture provides specifications promoting multi-vendor interoperability in the following key telephony areas:

- Network-based call signaling (NCS)
- Inter-domain call signaling (CMSS)
- PSTN gateway and interop functionality (TGCP, ISTP)
- Access-network and backbone QoS capabilities (DQoS, IQoS)
- Event messaging with telephony-specific extensions (EM)
- CPE endpoint provisioning and monitoring (Provisioning and SNMP MIBs)
- Comprehensive suite of interface security mechanisms (Security)
- Primary-line telephony characteristics
- Electronic surveillance

Since the intent of this specification is to extract the functional core of this architecture in support of the enhancement and deployment of other Multimedia services, support for these telephony-specific features is not required, while the core QoS, event messaging and security mechanisms are emphasized and generalized. The result is a framework that provides IP-based access to DOCSIS access-network QoS mechanisms complemented by secure and robust authorization and audit mechanisms.

In keeping with the IPCablecom 1.x Dynamic Quality of Service (DQoS) model, the primary logical construct in support of QoS-based service delivery is a Gate (along with its accompanying authorization token). An IPCablecom Gate represents a policy-based authorization for a specific envelope of network resources characterized by a suite of QoS parameters, as well as classifiers for originating and terminating IP addresses and ports, which define and limit the scope of the associated QoS-enhanced flow.

IPCablecom 1.x defines a pre-authorization model in which Gates are created and installed at policy enforcement points (e.g., a CMTS) prior to actual network resource reservation or activation requests. This process, termed Gate Control, is managed through a COPS-based policy interface on the CMTS. In this specification, this interface is generalized and enhanced to allow for full lifecycle management of QoS-based service flows through this policy interface. That is, Gate installation and management is supplemented with service flow creation, modification and deletion functions to provide for network-based QoS service delivery.

To complement these enhanced QoS policy and signaling capabilities, the RADIUS-based event messaging infrastructure of IPCablecom 1.x is carried over, but with telephony-specific primitives tagged as optional. Since the EM protocol was designed with a considerable amount of generality and abstraction, this approach provides continuity with existing protocol and Record Keeping Server (RKS) implementation while still allowing Multimedia vendors to select from a set of supplementary protocol primitives on an as-needed basis. For example, if a QoS-based video conferencing service were being developed based on this specification, a number of telephony-specific EM message types (e.g., Call_Answer and Call_Disconnect) may map directly to the service's audit requirements.

Finally, in order to provide robust security for this Multimedia framework, IPsec will be applied to the COPS and RADIUS interfaces in a manner analogous to the corresponding interfaces in the IPCablecom 1.x architecture.

1.4 Specification Language

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

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

2 REFERENCES

2.1 Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

- [1] ANSI/SCTE 135-2 2008, DOCSIS 3.0 Part 2: MAC and Upper Layer Protocols.
- [2] IETF RFC 1305, Network Time Protocol (Version 3) Specification, Implementation and Analysis, March 1992.
- [3] IETF RFC 2205, Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification, September 1997.
- [4] IETF RFC 2210, The Use of RSVP with IETF Integrated Services, September 1997.
- [5] IETF RFC 2211, Specification of the Controlled-Load Network Element Service, September 1997.
- [6] IETF RFC 2212, Specification of Guaranteed Quality of Service, September 1997.
- [7] IETF RFC 2216, Network Element Service Specification Template, September 1997.
- [8] IETF RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, December 1998.
- [9] IETF RFC 2670, Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS Compliant RF Interfaces, August 1999.
- [10] IETF RFC 2748, The COPS (Common Open Policy Service) Protocol, January 2000.
- [11] IETF RFC 2753, A Framework for Policy-based Admission Control, January 2000.
- [12] IETF RFC 2866, RADIUS Accounting, June 2000.
- [13] IETF RFC 3084, COPS Usage for Policy Provisioning (COPS-PR), March 2001.
- [14] SCTE 165-4 2009, IPCablecom 1.5 Part 4: Dynamic Quality-of-Service.
- [15] SCTE 165-9 2009, IPCablecom 1.5 Part 9: Event Messages.
- [16] PacketCable Multimedia Architecture Framework, PKT-TR-MM-ARCH-V03-091029, October 29, 2009, Cable Television Laboratories, Inc.
- [17] SCTE 165-10 2009, IPCablecom 1.5 Part 10: Security.
- [18] ANSI/SCTE 23-2 2007, DOCSIS 1.1 Part 2: Baseline Privacy Plus Interface.

2.2 Informative References

This document uses the following informative references.

- [19] IETF RFC 2751, S. Herzog. Signaled Preemption Priority Policy Element, January 2000.
- [20] SCTE 159-2 2010, Multimedia Application and Service Part 2: IPCablecom Multimedia Web Services.
- [21] IETF RFC 3168, The Addition of Explicit Congestion Notification (ECN) to IP, September 2001.
- [22] IETF RFC 3171, IANA Guidelines for IPv4 Multicast Address Assignment, August 2001.
- [23] IETF RFC 4291, IP Version 6 Addressing Architecture, February 2006.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; Internet: www.cablelabs.com, www.packetcable.com
- Internet Engineering Task Force (IETF) Secretariat c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, Phone +1-703-620-8990, Fax +1-703-620-9071, Internet: www.ietf.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Access Control	Limiting the flow of information from the resources of a system only to authorized persons, programs, processes, or other system resources on a network.
Active	A service flow is said to be "active" when it is permitted to forward data packets. A service flow must first be admitted before it is active.
Admitted	A service flow is said to be "admitted" when the CMTS has reserved resources (e.g., bandwidth) for it on the DOCSIS network.
Authentication	The process of verifying the claimed identity of an entity to another entity.
Authorization	The act of giving access to a service or device if one has permission to have the access.
DiffServ	Differentiated Services (RFC2475) refers to an architectural approach to providing QoS in which packets are tagged with a priority designation associated with various service levels on the network.
Downstream	The direction from the head-end toward the subscriber location.
Event Message	A message capturing a single step during the lifetime of a usage session. In this report event messages generally reference policy decisions or QoS changes.
Flow [DOCSIS Flow]	A unidirectional sequence of packets associated with a Service ID and a QoS. Multiple Multimedia streams may be carried in a single DOCSIS Flow. Also known as a DOCSIS "service flow"
Flow [IP Flow]	A unidirectional sequence of packets identified by OSI Layer 3 and Layer 4 header information. This information includes source/destination IP addresses, source/destination port numbers, protocol ID. Multiple Multimedia streams may be carried in a single IP Flow.
IntServ	Integrated Services (RFC1633) refers to an architectural approach to providing QoS in which multiple flows requiring real-time and non-real-time treatment are mixed over shared network links. Resources are reserved and state is maintained on a per-flow basis.
Originating Leg	Segment of a service session geographically associated with originating endpoint (i.e., calling party in the telephony model).
Terminating Leg	Segment of a service session geographically associated with the terminating endpoint (i.e., called party in the telephony model).
Upstream	The direction from the subscriber location toward the headend.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

AM	Application Manager. A system that interfaces to Policy Server(s) for requesting QoS-based service on behalf of an end-user or network management system.
BCID	Billing Correlation ID. Defined in the IPCablecom Event Messaging specification.
AS	Application Server. A server that interfaces to the IPCablecom Multimedia Application Manager that requests IPCablecom Multimedia services on behalf of clients.
CM	DOCSIS Cable Modem.
CMS	Call Management Server. Network element defined in the IPCablecom 1.0 Architecture technical report.
CMTS	Cable Modem Termination System. 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 Service. Defined in RFC2748.
DOCSIS[®]	Data-Over-Cable Service Interface Specifications. A set of interface specifications for transmitting data over cable television systems in a standard fashion.
DQoS	Dynamic Quality-of-Service.
DSx (Messaging)	DOCSIS QoS signaling mechanism providing Dynamic Service Add, Change and Delete semantics.
FQDN	Fully Qualified Domain Name.
HFC	Hybrid Fiber/Coax. An 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.
IETF	Internet Engineering Task Force. A body responsible for, among other things, developing standards used on the Internet. (See http://www.ietf.org .)
IGMP	Internet Group Management Protocol.
IP	Internet Protocol.
KDC	Key Distribution Center. Network element defined in the IPCablecom 1.0 Architecture technical report. Also defined in the IPCablecom Security specification.
MG	Media Gateway. Network element defined in the IPCablecom 1.0 Architecture technical report.
MGC	Media Gateway Controller. Network element defined in the IPCablecom 1.0 Architecture technical report.
MLD	Multicast Listener Discovery.
MSO	Multiple Service Operator.
MTA	Multimedia Terminal Adapter. Network element defined in the IPCablecom 1.0 Architecture technical report. Contains the interface to a physical voice device, a network interface, CODECs, and all signaling and encapsulation functions required for VoIP transport, class features signaling, and QoS signaling. May be implemented as a standalone (S-MTA) or embedded (E-MTA) device, depending upon whether a DOCSIS CM is integrated.
NAT	Network Address Translation. Function performed to convert IP addresses, and potentially transport ports (PAT), from one network address numbering convention to another.
PDP	Policy Decision Point. Defined in RFC2753.
PEP	Policy Enforcement Point. Defined in RFC2753.

PS	Policy Server. A system that primarily acts as an intermediary between Application Manager(s) and CMTS(s). It applies network policies to Application Manager requests and proxies messages between the Application Manager and CMTS.
PSTN	Public Switch Telephone Network.
QoS	Quality of Service. Method used to reserve network resources and guarantee availability for applications.
RADIUS	Remote Authentication Dial-In User Service. Defined in RFC2138 and RFC2139. An Internet protocol originally designed for allowing users dial-in access to the Internet through remote servers. Its flexible design has allowed it to be extended well beyond its original intended use.
RAP	Resource Allocation Protocol Working Group in the IETF. Responsible for the definition and maintenance of the COPS protocol.
RCD	Resource Control Domain. A logical grouping of elements that provide connectivity and network resource level policy management along the packet forwarding paths to and from an end host. The RCD consists of CMTS and Policy Server entities whose responsibilities include management of resources along the packet forwarding paths.
RFC	Request for Comments. Technical policy documents approved by the IETF which are available at http://www.ietf.org/rfc.html
RFI	DOCSIS Radio Frequency Interface specification, defining MAC and Physical layer interfaces between CMTS and CM network elements.
RKS	Record Keeping Server. Network element defined in IPCablecom 1.0 Architecture technical report. Also defined in the IPCablecom Event Messaging specification. The device which collects and correlates the various Event Messages.
RSVP	Resource Reservation Protocol. Defined in RFC2205.
RSVP+	IPCablecom profile and extension of RSVP, defined in the IPCablecom DQoS specification.
SCD	Session Control Domain. A logical grouping of elements that offer applications and content to service subscribers. The Application Manager resides in the SCD.
S-MTA	Standalone MTA. A single node that contains an MTA and a non-DOCSIS MAC (e.g., Ethernet).
TCP	Transmission Control Protocol.
TLV	Type-Length-Value. Technique used in formatting protocol elements.
UDP	User Datagram Protocol. A connectionless protocol built upon Internet Protocol (IP).
UGS	Unsolicited Grant Service. DOCSIS QoS scheduling type used for constant bit rate services (e.g., voice codecs).
UGS/AD	Unsolicited Grant Service with Activity Detection. DOCSIS QoS scheduling type used for constant bit rate services with periodic inactivity (e.g., voice codecs implementing silence suppression).
VoIP	Voice over IP.
VPN	Virtual Private Network.

5 TECHNICAL OVERVIEW

This section consists of background material which some readers may find to be useful context for the detailed protocol interface specifications that follow. The intent in this section is to provide a high-level overview of the IPCablecom Multimedia architecture and the fundamental technologies upon which it is based.

5.1 QoS Background

As noted throughout this specification, one of the primary features of the IPCablecom Multimedia service framework lies in the fact that it provides IP-layer access to sophisticated QoS capabilities defined in DOCSIS versions 1.1 and greater and IPCablecom 1.x. This subsection provides a brief overview of these capabilities as preparatory background for the detailed QoS policy and resource management discussion that follows.

5.1.1 DOCSIS QoS Summary

The DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification [1] defines a set of QoS facilities based on a fundamental network resource management construct known as a Service Flow. A Service Flow is defined as "a MAC-layer transport service which (1) provides unidirectional transport of packets from the upper layer service entity to the RF, and (2) shapes, polices, and prioritizes traffic according to QoS traffic parameters defined for the flow." In addition to this primary abstraction facilitating the reservation and scheduling of shared access network resources on a per-flow basis, a number of tangible supporting constructs are defined and used to manage these resources. Three of these are:

Service Flow Encodings: type-length-value (TLV) encoded parameters used to define QoS parameters associated with a Service Flow.

Classifier: type-length-value (TLV) encoded IP, Ethernet and IEEE802.1p/q parameters used to define and limit the scope of a flow in terms of originating and terminating endpoints.

Payload Header Suppression Rules: type-length-value (TLV) encoded parameters used by the sending entity to suppress repetitive payload headers following the DOCSIS Extended Header field and used by the receiving entity to restore the suppressed header information. IPCablecom Multimedia does not support this functionality at this time.

While DOCSIS supports provisioned (i.e., static, long-lived Service Flows that are established during the CM registration process) and dynamic (i.e., transient Service Flows that are added, modified and deleted on an as-needed basis) QoS models, the IPCablecom Multimedia framework is primarily concerned with the dynamic variety as this allows for optimal network resource management through statistical multiplexing as service requirements demand.

Service Flow management is performed through MAC-layer DOCSIS Dynamic Service Add/Change/Delete (DSA/DSC/DSD) messaging, which may be initiated either by the CM or by the CMTS. DSA and DSC transactions take the form of a three-way exchange in which a request (REQ) is followed by a response (RSP) which is then acknowledged (ACK). DSD messages are simple two-way exchanges. A specific attribute known as a Confirmation Code is provided in each DSx response message and indicates the success or failure status of a transaction.

One important point to note in reviewing the QoS capabilities provided in DOCSIS is that upstream and downstream Service Flows receive fundamentally different treatment at the CMTS. This is a result of the fact that upstream RF channels are contentious, shared-access mediums, taking the topological form of a many-to-one relationship between multiple CMs and a single CMTS. Conversely, the downstream RF channel behaves much more akin to a traditional IP router in which packets arrive (either from the access network or over backbone trunks), are queued, and are forwarded to one or more destinations. Consequently, distinct QoS mechanisms are applied depending upon whether a particular uni-directional Service Flow is oriented in the upstream or downstream direction.

Upstream Service Flows may be defined with one of five service flow scheduling types:

Best-Effort: a standard contention-based resource management strategy in which transmit opportunities are granted on a first-come-first-served basis, albeit under the coordination of the CMTS scheduler. This scheduling type may be supplemented with QoS characteristics in which, for example, maximum rate limits are applied to a particular Service Flow.

Non-Real-Time Polling: a reservation-based resource management strategy in which a particular CM is polled on a fixed interval to determine whether data has been queued for transmission on a particular service flow, and, if so, a transmission opportunity or grant for that service flow is provided by the scheduler.

Real-Time Polling: analogous to the Non-Real-Time-Polling scheduling type, except that the fixed polling interval is typically very short (<500ms). Polling scheduling types are most suitable for variable-bit-rate traffic that has inflexible latency and throughput requirements.

Unsolicited Grant: a reservation-based resource management strategy in which a fixed-size grant is provided to a particular Service Flow at (nearly) fixed intervals without additional polling or interaction. This scheduling type is most suitable for constant bit-rate traffic and eliminates much of the protocol overhead associated with the polling types.

Unsolicited Grant with Activity Detection: a reservation-based resource management strategy that represents a hybrid of the polling and unsolicited grant scheduling types in which fixed grants are provided at (nearly) fixed intervals so long as data is queued for transmission. During periods of inactivity, this scheduling type reverts to a polling mode in order to conserve unused bandwidth.

Due to the unique nature and specialized characteristics of each of these scheduling types, specific QoS parameters are associated with each. These parameters are outlined in detail in the section that follows.

Downstream Service Flows are defined using the same set of QoS parameters that are associated with the Best-Effort scheduling type on the upstream.

Regardless of the orientation of the flow or the particular scheduling type requested, all dynamic Service Flows procedure through three logical states, summarized below. While certain optimized signaling scenarios allow for a so-called "single-phase" commit operation, the request still proceeds logically through all three phases as it is serviced at the CMTS.

Authorized: requests are authenticated and network policy rules are applied resulting in an authorization envelop forming the boundary of subsequent reservation requests.

Admitted (or Reserved): an inactive Service Flow is constructed and resources are reserved by the scheduler so that subsequent activation requests are guaranteed to succeed; reserved resources may be used by best-effort traffic (from the same or different CMs) until committed.

Active (or Committed): the Service Flow is activated, along with corresponding Classifiers; QoS-enhanced packets are now able to traverse the flow.

Note: In a literal sense, DOCSIS does not define "states", but, rather, "attributes" of Service Flows which are completely replaced with each DSC transaction. The states described here are a logical construct used in a conceptual model describing the resource management process performed on the CMTS. Also, the DOCSIS RFI specification standardizes upon the terms "admitted" and "active" in defining the attributes of a service flow, while IPCablecom has adopted the equivalent terms "reserved" and "committed" in characterizing Gate states.

While DOCSIS does not define a specific authorization procedure to be applied to DSx messages, it does provide protocol support through a facility known as an Authorization Block for service-specific authorization schemes. Any credentials or authorization tokens that are presented via the Authorization Block are forwarded to an appropriate authorization module prior to the processing of the DSx request on the CMTS. IPCablecom makes extensive use of this authorization mechanism as described below.

5.1.1.1 DOCSIS 3.0 Multicast QoS Framework

In DOCSIS 3.0, a framework was developed for providing special QoS treatment for IP Multicast Sessions that were signaled via IP Multicast protocols such as IGMP and MLD. This framework relies on an incoming membership request or "JoinMulticastSession" to be evaluated against the CmtsGroupConfig table for the purposes of determining if special QoS handling needed for a specific multicast session.

This specification allows a Gate-Set message be treated as the functional equivalent of a "JoinMulticastSession". This specification replaces the use of the configured CmtsGroupConfigTable with dynamic IPCablecom Multimedia based Gate-Set messages for configuring the QoS of IP multicast replications to subscribers. Multicast Gates are expected to use application level encryption as IPCablecom Multimedia does not define encryption of IP multicast

flows. Furthermore for Multicast Gates, IPCablecom Multimedia overrides the use of the IP Multicast Join Authorization feature defined in [1].

Addition of non-IPCablecom Multimedia signaled IP Multicast Joiners to an existing IPCablecom Multimedia-initiated Group Service Flow (GSF) [1] is beyond the scope of this specification. The combination of non-IPCablecom Multimedia signaled IP Multicast Joiners and IPCablecom Multimedia-initiated Multicast joiners onto the same GSF is handled in a vendor specific manner.

An operator should configure the IP Multicast Join Authorization feature to reject unauthorized attempts by non-IPCablecom Multimedia authorized Multicast Joins to join multicast sessions intended to be authorized only by IPCablecom Multimedia.

IPCablecom Multimedia Multicast Gates are implemented using DOCSIS Single-Session Group Service Flows.

The operation of IPCablecom Multimedia Multicast Gates for subscribers reached through pre-3.0 CMs is beyond the scope of this specification.

5.1.2 IPCablecom 1.x QoS Summary

While the DOCSIS 3.0 MAC and Upper Layer Protocols Interface [1] specification defines the fundamental QoS mechanisms that form the core of the IPCablecom DQoS model, the IPCablecom DQoS specification [14] augments these capabilities with a COPS-based policy management framework. Just as the Service Flow represents the primary abstraction in the DOCSIS QoS model, the Gate plays a comparably significant role in the IPCablecom DQoS scheme. A Gate defines a resource authorization envelope consisting of IP-level QoS parameters as well as classifiers defining the scope of Service Flows that may be established against the Gate. In accordance with the DOCSIS authorization mechanisms described above, only DSx requests which conform to the following general relation on a parameter-by-parameter basis will be granted:

$$\text{Authorized Envelope} \geq \text{Reserved Envelope} \geq \text{Committed Envelope}$$

Based in this policy management model, IPCablecom 1.x defines a pre-authorization scheme in which network resources are authorized in advance of DSx messaging that requests establishment of a corresponding Service Flow. Consequently, the COPS interface used to install and manage Gates corresponds more closely with the COPS-PR model defined in RFC 3084 [12] than with the standard COPS scheme specified in RFC 2748 [10]. Also, in order to install and manage these Gates, the IPCablecom DQoS specification defines a set of COPS client-specific objects which constitute the primitives of a Gate Control signaling interface between the CMS and the CMTS.

Specifically, the CMS may be logically decomposed into a Call Agent, responsible for telephony call-state maintenance, and a Gate Controller, which receives authorization requests from the Call Agent (through an internal interface) and installs policy decisions in the form of Gates on the CMTS. In the IPCablecom Multimedia model, this decomposition is formalized through two separate network elements, the Policy Server (analogous to the IPCablecom 1.x Gate Controller) and the Application Manager (defining service-specific functionality similar to the Call Agent in the IPCablecom 1.x model).

As an illustration of this pre-authorization model and the use of the Gate Control interface on the CMTS, a typical single-zone (i.e., using a single CMS) on-net IPCablecom 1.x call flow proceeds as follows (some of these steps would normally happen in parallel):

- E-MTA_o boots, provisions and registers with CMS
- CMS issues request to E-MTA_o for notification of off-hook event and dialed-digits
- E-MTA_t boots, provisions and registers with CMS
- CMS issues request to E-MTA_t for notification of off-hook event and dialed-digits
- E-MTA_o goes off-hook, notifies CMS and delivers dialed-digits
- CMS issues a request to E-MTA_o to create a new logical connection and retrieves SDP_o
- CMS issues a request to E-MTA_t to create a new logical connection and retrieves SDP_t
- CMS installs a Gate on CMTS_o and retrieves corresponding GateID_o token

- CMS installs a Gate on CMTS_t and retrieves corresponding GateID_t token
- CMS issues a request (with GateID_o) to E-MTA_o to reserve resources and play ringback
- E-MTA_o issues a DSA-REQ to CMTS_o to establish service flows and reserve resources
- CMS issues a request (with GateID_t) to E-MTA_t to reserve resources and play an alerting tone
- E-MTA_t issues a DSA-REQ to CMTS_t to establish service flows and reserve resources
- E-MTA_t goes off-hook and notifies CMS
- CMS issues a request to E-MTA_o to stop playing ringback, commit resources and cut-through media path
- E-MTA_o issues a DSC-REQ to CMTS_o to commit resources
- CMS issues a request to E-MTA_t to commit resources and cut-through media path
- E-MTA_t issues a DSC-REQ to CMTS_t to commit resources
- Call proceeds

In contrast to the IPCablecom 1.x model in which the client device (i.e., E-MTA) initiates the resource reservation and activation procedures, the IPCablecom Multimedia resource management model allows for the proxying of these steps on behalf of the endpoint through an enhanced Gate Control interface.

This concludes this brief review of DOCSIS and IPCablecom 1.x QoS fundamentals. For further details related to either of these considerably complex topics, please consult the respective primary sources [1] and [14]. The next section provides a summary overview of the IPCablecom Multimedia architecture including each of the primary network elements and associated interfaces as further preparation for the technical protocol specification that follows.

5.2 Architecture

The IPCablecom Multimedia technical report [16] describes an architecture framework and reference model for IPCablecom Multimedia. This specification applies the model contained in the architectural framework and adds normative requirements to provide a scalable, interoperable solution suitable for deploying IPCablecom Multimedia services.

5.2.1 Client Types

The IPCablecom Multimedia tech report defines three kinds of client types:

- Client Type 1, which represents existing "legacy" endpoints (e.g., PC applications, gaming consoles) that lack specific QoS awareness or signaling capabilities. This client has no awareness of DOCSIS, CableHome, or IPCablecom messaging, and hence no related requirements can be placed upon it. Client Type 1 communicates with an Application Manager to request service, and does not (cannot) request QoS resources directly from the MSO access network.
- Client Type 2 is similar to an IPCablecom 1.x telephony MTA in that it supports QoS signaling based on the IPCablecom DQoS specification [14].
- Client Type 3 directly requests QoS treatment from the access network without Application Manager interaction. This client is aware of IETF standards-based RSVP and uses this protocol to request access network QoS resources directly from the CMTS.

In the current release of this specification support is limited to Client Type 1. Consequently, this release of this specification supports only Scenario 1, the "Proxied QoS with Policy Push" scenario, described in [16]. Under this scenario, the Application Manager is responsible for requesting QoS resources on behalf of the client, and a Policy Server pushes the request down to the CMTS, which is the device actually responsible for setting up and managing the DOCSIS service flows required by the application.

5.2.2 IPCablecom Multimedia Devices

In addition to the client (which typically resides at a subscriber's premises), IPCablecom Multimedia requires several network elements residing in, or accessible to and trusted by, the MSO's network. In the process of describing these network elements throughout this specification we borrow heavily from standard IETF terminology and concepts. For a more thorough treatment of the overall IPCablecom Multimedia architecture, including a discussion of underlying requirements and objectives, please refer to the accompanying technical report [16].

Since COPS [10] and COPS-PR [13] each use the terms Policy Enforcement Point (PEP) and Policy Decision Point (PDP) in substantially different interaction scenarios and since IPCablecom Multimedia adds further nuances to these concepts (particularly in the definition of the Policy Server), it is occasionally confusing to think solely in terms of PEPs and PDPs to understand the responsibilities of the various components of the IPCablecom Multimedia architecture. To attempt to alleviate this confusion, portions of this specification employ the notion of a Service Control Domain and a Resource Control Domain to draw distinctions in the type of policy that is being defined and enforced.

The Resource Control Domain (RCD) may be defined as a logical grouping of elements that provide connectivity and network resource level policy management along the packet forwarding paths to and from an end host. The RCD consists of CMTS and Policy Server entities whose responsibilities include management of resources along the packet forwarding paths.

The Service Control Domain (SCD) is defined as a logical grouping of elements that offer applications and content to service subscribers. The Application Manager resides in the SCD. Note that there may be one or more SCDs related to a single RCD. Conversely, each RCD may interact with one or more SCDs.

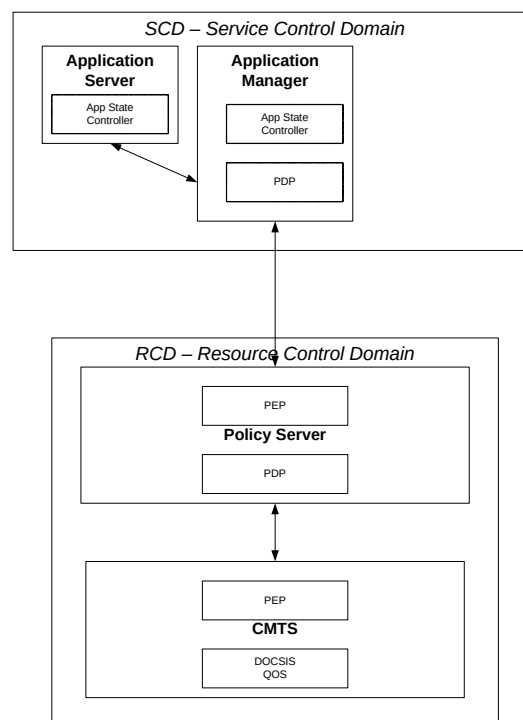


Figure 1 - Session & Resource Control Domains

In the IPCablecom Multimedia architecture the fundamental role of the Application Server is to process requests on behalf of clients for network services and to convert them into session requests. The session request is then forwarded to the Application Manager for additional Service Control Domain processing.

In the IPCablecom Multimedia architecture the fundamental role of the Application Manager is to maintain an application's session-level state and to enforce any SCD policies against client-driven session requests either directly from the client or via the Application Server. If the client's session requests pass the Application Manager's SCD policy checks, the Application Manager converts the session request into a resource request and passes it on to the

Policy Server for Resource Control Domain (RCD) policy checks. If the resource request fails the RCD policy check the Policy Server denies the resource request and the Application Manager consequently denies the client's session request. If, however, the resource request passes the Policy Server's RCD checks, the Policy Server forwards the request on to the CMTS for network-level admission control.

Fundamentally, the roles of the various IP-Cablecom Multimedia components are:

- The Application Server is responsible for relaying client requests to the Application Manager.
- The Application Manager is responsible for application or session-level state and for applying SCD policy.
- The Policy Server is responsible for applying RCD policy and for managing relationships between Application Managers and CMTSs.
- The CMTS is responsible for performing admission control and managing network resources through DOCSIS Service Flows.

It may be useful here to clarify our use of the terms "admission control" and "policy authorization." For the purposes of this specification admission control is generally understood as the process of managing a finite pool of network-level resources (e.g., access network bandwidth, scheduler DOCSIS minislots, or CMTS resources supporting Gates and timers, etc.) and admitting requests against this pool. For performance reasons admission control is usually performed directly on network elements managing the packet forwarding path (such as the CMTS), though some sophisticated Policy Server implementations may choose to maintain state associated with network resources, thus supplementing and contributing to the admission control process.

In contrast, policy authorization is used here to describe higher-level aggregate usage policies (e.g., number of concurrent authorizations for a particular subscriber or service) which constitute an MSO's network management strategy. Policy authorization is almost always defined and enforced at the Policy Server.

The rest of this section describes each of these architectural components and their associated interfaces in more detail.

5.2.2.1 Application Server (AS)

The Application Server is a network entity that interfaces with the Application Manager that requests IP-Cablecom Multimedia services on behalf of clients. The AS may reside on the MSO's network or it may reside outside of this domain and interact with the MSO network via a particular trust relationship. Similarly, the AS may be under direct control of the operator or it may be controlled by a third-party. Any given AS may communicate with one or more Application Managers.

The AS will communicate with a client via a signaling protocol that is outside the scope of this specification. Using this unspecified protocol, the AS authenticates and authorizes client requests based upon Service Control Domain policies. For client requests that pass these checks, the AS determines the particular QoS parameters necessary to deliver the service to the client, based upon its knowledge of the requested service. It then sends a request for these resources to the appropriate Application Manager, which may deny the request based upon additional Service Control Domain policies or may pass the request on to the Policy Server.

5.2.2.2 Application Manager (AM)

As noted in the preceding summary, the Application Manager is a network entity that defines SCD policies, coordinates subscriber-initiated requests for application sessions with access to the resources needed to meet those requests, and maintains application-level state.

The AM may reside on the MSO's network or it may reside outside this domain and interact with the MSO network via a particular trust relationship (typically defined by and enforced on the basis of a Service Level Agreement). Similarly, the AM may be under the direct control of the operator or it may be controlled by a third party. Any given Application Manager may communicate with one or more Policy Servers on the operator's network; likewise, one or more Application Managers may communicate with any given Policy Server on the operator's network (so long as an appropriate trust relationship exists).

The Application Manager may communicate with the Application Server using a signaling protocol as defined in [20]. For Application Server requests the AM authenticates and authorizes the Application Server requests based upon Service Control Domain policies.

The Application Manager may communicate with a client via a signaling protocol that is beyond the scope of this specification. Using this unspecified protocol, the AM authenticates and authorizes client requests based on Service Control Domain policies.

For client and Application Server requests that pass these checks, the AM determines the particular QoS parameters necessary to deliver the service to the client, based on its knowledge of the requested service. It then sends a request for these resources to the appropriate Policy Server, which may deny the request based on network or RCD policy or may pass the request on to the CMTS for admission control and enforcement.

5.2.2.3 Policy Server (PS)

As discussed in RFC 2753 [11], the policy management framework underlying IP-Cablecom Multimedia is based on the work of the IETF's Resource Allocation Protocol (RAP) working group. Since the Policy Server is situated between the Application Manager and the CMTS, it simultaneously plays a dual role as a "proxy" for AM-initiated session requests and as a "sentry" for defining and enforcing Resource Control Domain policy.

As described in [11] and in keeping with the IP-Cablecom 1.x DQoS model, the Policy Server serves as Policy Decision Point (PDP) in relation to the CMTS in that the Policy Server implements MSO-defined authorization and resource-management procedures. Conversely, the Policy Server assumes the role of Policy Enforcement Point (PEP) in relation to the Application Manager as it proxies Gate Control messages to and from the CMTS element.

To revisit the interaction scenario, the Application Manager issues policy requests to the Policy Server. The Policy Server acting as a "sentry" for these requests, and applies a set of policy rules that have been pre-provisioned by the MSO. Upon passing the checks, the Policy Server then acts as a "proxy" with respect to the Application Manager and the CMTS, forwarding the policy request and returning any associated response. Each policy request transaction must be processed individually.

Policy decisions may be based on a number of factors, such as:

- Parameters associated with the request and the status of available resources
- Identity of the particular client and associated profile information
- Application parameters
- Security considerations
- Time-of-day

The primary functions of the Policy Server include:

- A policy decision request mechanism, invoked by Application Managers
- A policy decision request 'policing' mechanism, enforcing installed Policy Rules
- A policy decision delivery mechanism, used to install policy decisions on the CMTS
- A mechanism to allow for the proxying of QoS management messages to the CMTS on behalf of the Application Manager
- An event recording interface to a Record Keeping Server that is used to log policy requests, which may in turn be correlated with network resource usage records

Since the Policy Server functions as a proxy between the AM and CMTS elements (with complementary client and server interfaces) some MSOs may elect to deploy multiple layers of Policy Servers and to delegate certain policy decisions among these servers in order to satisfy requirements associated with scalability and fault-tolerance.

5.2.2.3.1 Stateful & Stateless Policy Servers

There are two basic classes of Policy Servers – Stateful and Stateless. A Stateless Policy Server is a slight misnomer since it does maintain enough state to map Application Manager requests to the proper CMTS and maintain COPS

session state, while a pure Stateless Policy Server maintains no state on any of the media sessions. Stateful Policy Servers come in several varieties – some participate in admission control and thus monitor the QoS attributes of active media sessions, some leave QoS and admission control to the CMTS but monitor time-based or volume-based service requests from the Application Manager, and some Policy Servers are somewhere between these extremes.

The reason there is a variety of Policy Server types is that there is a variety of environments that operators are trying to support. For example, some operators may wish to support IPCablecom Multimedia over the same CMTSs that they use for IPCablecom telephony, and they may want a single CMS/Policy Server that has a more global view of the network resources being used. On the other hand, some operators may wish to run an IPCablecom Multimedia-only environment, or they may utilize simpler CMTS-driven mechanisms for partitioning IPCablecom Multimedia and telephony resources. These simpler configurations have more modest requirements on the amount of state that a Policy Server maintains.

Policy Server state requirements can also be driven by the level of trust between the Policy Server and Application Manager; a Stateful Policy Server can more readily police Application Manager session control behavior than can a Stateless Policy Server. So a Stateful Policy Server may be more appropriate for operators supporting third party Application Managers. Other operators may rely on economics to enforce their trust relationships with Application Managers, or they may control the Application Managers themselves. In such cases a Stateless Policy Server may be more appropriate.

Since it is impossible to categorize all the various components of media session and network QoS state that a Policy Server is maintaining, the protocol is designed to be independent of this complexity. A Stateful Policy Server gleans IPCablecom Multimedia media session information from the Application Manager requests it proxies; any other information it requires is gathered via mechanisms that are outside the scope of this specification. The CMTS and the Application Manager make no distinction as to the type of Policy Server to which they are connected, and the protocol is designed in such a manner that the type of Policy Server is transparent to the end point. The type of Policy Server is only of importance to the operator.

Since some types of Policy Servers attempt to assist with admission control and may have a larger view of the network and its resources, additional state synchronization issues may arise in design in a network which contains more than one of these types of Policy Servers. It is the responsibility of the operator to ensure that the efforts of these Policy Servers are not undermined by a network that includes other autonomous Policy Servers.

5.2.2.3.2 *Modification of Requests and Responses by Policy Servers*

Although nominally a part of the Resource Control Domain, the Policy Server can be an intermediary between the Service and the Resource Control Domains, in addition to its normal role of implementing MSO-defined authorization and resource management procedures. In either of these capacities it may modify the incoming request before forwarding it to the CMTS.

In acting as an intermediary between the SCD and RCD, the Policy Server may translate fields from formats or scales used in the SCD into formats or scales used in the RCD. For example, the Policy Server may modify the "priority" of a request coming from an Application Manager (especially important to do for an AM outside of the MSO network) so that this priority field uses a consistent scale throughout the operator's RCD. In its capacity as an intermediary, the Policy Server may use bidirectional translation – in other words, it should translate requests from the AM to the CMTS and "untranslate" the responses from the CMTS to the AM. This capability can be supported by stateful policy servers by remembering the original request, and it can be supported by stateless Policy Servers if the translation function is invertible.

Modification of certain objects, specifically the Classifier and Traffic Profile objects, may cause operational problems in the originating AM. As such, these objects MUST NOT be modified by the policy server. Aside from these exceptions, all other objects may be policed and modified at the PS's discretion based on provisioned policy rules.

5.2.2.4 *Cable Modem Termination System (CMTS)*

In describing the role of the CMTS network element, it is important to consider the relation among DOCSIS, IPCablecom 1.x and IPCablecom Multimedia functionality. While each of these suites of specifications addresses a specific set of functional requirements, each has also been defined in such a way that corresponding implementations may be constructed in a modular manner; either IPCablecom 1.x or IPCablecom Multimedia Gate

Control may be layered on top of a DOCSIS 1.1 or greater CMTS foundation, with the option of adding additional, complementary functionality as business indicates. Further, it should be emphasized that it is a significant asset of the IPCablecom architecture that both telephony and Multimedia variants employ considerable architectural similarity, leading to potential reuse in the underlying Gate management models.

The IPCablecom Multimedia CMTS is a generalized version of the IPCablecom 1.x CMTS that has been defined in order to deliver telephony services in IPCablecom 1.x networks. The CMTS is responsible for fulfilling requests for QoS that are received from one or more Policy Servers. It performs this function by installing Gates, which are similar to the Gates defined in [14]; Gates allow the subscriber's cable modem to request network resources from the CMTS through the creation of dynamic DOCSIS flows with guaranteed levels of QoS. The CMTS also sends Event Messages detailing actual usage of QoS resources to the Record Keeping Server.

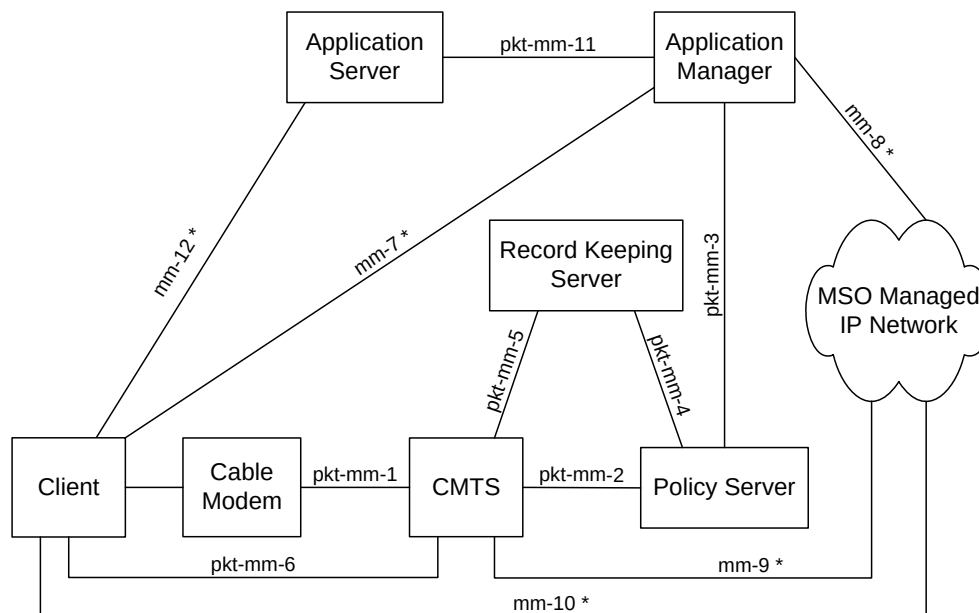
5.2.2.5 Record Keeping Server (RKS)

The IPCablecom Multimedia Record Keeping Server fulfills a similar role to the RKS in IPCablecom 1.x [15]. It receives Event Messages pertaining to policy decisions from the Policy Server and Event Messages pertaining to QoS resource usage from the CMTS.

In the IPCablecom Multimedia architecture, the Record Keeping Server does not receive messages directly from the Application Manager. However, the Application Manager can embed opaque data in messages that it sends to the Policy Server, and these data can then be included in Event Messages that are subsequently sent to the RKS.

5.2.3 IPCablecom Multimedia Interfaces

IPCablecom Multimedia builds on the IPCablecom 1.x suite of specifications. Where an IPCablecom Multimedia interface has a corollary in IPCablecom 1.x, IPCablecom Multimedia uses the same protocol, or an extension of the same protocol.



* Out of Scope

Figure 2 - IPCablecom Multimedia Architectural Framework

Table 1 - IPCablecom Multimedia Interfaces

Interface	Description	Notes
pkt-mm-1	CMTS – CM	The Cable Modem (CM) may request QoS from the CMTS via DOCSIS DSx signaling. Alternatively, the CMTS may instruct the CM to setup, teardown or change a DOCSIS service flow in order to satisfy a QoS request, again via DSx signaling.
pkt-mm-2	PS – CMTS	This interface is fundamental to the policy-management framework. It controls policy decisions, which may be: (a) pushed by the Policy Server (PS) onto the CMTS, or (b) pulled from the PS by the CMTS. The interface also allows for proxied QoS requests on behalf of a client. In some scenarios, this interface may also be used to inform the PS when QoS resources have become inactive.
pkt-mm-3	AM – PS	The Application Manager (AM) may request that the PS install a policy decision on the CMTS on behalf of the client. This interface may also be used to inform the AM of changes in the status of QoS resources.
pkt-mm-4	PS – RKS	The PS sends event messages to the Record Keeping Server (RKS) to track policy decisions related to QoS.
pkt-mm-5	CMTS – RKS	The CMTS sends the RKS event messages to track requests for and usage of QoS (e.g., service flow additions, changes, deletions, and volume metrics).
pkt-mm-6	Client – CMTS	The client may use this interface to directly request and manage QoS network resources. If authorized, these resources are provided by the CMTS.
mm-7	Client – AM	This interface may be used by the client to interact with the AM and to request and manage QoS resources indirectly. This interface is out of scope for this version of this specification.
mm-8	AM – Peer	The AM may use this interface to interact with some other entity that is part of the application in question. This interface is out of scope for this version of this specification.
mm-9	CMTS – MSO-Managed IP Network	This interface on the CMTS may be used in support of end-to-end QoS requests beyond the access network. This interface is out of scope for this version of this specification.
mm-10	Client – Peer	The Client may use this interface to interact with some other entity that is part of the application in question. This interface is out of scope for this version of this specification.
pkt-mm-11	AS - AM	The Application Server uses this interface to send network resource requests on behalf of the client to the AM. This interface may also be used to notify the AS of changes in the status of the network resources.
mm-12	Client – AS	This interface may be used by the client to interact with the AS and to request applications, which indirectly requests network resources. This interface is out of scope for the current effort.

5.2.3.1 Client and Application Manager Interface (mm-7)

The interface between the client and the Application Manager is out of scope. Typically, the Application Manager will, through some means beyond the scope of this specification, authenticate the client and ensure that the client is entitled to the Multimedia service. For example, the client may login to a web page and request service by providing a username and password. However it is accomplished, the Application Manager must be able to identify unambiguously the cable modem(s) to which service is to be delivered, since this information must be made available to the network operator before QoS can be delivered.

5.2.3.2 Application Manager and Policy Server Interface (pkt-mm-3)

This interface corresponds to the IPCablecom 1.x interface between a Call Agent and a Gate Controller. In IPCablecom 1.x this is a hidden, non-testable interface, and so there are no pre-existing protocol requirements on this interface.

IPCablecom Multimedia requires the use of COPS [10] on this interface. In order to simplify the architecture and to allow for multiple levels of Policy Server elements between the Application Manager and CMTS, this interface mirrors as far as possible the interface between the Policy Server and the CMTS. Although the Application Manager is the one requesting resource authorization from the Policy Server, it actually issues this request in a COPS Decision message, instead of a COPS Request message. This allows the interface between the Application Manager and the Policy Server to appear identical to the interface between the Policy Server and the CMTS. The Application Manager is the PDP relative to the Policy Server, and the Policy Server is the PEP relative to the Application Manager.

When an Application Manager agrees to provide service to a client, it sends a COPS Decision that contains (at least) the following information in the form of COPS objects:

- Identity of the Application Manager making the request
- Identity of the client(s) to whom service is to be provided
- RSVP FlowSpec(s) specifying traffic envelope(s) for the session

In the reply from the Policy Server, the server includes an authorization token, the Gate-ID, which is provided to it by the CMTS.

5.2.3.3 Policy Server and CMTS Interface (pkt-mm-2)

This interface is essentially identical to the equivalent interface (between CMTS and Gate Controller) in IPCablecom 1.x. As in IPCablecom 1.x, COPS is used to transfer policy information between the Policy Server and the CMTS. The CMTS acts as a COPS PEP and the Policy Server acts as a COPS PDP. Following the IPCablecom 1.x model, the Policy Server initiates communication for a Multimedia session by sending a DQoS Gate-Set message (which is an unsolicited COPS DECISION message) to the CMTS.

This message contains (at least):

- Application Manager ID
- Subscriber ID
- GateSpec
- FlowSpec(s)
- Classifier

The CMTS responds, as in DQoS, with either Gate-Set-Ack or Gate-Set-Err (both of which are COPS REPORT messages).

If the CMTS responds positively (i.e., with a Gate-Set-Ack), it includes a GateID. As in IPCablecom 1.x, the GateID acts as an authorization token. Unlike IPCablecom 1.x, the token is not ultimately passed to the client (since Client Type 1 endpoints have no knowledge of IPCablecom); instead it is held by the Policy Server (if stateful) and by the Application Manager, thus allowing them to issue commands pertaining to this session to the CMTS, either directly in the case of the Policy Server, or indirectly via the Policy Server in the case of the Application Manager.

5.2.3.4 Record Keeping Server and Policy Server Interface (pkt-mm-4) and Record Keeping Server and CMTS Interface (pkt-mm-5)

The interfaces into the Record Keeping Server from the Policy Server and the CMTS are identical to the equivalent interfaces (from CMS and CMTS, respectively) in IPCablecom 1.x (see [15]). These interfaces are used to carry IPCablecom Event Messages, which use RADIUS formatting. In IPCablecom Multimedia, Event Messages carry

detailed information pertaining to the service delivered, including the exact time that service flows are created and deleted and (optionally) the amount of traffic that passed over the service flow while it was in existence.

5.2.3.5 Client and Application Server Interface (mm-12)

The interface between the client and the Application Server is out of scope. Typically, the Application Server will, through some means beyond the scope of this specification, authenticate the client and ensure that the client is entitled to the Multimedia service. However it is accomplished, the Application Server must be able to identify the subscriber to which service is to be delivered, since this information must be made available to the network operator before QoS can be delivered.

5.2.3.6 Application Server and Application Manager Interface (pkt-mm-11)

The interface between the Application Server and the Application Manager is defined in [20]. IPCablecom Multimedia requires the use Web Services on this interface as specified in [20]. Application Servers use this interface to send network resource requests on behalf of clients to the AM. The interface may also be used by the AM to send notifications to the AS of changes in the status of network resources.

When an Application Server agrees to forward a service request for a client, it sends a Web Service message that contains (at least) the following information in the form of message arguments:

- Subscriber Identifier
- Service Name

The interface supports a number of optional arguments that may be used to provide more granularity when requesting the service.

5.2.4 State Information

In this section, we provide an overview of state location in an IPCablecom Multimedia system. In addition to maintaining detailed state information, devices send information about state transitions to the Record Keeping Server for purposes such as billing, fraud detection, session reconstruction, etc.

5.2.4.1 Application State

The Application Manager is at all times responsible for maintaining detailed knowledge of the state of the application media session. How it does so in detail is beyond the scope of this specification, but it is important to recognize that no devices other than the Application Manager are required or expected to maintain any knowledge of the application state.

The Application Manager may, however, report session state by proxying such information via the Policy Server to the Record Keeping Server. In addition, some coarse state information (such as the fact that resources have been requested) is automatically sent from the Policy Server to the Record Keeping Server.

5.2.4.2 QoS Resource State

The CMTS naturally is aware of the detailed state of flows that it manages. The Policy Server (if it is a stateful Policy Server) may also maintain a notion of the state of QoS resources on a single CMTS; it may also collate the state information over several CMTSs so that it (and only it) knows the QoS state of the entire system. This may be important, for example, if an operator has instituted a policy whereby a particular application is not to be permitted to consume more than a specified percentage of the total system resources. In a network with only stateless Policy Servers, the CMTSs are the only devices to maintain QoS state information. Since stateless Policy Servers do not maintain GateIDs, they cannot even interrogate a CMTS to obtain information about a particular Multimedia session.

Whenever a QoS Resource transitions from one state to another, and whenever a QoS Resource is deleted, a corresponding Event Message is sent from the CMTS to the Record Keeping Server.

6 AUTHORIZATION INTERFACE DESCRIPTION

This section describes the interface between the Application Manager and the Policy Server, and the interface between the Policy Server(s) and the CMTS.

The interface between the Application Manager and the Policy Server is translationally symmetric to the interface between the Policy Server and the CMTS. The interfaces are used to pass authorization, reservation and activation information to the CMTS, to provide state information from the CMTS to the Policy Server, and from the Policy Server to the Application Manager.

The Application Manager is the PDP for the Service Control Domain. The Policy Server is the PEP with respect to the Application Manager and applies Resource Control Domain policies. The Policy Server is a PDP with respect to the CMTS, and the CMTS is the PEP with respect to the Policy Server and lies in the actual packet forwarding path.

This section describes the use of the COPS protocol to transport IPCablecom QoS messages between the Application Manager and Policy Server, and between the Policy Server and CMTS.

6.1 Gates: The Framework for QoS Control

An IPCablecom Multimedia Gate is a logical representation of a policy decision that has been installed on the CMTS. A Gate is used to control access by a single IP flow to enhanced QoS Services provided by a DOCSIS cable network. Gates are unidirectional; a single Gate controls access to a flow in either the upstream or the downstream direction, but not both. For a bi-directional IP session, two Gates are required, one for upstream and one for downstream, each identified by a unique GateID. It is important to recognize that this is a fundamental difference from IPCablecom 1.x, in which a single GateID may reference both an upstream and a downstream Gate.

In IPCablecom Multimedia, each Gate has a separate GateID. The Gate defines the authorization, reservation and committed envelopes to be used by the CMTS to perform authorization, reservation and commit operations.

Gates can be further qualified as either unicast or multicast. Fundamentally the two Gates are as described above with one key differentiator between them - the destination IP address indicated in the classifier object. In a Unicast Gate the destination IP address (v4 or v6) contained in the classifier identifies a unicast IP address (or a range of unicast IP addresses). A Multicast Gate contains a destination IP address (v4 or v6) in the classifier identifying a multicast IP address. See RFC 4291 [23] for IPv6 unicast and multicast IP address definitions and RFC 3171 [22] for IPv4 multicast address assignments.

Throughout this document, unless specifically called out as either a Unicast Gate or Multicast Gate, the reference to Gate applies to both Gate types.

In all Scenarios, the CMTS MUST perform admission control checks of the envelopes to make sure that the committed envelope is less than or equal to the reserved envelope, and the reserved envelope is less than or equal to the authorized envelope. (See [1] for specific DOCSIS admission control requirements.)

In the 'Proxied QoS Policy Push' (Scenario 1) model, the information in a Gate is used by the CMTS to create the DOCSIS Service Flow directly, after the CMTS performs the necessary admission control checks of the envelopes. In the other two models outlined in [16], 'Client requested QoS with Policy Push' (Scenario 2) and 'Client requested QoS Policy Pull' (Scenario 3), the CMTS uses the Gate information to perform admission control of the client requested resources; the CMTS does not initiate creation of the flows. The Application Manager is responsible for issuing Gate messages to the Policy Server and the Policy Server is responsible for applying policy rules, and then issuing Gate Control messages to the CMTS.

A Gate consists of the following elements, which are described later in this section:

- GateID
- AMID
- SubscriberID
- GateSpec
- Classifier

- Traffic Profile
- Event Generation Info (optional)
- Time-Based Usage Limit (optional)
- Volume-Based Usage Limit (optional)
- Opaque Data (optional)
- UserID (optional)
- SharedResourceID (optional)

GateID is the handle for the Gate. The GateID is assigned by the CMTS and is used by the Application Manager, Policy Server, and client to reference the Gate.

AMID is the handle that identifies the Application Manager and Application Type.

SubscriberID uniquely identifies the Client for which the policy is being set.

GateSpec describes specific authorization parameters defining a Gate (i.e., QoS limits, timers, etc.).

Classifier describes the IP flow(s) that will be mapped to the DOCSIS Service Flow.

Traffic Profile describes the QoS attributes of the Service Flow used to support the IP flow.

Event Generation Information contains information used by the CMTS for the purpose of accounting and usage reporting.

Volume-Based Usage Limit defines a volume cap for traffic traversing the flow associated with the Gate.

Time-Based Usage Limit describes a time cap limiting the duration of the flow associated with the Gate.

Opaque Data represents a general-purpose object that remains opaque to the CMTS and PS elements, but which may contain data that is significant to the AM. This optional object, if provided by the AM, is retained at the CMTS and returned in all associated responses (see Section 6.4.2.11).

UserID identifies the User for which the policy is being set.

SharedResourceID indicates the use of a shared underlying resource in fulfilling the request, and uniquely identifies that resource. The SharedResourceID could be used by the Application Manager and Policy Server to aid in understanding true resource allocation. A single SharedResourceID may be associated with multiple gates, allowing an accurate summary of resource allocations while preserving the gate's role in identifying a specific policy decision.

These elements are communicated to the Policy Server and the CMTS via COPS objects and are described in greater detail later in this section. During the installation of the Gate, the information above is communicated to the CMTS. After the installation is complete, a DOCSIS Service Flow can be created. After the creation of the DOCSIS Service Flow, the Gate has associated with it an additional element, the DOCSIS Service Flow.

A Gate transitions through multiple states. In Scenarios 2 and 3, where the client entity is responsible for reserving and then activating the DOCSIS Service Flows, a Multimedia Gate behaves in a manner very similar to an IPCom 1.x DQoS Gate. When the Policy Server installs the Gate onto the CMTS, the Gate is said to be in an 'Authorized' state. It remains in this state until explicitly deleted by the Policy Server (or, less likely, it is deleted for some reason by the CMTS itself) or until a dynamic flow request from the client arrives.

When the client requests that a dynamic Service Flow be added, it presents the GateID as an authorization token. The CMTS uses the GateID to perform admission control on the DOCSIS dynamic flow against the authorized envelope defined by the Gate. In Scenario 1, the Policy Server instructs the CMTS to transition between the states on behalf of the Application Manager, and the CMTS is the entity responsible for initiating and tearing down DOCSIS Service Flows. The State Transition section in this document describes this behavior. When the CMTS is instructed to tear down a DOCSIS Service Flow, its associated Gate remains until explicitly deleted by the PS/AM or until it times out and its resources are reclaimed by the CMTS (see Section 6.5.8). In contrast, however, when the PS/AM deletes a Gate, the CMTS will delete the associated DOCSIS Service Flow.

6.1.1 Gate Identification (GateID)

A GateID is an identifier that is locally allocated by the CMTS where the Gate resides. A GateID MUST be associated with only one Gate. Whereas the IP-Cablecom 1.x DQoS Gate Control model generally assumed a pair of unidirectional Gates (one upstream and one downstream) per GateID in support of a typical two-way voice session, here the Gate/GateID relationship is explicitly one-to-one, so that it is easier to support a wide range of Multimedia services.

When the Application Manager issues a Gate-Set request, this triggers the Policy Server to issue a Gate-Set message to the CMTS. When the CMTS responds with an acknowledgment containing the GateID, the Policy Server forwards this response including the GateID back to the Application Manager. Note that since there can be a many-to-many relationship between a PS and CMTS, the GateID assigned by one CMTS cannot be guaranteed to be unique across the network, so the PSs may use the AMID of the AM along with the SubscriberID and GateID in order to uniquely identify the Gate.

An algorithm that may be used to assign values of GateIDs is as follows. Partition the 32-bit word into two parts, an index part, and a random part. The index part identifies the Gate by indexing into a small table, while the random part provides some level of obscurity to the value. Regardless of the algorithm chosen, the CMTS SHOULD attempt to minimize the possibility of GateID ambiguities by ensuring that no GateID gets reused within three minutes of its prior closure or deletion. For the algorithm suggested this could be accomplished by simply incrementing the index part for each consecutively assigned GateID, wrapping around to zero when the maximum integer value of the index part is reached.

6.1.2 Application Manager Identification (AMID)

The AMID consists of two fields: the Application Manager Tag and Application Type. Each Application Manager is pre-provisioned with an Application Manager Tag that is unique within the universe of a single service provider. The Application Manager may also be pre-provisioned with a set of Application Type values that can be used to identify the particular application that a gate is associated with. The Application Manager includes the AMID in all messages that it issues to the Policy Server. The Policy Server transparently passes this information to the CMTS via Gate Control messages. The CMTS MUST return the AMID associated with the Gate to the Policy Server. The Policy Server uses this information to associate Gate messages with a particular Application Manager and Application Type.

The Application Manager Tag MUST be a globally unique value assigned to the Application Manager by the service provider. The Application Manager MUST use the assigned Application Manager Tag in all its interactions with Policy Servers. Note that since the Application Manager may be operated by a third party, and a single Application Manager could interact with multiple service provider operators, a single physical Application Manager may be provisioned with multiple Application Manager Tags, and multiple Application Type sets (one for each configured Application Manager Tag).

6.1.3 Subscriber Identification (SubscriberID)

The SubscriberID, consisting of the IPv4 or IPv6 address of either the CM or client CPE device (either directly connected to the CM or on a routable network behind the CM), identifies the user requesting the service. In complex network environments this address may be used to route Gate Control messages between a number of Policy Servers and to determine which CMTS is providing service to a particular endpoint. In addition to the IPv4 or IPv6 address, a subscriber may also be identified via a FQDN or some opaque data (object defined below) relevant to the service in question.

For a Multicast Gates the CMTS uses the SubscriberID to decide where the Multicast replication needs to be created. The CMTS treats the SubscriberID as the source IP address of a JoinMulticastSession [1]. If the SubscriberID is on an IP subnet that is not directly connected to the CMTS, the CMTS MAY reject the Gate as having an invalid SubscriberID see 6.4.2.14 IP-Cablecom Error.

6.1.4 Gate Specification (GateSpec)

The GateSpec describes some high-level attributes of the Gate, and contains information regarding the treatment of other objects specified in the Gate message. Information contained in a GateSpec is outlined below:

- SessionClassID
- Direction
- Authorized Timer
- Reserved Timer
- Committed Timer
- Committed Recovery Timer
- DSCP/TOS Overwrite
- DSCP/TOS Mask

SessionClassID provides a way for the Application Manager and the Policy Server to group Gates into different classes with different authorization characteristics. For example, one could use the SessionClassID to represent some prioritization or preemption scheme that would allow either the Policy Server or the CMTS to preempt a pre-authorized Gate in favor of allowing a new Gate with a higher priority to be authorized.

Direction indicates whether the Gate is for an upstream or downstream flow. Depending on this direction, the CMTS MUST reserve and activate the DOCSIS flows accordingly. For Multicast Gates the CMTS needs to only support flows or gates in the downstream direction.

Authorized Timer limits the amount of time the authorization must remain valid before it is reserved (see Section 6.2).

Reserved Timer limits the amount of time the reservation must remain valid before the resources are committed (see Section 6.2).

Committed Timer limits the amount of time a committed service flow may remain idle.

Committed Recovery Timer limits the amount of time that a committed service flow can remain without a subsequent refresh message from the PS/AM once the PS/AM has been notified of inactivity (see Section 6.2).

DSCP/TOS Overwrite field can be used to overwrite the DSCP/TOS field of packets associated with the DOCSIS Service Flow that corresponds to the Gate. This field may be unspecified in which case the DSCP/TOS field in the packet is not over-written by the CMTS. This field may be used in both the upstream and downstream directions. The CMTS MUST support DSCP/TOS overwrite for upstream gates. The CMTS MAY support DSCP/TOS overwrite for downstream gates. If DSCP/TOS is enabled in a downstream gate and the CMTS does not support that function, then the field is ignored.

6.1.5 Classifier

A classifier MUST be defined for a Gate. Additional classifiers may be included in the original Gate-Set. Classifiers may be added or deleted in a subsequent Gate-Set. Compliant implementations MUST be able to support a minimum of four classifiers when processing a Gate-Set message for Unicast Gates. Compliant implementations MUST support a minimum of one classifier when processing a Gate-Set message for Multicast Gates. The AM and PS MUST NOT mix unicast and multicast destination IP addresses for a single Gate. The classifier identifies the IP flow that will be mapped to the DOCSIS service flow associated with the Gate. In Scenario 1, when the CMTS creates the dynamic flow, it MUST use the Gate classifier as the classifier for the DOCSIS Service Flow according to the rules specified in Sections 9.3.4.1 and 9.4.3.1.

There are three types of classifiers defined in this specification; a Classifier, an Extended Classifier, and an IPv6 classifier, which are described below. For IPv4 traffic, an Application Manager SHOULD use the Extended Classifier to properly match the traffic it is interested in unless it receives a Gate-Set-Err with either Error-Code 6 (Missing Required Object) and Error-Subcode of 6 | 1 (S-Num | S-Type) or Error-Code 7 (Invalid Object) and Error-Subcode of 6 | 2. This would occur if the AM is interacting directly with a Policy Server or indirectly with a CMTS that does not support the Extended Classifier. In this case, the AM may resend the Gate using the Classifier object instead of the Extended Classifier object. The Classifier object is retained for legacy purposes. That being said, the legacy Classifier object will be deprecated from the specification in a future release.

For IPv6 traffic, an Application Manager MUST use an IPv6 Classifier.

For Multicast Gates (where the Destination IP Address in the classifier identifies a specific Multicast IP address) the AM encodes the classifier (any of the three classifier types) as follows:

- The Source IP address & the Mask/Prefix Length combination MUST resolve to a specific valid Unicast IP address value i.e., not resolve into a range of addresses. If the Source IP address and the Mask/Prefix Length combination resolve to the All-Zeros IP Address, it indicates that all Source IP addresses are permitted, i.e., Any Source Multicast (ASM).
- The Destination IP address and the Mask/Prefix Length combination MUST resolve to a single specific Multicast address.
- The Source and Destination Port Number fields in the classifier MUST be ignored by the CMTS for a Multicast Gate.

The CMTS MUST reject a Multicast gate request which does not meet the above stated Multicast gate classifier rules with error code 7 (invalid object) and the Error-subcode identifying the invalid classifier object.

A PS MUST support all three types of classifiers specified here. A CMTS MUST support both legacy and Extended Classifiers. A CMTS that supports DOCSIS 3.0 MUST support the IPv6 Classifier. An Application Manager MUST use only one classifier type per Gate-Set, and MUST continue to use that type for all subsequent Gate-Sets for that Gate.

Throughout this specification, unless stated otherwise, the "classifier" refers generically to either the Classifier object, Extended Classifier object, or IPv6 Classifier object, depending on which type was used in the Gate-Set.

A (legacy) Classifier is an eight-tuple:

- Protocol
- Source IP
- Source Port
- Destination IP
- Destination Port
- Priority
- DSCP/TOS Field
- DSCP/TOS Mask

An Extended Classifier is defined by the following tuple:

- Protocol
- IP Source Address
- IP Source Mask
- Source Port Start
- Source Port End
- IP Destination Address
- IP Destination Mask
- Destination Port Start
- Destination Port End
- Priority
- DSCP/TOS Field

- DSCP/TOS Mask
- ClassifierID
- Activation State
- Action

An IPv6 Classifier is defined by the following tuple:

- Tc-low
- Tc-high
- Tc-mask
- Flow Label
- Next Header Type
- Source Prefix Length
- Destination Prefix Length
- IPv6 Source Address
- IPv6 Destination Address
- Source Port Start
- Source Port End
- Destination Port Start
- Destination Port End
- ClassifierID
- Priority
- Activation State
- Action

Protocol field, in a legacy Classifier or Extended Classifier, identifies the type of protocol (e.g., IP, ICMP, etc). The Next Header Type field serves a similar function in the IPv6 Classifier.

Source IP, IP Source Address, or IPv6 Source Address (in the case of Extended Classifier or IPv6 Classifier) is the IP address (as seen at the CMTS) of the originator of the IP flow, while Destination IP, IP Destination Address or IPv6 Destination Address is the termination point for the IP flow. When using an Extended Classifier, a subnet can be used to enable classifying multiple IP addresses with a single Extended Classifier object by the use of the IP Source Mask and/or IP Destination Mask fields. Similarly, when using the IPv6 Classifier, the Source Prefix Length and Destination Prefix Length indicate how many high order bits in the corresponding IPv6 Address to consider in determining a match.

Source Port and Destination Port specify the UDP or TCP ports for the IP flow. When using an Extended Classifier or IPv6 Classifier, the Source Port (Start/End) and Destination Port (Start/End) specify the UDP or TCP port ranges for matching the IP flow.

Priority may be used to distinguish between multiple classifiers that match a particular packet. This is typically set to a default value since classifiers are generally intended to be unique.

When using the legacy Classifier or Extended Classifier, DSCP/TOS field identifies the DSCP/TOS field that must be matched for packets to be classified onto the IP flow. To provide for maximum flexibility in defining a network management strategy, an accompanying mask is defined which determines which bits in the DSCP/TOS byte are to be used as filters in classifying packets. This allows for both DiffServ and TOS strategies (which each define and use separate bits within this byte).

When using IPv6 Classifier the Traffic Class Range and Mask fields, tc-low, tc-high, and tc-mask, allow matching on the IPv6 Traffic Class value. An IPv6 packet with IPv6 Traffic Class value "ip-tc" matches this parameter if $tc\text{-}low \leq (ip\text{-}tc \text{ AND } tc\text{-}mask) \leq tc\text{-}high$. If the tc-mask field is set to 0, then comparison of the IPv6 packet Traffic Class byte for this entry is irrelevant.

Note: The value 0x3F for tc-mask will exclude the Explicit Congestion Notification [21] bits from the comparison, and hence will result in classification based on DSCP [8].

The Flow Label field matches the Flow Label used in the IPv6 header. The 20 least significant bits represent the 20-bit IPv6 Flow Label while the 12 most significant bits are ignored. If this parameter is set to 0, then comparison of IPv6 flow label for this entry is irrelevant.

A classifier MAY have wild-carded fields (indicated by zeroed fields, unless otherwise specified), but care must be taken so that multiple IP flows do not unintentionally match the same classifier, which can lead to unexpected results.

Fields unique to the Extended Classifier and IPv6 Classifier:

- The ClassifierID field is used as an identifier for the classifier. It can be used to reference an existing classifier when making changes to that classifier.
- The Activation State field is used to instruct the CMTS to either activate or inactivate the classifier.
- The Action field is used to add, replace, or delete the classifier.

6.1.6 Traffic Profile

There are four basic ways to express the Traffic Profile for a Gate:

1. FlowSpec
2. DOCSIS Service Class Name, or
3. DOCSIS-Specific Parameterization
4. Upstream Drop

The Policy Server or Application Manager MUST define the Traffic Profile for a Gate using one of the following: (1) the FlowSpec, (2) DOCSIS Service Class Names, (3) DOCSIS-Specific Parameters, or (4) Upstream Drop. All envelopes used in a Traffic Profile MUST be the same type, i.e., either FlowSpec, DOCSIS Service Class Names, DOCSIS-Specific Parameters or Upstream Drop.

There MUST be at least one set of Traffic Profile parameters specified when the Gate is first being installed. The Policy Server and Application Manager MAY specify a second set to represent the reserved envelope, and a third set to represent the committed envelope. If the CMTS is told to immediately create a dynamic flow upon receipt of a Gate-Set message (via the presence of the reserved or committed envelopes), the CMTS MUST use the reserved and committed envelope Traffic Profile parameters to perform the DOCSIS messaging to create the flow, in the direction specified by the Direction field in the GateSpec (provided the request is authorized and sufficient resources exist to satisfy the request). When told to transition into the Committed state, the CMTS MUST use the Traffic Profile to activate the DOCSIS Service Flow. As an optimization, the Policy Server MAY tell the CMTS to perform all three actions (authorize, reserve and commit) on behalf of the Application Manager via a single Gate Control message. Alternatively, the PS/AM MAY issue separate Gate-Set messages to tell the CMTS to authorize and reserve and then to commit via a subsequent Gate-Set message.

6.1.6.1 FlowSpec

The FlowSpec object contains RSVP FlowSpecs that are used to describe the Traffic Profile of the IP flow. The FlowSpec object can contain multiple RSVP FlowSpecs:

- FlowSpec that defines the authorization resource envelope against which future reservations can be made.
- FlowSpec that defines the reserved envelope against which future commit requests can be made.
- FlowSpec that defines the resources to be committed.

RSVP FlowSpecs support two types of services: controlled load [5] and guaranteed [6]. The main difference between the two types of services is discussed in Section 8. The two types of services are distinguished based on the FlowSpec Service number, which is specified in the RSVP FlowSpec. Service number 5 is for controlled load, and Service number 2 is for guaranteed. A controlled load service MUST contain only the TSpec token bucket parameters, and not the RSpec. A guaranteed service MUST contain both the TSpec and the RSpec.

Please refer to Section 8 for information on how to explicitly map RSVP parameters into DOCSIS parameters. When deriving the DOCSIS parameters using the RSVP FlowSpec parameters, there are some DOCSIS parameters that are highly approximated. If the approximations do not give the Policy Server or Application Manager the control it desires, the PS/AM MAY use the other methods of defining the Traffic Profile, which includes the ability to define some DOCSIS-specific parameters. These parameters allow the Policy Server or Application Manager to fine-tune the standard mapping of FlowSpecs to DOCSIS parameters.

6.1.6.2 DOCSIS Service Class Name

The DOCSIS Service Class Name indicates the DOCSIS Service Class to be used to describe the QoS attributes. A CMTS MUST support DOCSIS Service Class Names.

DOCSIS Service Class Name enables one to use pre-provisioned DOCSIS QoS parameters on the CMTS. On the CMTS, one can configure DOCSIS Named Service Classes with different DOCSIS QoS profiles, then reference the DOCSIS Service Class Name in the Gate to indirectly associate a QoS profile with a particular Gate. When using the DOCSIS Service Class Name traffic profile, values in the GateSpec, with the exception of the direction flag, MUST overwrite the equivalent defined parameter value in the Service Class Name. For example, if T3 is 30 seconds in the GateSpec and the Service Class Name defined the Timeout for Active QoS Parameters to 20, the value in the GateSpec, i.e., 30, will be used. This currently applies to the GateSpec DSCP/TOS Overwrite, DSCP/TOS Mask, T2 and T3+T4, which map to the DOCSIS tos-or-mask, tos-and-mask, Timeout for Admitted QoS Parameters and Timeout for Active QoS Parameters respectively. If the CMTS receives a GateSpec with a direction flag that does not match the direction associated with the Service Class Name specified in the Gate-Set, the CMTS MUST send a Gate-Set-Err with error code 11 (Undefined Service Class Name).

For more information on DOCSIS Service Classes please refer to section 7.5.3 of the DOCSIS specification [1].

6.1.6.3 DOCSIS Specific Parameterization

The third way of defining the Traffic Profile consists of using DOCSIS-specific Traffic Profile; this allows the Application Manager to explicitly specify the DOCSIS parameters of the DOCSIS flow. If the Application Manager wishes to use this third way of defining a Traffic Profile, it MUST include an object containing the DOCSIS Specific Parameters.

All DOCSIS Service Flow Scheduling types are supported via several different Traffic Profile S-Types. Each S-Type has a different encoding of the DOCSIS-specific parameters relevant to that Service Flow Scheduling type. For further details regarding DOCSIS-specific parameterization refer to Section 6.4.2.7.

6.1.6.4 Upstream Drop

An Upstream Drop Traffic Profile represents a null service flow for upstream traffic filtering at the CM. DOCSIS 3.0 refers to a null service flow at the CM as an Upstream Drop Classifier (UDC). An AM can push an Upstream Drop Traffic Profile to the CMTS, via the PS, which pushes the UDC to the CM. When a CM installs a UDC all data packets meeting the criteria of the classifier are dropped at the CM.

6.1.7 Event Generation Info

This object contains information relevant to the CMTS in support of accounting and billing functions. Attributes include:

- Primary Address: Port of the Primary Record Keeping Server to which the CMTS MUST send event records.
- Secondary Address: Port of the Secondary Record Keeping Server, the CMTS MUST use as specified in [15] if the primary is unavailable.
- BillingCorrelationID, which the CMTS MUST pass to the Record Keeping Server with each event record.

Omission of the Event Generation Info object indicates that the CMTS MUST NOT generate Event Messages for a specific Gate.

There are two versions of the Event Generation Info object: an IPv4 Event Generation Info object and an IPv6 Event Generation Info object to accommodate IPv4 and IPv6 Address values respectively.

6.1.8 Time-Based Usage Limit

This object specifies amount of time a Gate can remain committed before meeting the time limit threshold for this Gate. The CMTS is not responsible for enforcing time limits, but MUST store this object and return it upon request.

6.1.9 Volume-Based Usage Limit

The Application Manager uses the Volume-Based Usage Limit to signal the CMTS to generate a Gate Control message when the specified amount of data has traversed the Gate. The CMTS is not responsible for enforcing volume limits, but MUST signal to the PS/AM when a volume limit is reached.

6.1.10 Opaque Data

Opaque Data consists of general information that a Policy Server or Application Manager can store on a CMTS. This data remains opaque to the CMTS, but contains information useful for the PS or AM. If provided by the PS/AM, the CMTS will return this object in all responses (see Section 6.4.2.11).

6.1.11 Gate Time Info

Gate Time Info contains a timestamp representing the time the Gate was committed. This may be queried and used by a Policy Server or Application Manager to enforce time-based network policy.

6.1.12 Gate Usage Info

Gate Usage Info consists of an octet counter indicating the number of bytes of data transmitted over this Gate (see Section 6.4.2.13). Analogous to the Gate Time Info object, this information may be used by a Policy Server or Application Manager to enforce volume-based network policy.

6.1.13 User Identification (UserID)

The UserID is a UTF-8 string value that identifies the user requesting the service. As defined earlier the SubscriberID defines the actual device requesting the service, but there may be multiple users that share that device. The UserID field allows the elements that manage the QoS to associate a request with a specific user.

6.1.14 Shared Resource Identification (SharedResourceID)

The SharedResourceID is only applicable to Multicast Gates and is locally generated by the CMTS. When a Multicast Gate request is successfully processed by the CMTS, the CMTS MUST return a SharedResourceID in the Gate-Set-Ack to the Policy Server.

The CMTS MUST NOT include a SharedResourceID in Gate-Set-Ack messages in response to a Unicast Gate request.

The presence of the SharedResourceID indicates to the PS and AM that resources used in fulfilling the Multicast Gate request are either already shared with a previous Multicast Gate request, or may be shared by future Multicast Gate requests. The SharedResourceID essentially identifies the Multicast Group Service Flow on the DOCSIS network.

The method of allocation of SharedResourceIDs is an implementation option of the CMTS. However, the CMTS MUST ensure the uniqueness of all current SharedResourceID values within the scope of the CMTS. The CMTS SHOULD attempt to minimize the possibility of SharedResourceID ambiguities by ensuring that no SharedResourceID is reused for a different underlying resource within a reasonable duration after the prior closure or deletion of the last gate associated with the SharedResourceID in question.

While each PS may communicate with many CMTSs, a SharedResourceID assigned by one CMTS cannot be guaranteed to be unique across the network. The PS may use additional information, such as the associated CMTS identity, in order to uniquely identify the shared resource.

Note: The SharedResourceID does not replace the GateID, but instead supplements it by providing information about the underlying resource being utilized for a Multicast Gate. The GateID remains the primary and unique identifier of the Gate.

6.2 Gate Transitions

As briefly outlined earlier, a Gate may reside in the following logical states:

- Authorized – a Policy Server has authorized the flow with resource limits defined
- Reserved – resources have been reserved for the flow
- Committed – resources are active and are being used
- Committed-Recovery – inactivity has been detected on the flow; resource recovery pending

For the state machine depicted in Figure 3, the CMTS MUST complete the triggering event with a successful outcome before it transitions a gate from one state to another. For Gate Control events, the CMTS MUST not change state until the request has been fully processed (including any resulting flow transitions) and the CMTS has determined that a success acknowledgement is to be transmitted.

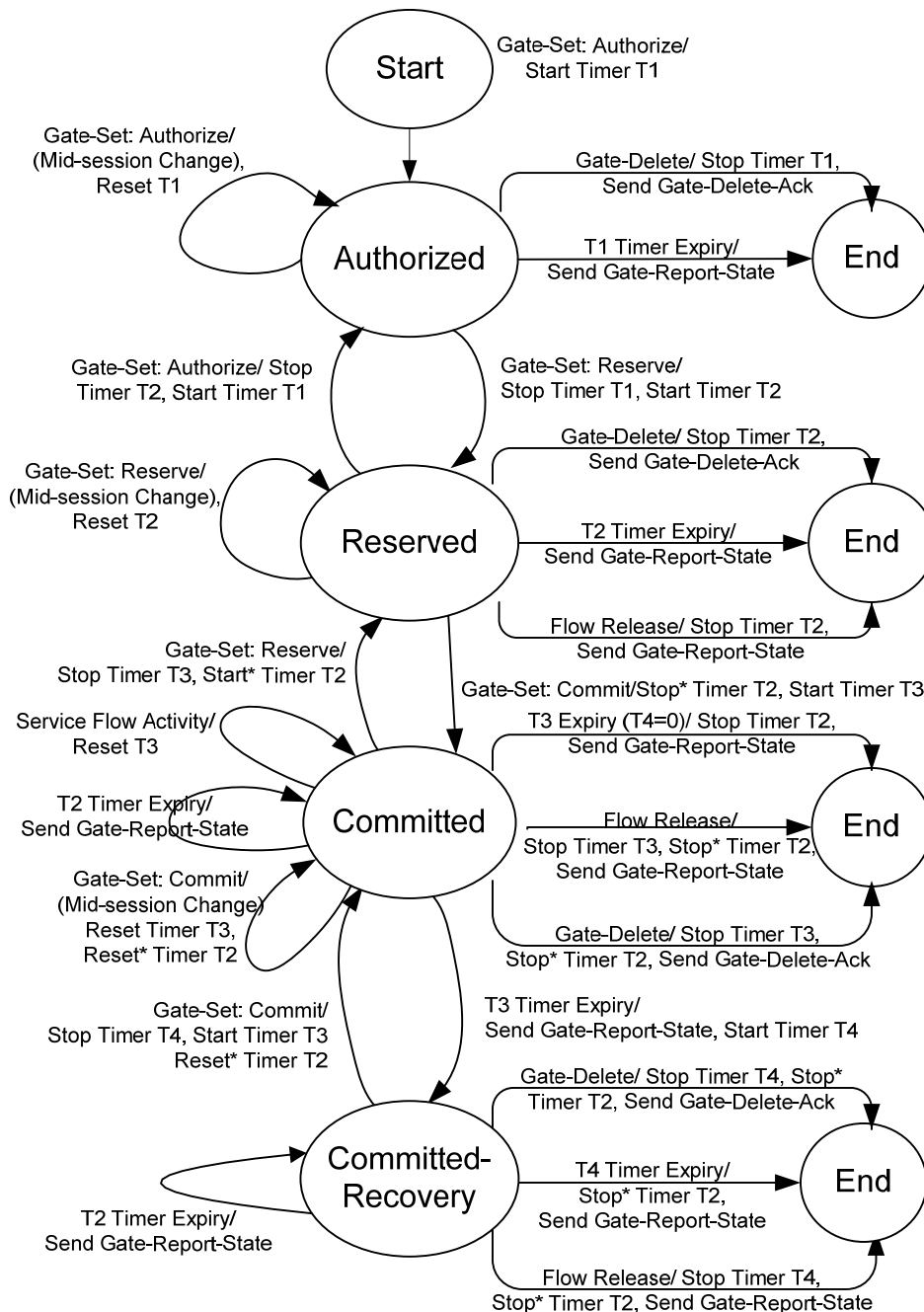


Figure 3 - Gate State Transitions

The CMTS MUST support Gate states and transitions as shown in Figure 3 and described in this section. The CMTS MUST also implement transitions for protocol error processing.

In this section we describe the state transitions of the Gate in the CMTS that result from external events (Gate Control messages from the Policy Server), as well as transitions that result from internal events (e.g., timer expiration). Note that the Policy Server is not the source of the external events; instead the Policy Server is merely acting as a proxy for the Application Manager, which is the trigger for the events.

6.2.1 Authorized

A Gate is created in the CMTS by a Gate-Set command from the Policy Server. The CMTS allocates a unique identifier called a GateID. The Gate is now said to be in the Authorized state and the CMTS MUST start Timer T1. Timer T1 limits the amount of time the authorization remains valid.

A Gate in the Authorized state MUST be deleted upon receipt of a Gate-Delete message. When this happens, the CMTS MUST respond with a Gate-Delete-Ack message and MUST stop Timer T1.

The CMTS is required to support the following state transitions while a Gate is in the Authorized state:

Authorized State Transitions:

- Authorized Loopback to Authorized: Modify Authorized Envelope
- Authorized to Reserved (defines Reserved Envelope $\leq$ Authorized Envelope)
- Authorized to End (deletes Authorized Envelope)

The CMTS MUST NOT support any other state transitions for a Gate in the Authorized state, but a number of separate stimuli may result in the transitions described.

When the Gate is installed, it is said to be in an Authorized State. While in the Authorized state, the Policy Server MAY modify any of the parameters associated with a Gate (e.g., Traffic Profile, Classifier, etc). If in a Authorized state a Gate-Set message is received that does not transition the Gate to the Reserved or Committed states, then the CMTS MUST restart timer T1.

While in the Authorized state, the CMTS MUST transition the Gate into the Reserved state upon successful request of the Policy Server. The CMTS MUST transition the Gate into the End state upon receipt of a Gate-Delete from the Policy Server or upon T1 timer expiration.

6.2.2 Reserved

A Gate in the Authorized state is expecting the client to attempt to reserve resources. In Scenario 1, the Policy Server reserves the resources on the client's behalf. To reserve resources, the Policy Server MUST issue a subsequent Gate-Set message with a Traffic Profile that includes the Reserved Envelope. On receipt of this reserve request, the CMTS MUST verify the request is within the authorization limits established for the Gate and perform admission control procedures.

If the Reserve request does not arrive before Timer T1 expires, the CMTS MUST delete the Gate, and notify the Policy Server of the state change. If admission control succeeds and only resource reservation was requested, the CMTS MUST put the Gate in the Reserved state. Simultaneously, the CMTS MUST also stop timer T1 and start timer T2 (Reserved Timer). If admission control procedures are not successful, the CMTS MUST maintain the Gate in the Authorized state and provide a Gate-Set-Err response to the PS.

The CMTS is required to support the following state transitions while a Gate is in the Reserved state:

Reserved State Transitions:

- Reserved Loopback to Reserved: Modify Authorized Envelope ($\geq$ Reserved Envelope)
- Reserved Loopback to Reserved: Modify Reserved Envelope ($\leq$ Authorized Envelope)
- Reserved to Authorized (deletes Reserved Envelope and replaces Authorized Envelope)
- Reserved to Committed (defines Committed Envelope $\leq$ Reserved Envelope)
- Reserved to End (deletes Reserved and Authorized Envelopes)

The CMTS MUST NOT support any other state transitions for a Gate in the Reserved state, but a number of separate stimuli may result in the transitions described.

From the Authorized state, the CMTS MUST transition the Gate into the Reserved state, if requested by the Policy Server as long as the Reserved envelope is less than or equal to the Authorized envelope, the request passes admission control, and the flow is successfully reserved. Once in the Reserved state, the Gate's Authorized envelope

MAY be modified via a Gate-Set message. The Gate's Reserved envelope can also be modified in the Reserved State (see Section 6.5.6). If in a Reserved state a Gate-Set message is received that does not transition the Gate to the Authorized or Committed states, then the CMTS MUST restart timer T2.

If the Commit request does not arrive before timer T2 expires, the CMTS MUST delete the Gate, and notify the Policy Server of the state change.

The Reserved envelope MUST always be less than or equal to the Authorized envelope. In the Reserved state, for a CMTS to transition a Gate to the Committed state, the Committed Envelope MUST be less than or equal to the Reserved Envelope (see Section 6.5.3).

While in the reserved state, the Policy Server may modify the Authorized envelope by specifying a new Traffic Profile in a Gate-Set message. The new Traffic Profile will define a modified Authorized envelope, and the same Reserved Envelope that was used previously to transition the Gate into the Reserved state. However, all requests to modify Authorized, Reserved, or Committed envelopes MUST conform to the general rule:

$$\text{Authorized Envelope} \geq \text{Reserved Envelope} \geq \text{Committed Envelope}$$

The Policy Server MAY delete a Gate in the Reserved state by issuing a Gate-Delete message.

6.2.3 Committed

In the Reserved state the Gate is expecting the client to commit resources, and thereby activate them. In Scenario 1, the Policy Server commits the resources on the client's behalf. To commit resources, the Policy Server MUST issue a Gate-Set command with a Traffic Profile that includes the Committed Envelope. The CMTS MUST authorize the requested Committed envelope against the Reserved envelope, assuming the Reserved and Authorized envelopes remain unchanged. If the authorization succeeds, the CMTS MUST start timer T3, and either stop timer T2 if the Reserved envelope equals the Committed envelope or restart timer T2 if the Reserved Envelope is greater than the Committed envelope. If the authorization fails, the CMTS MUST leave the state of T2 timer unmodified.

Note that once the DOCSIS Service Flow has been activated, the CMTS MUST refresh timer T3 when data is transferred over the flow. If there is no activity on the flow for time equal to T3, the CMTS MUST notify the Policy Server of the state change. Likewise, the Policy Server MUST notify the Application Manager of the state change.

In the Committed state, the Application Manager MAY delete the Gate by issuing a Gate-Delete message to the Policy Server, which in turn MUST relay the message onto the CMTS. If the Policy Server issues a Gate-Delete message to the CMTS, the CMTS MUST delete the Gate and the corresponding Service Flow, and stop timers T2 and T3 if they are running.

The CMTS is required to support the following state transitions while a Gate is in the Committed state:

Committed State Transitions:

- Committed Loopback to Committed: Modify Authorized Envelope ($\geq$ Reserved Envelope)
- Committed Loopback to Committed: Modify Reserved Envelop ($\geq$ Committed Envelope and $\leq$ Authorized Envelope)
- Committed Loopback to Committed: Modify Committed Envelop ($\leq$ Reserved Envelope)
- Committed to Reserved (deletes Committed Envelope)
- Committed to Committed-Recovery: (initiate resource recovery process)
- Committed to End (deletes Committed, Reserved and Authorized Envelopes)

The CMTS MUST NOT support any other state transitions for a Gate in the Committed state, but a number of separate stimuli may result in the transitions described.

While in the Reserved State, the CMTS MUST transition a Gate to the Committed state, if requested by the Policy Server as long as the Committed Envelope is less than or equal to the Reserved Envelope (see Section 6.5.3). While in the Committed state, the Policy Server MAY modify the Authorized Envelope of the Gate via a Gate-Set message, as long as the Authorized Envelope is greater than or equal to the Reserved Envelope. In this state, the Policy Server MAY also modify the Reserved envelope, as long as the reserved envelope is greater than or equal to

the Committed Envelope. In this state, the Policy Server MAY even modify the Committed envelope, as long as the new envelope is less than or equal to the Reserved Envelope. If a request is received to de-commit all resources (but keep them reserved) while the Gate is in the Committed state, the CMTS MUST stop timer T3, (re-)start timer T2, and transition back to the Reserved state. In Scenario 1, the Policy Server MAY request this action by issuing a Gate-Set message with the a Traffic Profile that includes the Authorized and Reserved Envelopes, but does not include a Committed Envelope.

While in the Committed state, the CMTS MUST transition a Gate to the End state upon receiving a Gate-Delete message from the Policy Server. While in the Committed state, the Policy Server MAY modify the Authorized or Reserved envelope by simply specifying the new Traffic Profile; the new Traffic Profile MUST contain modified Authorized or Reserved envelopes, and the same Committed Envelope that was used previously to transition the Gate into the Committed state.

When (re-)entering the Committed state due to the receipt of a Gate-Set message, the CMTS MUST (re-)start timer T2 if the Reserved Envelope is greater than the Committed Envelope, or MUST stop timer T2 if it was previously running and the Reserved Envelope is now equal to the Committed Envelope.

While in the Committed state: 1) If the timer T2 expires the Gate MUST remain in the Committed state, the T3 timer MUST be reset, and the CMTS MUST send a Gate-Report-State with Reason code 9 (Gate state unchanged, but T2 timer expiration caused reservation reduction) to the Policy Server indicating the reduction in reserved resources. 2) If a Gate-Set transitions the Gate back to the Reserved state the timer T2 MUST be started if it was not running and restarted if had previously been running, and 3) If a transition to the End state occurs both the timer T2 and T3 MUST be stopped if they are running.

As an optimization, for Scenario 1, the Policy Server MAY Authorize, Reserve, and Commit at the same time by issuing a Gate-Set message with the Traffic Profile that includes all three envelopes set such that the CMTS is told to execute all three actions sequentially without any further interaction from the Policy Server – that is, they must all succeed (if so, the CMTS MUST indicate this by a Gate-Set-Ack) or fail (if so, the CMTS MUST indicate this by a Gate-Set-Err).

From the Committed state, the Gate transition to the Committed-Recovery state due to the expiration of the T3 timer. If the CMTS detects that there has been no activity on the associated flow for a duration of T3, the CMTS MUST start the T4 timer, generate a Gate-Report-State message to the Policy Server indicating that the flow has been inactive for time duration defined by T3 and transition to the Committed-Recovery state, while leaving the associated flow in the activated state. The Policy Server MUST relay the Gate-Report-State message to the Application Manager. The Application Manager MUST either refresh the policy by issuing a Gate-Set message, or remove the Gate by issuing a Gate-Delete message.

Certain applications may not wish to be notified when activity on the flow ceases. In this case, the Application Manager may set the T3 timer (which corresponds to the DOCSIS Active Timer) to 0. As specified in [1] a value of 0 for the DOCSIS Active Timer indicates that activity detection on the CMTS is turned off for that flow. Thus, the Gate will remain in the Committed state forever until either a Gate-Delete is received or the CM goes offline.

6.2.4 Committed-Recovery

In the Committed state the flow associated with the Gate is active. If the CMTS detects that the flow is unused for a time in excess of the T3 timer, the CMTS notifies the PS (who notifies the AM) that the service-flow associated with the gate has been unused, starts the T4 timer and transitions the Gate to the Committed-Recovery state. (Note: If the T2 timer is running it will continue running. It MUST NOT be reset). The AM must decide to either refresh the policy by issuing a Gate-Set message to the PS or delete the Gate by issuing a Gate-Delete message to the PS. The Policy Server MUST forward the Gate-Set message or Gate-Delete message to the CMTS.

If, while in the Committed-Recovery state, the CMTS receives a Gate-Set message for the Gate before the timer T4 expires, then the CMTS MUST stop the T4 timer, restart the T3 timer, transition the Gate back to the Committed state and (re-)start the T2 timer if the Reserved Envelope is greater than the Committed Envelope or stop the T2 timer if it is running and the new Reserved Envelope is equal to the Committed Envelope. If the authorization fails, the CMTS MUST leave the state of the T4 and T2 timers unmodified.

If, while in the Committed-Recovery state, the CMTS receives a Gate-Delete message before the timer T4 expires, then the CMTS MUST stop the T4 Timer, delete the Gate and the corresponding service flow, and stop the T2 timer if it is running.

If the timer T4 expires while in the Committed-Recovery state, then the CMTS MUST send a Gate-Report-State message to the PS, stop the T2 timer if it is running, tear down the service flow associated with the Gate, and then delete the Gate. Likewise, the Policy Server MUST notify the Application Manager of the state change.

If the timer T2 expires while in the Committed-Recovery state, then the Gate MUST remain in the Committed-Recovery state and the CMTS MUST send a Gate-Report-State with Reason code 9 (Gate state unchanged, but T2 timer expiration caused reservation reduction) to the Policy Server indicating the reduction in reserved resources.

The CMTS is required to support the following state transitions while a Gate is in the Committed-Recovery state:

Committed-Recovery State Transitions:

- Committed-Recovery to Committed: (policy has been refreshed)
- Committed-Recovery to End (deletes Committed, Reserved and Authorized Envelopes)
- Committed-Recovery to Committed-Recovery (Timer T2 Expiry; send Gate-Report-State)

The CMTS MUST NOT support any other state transitions for a Gate in the Committed-Recovery state, but a number of separate stimuli may result in the transitions described.

There may be applications that do not wish to keep a Gate once inactivity has been detected. In this instance, the AM may set the T4 timer to zero. When T4 is set to zero, the Committed-Recovery state is bypassed. In this case, when the T3 timer expires the CMTS MUST send a Gate-Report-State message to the PS (with reason code = 5), stop the T2 timer if it is running, tear down the service flow associated with the Gate, and then delete the Gate. Likewise, the Policy Server MUST notify the Application Manager of the state change.

6.3 COPS Profile for IPCablecom Multimedia

As defined earlier, admission control involves the process of managing QoS resource requests based on administrative policies and available resources. High-level operational modules associated with this process are depicted in [16]. Under this model administrative policies are stored in a policy database and controlled by the Policy Server.

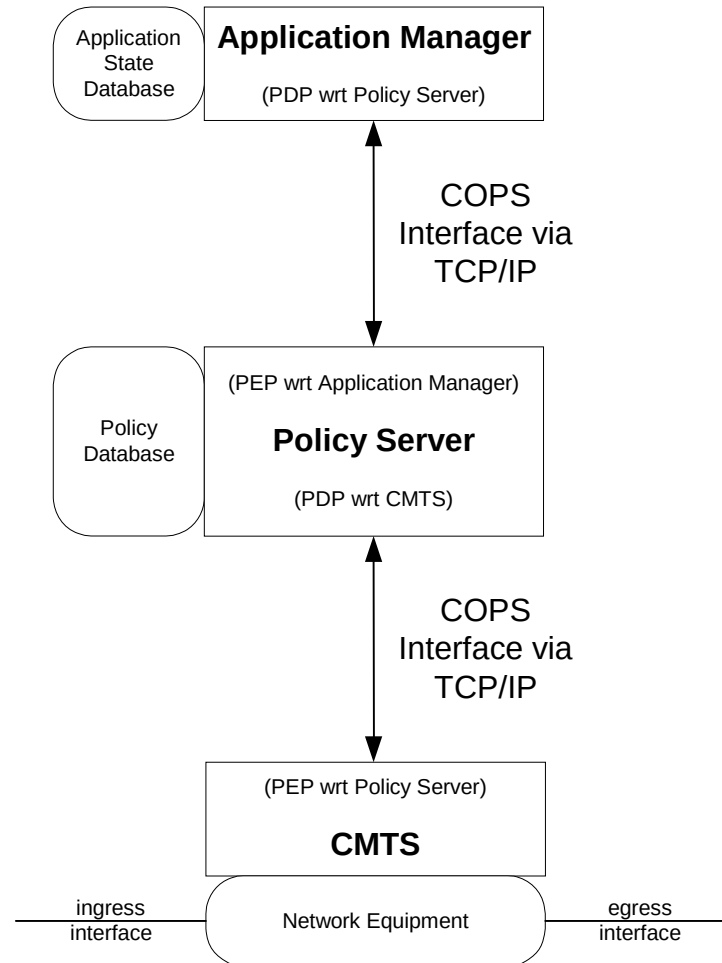


Figure 4 - QoS Admission Control Layout

Admission control decisions made by the Policy Server **MUST** be communicated to the CMTS or Application Manager using COPS. The CMTS **MAY** make QoS Admission Control requests to the COPS Server based on network events triggered by either the QoS signaling protocol, or via data flow detection mechanisms. The network event can also be in need of QoS bandwidth management, e.g., a new QoS capable interface becomes operational.

QoS policy decisions made by the Policy Server **MAY** be pushed to the CMTS based on a request from the Application Manager. The CMTS **MAY** access that decision information to make policy enforcement decisions on incoming session requests received at the CMTS. The CMTS **MUST NOT** support CM-initiated DSx messages in IPCablecom Multimedia. The CMTS **MUST** treat a CM-initiated DSx message as a request with an invalid GateID.

A COPS client/server configuration supporting QoS Admission Control is specified in the IETF's COPS protocol [10]. This protocol includes the following operations:

- Client-Open (OPN)/Client-Accept (CAT)/Client-Close (CC). The COPS client (PEP) sends an OPN message to initiate a connection with the COPS server (PDP), and the server responds with a CAT message to accept the connection. The server or client sends a CC message to terminate the connection.
- Request (REQ). The client sends a REQ message to the server to request admission control decision information or device configuration information. The REQ message may contain client-specific information that the server uses, together with data in the session admission policy database, to make policy-based decisions.
- Decision (DEC). The server responds to REQs by sending a DEC back to the client that initiated the original request. DEC messages may be sent immediately in response to a REQ (i.e., a solicited DEC) or at any time after to change or update a previous decision (i.e., an unsolicited DEC).
- Report-State (RPT). The client sends a RPT message to the server indicating changes to the request state in the client. The client sends this to inform the server of the actual resource reserved after the server has granted admission. The client can also use Report-State to periodically inform the server the current state of the client.
- Delete-Request-State (DRQ). The client sends a DEL message to the server to request state cleanup. This may be the result of QoS resource release by the client.
- Keep-Alive (KA). Sent by both the client and server for communication failure detection.

Within the IPCablecom Multimedia architecture, the PDP-PEP relations are as follows:

- The Application Manager is a COPS Policy Decision Point (PDP) relative to the Policy Server.
- The Policy Server is a PEP relative to the Application Manager.
- The Policy Server is a PDP relative to the CMTS.
- The CMTS is a PEP relative to the Policy Server.

Although the content of COPS messages required for IPCablecom Multimedia are consistent with the COPS protocol, there is a slight difference in the way the COPS session starts, and a relaxation of response ordering requirements. RFC2748 [10] states:

"The COPS protocol uses a single persistent TCP connection between the PEP and a remote PDP. One PDP implementation per server MUST listen on a well-known TCP port number (COPS=3288 [IANA]). The PEP is responsible for initiating the TCP connection to a PDP."

The last line of the statement says that the PEP is responsible for initiating the TCP connection. In contrast, in the IPCablecom model, the CMTS (PEP) is the one that listens on the assigned port 3918, and it is the Policy Server that MUST initiate the TCP connection to the CMTS. This is the opposite of the model described in the RFC. However, once the TCP connection is in place, the CMTS behaves in a manner consistent with the client, or PEP, in the COPS protocol. Similarly, the Policy Server (PEP) listens on the assigned port 3918 and it is the Application Manager that MUST initiate the TCP connection to the Policy Server.

Note that IPCablecom Multimedia and IPCablecom 1.x DQoS listen on different ports so that the CMTS may initiate the COPS session with the proper Client-Type.

RFC2748 [10] also states:

"RPT messages solicited by decisions for a given Client Handle MUST set the Solicited Message flag and MUST be sent in the same order as their corresponding Decision messages were received."

The COPS protocol uses the order of RPT and Decision messages to match requests to responses. In contrast, IPCablecom Multimedia implementations MUST use the TransactionID object to match responses with requests and SHOULD send RPT messages as soon as they are ready.

The details of the COPS protocol are provided in [10]. This IETF RFC provides a description of the base COPS protocol, independent of Client-Type. The IPCablecom architecture is also aligned with the IETF RFC 3084 [12] document. COPS-PR states:

"In COPS-PR, policy requests describe the PEP and its configurable parameters (rather than an operational event). If a change occurs in these basic parameters, an updated request is sent. Hence, requests are issued quite infrequently. Decisions are not necessarily mapped directly to requests, and are issued mostly when the PDP responds to external events or PDP events (policy updates)."

When this concept is mapped to the IPCablecom Multimedia architecture, the PEP issues a Request to the PDP, specifying a Client-Handle. This Client-Handle is then used in all future Decision messages from the PDP to the PEP. These Decision messages carry the Gate Control messages (i.e., Gate-Set, Gate-Info and Gate-Delete) defined for the DQoS and Multimedia Client-Types. The Client-Handle is used to uniquely identify the PDP-PEP association.

In the IPCablecom Multimedia architecture, there can be multiple Application Managers interacting with one or more Policy Servers. There is a single instance of an IPCablecom Multimedia COPS session per TCP connection; where an IPCablecom Multimedia COPS session refers to the Gate messages between the PDP and PEP associated with a single Client-Handle. This means there is one COPS-TCP connection between an Application Manager and a Policy Server. Similarly, there can be one or more Policy Servers talking to one or more CMTSSs. When connected to multiple PDPs, the PEP MUST ensure that the Client-Handle used is unique per association.

6.4 Gate Control Protocol Message Formats

Protocol messages for Gate Control MUST be transported within the COPS protocol messages. The PDP and PEP MUST establish and use a TCP connection for communication, and utilize the mechanisms specified in [17] to secure the communication path.

6.4.1 COPS Common Message Format

Each COPS message consists of the COPS header followed by a number of typed objects. The Application Manager, Policy Server and CMTS MUST use the COPS Common Message format as defined below as the message format for all message exchanges. In the object specifications that follow, each row represents a 4-byte word as all objects align on 4-byte word boundaries.

0		1	2	3
Version	Flags	Op-Code	Client-Type	
Message Length				

Version is a 4-bit field giving the current COPS version number. This field MUST be set to 1.

Flags is a 4-bit field. The least significant bit is the solicited message flag. When a COPS message is sent in response to another message (e.g., a solicited decision sent in response to a request) this flag MUST be set to 1. In other cases (e.g., an unsolicited decision) the flag MUST NOT be set (value = 0). In keeping with the DQoS model, the first Decision message sent in response to a Request message is a solicited response and its solicited message flag MUST be set. All other Decision messages are unsolicited and the solicited message flag MUST be cleared. All other flags MUST be set to zero.

Op-code is a 1-byte unsigned integer field that gives the COPS operation to be performed. COPS operations used in this IPCablecom specification are:

- 1 = Request (REQ)
- 2 = Decision(DEC)
- 3 = Report-State (RPT)
- 4 = Delete Request State (DRQ)
- 6 = Client-Open (OPN)
- 7 = Client-Accept (CAT)
- 8 = Client-Close (CC)
- 9 = Keep-Alive (KA)

Client-Type is a 2-byte unsigned integer identifier. For IPCablecom Multimedia use, the Client-Type MUST be set to IPCablecom Multimedia client (0x800A). For Keep-Alive messages (Op-code = 9) the Client-Type MUST be set to zero, as the KA is used for connection verification rather than per-client session verification.

Message Length is a 4-byte unsigned integer value giving the size of the overall message in octets. Messages MUST be aligned on 4-byte boundaries, so the length MUST be a multiple of four.

Following the COPS common header are one or more objects. All the objects MUST conform to the same object format where each object consists of one or more 4-byte words with a four-octet header, using the following format.

0	1	2	3
Length		C-Num	C-Type
Object Contents			

Length is a 2-byte unsigned integer value that MUST give the number of bytes (including the header) that compose the object. If the original length in octets is not a multiple of four, padding MUST be added to the end of the object so that it is aligned to the next 4-byte boundary.

C-Num identifies the class of information contained in the object, and the C-Type identifies the subtype or version of the information contained in the object. Standard COPS objects (as defined in [10]) used in this specification, and their C-Num values, are:

- 1 = Handle
- 2 = Context
- 6 = Decision
- 8 = Error
- 9 = Client Specific Info
- 10 = Keep-Alive-Timer
- 11 = PEP Identification
- 12 = Report Type

Each of these objects MUST conform to the format and rules relating to the individual object as defined in [10].

6.4.2 Additional COPS Objects for Gate Control

As with the COPS-PR and COPS-RSVP profiles, the IPCablecom Client-Type defines a number of additional object formats. These objects MUST be placed inside a Decision object, C-Num = 6, C-Type = 4 (Client Specific Decision Data) when carried from PDP to PEP in a Decision message. They MUST also be placed in a ClientSI object, C-Num = 9, C-Type = 1 (Signaled ClientSI) when carried from the PEP to the PDP in a Report State message or Client-Open message.

These objects are encoded similarly to the client-specific objects for COPS-PR and as in COPS-PR these objects are numbered using a client-specific number space, which is independent of the top-level COPS object number space. For this reason, the object numbers and types are given as S-Num and S-Type, respectively. S-Num and S-Type MUST be one octet. The COPS Length field MUST be two octets. Additional COPS objects are defined for use by IPCablecom Multimedia in the following sections.

6.4.2.1 TransactionID

TransactionID is a 2-byte unsigned integer quantity, which contains a token that is used by the Application Manager to match responses from the Policy Server and by the Policy Server to match responses from the CMTS to the previous requests. The TransactionID MUST also contain the command type that identifies the action to be taken or response. The TransactionID Object MUST conform to the following format.

Length = 8	S-Num = 1	S-Type = 1
Transaction Identifier	Gate Command Type	

The Transaction Identifier MUST be set to 0 when included in a Gate-Report-State message.

Gate Command Type is a 2-byte unsigned integer value which identifies the Gate Control message type and MUST be one of the following:

<Reserved>	1-3
Gate-Set	4
Gate-Set-Ack	5
Gate-Set-Err	6
Gate-Info	7
Gate-Info-Ack	8
Gate-Info-Err	9
Gate-Delete	10
Gate-Delete-Ack	11
Gate-Delete-Err	12
<Reserved>	13
<Reserved>	14
Gate-Report-State	15
Gate-Cmd-Err	16
PDP-Config	17
PDP-Config-Ack	18
PDP-Config-Err	19
Synch-Request	20
Synch-Report	21
Synch-Complete	22
Msg-Receipt	23

6.4.2.2 AMID

AMID, the Application Manager ID, consists of two fields, the Application Manager Tag and Application Type. The Application Manager Tag is a 2-byte unsigned integer value which identifies the Application Manager responsible for handling the session. The Application Type is a 2-byte unsigned integer value which identifies the type of application that this gate is associated with. The Application Manager MUST include this object in all messages it issues to the Policy Server. The Policy Server MUST include the received AMID in all messages it issues down to the CMTS in response to the messages it receives from the Application Manager. The CMTS MUST include the received AMID object in all messages it issues to the Policy Server, and likewise, the Policy Server MUST include the received AMID to the Application Manager. The Policy Server may use the AMID in messages from the CMTS to resolve the Application Manager to which it may need to generate a message.

Application Type is used by an Application Manager to identify the application that a gate is associated with. There is only one reserved value, zero, which indicates that there is "no defined application association". An Application Manager MAY pass a non-zero value to the Policy Server to indicate the particular application that a gate is associated with. The Policy Server MAY use this value when applying policies to this gate. The CMTS MAY use this value when performing admission control for this gate. The Application Type values are configured by the service provider in the Application Managers and Policy Servers. In order for the CMTS to utilize the Application Type in its admission control logic, the service provider would need to configure the CMTS with the allowed Application Types and their associated meaning. These values are unique within the domain of the service provider,