INTERNATIONAL TELECOMMUNICATION UNION

ITU-T J.179

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(04/2004)

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

IPCablecom

IPCablecom support for multimedia

ITU-T Recommendation J.179

ITU-T Recommendation J.179

IPCablecom support for multimedia

Summary

This Recommendation supports the deployment of general multimedia services by providing a technical definition of several IP-based signalling interfaces that leverage core QoS and policy management capabilities native to CableModems.

Source

ITU-T Recommendation J.179 was approved on 22 April 2004 by ITU-T Study Group 9 (2001-2004) under the ITU-T Recommendation A.8 procedure.

FOREWORD

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

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

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

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

NOTE

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

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

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

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

CONTENTS

		Page
1	Scope	1
2	References	1
	2.1 Normative references	1
	2.2 Informative references	2
3	Terms and definitions	2
4	Abbreviations, acronyms and conventions	3
	4.1 Abbreviations and acronyms	3
	4.2 Conventions	4
5	Technical overview	4
	5.1 QoS background	5
	5.2 Architecture	8
6	Authorization interface description	16
	6.1 Gates: The framework for QoS control	16
	6.2 Gate transitions	22
	6.3 COPS profile for IPCablecom multimedia	26
	6.4 Gate Control protocol message formats	29
	6.5 Gate control protocol operation	49
7	Event messaging interface description	58
	7.1 Introduction	58
	7.2 Record Keeping Server requirements	60
	7.3 General IPCablecom multimedia network element requirements	60
	7.4 Event Messages for IPCablecom Multimedia	61
	7.5 Event messaging attributes for IPCablecom multimedia	67
	7.6 RADIUS accounting protocol	73
8	Security requirements	75
	8.1 CMTS – CM QoS Interface (pkt-mm-1)	76
	8.2 Policy server – CMTS COPS Interface (pkt-mm-2)	76
	8.3 Application Manager – Policy Server COPS Interface (pkt-mm-3)	76
	8.4 Policy Server – RKS Event Message Interface (pkt-mm-4)	76
	8.5 CMTS – RKS Event Message Interface (pkt-mm-5)	76
9	Mapping a FlowSpec Traffic Profile to DOCSIS	77
	9.1 Mapping FlowSpecs to DOCSIS scheduling types	77
	9.2 Mapping FlowSpecs to DOCSIS traffic parameters	78
	9.3 DOCSIS upstream parameters	80
	9.4 DOCSIS downstream parameters	83
10	Message flows	85
	10.1 Basic message sequence	85

Page

10.2 Detailed message sequence 87

11 Issues for future study..... 113

Appendix I – Background information 113

I.1 Introduction 113

I.2 IPCablecom Multimedia objectives and scope..... 114

I.3 IPCablecom multimedia framework..... 117

I.4 Proxied QoS with Policy Push (Scenario 1)..... 122

I.5 Client-requested QoS with policy-push (Scenario 2)..... 129

I.6 Client-requested QoS with policy-pull (Scenario 3) 137

I.7 Comparison of IPCablecom-T and IPCablecom Multimedia 139

Introduction

Since its inception, the IPCablecom initiative has been positioned as an IP-based multimedia service deployment infrastructure, exploiting and enhancing the underlying QoS capabilities of the CableModem access network. Based on market demand and technology readiness, voice telephony was chosen as the first IP-based service to leverage these unique broadband access network capabilities, as the IPCablecom-T suite of Recommendations simultaneously defines both a general QoS-based service delivery framework and a number of telephony-specific functional elements and mechanisms. Since the intent of this Recommendation 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 CableModem access-network QoS mechanisms complemented by secure and robust authorization and audit mechanisms.

ITU-T Recommendation J.179

IPCablecom support for multimedia

1 Scope

This Recommendation supports the deployment of general multimedia services by providing a technical definition of several IP-based signalling interfaces that leverage core QoS and policy management capabilities native to CableModems. Multimedia services are defined as IP-based services (e.g., online gaming, videoconferencing, 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 IPCablecom-T set of Recommendations provide coverage specific to this type of service delivery, and, therefore, those Recommendations should be consulted as appropriate.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

2.1 Normative references

[1] ITU-T Recommendation J.112 Annex B (2004), *Data-over-cable service interface specifications: Radio-frequency interface specification*.

[2] IETF RFC 1305 (1992), *Network Time Protocol (Version 3) Specification, Implementation and Analysis*.

[3] IETF RFC 2210 (1997), *The Use of RSVP with IETF Integrated Services*.

[4] IETF RFC 2211 (1997), *Specification of the Controlled-Load Network Element Service*.

[5] IETF RFC 2212 (1997), *Specification of Guaranteed Quality of Service*.

[6] IETF RFC 2474 (1998), *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*.

[7] IETF RFC 2748 (2000), *The COPS (Common Open Policy Service) Protocol*.

[8] IETF RFC 2866 (2000), *RADIUS Accounting*.

[9] ITU-T Recommendation J.163 (2004), *Dynamic quality of service for the provision of real-time services over cable television networks using cable modems*.

[10] ITU-T Recommendation J.164 (2001), *Event message requirements for the support of real-time services over cable television networks using cable modems*.

[11] ITU-T Recommendation J.170 (2002), *IPCablecom security specification*.

[12] ITU-T Recommendation J.125 (2004), *Link privacy for cable modem implementations*.

2.2 Informative references

[13] IETF RFC 1633 (1994), *Integrated Services in the Internet Architecture: An Overview*.

[14] IETF RFC 2205 (1997), *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*.

[15] IETF RFC 2216 (1997), *Network Element Service Specification Template*.

[16] IETF RFC 2475 (1998), *An Architecture for Differentiated Services*.

[17] IETF RFC 2670 (1999), *Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces*.

[18] IETF RFC 2753 (2000), *A Framework for Policy-based Admission Control*.

[19] IETF RFC 3084 (2001), *COPS Usage for Policy Provisioning (COPS-PR)*.

[20] IETF RFC 3175 (2001), *Aggregation of RSVP for IPv4 and IPv6 Reservations*.

[21] CableLabs (http://www.cablemodem.com/specifications/).

3 Terms and definitions

This Recommendation defines the following terms:

3.1 client type 1: Client Type 1 represents existing "legacy" endpoints (e.g., PC applications, gaming consoles) which lack specific QoS awareness or signalling capabilities. This Client knows nothing about CableModem, IPCable2Home, or IPCablecom messaging, and hence no related requirements can be placed upon it. Such clients may range from simple analog audio and video presentation devices to complex networked peripherals and consumer electronics, such as set-top boxes or gaming consoles. This Client communicates with an Application Manager to request service, and does not request QoS resources directly from the operator access network. This Recommendation supports only client type 1.

3.2 client type 2: Client Type 2 is similar to an IPCablecom-T telephony MTA in that it supports QoS signalling based on IPCablecom DQoS. This Client is aware of IPCablecom Multimedia QoS, and communicates with an Application Manager to request service and obtain a token for access-network resources. The client then presents this token when requesting QoS resources from the access network (pkt-mm-1, pkt-mm-6). This Recommendation support for client type 2 remains as for further study.

3.3 client type 3: Client Type 3 requests QoS based on RSVP without Application Manager interaction. This Client is aware of IETF standards-based RSVP and uses this protocol to request QoS resources from the access network directly from the CMTS. This Recommendation support for client type 3 remains as for further study.

3.4 DOCSIS: Describes a specific CableModem technology as developed by Cable Television Laboratories, Inc. ("CableLabs") located at: http://www.cablemodem.com/specifications/. The international version is defined in J.112 Annex B.

3.5 IPCablecom-T: The suite of IPCablecom ITU-T Recommendations that support telephone service.

4 Abbreviations, acronyms and conventions

4.1 Abbreviations and acronyms

This Recommendation 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 Recommendation.
CM	Cable Modem
CMS	Call Management Server
CMTS	Cable Modem Termination System
COPS	Common Open Policy Service. Defined in RFC 2748.
DQoS	Dynamic Quality-of-Service
DSx (Messaging)	J.112 Annex B QoS signalling mechanism providing Dynamic Service Add, Change and Delete semantics.
FQDN	Fully Qualified Domain Name
HFC	Hybrid Fibre/Coax
IETF	Internet Engineering Task Force
IP	Internet Protocol
KDC	Key Distribution Centre
MG	Media Gateway
MGC	Media Gateway Controller
MTA	Multimedia Terminal Adapter
NAT	Network Address Translation
PDP	Policy Decision Point. Defined in RFC 2753.
PEP	Policy Enforcement Point. Defined in RFC 2753.
PS	Policy Server
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service. Defined in RFC 2138 and RFC 2139.
RAP	Resource Allocation Protocol Working Group in the IETF. Responsible for the definition and maintenance of the COPS protocol.
RCD	Resource Control Domain
RFC	Request for Comments. Technical policy documents approved by the IETF which are available at http://www.ietf.org/rfc.html.
RFI	Radio Frequency Interface specification, defining MAC and Physical layer interfaces between CMTS and CM network elements.
RKS	Record Keeping Server

RSVP	Resource ReSerVation Protocol. Defined in RFC 2205.
RSVP+	IPCablecom profile and extension of RSVP, defined in the IPCablecom DQoS Recommendation.
SCD	Service Control Domain
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. J.112 Annex B QoS scheduling type used for constant bit rate services (e.g., voice codecs).
UGS/AD	Unsolicited Grant Service with Activity Detection
VoIP	Voice over IP
VPN	Virtual Private Network

4.2 Conventions

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

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

5 Technical overview

This clause consists of background material which some readers may find to be useful context for the detailed protocol interface Recommendations that follow. The intent in this clause is to provide a high-level overview of the IPCablecom Multimedia architecture and the fundamental technologies upon which it is based. For more details on Multimedia architecture, see Appendix I.

5.1 QoS background

As noted throughout this Recommendation, 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 J.112 Annex B and IPCablecom-T. This clause provides a brief overview of these capabilities as preparatory background for the detailed QoS policy and resource management discussion that follows.

5.1.1 J.112 Annex B QoS summary

The J.112 Annex B RFI Recommendation [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. Two 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.

While J.112 Annex B 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 J.112 Annex B 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 unidirectional 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 (<500 ms). 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 following clause.

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 proceed through three logical states, summarized below. While certain optimized signalling 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 Recommendation 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.

5.1.2 IPCablecom-T QoS summary

While the J.112 Annex B RFI Recommendation defines the fundamental QoS mechanisms that form the core of the IPCablecom DQoS model, the IPCablecom DQoS Recommendation [9] augments these capabilities with a COPS-based policy management framework. Just as the Service Flow represents the primary abstraction in the J.112 Annex B 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 J.112 Annex B authorization mechanisms described above, only DSx requests which conform to the following general relation on a parameter-by-parameter basis will be granted:

Authorized Envelope ≥ Reserved Envelope ≥ Committed Envelope

Based on this policy management model, IPCablecom-T 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 [19] than with the standard COPS scheme specified in RFC 2748 [7]. Also, in order to install and manage these Gates, the IPCablecom DQoS Recommendation defines a set of COPS client-specific objects which constitute the primitives of a Gate Control signalling 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-T Gate Controller) and the Application Manager (defining service-specific functionality similar to the Call Agent in the IPCablecom-T 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-T 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 dialled-digits;
- E-MTA$_t$ boots, provisions and registers with CMS;
- CMS issues request to E-MTA$_t$ for notification of off-hook event and dialled-digits;
- E-MTA$_o$ goes off-hook, notifies CMS and delivers dialled-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 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 J.112 Annex B and IPCablecom QoS fundamentals. For further details related to either of these considerably complex topics, please consult the respective primary sources [1] and [9]. Clause 5.2 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 Recommendation that follows.

5.2 Architecture

Appendix I describes an architecture framework and reference model for IPCablecom Multimedia. This Recommendation applies the model contained in the architectural framework and adds normative requirements to provide a scaleable, interoperable solution suitable for deploying IPCablecom Multimedia services.

5.2.1 Client types

The IPCablecom Multimedia technical 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 signalling capabilities. This client has no awareness of CableModem, IPCable2Home, 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 cable operator access network.
- Client Type 2 is similar to an IPCablecom-T telephony MTA in that it supports QoS signalling based on the IPCablecom DQoS Recommendation.
- 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 Recommendation support is limited to Client Type 1. Consequently, this release of this Recommendation supports only Scenario 1, the "Proxied QoS with Policy Push" scenario, described in Appendix I. 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 cable operator's network. In the process of describing these network elements throughout this Recommendation, 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 Appendix I.

Since COPS [7] and COPS-PR [19] 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 Recommendation 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.

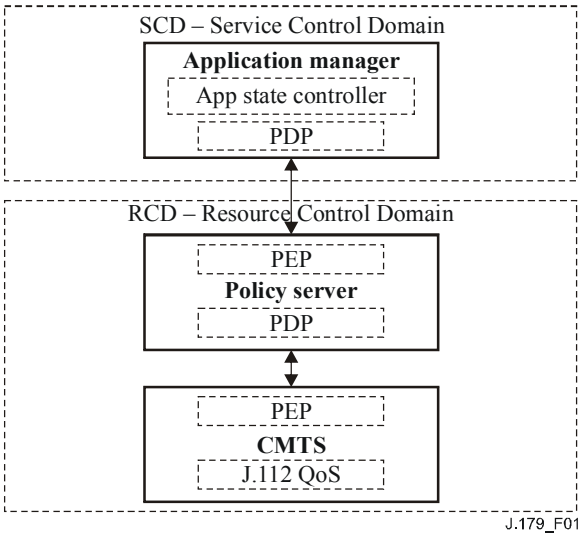

Figure 1/J.179 – Service and resource control domains

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 Service Control Domain (SCD) policies against client-driven session requests. 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 IPCablecom Multimedia components are:

- 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 Recommendation, 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 a cable operator's network management strategy. Policy authorization is almost always defined and enforced at the Policy Server.

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

5.2.2.1 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 cable operator's network or it may reside outside this domain and interact with the cable operator 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).

In most anticipated service deployment scenarios, the Application Manager will communicate with a client via a signalling protocol that is beyond the scope of this Recommendation. Using this unspecified protocol, the AM authenticates and authorizes client requests based on Service Control Domain policies. For client 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.

5.2.2.2 Policy Server (PS)

As discussed in RFC 2753 [18], the policy management framework underlying IPCablecom 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 [18] and in keeping with the IPCablecom-T DQoS model, the Policy Server serves as Policy Decision Point (PDP) in relation to the CMTS in that the Policy Server implements cable operator-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 cable operator. 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 cable operators 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.

5.2.2.2.1 Stateful and 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 behaviour 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 Recommendation. 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 Cable Modem Termination System (CMTS)

In describing the role of the CMTS network element, it is important to consider the relation among CableModem, IPCablecom-T and IPCablecom Multimedia functionality. While each of these suites of Recommendations 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-T or IPCablecom Multimedia Gate Control may be layered on top of a J.112 Annex B 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-T CMTS that has been defined in order to deliver telephony services in IPCablecom-T 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 [9]; 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.

5.2.2.4 Record Keeping Server (RKS)

The IPCablecom Multimedia Record Keeping Server fulfils a similar role to the RKS in IPCablecom-T [10]. 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-T suite of Recommendations. Where an IPCablecom Multimedia interface has a corollary in IPCablecom-T, IPCablecom Multimedia uses the same protocol, or an extension of the same protocol.

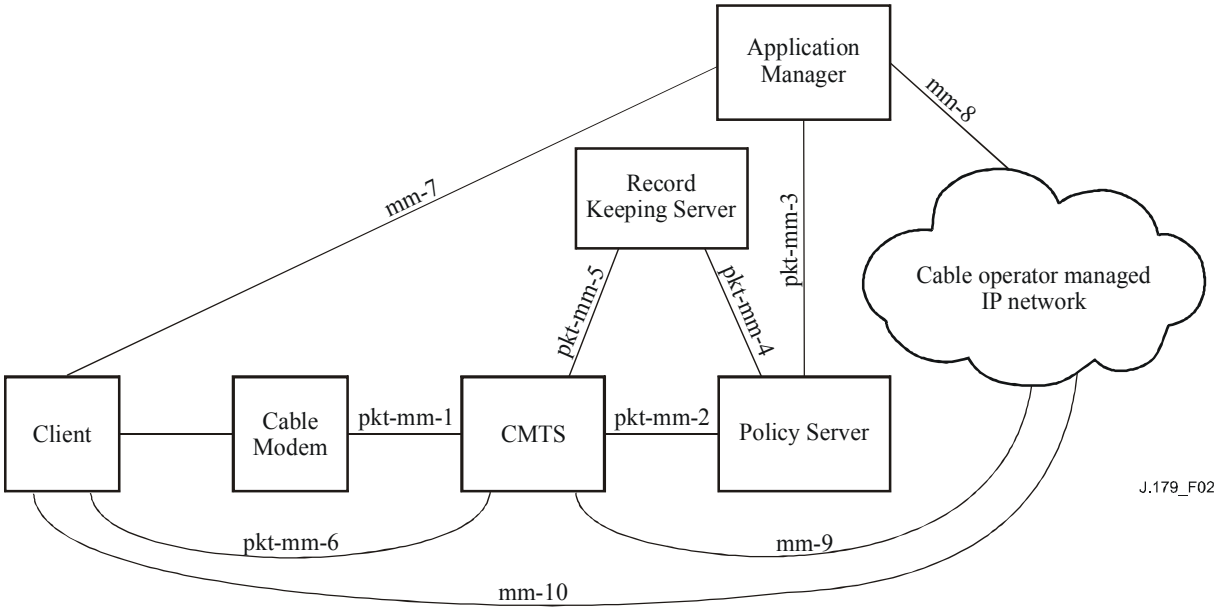

Figure 2/J.179 – IPCablecom multimedia architectural framework

Table 1/J.179 – IPCablecom multimedia interfaces

Interface	Description	Notes
pkt-mm-1	CMTS – CM	The Cable Modem (CM) may request QoS from the CMTS via J.112 Annex B DSx signalling. Alternatively, the CMTS may instruct the CM to set up, tear down or change a DOCSIS service flow in order to satisfy a QoS request, again via DSx signalling.
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 Recommendation.

Table 1/J.179 – IPCablecom multimedia interfaces

Interface	Description	Notes
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 Recommendation.
mm-9	CMTS – cable operator-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 Recommendation.
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 Recommendation.

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 Recommendation, 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-T interface between a Call Agent and a Gate Controller. In IPCablecom-T 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 [7] 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-T. As in IPCablecom-T, 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-T 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-T, the GateID acts as an authorization token. Unlike IPCablecom-T, 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-T (see [10]). 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.

5.2.4 State information

In this clause, 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 Recommendation, 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 clause 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 clause 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 J.112 Annex B 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 bidirectional 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-T, 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.

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 J.112 Annex B Service Flow directly, after the CMTS performs the necessary admission control checks of the envelopes. In the other two models outlined in Appendix I, '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 clause:

- GateID;
- AMID;
- SubscriberID;

- GateSpec;
- Classifier;
- Traffic Profile;
- Event Generation Info (optional);
- Time-Based Usage Limit (optional);
- Volume-Based Usage Limit (optional);
- Opaque Data (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.

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 6.4.2.11).

These elements are communicated to the Policy Server and the CMTS via COPS objects and are described in greater detail later in this clause. 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. There is a strict one-to-one mapping between a DOCSIS Service Flow and a Gate.

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 IPCablecom-T 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 clause in this Recommendation describes this behaviour. 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 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 IPCablecom-T 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. This identifier MUST be unique within the context of the IPCablecom Multimedia COPS session. 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 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)

Each Application Manager is pre-provisioned with an AMID that is unique within the universe of a single service provider; the Application Manager includes this identifier 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.

The AMID MUST be a globally unique value assigned to the Application Manager by the service provider. The Application Manager MUST use the assigned AMID 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 AMIDs.

6.1.3 Subscriber Identification (SubscriberID)

The SubscriberID, consisting of the IP address of either the client CPE device or 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 IP address, a subscriber may also be identified via a FQDN or some opaque data (object defined below) relevant to the service in question.

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:

- GateID;
- SessionClassID;
- Direction;

- Authorized Timer;
- Reserved Timer;
- Committed Time;
- DSCP/TOS Override.

GateID uniquely identifies the Gate for which the operation should be performed.

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 favour 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.

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

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

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

DSCP/TOS Override field can be used to override 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 overwritten by the CMTS. This field MAY be used in both the upstream and downstream directions.

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. The Classifier identifies the IP flow that will be mapped to the DOCSIS service flow associated with the Gate. The Classifier used to construct a Service Flow MUST match the Classifier specified for 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.

A classifier is a seven-tuple:

- Protocol;
- Source IP;
- Source Port;
- Destination IP;
- Destination Port;
- Priority;
- DSCP/TOS.

Protocol field identifies the type of protocol (e.g., IP, ICMP, etc.).

Source IP is the IP address (as seen at the CMTS) of the originator of the IP flow, while Destination IP is the termination point for the IP flow.

Source Port and Destination Port specify the UDP or TCP ports for the IP flow.

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

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 with 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).

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

6.1.6 Traffic profile

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

1) FlowSpec;
2) DOCSIS Service Class Name; or
3) DOCSIS-Specific Parameterization.

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; or
3) DOCSIS-Specific Parameters.

If the Policy Server or Application Manager uses FlowSpecs to define the authorized envelope, then the reserved and committed envelopes MUST also be defined using FlowSpecs. Alternatively, if the Policy Server or Application Manager uses DOCSIS Service Class Names to define the authorized envelope, then the reserved and committed envelopes MUST also be defined using DOCSIS Service Class Names.

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 J.112 Annex B 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 [4] and guaranteed [5]. The main difference between the two types of services is discussed in clause 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 clause 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. DOCSIS also allows the parameters to be modified through the use of TLVs. Within IPCablecom Multimedia, DOCSIS Service Class Name QoS parameters MUST NOT be modified through the use of TLVs. A CMTS MUST return error "Undefined Service Class Name" if modifications to the Service Class Name QoS parameters are requested (see 8.4.2.14).

For more information on DOCSIS Service Classes please refer to B.10.1.3 of the J.112 Annex B RFI Recommendation [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 6.4.2.7.

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 [10] if the primary is unavailable.
- Flag indicating whether the CMTS MUST send Event Messages to the Record Keeping Server in real-time, or whether the CMTS MUST batch Event Messages and send them at periodic intervals.

- 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.

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. This object is opaque to the CMTS. 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 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 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.

6.2 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.

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.

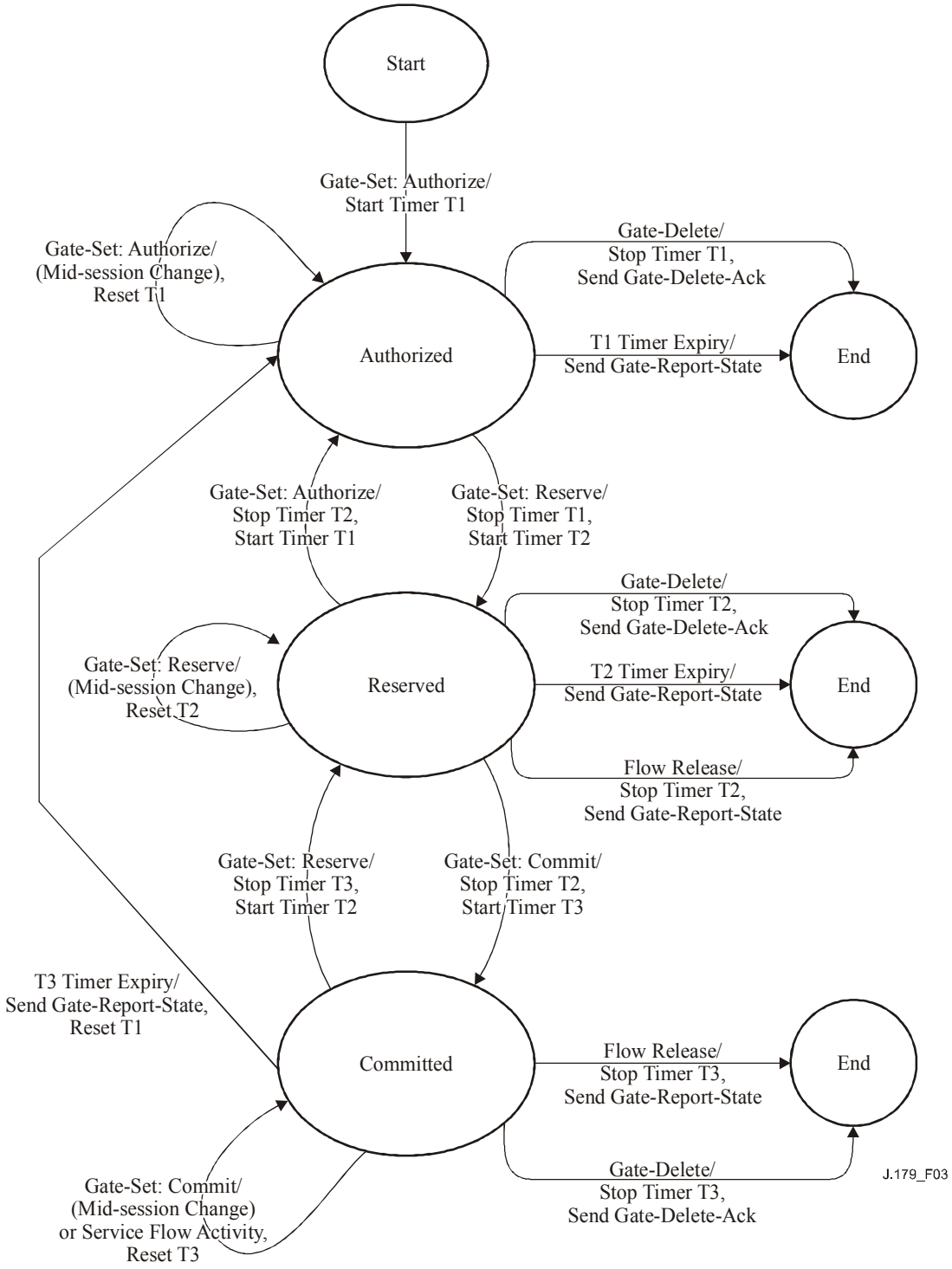

Figure 3/J.179 – Gate state transitions

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

In this clause, 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 ≤ 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 an Authorized state a Gate-Set message is received which 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, stop timer T1, 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 (≥ Reserved Envelope);
- Reserved Loopback to Reserved: Modify Reserved Envelope (≤ Authorized Envelope);
- Reserved to Committed (defines Committed Envelope ≤ 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 6.5.6). If in a Reserved state a Gate-Set message is received which does not transition the Gate to the Authorized or Committed states, then the CMTS MUST restart timer T2.

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 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 again authorize the requested QoS against the Reserved envelope. If the authorization succeeds, the CMTS MUST stop timer T2 and start timer T3. If the authorization fails, the CMTS MUST reset timer T2.

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

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 delete the Service Flow as well as the corresponding Gate, and the Policy Server MUST be notified 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 before timer T2 expires, the CMTS MUST delete the Gate and the corresponding Service Flow and stop timer T2.

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 (≥ Reserved Envelope);
- Committed Loopback to Committed: Modify Reserved Envelop (≥ Committed Envelope);
- Committed Loopback to Committed: Modify Committed Envelop (≤ Reserved Envelope);
- Committed to Reserved (deletes Committed Envelope);
- 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 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. While in the Committed state, the CMTS MUST transition a Gate back into the Reserved state if requested. In Scenario 1, the Policy Server MAY request this action by issuing a Gate-Set message with 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.

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 – 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 may be returned to the Authorized 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, it MUST generate a Gate-Report-State message to the Policy Server indicating that the flow has been inactive for time duration defined by T3. The Policy Server MUST relay the message to the Application Manager. The Application Manager MUST either reactivate the flow and reset the timer by issuing a new Gate-Set message, or remove the Gate by issuing a Gate-Delete message.

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 Appendix I. Under this model, administrative policies are stored in a policy database and controlled by the Policy Server.

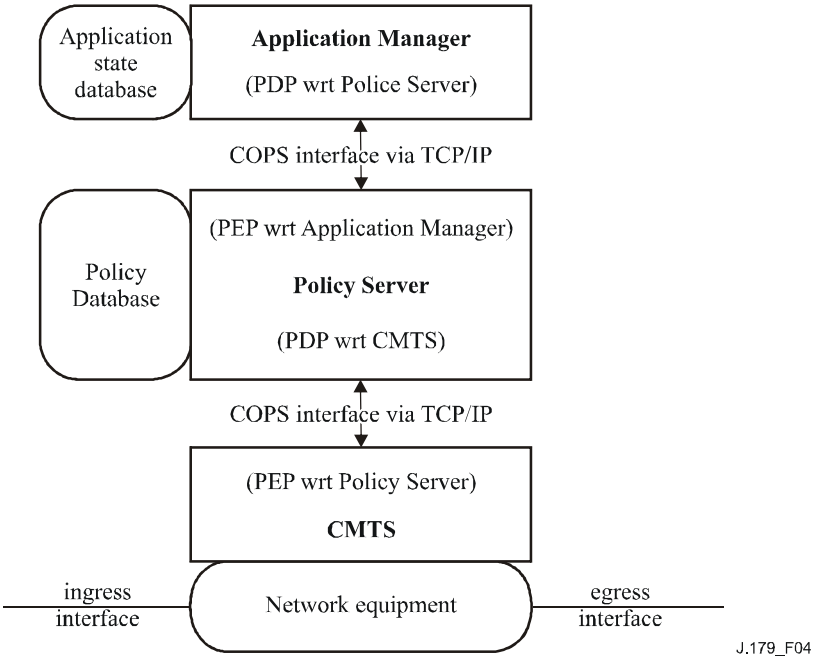

Figure 4/J.179 – QoS admission control layout

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 signalling 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 [7]. 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 of 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.
- Synchronize-State-Request (SSQ)/Synchronize-State-Complete (SSC): The server sends the SSQ message to the client requesting current state information. The client re-issues request queries to the server to perform the synchronization, then sends a SSC message to indicate the completion of the synchronization event. A Policy Server MAY support SSQ/SSC synchronization functions if necessary for the Policy Server to acquire or rebuild state from the CMTS. The CMTS MUST support the SSQ/SSC synchronization functions.

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 COPS message exchanges required for IPCablecom Multimedia are consistent with the COPS protocol, there is a slight difference in the way the COPS session starts. RFC 2748 [7] 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 an 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-T DQoS listen on different ports so that the CMTS may initiate the COPS session with the proper Client-Type.

The details of the COPS protocol are provided in [7]. This IETF RFC provides a description of the base COPS protocol, independent of Client-Type. The IPCablecom architecture is also aligned with IETF RFC 3084 [19]. 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 CMTSs. 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 [11] 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 Recommendation are:

1 = Request (REQ)

2 = Decision (DEC)

3 = Report-State (RPT)

4 = Delete Request State (DRQ)

5 = Synchronize State Request (SSQ)

6 = Client-Open (OPN)

7 = Client-Accept (CAT)

9 = Keep-Alive (KA)

10 = Synchronize State Complete (SSC)

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 [7]) used in this Recommendation, and their C-Num values, are:

1 = Handle

6 = Decision

8 = Error

9 = Client Specific Info

10 = Keep-Alive-Timer

11 = PEP Identification

Each of these objects MUST conform to the format and rules relating to the individual object as defined in [7].

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 (Signalled ClientSI) when carried from the PEP to the PDP in a Report 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 subclauses.

6.4.2.1 TransactionID

TransactionID 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	

Transaction Identifier is a 2-byte unsigned integer quantity that MUST be used by the Policy Server and Application Manager to match responses to commands. Transaction Identifier MUST be set to 0 when included in a Gate-State-Report 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
Gate-Open	13
Gate-Close	14
Gate-Report-State	15

6.4.2.2 AMID

AMID, the Application Manager ID, is a 4-byte unsigned integer value which identifies the Application Manager responsible for handling the session. 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. 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. The AMID object MUST conform to the following format.

Length = 8	S-Num = 2	S-Type = 1
AMID		

6.4.2.3 SubscriberID

SubscriberID is a 4-byte value giving the IPv4 address (represented as four concatenated octet values) of the subscriber for this service request. This address may be the actual IP address of the subscriber CPE device requesting service (if this address is routable and visible from the head-end) or this address may be the IP address of the CM serving this subscriber (if NAT is performed behind the CM). This object is used to route Gate Control messages within a complex network of PS and CMTS elements. It may also be used in the definition and enforcement of per-subscriber policy rules. The SubscriberID object MUST conform to the following format.

Length = 8	S-Num = 3	S-Type = 1
SubscriberID (4-octet IPv4 Address)		

6.4.2.4 GateID

GateID is a 4-byte unsigned integer value which identifies the Gate referenced in the command message, or referenced by the CMTS for a response message. The CMTS MUST ensure the GateID is unique. If the CMTS also supports IPCablecom-T, the GateID MUST NOT duplicate an IPCablecom-T GateID currently in use. The GateID object MUST conform to the following format.

Length = 8	S-Num = 4	S-Type = 1
GateID		

6.4.2.5 GateSpec

GateSpec defines a specific set of attributes associated with a Gate. The GateSpec object MUST conform to the following format.

Length = 16		S-Num = 5	S-Type = 1
Flags	DSCP/TOS Field	DSCP/TOS Mask	SessionClassID
Timer T1		Timer T2	
Timer T3		Reserved	

Flags is a 1-byte bit-field value defined as follows:

Bit 0: direction bit, MUST be either zero for a downstream Gate, or one for an upstream Gate.

Bit 1: DSCP/TOS enable bit, MUST be either zero to disable DSCP overwrite, or one to enable.

Bits 2-7: reserved, MUST be zero.

SessionClassID is a 1-byte unsigned integer value which identifies the proper admission control policy or parameters to be applied for this Gate. The meaning and interpretation of this field is left to the policies of the administrator. The Policy Server and the CMTS MAY support configurable policies based on the SessionClassID. Such policies can be used to limit bandwidth based on different types of services or sessions being administered.

The DSCP/TOS Field is a 1-byte bit field [6] defined by the following alternative structures, depending upon network management strategy. This field, combined with the 1-byte DSCP/TOS Mask is used to identify particular bits within the IPv4 DSCP/TOS byte.

0	1	2	3	4	5	6	7
Differentiated Services Code Point (DSCP)						Not Used	Not Used

0	1	2	3	4	5	6	7
IP Precedence			IP TOS				Not Used

If the 'Enable' bit in the GateSpec Flags field is set, then the CMTS MUST mark the packets traversing the CMTS DSCP/TOS value. If the 'Enable' bit is cleared, then the CMTS MUST NOT perform any marking.

Timers T1, T2 and T3 are 2-byte unsigned integers specified in seconds, and MUST be used as outlined in the Gate Transition Diagram as described in 6.2. A value of zero for T1 indicates that the CMTS provisioned value for the timer MUST be used. T2 corresponds to the DOCSIS Admitted timer and T3 corresponds to the DOCSIS Active timer. All corresponding DOCSIS requirements apply to these timers. Specifically, a zero value for either of these timers indicates that the corresponding timer MUST be disabled.

6.4.2.6 Classifier

The Classifier object specifies the packet matching rules associated with a Gate. As defined in 6.4.3.1 and 6.4.3.2, multiple Classifier objects may be included in the Gate-Set to allow for complex classifier rules. The Classifier object MUST conform to the following format.

Length = 24		S-Num = 6	S-Type = 1
Reserved	Protocol ID	DSCP/TOS Field	DSCP/TOS Mask
Source IP Address (4-octets)			
Destination IP Address (4-octets)			
Source Port		Destination Port	
Priority	Reserved		

Source IP Address and Destination IP Address MUST be a pair of 4-octet IPv4 addresses, or zero for no match (i.e., a wildcard specification that will match any request from the MTA).

Source Port and Destination Port MUST be a pair of 2-byte unsigned integer values, or zero for no match.

DSCP/TOS Field is a 1-byte bit field which MUST conform to the following alternative structures:

0	1	2	3	4	5	6	7
Differentiated Services Code Point (DSCP)						Not Used	Enable

0	1	2	3	4	5	6	7
IP Precedence			IP TOS				Enable

DSCP/TOS Mask is a 1-byte bit field providing a bit mask used to select relevant bits from the accompanying DSCP/TOS Field value.

If the 'Enable' bit is set, then the CMTS MUST use these values to construct the IP TOS Range and Mask field specified in its DSx messaging. If the 'Enable' bit is cleared, then the CMTS MUST omit the IP TOS Range and Mask values from its DSx messaging and exclude the IP TOS byte from the packet classification process.

Priority is a 1-byte field that allows differentiation between classifiers that might overlap. A default value of 64 SHOULD be used if a specific priority value is not required. For further discussion of the Priority field, refer to B.C.2.1.3.5 of [1].

6.4.2.7 Traffic profiles

There are three different ways to express a traffic profile. The traffic profile can be expressed via a FlowSpec, a DOCSIS Service Class Name, or DOCSIS-specific parameters. The three methods are distinguished via a different S-Type value in the Traffic Profile (S-Num = 7) object. S-Type of 1 indicates the object contains a traffic profile specified in RSVP FlowSpec format. S-Type of 2 indicates the object contains a traffic profile specified in DOCSIS Service Class Name format. S-Type of 3 indicates the object contains a traffic profile that is specified via a combination of RSVP FlowSpec and DOCSIS-specific parameters.

All Traffic Profiles utilize "replace" semantics, meaning that the envelopes present in this Traffic Profile replace all existing envelopes associated with the Gate and corresponding Service Flow. Thus, all traffic parameters associated with a given Gate MUST be included in every message that includes a Traffic Profile.

All Traffic Profiles share a common field known as the Envelope Field. This field is a bit field that signals the envelope types (i.e., Authorized, Reserved, and Committed) that are present in the object. A value of 1 in a given bit field indicates that the envelope type is present in the Traffic Profile.

- Bit 0: Authorized Envelope;
- Bit 1: Reserved Envelope;

- Bit 2: Committed Envelope.

Thus, a bit pattern of 001 (or 0x01) indicates the presence of only the Authorized Envelope, while a value of 111 (or 0x7) indicates the presence of all three envelopes. Only the following values are legal: 001, 011 and 111; the Envelope Field MUST be set to one of these three legal values. Further limitations on the value of the Envelope Field may be a function of the current state of the Gate. Refer to 6.2 for more information.

While all Traffic Profiles end up providing QoS on the access network, it is important to note several subtle differences between the signalling mechanisms. As noted previously, the conversion of a FlowSpec (S-Type 1) to DOCSIS parameters by the CMTS is generally less efficient than specifying the DOCSIS parameters themselves. That said, specifying DOCSIS parameters explicitly (S-Types 3-7) is not a panacea either, the QoS MIB only logs QoS information about named Service Flows in its ServiceFlowLogTable. Thus, only flows created via S-Type 2 will have logged QoS information in this table. For some this may not be a major issue, but for debugging and just general operational tracking this subtlety should be taken into account by operators and Application Manager vendors evaluating the Traffic Profile signalling alternatives provided by this Recommendation.

6.4.2.7.1 Flow Spec

The FlowSpec object defines the Traffic Profile associated with a Gate through an RSVP-like parameterization scheme. The mapping of these parameters to DOCSIS parameters is specified in clause 8. The FlowSpec object MUST conform to the following specification:

Length = 36 or 64 or 92		S-Num = 7	S-Type = 1
Envelope	Service Number	Reserved	Reserved
Authorized Envelope			
Token Bucket Rate [r] (IEEE floating point number)			
Token Bucket Size [b] (IEEE floating point number)			
Peak Data Rate (p) (IEEE floating point number)			
Minimum Policed Unit [m] (integer)			
Maximum Packet Size [M] (integer)			
Rate [R] (IEEE floating point number)			
Slack Term [S] (integer)			
Reserved Envelope (optional)			
Token Bucket Rate [r] (IEEE floating point number)			
Token Bucket Size [b] (IEEE floating point number)			
Peak Data Rate (p) (IEEE floating point number)			
Minimum Policed Unit [m] (integer)			
Maximum Packet Size [M] (integer)			
Rate [R] (IEEE floating point number)			
Slack Term [S] (integer)			

Committed Envelope (optional)
Token Bucket Rate [r] (IEEE floating point number)
Token Bucket Size [b] (IEEE floating point number)
Peak Data Rate (p) (IEEE floating point number)
Minimum Policed Unit [m] (integer)
Maximum Packet Size [M] (integer)
Rate [R] (IEEE floating point number)
Slack Term [S] (integer)

The Service Number field corresponds to the RSVP FlowSpec service number as defined in [3]. If Service Number is set to five, this indicates Controlled Load service and the CMTS MUST utilize only the TSpec values (i.e., token bucket parameters) to perform the necessary authorization, reservation and commit operations. For Controlled Load service, the CMTS MUST ignore the RSpec R and S fields.

If Service Number is set to two, this signals Guaranteed service and the CMTS MUST utilize both the TSpec and RSpec values to perform the necessary authorization, reservation and commit operations.

The values r, b, p, m, M, R, and S are defined and described in clause 9.

6.4.2.7.2 DOCSIS Service Class Name

The DOCSIS Service Class Name object defines the preconfigured Service Class Name associated with a Gate. The DOCSIS Service Class Name object MUST conform to the following specification:

Length = 12 or 16 or 20 or 24		S-Num = 7	S-Type = 2
Envelope	Reserved	Reserved	Reserved
Service Class Name			

The Service Class Name MUST be 2-16 bytes of null-terminated ASCII string. (Refer to B.C.2.2.3.4 of [1]). This name MUST be padded with null bytes to align on a 4-byte boundary.

Note that unlike a FlowSpec Traffic Profile which allows for different parameters to be associated with each Envelope, the DOCSIS Service Class Name Traffic Profile supports different Gate states as specified by the Envelope field, but each Envelope is defined by the same associated DOCSIS Service Class Name. This allows for having two-phase commit operations utilizing the DOCSIS Service Class Names, but each Envelope must be identical. Also note, that it is possible to change the DOCSIS Service Class Name associated with a Gate, but that such a change applies to all Envelopes associated with a given Gate.

6.4.2.7.3 Best Effort service

The Best Effort object defines the Traffic Profile associated with a Gate through an upstream DOCSIS-specific parameterization scheme. The Best Effort object MUST conform to the following specification:

Length = 32, 56 or 80		S-Num = 7	S-Type = 3
Envelope	Reserved	Reserved	Reserved
Authorized Envelope			
Traffic Priority	Reserved		
Request Transmission Policy			
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Reserved Envelope (optional)			
Traffic Priority	Reserved		
Request Transmission Policy			
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Committed Envelope (optional)			
Traffic Priority	Reserved		
Request Transmission Policy			
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	

Traffic Priority is a 1-byte unsigned integer field specifying the relative priority assigned to the Service Flow in comparison with other flows. This field is fully defined in B.C.2.2.5.1 of [1]. A default Traffic Priority of 0 SHOULD be used if a specific Traffic Priority value is not required.

Request/Transmission Policy is a 4-byte bit field as defined in B.C.2.2.6.3 of [1]. A default Request/Transmission policy of 0 SHOULD be used if a specific Request/Transmission Policy value is not required.

Maximum Sustained Traffic Rate is a 4-byte unsigned integer field specifying the rate parameter, in bits/s, for a token-bucket-based rate limit for this Service Flow. This field is fully defined in B.C.2.2.5.2 of [1]. A value of 0 indicates that no explicitly-enforced Maximum Sustained Rate is requested. A default Maximum Sustained Traffic Rate of 0 SHOULD be used if a specific Maximum Sustained Traffic Rate is not required.

Maximum Traffic Burst is a 4-byte unsigned integer field specifying the token bucket size, in bytes, for a token-bucket-based rate limit for this Service Flow. This field is fully defined in B.C.2.2.5.3 of [1]. A default Maximum Traffic Burst of 3044 bytes SHOULD be used if a specific Maximum Traffic Burst is not required. The value of this parameter has no effect unless a non-zero value has been provided for the Maximum Sustained Traffic Rate parameter.

Minimum Reserved Traffic Rate is a 4-byte unsigned integer field specifying the minimum rate, in bits/s, reserved for this Service Flow. This field is fully defined in B.C.2.2.5.4 of [1]. A default Minimum Reserved Traffic Rate of 0 SHOULD be used if a specific Minimum Reserved Traffic Rate is not required.

Assumed Minimum Reserved Traffic Rate Packet Size is a 2-byte unsigned integer field specifying an assumed minimum packet size, in bytes, for which the Minimum Reserved Traffic Rate will be provided for this Service flow. This field is fully defined in B.C.2.2.5.5 of [1]. A default Assumed Minimum Reserved Traffic Rate Packet Size of 0 SHOULD be used if a specific Assumed Minimum Reserved Traffic Rate Packet size is not required. Upon receipt of a value of 0 the CMTS MUST utilize its implementation-specific default size for this parameter, not 0 bytes.

6.4.2.7.4 Non-Real-Time Polling service

The Non-Real-Time Polling object defines the Traffic Profile associated with an upstream Gate through a DOCSIS-specific parameterization scheme. The Non-Real-Time Polling object MUST conform to the following specification:

Length = 36, 64 or 92		S-Num = 7	S-Type = 4
Envelope	Reserved	Reserved	Reserved
Authorized Envelope			
Traffic Priority	Reserved		
Request Transmission Policy			
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Nominal Polling Interval			
Reserved Envelope (optional)			
Traffic Priority	Reserved		
Request Transmission Policy			
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Nominal Polling Interval			
Committed Envelope (optional)			
Traffic Priority	Reserved		
Request Transmission Policy			
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Nominal Polling Interval			

Traffic Priority is a 1-byte unsigned integer field specifying the relative priority assigned to the Service Flow in comparison with other flows. This field is fully defined in B.C.2.2.5.1 of [1]. A default Traffic Priority of 0 SHOULD be used if a specific Traffic Priority value is not required.

Request/Transmission Policy is a 4-byte bit field as defined in B.C.2.2.6.3 of [1].

NOTE – For this Service Flow Scheduling Type, there is no default value for Request/Transmission Policy and all values (including 0) have meaning in DOCSIS.

Maximum Sustained Traffic Rate is a 4-byte unsigned integer field specifying the rate parameter, in bits/s, for a token-bucket-based rate limit for this Service Flow. This field is fully defined in B.C.2.2.5.2 of [1]. A value of 0 indicates that no explicitly-enforced Maximum Sustained Rate is requested. A default Maximum Sustained Traffic Rate of 0 SHOULD be used if a specific Maximum Sustained Traffic Rate is not required.

Maximum Traffic Burst is a 4-byte unsigned integer field specifying the token bucket size, in bytes, for a token-bucket-based rate limit for this Service Flow. This field is fully defined in B.C.2.2.5.3 of [1]. A default Maximum Traffic Burst of 3044 bytes SHOULD be used if a specific Maximum Traffic Burst is not required. The value of this parameter has no effect unless a non-zero value has been provided for the Maximum Sustained Traffic Rate parameter.

Minimum Reserved Traffic Rate is a 4-byte unsigned integer field specifying the minimum rate, in bits/s, reserved for this Service Flow. This field is fully defined in B.C.2.2.5.4 of [1]. A default Minimum Reserved Traffic Rate of 0 SHOULD be used if a specific Minimum Reserved Traffic Rate is not required.

Assumed Minimum Reserved Traffic Rate Packet Size is a 2-byte unsigned integer field specifying an assumed minimum packet size, in bytes, for which the Minimum Reserved Traffic Rate will be provided for this Service flow. This field is fully defined in B.C.2.2.5.5 of [1]. A default Assumed Minimum Reserved Traffic Rate Packet Size of 0 SHOULD be used if a specific Assumed Minimum Reserved Traffic Rate Packet size is not required. Upon receipt of a value of 0 the CMTS MUST utilize its implementation-specific default size for this parameter, not 0 bytes.

Nominal Polling Interval is a 4-byte unsigned integer field specifying the nominal interval (in units of microseconds) between successive unicast request opportunities for this Service Flow on the upstream channel. This field is fully defined in B.C.2.2.6.4 of [1]. A default Nominal Polling Interval of 0 SHOULD be used if a specific Nominal Polling Interval is not required. Upon receipt of a value of 0 the CMTS MUST utilize its implementation-specific default size for this parameter – not 0 microseconds.

6.4.2.7.5 Real-Time Polling service

The Real-Time Polling object defines the Traffic Profile associated with an upstream Gate through a DOCSIS-specific parameterization scheme. The Real-Time Polling object MUST conform to the following specification:

Length = 36, 64 or 92		S-Num = 7	S-Type = 5
Envelope	Reserved	Reserved	Reserved
Authorized Envelope			
Request Transmission Policy			
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Nominal Polling Interval			
Tolerated Poll Jitter			

Reserved Envelope (optional)	
Request Transmission Policy	
Maximum Sustained Traffic Rate	
Maximum Traffic Burst	
Minimum Reserved Traffic Rate	
Assumed Minimum Reserved Traffic Rate Packet Size	Reserved
Nominal Polling Interval	
Tolerated Poll Jitter	
Committed Envelope (optional)	
Request Transmission Policy	
Maximum Sustained Traffic Rate	
Maximum Traffic Burst	
Minimum Reserved Traffic Rate	
Assumed Minimum Reserved Traffic Rate Packet Size	Reserved
Nominal Polling Interval	
Tolerated Poll Jitter	

Request/Transmission Policy is a 4-byte bit field as defined in B.C.2.2.6.3 of [1].

NOTE – For this Service Flow Scheduling Type, there is no default value for Request/Transmission Policy and all values (including 0) have meaning in DOCSIS.

Maximum Sustained Traffic Rate is a 4-byte unsigned integer field specifying the rate parameter, in bits/s, for a token-bucket-based rate limit for this Service Flow. This field is fully defined in B.C.2.2.5.2 of [1]. A value of 0 indicates that no explicitly-enforced Maximum Sustained Rate is requested. A default Maximum Sustained Traffic Rate of 0 SHOULD be used if a specific Maximum Sustained Traffic Rate is not required.

Maximum Traffic Burst is a 4-byte unsigned integer field specifying the token bucket size, in bytes, for a token-bucket-based rate limit for this Service Flow. This field is fully defined in B.C.2.2.5.3 of [1]. A default Maximum Traffic Burst of 3044 bytes SHOULD be used if a specific Maximum Traffic Burst is not required. The value of this parameter has no effect unless a non-zero value has been provided for the Maximum Sustained Traffic Rate parameter.

Minimum Reserved Traffic Rate is a 4-byte unsigned integer field specifying the minimum rate, in bits/s, reserved for this Service Flow. This field is fully defined in B.C.2.2.5.4 of [1]. A default Minimum Reserved Traffic Rate of 0 SHOULD be used if a specific Minimum Reserved Traffic Rate is not required.

Assumed Minimum Reserved Traffic Rate Packet Size is a 2-byte unsigned integer field specifying an assumed minimum packet size, in bytes, for which the Minimum Reserved Traffic Rate will be provided for this Service flow. This field is fully defined in B.C.2.2.5.5 of [1]. A default Assumed Minimum Reserved Traffic Rate Packet Size of 0 SHOULD be used if a specific Assumed Minimum Reserved Traffic Rate Packet size is not required. Upon receipt of a value of 0, the CMTS MUST utilize its implementation-specific default size for this parameter, not 0 bytes.

Nominal Polling Interval is a 4-byte unsigned integer field specifying the nominal interval (in units of microseconds) between successive unicast request opportunities for this Service Flow on the upstream channel. This field is fully defined in B.C.2.2.6.4 of [1]. For this Service Flow Scheduling Type, there is no default value for Nominal Polling Interval.

Tolerated Polling Jitter is a 4-byte unsigned integer field specifying the maximum amount of time that the unicast request interval may be delayed from the nominal periodic schedule (measured in

microseconds). This field is fully defined in B.C.2.2.6.5 of [1]. A default Tolerated Polling Jitter of 0 SHOULD be used if a specific Tolerated Polling Jitter is not required. Upon receipt of a value of 0, the CMTS MUST utilize its implementation-specific default size for this parameter – not 0 microseconds.

6.4.2.7.6 Unsolicited Grant service

The Unsolicited Grant object defines the Traffic Profile associated with an upstream Gate through a DOCSIS-specific parameterization scheme. The Unsolicited Grant object MUST conform to the following specification:

Length = 36, 64 or 92		S-Num = 7	S-Type = 6
Envelope	Reserved	Reserved	Reserved
Authorized Envelope			
Request Transmission Policy			
Unsolicited Grant Size		Grants/Interval	Reserved
Nominal Grant Interval			
Tolerated Grant Jitter			
Reserved Envelope (optional)			
Request Transmission Policy			
Unsolicited Grant Size		Grants/Interval	Reserved
Nominal Grant Interval			
Tolerated Grant Jitter			
Committed Envelope (optional)			
Request Transmission Policy			
Unsolicited Grant Size		Grants/Interval	Reserved
Nominal Grant Interval			
Tolerated Grant Jitter			

Request/Transmission Policy is a 4-byte bit field as defined in B.C.2.2.6.3 of [1].

NOTE – For this Service Flow Scheduling Type, there is no default value for Request/Transmission Policy and all values (including 0) have meaning in DOCSIS.

Unsolicited Grant Size is a 2-byte unsigned integer field specifying the grant size (in bytes) as defined in B.C.2.2.6.6 of [1]. There is no default value of Unsolicited Grant Size.

Grants per Interval is a 1-byte unsigned integer field specifying the number of grants per Nominal Grant Interval as defined in B.C.2.2.6.9 of [1]. There is no default value of Grants per Interval, but a value of 1 is recommended.

Nominal Grant Interval is a 4-byte unsigned integer field specifying the nominal time between successive data grant opportunities for this Service Flow (in units of microseconds) as defined in B.C.2.2.6.7 of [1]. There is no default value of Nominal Grant Interval.

Tolerated Grant Jitter is a 4-byte unsigned integer field specifying the maximum amount of time that transmission opportunities may be delayed from the nominal periodic schedule (in units of microseconds) as defined in B.C.2.2.6.8 of [1]. There is no default value for Tolerated Grant Jitter.

6.4.2.7.7 Unsolicited Grant service with Activity Detection

The Unsolicited Grant with Activity Detection object defines the Traffic Profile associated with an upstream Gate through a DOCSIS-specific parameterization scheme. The Unsolicited Grant with Activity Detection object MUST conform to the following specification:

Length = 36, 64 or 92		S-Num = 7	S-Type = 7
Envelope	Reserved	Reserved	Reserved
Authorized Envelope			
Request Transmission Policy			
Unsolicited Grant Size		Grants/Interval	Reserved
Nominal Grant Interval			
Tolerated Grant Jitter			
Nominal Polling Interval			
Tolerated Poll Jitter			
Reserved Envelope (optional)			
Request Transmission Policy			
Unsolicited Grant Size		Grants/Interval	Reserved
Nominal Grant Interval			
Tolerated Grant Jitter			
Nominal Polling Interval			
Tolerated Poll Jitter			
Committed Envelope (optional)			
Request Transmission Policy			
Unsolicited Grant Size		Grants/Interval	Reserved
Nominal Grant Interval			
Tolerated Grant Jitter			
Nominal Polling Interval			
Tolerated Poll Jitter			

Request/Transmission Policy is a 4-byte bit field as defined in B.C.2.2.6.3 of [1].

NOTE – For this Service Flow Scheduling Type, there is no default value for Request/Transmission Policy and all values (including 0) have meaning in DOCSIS.

Unsolicited Grant Size is a 2-byte unsigned integer field specifying the grant size (in bytes) as defined in B.C.2.2.6.6 of [1]. There is no default value of Unsolicited Grant Size.

Grants per Interval is a 1-byte unsigned integer field specifying the number of grants per Nominal Grant Interval as defined in B.C.2.2.6.9 of [1]. There is no default value of Grants per Interval, but a value of 1 is recommended.

Nominal Grant Interval is a 4-byte unsigned integer field specifying the nominal time between successive data grant opportunities for this Service Flow (in units of microseconds) as defined in B.C.2.2.6.7 of [1]. There is no default value of Nominal Grant Interval.

Tolerated Grant Jitter is a 4-byte unsigned integer field specifying the maximum amount of time that transmission opportunities may be delayed from the nominal periodic schedule (in units of microseconds) as defined in B.C.2.2.6.8 of [1]. There is no default value of Tolerated Grant Jitter.

Nominal Polling Interval is a 4-byte unsigned integer field specifying the nominal interval (in units of microseconds) between successive unicast request opportunities for this Service Flow on the upstream channel. This field is fully defined in B.C.2.2.6.4 of [1]. There is no default value of Nominal Polling Interval.

Tolerated Polling Jitter is a 4-byte unsigned integer field specifying the maximum amount of time that the unicast request interval may be delayed from the nominal periodic schedule (measured in microseconds). This field is fully defined in B.C.2.2.6.5 of [1]. Upon receipt of a value of 0 the CMTS MUST utilize its implementation-specific default size for this parameter, not 0 microseconds.

6.4.2.7.8 Downstream service

The Downstream object defines the Traffic Profile associated with a Gate through a downstream DOCSIS-specific parameterization scheme. The Downstream object MUST conform to the following specification:

Length = 32, 56 or 80		S-Num = 7	S-Type = 8
Envelope	Reserved	Reserved	Reserved
Authorized Envelope			
Traffic Priority	Reserved		
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Maximum Downstream Latency			
Reserved Envelope (optional)			
Traffic Priority	Reserved		
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Maximum Downstream Latency			
Committed Envelope (optional)			
Traffic Priority	Reserved		
Maximum Sustained Traffic Rate			
Maximum Traffic Burst			
Minimum Reserved Traffic Rate			
Assumed Minimum Reserved Traffic Rate Packet Size		Reserved	
Maximum Downstream Latency			

Traffic Priority is a 1-byte unsigned integer field specifying the relative priority assigned to the Service Flow in comparison with other flows. This field is fully defined in B.C.2.2.5.1 of [1]. A default Traffic Priority of 0 SHOULD be used if a specific Traffic Priority value is not required.

Maximum Sustained Traffic Rate is a 4-byte unsigned integer field specifying the rate parameter, in bits/s, for a token-bucket-based rate limit for this Service Flow. This field is fully defined in B.C.2.2.5.2 of [1]. A value of 0 indicates that no explicitly-enforced Maximum Sustained Rate is

requested. A default Maximum Sustained Traffic Rate of 0 SHOULD be used if a specific Maximum Sustained Traffic Rate is not required.

Maximum Traffic Burst is a 4-byte unsigned integer field specifying the token bucket size, in bytes, for a token-bucket-based rate limit for this Service Flow. This field is fully defined in B.C.2.2.5.3 of [1]. A default Maximum Traffic Burst of 3044 bytes SHOULD be used if a specific Maximum Traffic Burst is not required. The value of this parameter has no effect unless a non-zero value has been provided for the Maximum Sustained Traffic Rate parameter.

Minimum Reserved Traffic Rate is a 4-byte unsigned integer field specifying the minimum rate, in bits/s, reserved for this Service Flow. This field is fully defined in B.C.2.2.5.4 of [1]. A default Minimum Reserved Traffic Rate of 0 SHOULD be used if a specific Minimum Reserved Traffic Rate is not required.

Assumed Minimum Reserved Traffic Rate Packet Size is a 2-byte unsigned integer field specifying an assumed minimum packet size, in bytes, for which the Minimum Reserved Traffic Rate will be provided for this Service Flow. This field is fully defined in B.C.2.2.5.5 of [1]. A default Assumed Minimum Reserved Traffic Rate Packet Size of 0 SHOULD be used if a specific Assumed Minimum Reserved Traffic Rate Packet size is not required. Upon receipt of a value of 0 the CMTS MUST utilize its implementation-specific default size for this parameter, not 0 bytes.

Maximum Downstream Latency is a 4-byte unsigned integer field specifying the maximum latency between reception of a packet on the CMTS's NSI and the forwarding of the packet on its RF interface as defined in B.C.2.2.7.1 of [1]. A default Maximum Downstream Latency of 0 SHOULD be used if a specific Maximum Downstream Latency is not required. Upon receipt of a value of 0, the CMTS MUST NOT include this parameter in its DOCSIS signalling for this Service Flow.

6.4.2.8 Event Generation Info

The Event Generation Info object contains all the information necessary to support the event messages as specified and required in ITU-T Rec. J.164. The Event Generation Info object MUST conform to the following specification:

Length = 44	S-Num = 8	S-Type = 1
Primary-Record-Keeping-Server-IP-Address (4-octets)		
Primary-Record-Keeping-Server-Port	Reserved	
Secondary-Record-Keeping-Server-IP-Address (4-octets)		
Secondary-Record-Keeping-Server-Port	Reserved	
Billing-Correlation-ID (24-bytes)		

Primary-Record-Keeping-Server-IP-Address is a 4-byte field which MUST contain the IPv4 address of the primary RKS to whom event records are to be sent.

Primary-Record-Keeping-Server-Port field is a 2-byte unsigned integer which MUST contain the port number on the primary RKS where event records are to be sent.

Secondary-Record-Keeping-Server-IP-Address is a 4-byte field which MUST contain the IPv4 address of the secondary RKS to whom records are to be sent if the primary RKS is unavailable.

Secondary-Record-Keeping-Server-Port is a 2-byte unsigned integer which MUST contain the port number on the secondary RKS where event records are to be sent.

Billing-Correlation-ID is a 24-byte field which MUST contain the identifier assigned by the AM or PS for all records related to this session. See [10] for the definition and format of this attribute.

6.4.2.9 Volume-Based Usage Limit

The Volume-Based Usage Limit object specifies the amount of data that can be transmitted over this Gate before meeting a volume threshold. This object is OPTIONAL in a Gate-Set and Gate-Info-Ack message. It MUST NOT be used in any other messages. The Volume-Based Usage Limit object MUST conform to the following specification:

Length = 12	S-Num = 9	S-Type = 1
Usage Limit		

Usage Limit is an 8-byte unsigned integer defined in units of kilobytes. A value of zero indicates no volume limit is imposed. The bytes counted towards the limit are from the byte after the DOCSIS MAC Header HCS to the end of the CRC for all packets transmitted on the Service Flow associated with this Gate.

6.4.2.10 Time-Based Usage Limit

The Time-Based Usage Limit object specifies the amount of time a Gate can remain committed before meeting a time-limit threshold. The Time-Based Usage Limit object MUST conform to the following specification:

Length = 8	S-Num = 10	S-Type = 1
Time-Limit		

Time-Limit is a 4-byte unsigned integer defined in units of seconds. This is the limit on the amount of time a Gate can be in a committed state. This object is OPTIONAL in a Gate-Set message. If included in a Gate-Set, this object MUST be stored by the CMTS and provided in response to any subsequent Gate queries. While the Application Manager is REQUIRED to delete Gates associated with a media session that has exceeded its Time-Based Usage Limit, the CMTS or Policy Server MAY use this object to police Application Manager enforcement of Time-Based Usage Limits. The Application Manager or Policy Server MAY also query for this object as part of a failure recovery or other mechanism.

A value of zero indicates no time-limit for the associated Gate.

6.4.2.11 Opaque Data

The Opaque Data object contains information that a Policy Server or Application Manager MAY store on a CMTS that remains opaque to the CMTS. The opaque data object is OPTIONAL in a Gate-Set message. It MUST NOT be used in any other messages issued by the PDP to the PEP. If the object is present, the CMTS MUST store the Opaque Data locally, and include it in all messages it generates to the Policy Server associated with the Gate.

If the Opaque Data object is included in a Gate-Set message from the Application Manager to a Policy Server, the Policy Server MUST forward this object to the CMTS. The length of the Opaque Data is fixed at 8 bytes.

Length = 12	S-Num = 11	S-Type = 1
Opaque Data		

6.4.2.12 Gate Time Info

The Gate Time Info object contains a timestamp representing the time the Gate last entered the committed state. The Gate Time Info object MUST conform to the following specification:

Length = 8	S-Num = 12	S-Type = 1
Time Committed		

Time Committed is a 4-byte unsigned integer indicating the number of seconds this Gate has been in the Committed state.

NOTE – This is intended to be identical to docsQosServiceFlowTimeActive from the QoS MIB [17].

6.4.2.13 Gate Usage Info

The Gate Usage Info object contains a counter indicating the number of kilobytes transmitted over this Gate. The Gate Usage Info object MUST conform to the following specification:

Length = 8	S-Num = 13	S-Type = 1
Octet Count		

Octet Count is a 4-byte unsigned integer which represents the number of bytes (counted from the DOCSIS MAC Header HCS to the end of the CRC) which have traversed the Service Flow associated with the Gate in units of 1024 bytes.

6.4.2.14 IPCablecom Error

The IPCablecom Error object contains information on the type of error that has occurred. The error is generated in response to a Gate Control command and is contained in Gate-Set-Err, Gate-Info-Err and Gate-Delete-Err messages. The IPCablecom Error object MUST conform to the following specification:

Length = 8	S-Num = 14	S-Type = 1
Error-Code	Error-Subcode	

Error-Code is a 2-byte unsigned integer representing a specific error and MUST be one of the following:

1 = Insufficient Resources

2 = Unknown GateID

6 = Missing Required Object

7 = Invalid Object

8 = Volume-Based Usage Limit Exceeded

9 = Time-Based Usage Limit Exceeded

10 = Session Class Limit Exceeded

11 = Undefined Service Class Name

12 = Incompatible Envelope

13 = Invalid SubscriberID

14 = Unauthorized AMID

15 = Number of Classifiers Not Supported

127 = Other, Unspecified Error

Error-Subcode is a 2-byte unsigned integer field used to provide further information about the error. In the case of Error-Codes 6 and 7, this field MUST contain the S-Num and S-Type of the object that is missing or in error. The order of the S-Num and S-Type values within the Error-Subcode MUST be the same as in the original message. In cases where multiple valid alternatives exist for the S-Type of a missing object, this portion of the Error-Subcode MUST be set to zero. In the case of Error-Code 15, the Error-Subcode field MUST contain the number of Classifiers supported per Gate.

Error-Codes 8, 9 and 10 are generated as a result of a Policy Request failing to meet the requirements of a Policy Server's authorization. When the Application Manager issues a Gate-Set message with a volume or time-based limit to the Policy Server, the Policy Server MAY reject the request based on policy rules installed on the Policy Server. For example, such a policy rule may state that if a volume limit request exceeds a maximum value, the Policy Server must reject the request.

6.4.2.15 Gate State

The information in the Gate State object reflects the current state of the Gate. The CMTS MUST include the Gate State object in any unsolicited messages that it sends to the Policy Server. The Policy Server may use this information to report state to the Application Manager, or for enforcing complex rules that might require state knowledge of the Gate.

Typically, the Policy Server is aware of state transitions since it usually provides the stimulus for these transitions to the CMTS, but in some cases the Gate may transition locally on the CMTS without the Policy Server's involvement. In these cases, the CMTS MUST report the state transition to the Policy Server via unsolicited Gate-Report-State messages. When issuing Gate-Report-State messages, the PEP MUST make sure the Solicited Flag in the COPS message header is cleared, and the Report Type in the header is set to Accounting. The Gate State object MUST conform to the following specification:

Length = 8	S-Num = 15	S-Type = 1
State	Reason	

State is a 2-byte unsigned integer field which MUST indicate one of the following states:

1 = Idle/Closed

2 = Authorized

3 = Reserved

4 = Committed

Reason is a 2-byte unsigned integer field which MUST indicate one of the following reasons for this update:

1 = Close initiated by CMTS due to reservation reassignment

2 = Close initiated by CMTS due to lack of DOCSIS MAC-layer responses

3 = Close initiated by CMTS due to timer T1 expiration

4 = Close initiated by CMTS due to timer T2 expiration

5 = Inactivity timer expired due to Service Flow inactivity (timer T3 expiration)

6 = Close initiated by CMTS due to lack of Reservation Maintenance

7 = Gate state unchanged, but volume limit reached

65535 = Other

6.4.3 Gate Control messages

There are two separate profiles for Gate Control messages: one for messages exchanged between the Application Manager and the Policy Server, and one for messages between the Policy Server and the CMTS. While similar, these two profiles do exhibit some minor differences.

6.4.3.1 Profile for Application Manager to Policy Server interface

Messages that perform gate control between the Application Manager and Policy Server are defined and MUST be formatted as follows.

Note that messages from the Application Manager to Policy Server MUST be formatted as COPS Decision messages, and messages from Policy Server to Application Manager MUST be formatted as COPS Report-State messages.

<Gate Control Command> = <COPS Common Header> <Client Handle> <Context>

<Decision Flags> <ClientSI Data>

<ClientSI Data> = <Gate-Set> | <Gate-Info> | <Gate-Delete>

<Gate Control Response> = <COPS Common Header> <Client Handle> <Report Type>

<ClientSI Object>

<ClientSI Object> = <Gate-Set-Ack> | <Gate-Set-Err> | <Gate-Info-Ack> | <Gate-Info-Err> |

<Gate-Delete-Ack> | <Gate-Delete-Err> | <Gate-State-Report>

<Gate-Set> = <Decision Header> <TransactionID> <AMID> <SubscriberID> [<GateID>]

<GateSpec> <Traffic Profile> <Classifier> [<Classifier>]

[<Volume-Based Usage Limit>] [<Time-Based Usage Limit>] [<Opaque Data>]

<Gate-Set-Ack> = <ClientSI Header> <TransactionID> <AMID> <SubscriberID> <GateID>

[<Opaque Data>]

<Gate-Set-Err> = <ClientSI Header> <TransactionID> <AMID> <SubscriberID>

<IPCablecom Error> [<Opaque Data>]

<Gate-Info> = <Decision Header> <TransactionID> <AMID> <SubscriberID> <GateID>

<Gate-Info-Ack> = <ClientSI Header> <TransactionID> <AMID> <SubscriberID> <GateID> <GateSpec> <GateState> <Classifier> <Traffic Profile> <Gate Time Info> <Gate Usage Info> [<Volume-Based Usage Limit>] [<Time-Based Usage Limit>] [<Opaque Data>]

<Gate-Info-Err> = <ClientSI Header> <TransactionID> <AMID> <GateID> <IPCablecom Err> [<Opaque Data>]

<Gate-Delete> = <Decision Header> <TransactionID> <AMID> <SubscriberID> <GateID>

<Gate-Delete-Ack> = <ClientSI Header> <TransactionID> <AMID> <GateID> [<Opaque Data>]

<Gate-Delete-Err> = <ClientSI Header> <TransactionID> <AMID> <GateID>

<IPCablecom Error> [<Opaque Data>]

<Gate-State-Report> = <ClientSI Header> <TransactionID> <AMID> <SubscriberID> <GateID> <GateState>

<Gate Time Info> <Gate Usage Info> [<Opaque Data>]

6.4.3.2 Profile for Policy Server to CMTS interface

Messages that perform Gate Control between the Policy Server and CMTS are defined and MUST be formatted as follows.

Note that messages from Policy Server to CMTS MUST be formatted as COPS Decision messages, and messages from CMTS to Policy Server MUST be formatted as COPS Report-State messages.

<Gate Control Command> = <COPS Common Header> <Client Handle> <Context>

<Decision Flags> <ClientSI Data>

<ClientSI Data> = <Gate-Set> | <Gate-Info> | <Gate-Delete>

<Gate Control Response> = <COPS Common Header> <Client Handle> <Report Type>

<ClientSI Object>

<ClientSI Object> = <Gate-Set-Ack> | <Gate-Set-Err> | <Gate-Info-Ack> | <Gate-Info-Err> | <Gate-Delete-Ack> | <Gate-Delete-Err> | <Gate-State-Report>

<Gate-Set> = <Decision Header> <TransactionID> <AMID> <SubscriberID> [<GateID>]

<GateSpec> <Traffic Profile> <Classifier> [<Classifier>] [<Event Generation Info>]

[<Volume-Based Usage Limit>] [<Time-Based Usage Limit>] [<Opaque Data>]

<Gate-Set-Ack> = <ClientSI Header> <TransactionID> <AMID> <SubscriberID> <GateID> [<Opaque Data>]

<Gate-Set-Err> = <ClientSI Header> <TransactionID> <AMID> <SubscriberID>

<IPCablecom Error> [<Opaque Data>]

<Gate-Info> = <Decision Header> <TransactionID> <AMID> <SubscriberID> <GateID>

<Gate-Info-Ack> = <ClientSI Header> <TransactionID> <AMID> <SubscriberID> <GateID>

[<Event Generation Info>] <Gate-Spec> <Classifier> <Traffic Profile>

<Gate Time Info> <Gate Usage Info> [<Volume-Based Usage Limit>]

[<Time-Based Usage Limit>] [<Opaque Data>]

<Gate-Info-Err> = <ClientSI Header> <TransactionID> <AMID> <GateID> <IPCablecom Err> [<Opaque Data>]

<Gate-Delete> = <Decision Header> <TransactionID> <AMID> <SubscriberID> <GateID>

<Gate-Delete-Ack> = <ClientSI Header> <TransactionID> <AMID> <GateID> [<Opaque Data>]

<Gate-Delete-Err> = <ClientSI Header> <TransactionID> <AMID> <GateID>

<IPCablecom Error> [<Opaque Data>]

<Gate-State-Report> = <ClientSI Header> <TransactionID> <AMID> <SubscriberID> <GateID> <GateState>

<Gate Time Info> <Gate Usage Info> [<Opaque Data>]

There are three basic Gate Control command messages: Gate-Set, Gate-Info and Gate-Delete. These messages are embedded in the Client-Specific Decision Data in a COPS Decision message. For Gate Control command messages, the Context object (C-Num = 2, C-Type = 1) in the COPS Decision message MUST have the R-Type (Request Type Flag) value set to 0x08 (Configuration Request) and the M-Type set to zero. The Command-Code field in the mandatory Decision-Flags object (C-Num = 6, C-Type = 1) MUST be set to 1 (Install Configuration). Other values MUST cause the CMTS to generate a Report-State message indicating failure. The Gate Command Type field in the TransactionID object distinguishes the type of command being issued.

There are seven Gate Control response messages: Gate-Set-Ack, Gate-Set-Err, Gate-Info-Ack, Gate-Info-Err, Gate-Delete-Ack, Gate-Delete-Err, and Gate-State-Report. The first six Gate Control response messages are solicited responses to Gate Control command messages. The seventh, Gate-State-Report, is an unsolicited response to the PS from the CMTS to inform of a state change.

These messages are embedded in the Client-Specific Information Object in COPS Report-State messages. The Report-Type object (C-Num = 12, C-Type = 1) included in the COPS Report-State message for Gate Control responses MUST have the Report-Type field set to 1 (Success) or 2 (Failure) depending on the outcome of the Gate Control command. Report-State messages in response to a Gate Control command MUST have the solicited message flag bit set in the COPS header. The Gate Command Type field in the TransactionID object distinguishes the type of response being issued.

The CMTS generates the Gate-State-Report message when there is a state transition on the Gate that is not due to a Decision message, or when some policy limit has been reached. For the Gate-Report-State message, the Report-Type field MUST be set to 3 (Accounting), and the solicited flag field in the common header MUST be cleared.

If an object that is received in a Gate Control message contains an S-Num or S-Type that is not recognized, that object MUST be ignored. The presence of such an object within a Gate Control message MUST NOT be treated as an error provided that after such parameter is dropped, all required objects are present in the message.

6.5 Gate control protocol operation

6.5.1 Initialization sequence

When a PEP (Policy Server or CMTS) boots, it MUST listen for incoming COPS connections on the IANA-assigned TCP port number 3918. Any Application Manager or Policy Server (PDP) with a need to contact a PEP MUST initiate a TCP connection to the PEP on that port. It is expected that multiple Application Managers will establish COPS connections with multiple Policy Servers, and multiple Policy Servers will establish COPS connections with multiple CMTSs. When the TCP connection between the PEP and the PDP is established, the PEP MUST send information about itself to the PDP in the form of a Client-Open message.

Upon successful receipt of the Client-Open message, the PDP MUST send a Client-Accept message. This message MUST include the Keep-Alive-Timer object, which tells the PEP the maximum interval between Keep-Alive messages.

Upon successful receipt of the Client-Accept message, the PEP MUST send a Request message, including the Client-Handle and Context objects. The Context object (C-Num = 2, C-Type = 1)

MUST have the R-Type (Request Type Flag) value set to 0x08 (Configuration Request) and M-Type set to zero. The Client-Handle object contains a number that MUST be chosen by the PEP. The only requirement imposed on this number is that a PEP MUST NOT use the same number for two different Requests on a single TCP connection. This completes the initialization sequence, which is visually depicted below.

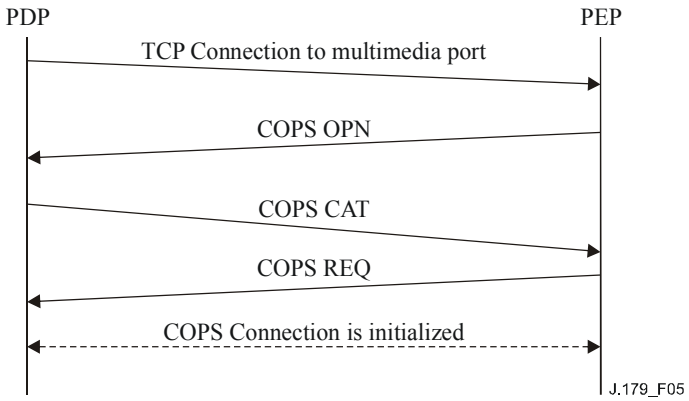

Figure 5/J.179 – COPS connection establishment

Periodically, the PEP MUST send a COPS Keep-Alive (KA) message to the PDP. Upon receipt of the COPS KA message, the PDP MUST echo a COPS KA message back to the PDP. This transaction is shown in Figure 6 and is fully documented in [7]. The PEP MUST send a Keep-Alive message at least as often as specified in the Keep-Alive-Timer object returned in the Client-Accept message. The Keep-Alive message MUST be sent with Client-Type set to zero and the solicited flag cleared.

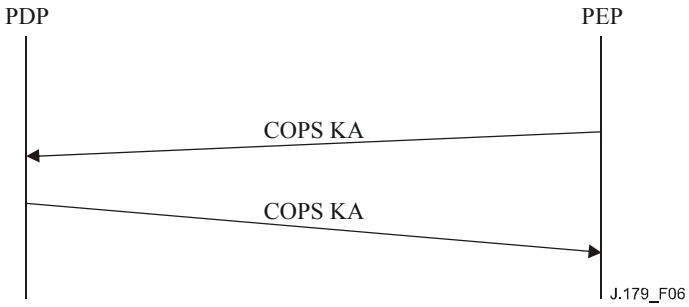

Figure 6/J.179 – COPS Keep-Alive exchange

6.5.2 Operation sequence

The protocol between the PDP and PEP is used for purposes of resource control and resource allocation policy. The Application Manager requests policy decisions from the Policy Server, and the Policy Server authorizes the requests and install them on the CMTS for enforcement through the use of Gates.

Messages that MAY be initiated by the Application Manager and Policy Server include Gate-Set, Gate-Info and Gate-Delete. The CMTS MAY initiate Gate-Report-State messages. The procedures for these messages are described in the following clauses. All messages from the PDP to the PEP MUST be sent using Client-Specific objects within the Decision object of a COPS Decision message. Solicited responses from the PEP MUST be sent as a Report-State message with Client-Specific objects in the ClientSI object, and the solicited flag MUST be set. Gate-Report-State

messages from the CMTS MUST be sent as unsolicited Report-State messages via Client-Specific objects in the ClientSI object.

The Decision messages and Report-State messages MUST contain the same Client-Handle as provided in the initial Request sent by the CMTS when the COPS connection was initiated.

Gate-Set initializes and modifies all the policy and traffic parameters for the Gate and establishes billing information. Gate-Set may also be used to control and update the state of a Gate on the CMTS.

Gate-Info is a mechanism by which the Policy Server may query all the current state and parameter settings of an existing Gate.

Gate-Delete allows a Policy Server to delete a specific Gate and any associated Service Flow.

Gate-Report-State allows the CMTS to inform the Policy Server that the Gate has transitioned into a new state. Gate-Report-State messages MUST be generated when the state transition happens asynchronously (i.e., not as a response to a Gate-Set message). Gate-Report-State messages MUST NOT be generated when the state transition happens synchronously.

The PEP MUST periodically send a Keep-Alive (KA) message to the PDP to facilitate the detection of TCP connection failures. The PDP MUST keep track of when KAs are received. If the PDP has not received a KA from the PDP in the time interval specified in [7] or the PDP has not received an error indication from the TCP connection, then the PDP MUST tear down the TCP connection and attempt to re-establish the TCP connection.

The following rules are used to route Gate Control messages through the IPCablecom Multimedia framework. In particular, provisions are provided for passing Gate Control messages forward (i.e., AM-to-PS-to-CMTS) and backward (i.e., CMTS-to-PS-to-AM) through a complex layered network in which multiple instances of each element are interacting with elements in the adjoining layer(s).

As described in 6.4.3.1, each Gate Control request that is initiated by an AM (i.e., Gate-Set, Gate-Info, and Gate-Delete) MUST include (in addition to other mandatory objects) both AMID and SubscriberID objects.

Upon receipt of a Gate Control message from an AM, a PS will apply any provisioned policy rules and determine whether to admit or reject the request. If the request is successfully admitted, the PS MUST route the message to the appropriate CMTS based on the SubscriberID included in the message. This SubscriberID-to-CMTS mapping MAY be performed dynamically based on a query to the OSS infrastructure or MAY reflect pre-provisioned routing information related to the range(s) of IP subnets that are associated with each CMTS.

If a Gate Control request is rejected by the PS, an error response MUST be returned to the issuing AM over the connection on which the original request was received. If a failure is detected on this connection between the time that a request is received and the response is delivered, the PS MUST discard the response.

Upon receiving a Gate Control message from a PS, a CMTS will execute the requested operation. If this operation is successful involving either a Gate-Set or Gate-Info operation, the CMTS MUST record the AMID and SubscriberID included in the message and maintain an association with the referenced Gate. This information must be used to ensure that only the AM which originally created the Gate is allowed to query or modify it. Any Gate Control messages that reference a Gate but which contain an AMID other than the one associated with the Gate MUST be rejected by the CMTS with error "Unauthorized AMID". Finally, Gate-State-Report messages MUST be delivered to the PS element, identified through its IP address, that originally created the Gate. If a connection to this PS is not available, then the CMST MUST suppress Gate-Report-State messages.

When a PS receives a Gate-State-Report message from a CMTS, the PS MUST forward this message to the AM associated with the AMID included in the message. In order to maintain a level of abstraction between non-adjacent layers and to hide information related to network topology from the AM layer, the PS MUST NOT include information directly identifying a particular CMTS to the AM layer.

6.5.3 Procedures for validating resource envelopes

The set of data service flow characteristics that are important for the purposes of providing enhanced Quality of Service is known as the envelope. An IPCablecom Multimedia Gate contains up to three envelopes – one that indicates the Authorized resources, one that indicates the Reserved resources, and one that indicates the Committed resources for the service flow corresponding to the Gate. At any one point in time, the Committed envelope MUST fit within the Reserved Envelope which MUST fit within the Authorized envelope.

When a CMTS receives a Gate-Set message, it MUST validate the relation between the Committed, Reserved, and Authorized envelopes of the Gate. If the envelope relation is invalid, the CMTS MUST reply with a Gate-Set-Err message with an IPCablecom Error-Code of "Incompatible Envelope".

The CMTS MUST also perform admission control whenever a change (including an addition) of the reserved envelope is requested. Admission control is the process of assigning resources for the flow corresponding to the gate. If the resources cannot be assigned, the CMTS MUST reply with a Gate-Set-Err message with an IPCablecom Error-Code of "Insufficient Resources".

6.5.3.1 FlowSpec

In Table 2, the second column indicates the operation that should be used to compare a parameter of A's envelope to a corresponding parameter in B's envelope. In other words, envelope A fits within envelope B if each of A's parameters meets the criteria specified in the table.

Table 2/J.179 – Envelope comparison rules

Parameter	A {OP} B
Token Bucket Rate [r]	≤
Token Bucket Size [b]	≤
Peak Data Rate [p]	≤
Minimum Policed Unit [m]	≥
Maximum Packet Size [M]	≤
Rate [R]	≤
Slack Term [S]	≥

6.5.3.2 DOCSIS Service Class Name

For traffic profiles in the form of a Service Class Name, the Service Class Name string MUST exactly match the preexisting Service Class Name on the CMTS. No envelope comparison is necessary as all three envelopes must share the same envelope parameters.

6.5.3.3 DOCSIS service flow parameters

6.3.3.3.1 Upstream encodings

In Table 3, the second column indicates the operation that should be used to compare a parameter of A's envelope to a corresponding parameter in B's envelope. In other words, envelope A fits within envelope B if each of A's parameters meets the criteria specified in the table.

Table 3/J.179 – Upstream envelope comparison

Parameter	A {OP} B
Traffic Priority (BE & NRTPS)	≤
Request Transmission Policy (all)	==
Maximum Sustained Traffic Rate (BE, NRTPS, RTPS)	≤
Maximum Traffic Burst (BE, NRTPS, RTPS)	≤
Minimum Reserved Traffic Rate (BE, NRTPS, RTPS)	≤
Assumed Minimum Reserved Traffic Rate Packet Size (BE, NRTPS, RTPS)	≥
Nominal Polling Interval (NRTPS, RTPS, UGS/AD)	See interval description below
Tolerated Poll Jitter (RTPS, UGS/AD)	≥
Unsolicited Grant Size (UGS & UGS/AD)	≤
Grants per Interval (UGS & UGS/AD)	≤
Nominal Grant Interval (UGS & UGS/AD)	See interval description below
Tolerated Grant Jitter (UGS & UGS/AD)	≥

Intervals – A is a subset of B if the parameter in A is an integer multiple of the same parameter in B.

6.5.3.3.2 Downstream encodings

In Table 4, the second column indicates the operation that should be used to compare a parameter of A's envelope to a corresponding parameter in B's envelope. In other words, envelope A fits within envelope B if each of A's parameters meets the criteria specified in the table.

Table 4/J.179 – Downstream envelope comparison

Parameter	A {OP} B
Traffic Priority	≤
Maximum Sustained Traffic Rate	≤
Maximum Traffic Burst	≤
Minimum Reserved Traffic Rate	≤
Assumed Minimum Reserved Traffic Rate Packet Size	≥
Maximum Downstream Latency	≥

6.5.4 Procedures for authorizing resources through a gate

The Gate-Set message MAY be sent by the PDP to the PEP to initialize or modify the operational parameters of a Gate. Figure 7 below provides an example of Gate-Set signalling.

NOTE – As an example, the "Start Session" message can be used to indicate to the Client that the resources have been authorized.

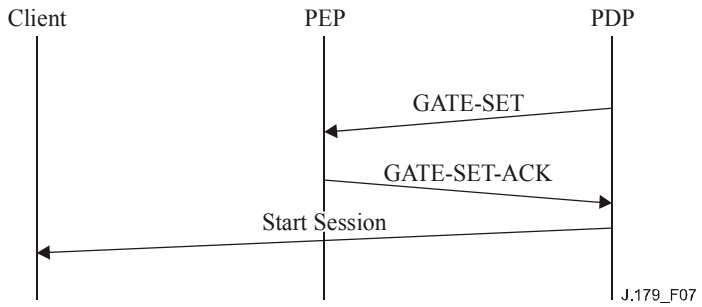

Figure 7/J.179 – Sample signalling of Gate-Set

If a GateID object is present in the Gate-Set message, then the request is to modify an existing Gate. If the GateID object is missing from the Gate-Set message, then it is a request to allocate a new Gate. The Gate-Set message MUST contain exactly one GateSpec object, describing one upstream or downstream Gate.

The Gate-Set message also contains the SubscriberID. The CMTS MUST use this IP address (i.e., the SubscriberID) to determine the servicing CM and MUST use the MAC address of the CM for subsequent MAC-layer messaging.

The PEP MUST respond to a Gate-Set message with either a Gate-Set-Ack, indicating success, or a Gate-Set-Err, indicating failure. The TransactionID in the response MUST match the TransactionID in the request. Errors in allocating or authorizing Gates MUST be reported by a Gate-Set-Err response. Refer to 6.4.2.14.

In Scenario 1, the Policy Server MAY specify the Authorized, Reserved and Committed Envelopes via a Traffic Profile sent in the Gate-Set message. It MAY simultaneously instruct the CMTS to authorize, reserve and commit resources.

Upon the receipt of a Gate-Set, the CMTS must first meet the requirements specified in 6.5.3, and then perform the requested actions. Upon successful completion of the actions requested in the Gate-Set (e.g., creation of a DOCSIS Service Flow), the CMTS MUST respond with a Gate-Set-Ack. The CMTS MUST NOT respond with a Gate-Set-Ack until it has completed sufficient steps to ensure that any subsequent requests to Admit or Commit the Gate will not fail due to a lack of resources.

A CMTS MAY perform complex authorization based not only on the requested QoS and the Gate's authorized FlowSpec, but also based on the SessionClassID specified in the GateSpec. The CMTS MAY have provisioned policies that define the amount of resources allocated exclusively to the particular Session Class, as well as 'borrow' and 'preemption' rules that apply to the use of the resources. The specifics of these types of policies and rules on the CMTS are out of scope for this Recommendation.

Upon receipt of a Gate-Set-Ack or Gate-Set-Err from a CMTS, the Policy Server MUST forward the message to the Application Manager corresponding to the AMID in the Gate-Set-Ack. The Policy Server MUST NOT transmit a Gate-Set-Ack to an Application Manager prior to receiving a Gate-Set-Ack from the CMTS. If the Application Manager requests a service that does not pass the Policy Server's policy checks, however, the Policy Server MUST NOT send the Gate-Set to the CMTS and MUST send a Gate-Set-Err to the Application Manager with the appropriate errors set.

6.5.5 Procedures for querying a Gate

When a Policy Server or Application Manager wishes to query the current parameter settings of a Gate, it sends to the CMTS a Gate-Info message. The CMTS MUST respond to a Gate-Info message with either a Gate-Info-Ack, indicating success, or a Gate-Info-Err, indicating failure. A

Gate-Info-Ack MUST contain information on the Gate associated with the GateID in the Gate-Info message. If the Gate being queried has an existing Volume-Based and/or Time-Based Usage Limit, then the CMTS MUST include these objects in the Gate-Info-Ack. A PS or AM can utilize this information to recover Gate state information from the CMTS for policing, error recovery or other purpose. The TransactionID in the response MUST match the TransactionID in the request.

Errors in querying gates MUST be reported by a Gate-Info-Err response. The Error object in a Gate-Info-Err message MUST contain one of the following Error-Codes:

2 = Unknown GateID

127 = Other, Unspecified Error

6.5.6 Procedures for modifying a Gate

To modify the Traffic Profile associated with an existing Gate, an Application Manager MAY send a Gate-Set message with the GateID of the Gate to be modified and the new Traffic Profile. If the Gate-Set fails the Policy Server's checks, the Policy Server MUST send a Gate-Set-Err to the Application Manager and MUST NOT send a Gate-Set to the CMTS. However, if the Gate-Set passes the Policy Server's policy checks, the Policy Server MUST send the Gate-Set to the CMTS unmodified. The TransactionID in the Policy Server Gate-Set MUST match the TransactionID in the Application Manager Gate-Set.

Upon the receipt of a Gate-Set, the CMTS must first meet the requirements specified in 6.5.3 and then perform the requested actions. As with the creation of a new Gate, upon successful completion of the actions requested in the Gate-Set (e.g., modification of a DOCSIS service flow) the CMTS MUST respond with a Gate-Set-Ack. The CMTS MUST NOT respond with a Gate-Set-Ack until it has completed a sufficient number of steps to ensure that any subsequent requests to Admit or Commit the Gate will not fail due to lack of resources.

Upon receipt of a Gate-Set-Ack or Gate-Set-Err from the CMTS, the Policy Server MUST forward the response unmodified to the Application Manager.

To modify the Usage Limits associated with an existing Gate, an Application Manager MAY send a Gate-Set message with the GateID of the Gate to be modified. If the Traffic Profile in the Gate-Set is different from the Traffic Profile currently associated with the Gate, then the previous rules apply. In either case, if the Time-Based Usage Limit or Volume-Based Usage Limit of the flow is present, then the existing limits associated with this/these parameter(s) MUST be replaced with the new parameter(s) and any existing counters or timers MUST be reset. However, the absence of these parameters from a Gate-Set message indicates that even if the Traffic Profile for the Gate is being modified, the existing Time-Based or Volume-Based Usage Limit(s) of the Gate still apply. If these parameters are not present in a Gate-Set message, the existing limits MUST be maintained and their associated counters/timers MUST continue from their current value without resetting.

6.5.7 Procedures for supporting Usage Limits

The Application Manager, Policy Server and CMTS all have a role in enforcing Usage Limits. There are subtle differences between Time-Based and Volume-Based Limits so each of these is described separately.

6.5.7.1 Procedures upon reaching a Volume-Based Usage Limit

Since the CMTS is the only trusted IPCablecom Multimedia device in the packet path, it is the only device capable of accurately tracking the usage of individual Gates. Thus, the CMTS MUST track the usage of all Gates regardless of whether or not they have an associated Volume-Based Usage Limit. The CMTS MUST report the amount of data transferred via a Gate in all Gate-Info-Ack and all Gate-State-Report messages.

If the Gate has an associated Volume-Based Usage Limit when the amount of data that has traversed the Gate equals the Volume-Based Usage Limit, the CMTS MUST send a Gate-State-Report message with the Solicited bit set to 0. The Gate-State-Report message MUST include a Gate State object with the Reason set to 7 (Gate state unchanged, but volume limit reached). Upon receipt of a Gate-State-Report message, the Policy Server MUST forward the Gate-State-Report message to the Application Manager unmodified. Upon receipt of a Gate-State-Report message with the Reason set to 7, the Application Manager MUST respond by performing one of the following actions:

- Send a Gate-Set message with a new Volume-Based Usage Limit object, which the CMTS must use to 'restart' accounting for this Gate.
- Send a Gate-Set message with a Volume-Based Usage Limit set to 0 to disable the feature and allow the CMTS to continue giving service to the session.
- Close the Gate by issuing a Gate-Delete command.

6.5.7.2 Procedures upon reaching a Time-Based Usage Limit

While it is a desirable design goal to keep the Volume-Based and Time-Based Usage Limit procedures as similar as possible, the number of CMTS interrupts required to support enforcement of Time-Based Usage Limits by the CMTS makes this approach impossible. Thus, the Application Manager MUST enforce the Time-Based Usage Limit of the Gate. Upon receiving the Gate-Set-Ack for a Gate with a Time-Based Usage Limit, the AM MUST start an application timer. When the application timer is equal to the Time-Based Usage Limit, the Application Manager MUST respond by performing one of the following actions:

- send a Gate-Set message with a new Time-Based Usage Limit object and reset its application timer;
- send a Gate-Set message with a Time-Based Usage Limit set to 0 to disable the feature;
- close the Gate by issuing a Gate-Delete command.

NOTE – In some ways it makes more sense for the Application Manager to enforce Usage Limits, since the Time-Based Usage Limit and Volume-Based Usage Limit are a reflection of the service being offered and are the responsibility of the Service Control Domain. It is really the Volume-Based Usage Limit procedure which is unusual, but the CMTS is the only device that can accurately enforce this limit.

6.5.7.3 Resource and error recovery

While it is required that the Application Manager perform one of several actions when a Gate's Usage Limit has been reached, there is always the possibility that the Application Manager will not respond appropriately. In this case, the RKS will still be recording the usage of this Gate so this activity will still be billable, but in some instances it may be useful to recover the resources that are being used 'illegally' by the Application Manager. A Policy Server MAY glean the fact that the Volume-Based or Time-Based Usage Limit of a Gate has been exceeded based on the messages it is proxying between the AM and CMTS. Using the 'gleaning' technique implies that the Policy Server is stateful, but a Policy Server that is not stateful can still recover resources via a second technique described below.

Alternatively, a Policy Server MAY occasionally query the CMTS with a Gate-Info message. The response will contain any associated Volume-Based Usage Limit and Gate Usage Info (or Time-Based Usage Limit and Gate Time Info). The Policy Server can then compare these values. Regardless of how a Policy Server gains the knowledge that a Gate is over-limit, it MAY issue a Gate-Delete for over-limit Gates. Upon receipt of the Gate-Set-Ack (or Gate-Set-Err) from the CMTS, the Policy Server MUST send the message to the Application Manager unmodified except for the transactionID.

Similarly, while not required to recover resources from over-limit Gates, a CMTS MAY perform the same comparisons itself and MAY delete over-limit Gates. Additional requirements for this scenario are described in 6.5.8.

6.5.7.4 Tracking Time-Based and Volume-Based Usage Limits

IPCablecom Multimedia Gates can enter and leave the Committed State multiple times (to support, for example, a 'pause' function on a game or streaming media). Since a subscriber cannot transmit/receive data while the Gate is not in the Committed State, these periods should not be counted against them. For Volume-Based Limits this requirement has no effect – there are no packets that might be over counted since no packets can be sent if a Gate is not Committed. However, for Time-Based Usage Limits the CMTS MUST stop its Gate Time Info timer when the Gate is not in the Committed State. If the Gate is re-Committed without changes to the Time-Based Limit, the Gate Time Info timer MUST be restarted from its stopped count. If changes are made to the Time-Based Limit, the Gate Time Info timer MUST be reset to 0 and restarted when the Gate is re-Committed.

NOTE – The Application Manager is required to maintain a timer independent of the CMTS timer to enforce the Time-Based Usage Limit. Since this timer is separated from the CMTS itself, messaging delays could cause discrepancies between these two timers. For applications that require a high-degree of time accuracy, the AM MAY query the CMTS for its Gate Time Info object after it moves a Gate into or out of the Committed state.

6.5.8 Procedures for deleting a Gate

Normally, when a Multimedia session ends, the Application Manager tells the Policy Server that the session has ended, and the Policy Server in turn instructs the CMTS to remove the Gate via a Gate-Delete message. The CMTS MUST respond to a Gate-Delete message with a Gate-Delete-Ack, indicating success, or a Gate-Delete-Err, indicating failure. The TransactionID in the response MUST match the TransactionID in the request.

Errors in deleting Gates MUST be reported by a Gate-Delete-Err response. The Error object MUST contain one of the following Error-Codes:

2 = Unknown GateID

127 = Other, Unspecified Error

At the CMTS, if timer T1 or T2 expires, the Gate MUST be deleted. When a CMTS deletes a Gate without being solicited by the Policy Server, the CMTS MUST send a Gate-Report-State message (with the Solicited bit set to 0) to the Policy Server indicating that the Gate was deleted. If the T2 timer expires, the CMTS MUST delete the DOCSIS flow through DOCSIS mechanisms (i.e., a DSD message), then the CMTS MUST return the Gate to an Authorized state, restart the T1 timer, and issue a Gate-Report-State message (with the Solicited bit set to 0) to the PS informing of this state transition. Upon receipt of a Gate-Report-State message, the Policy Server MUST forward it unmodified to the Application Manager.

6.5.9 Procedure for committing a Gate

In Scenario 1, the Policy Server is responsible for committing a Gate through a Traffic Profile containing a Committed Envelope. The CMTS commits the Gate and activates the DOCSIS Service Flow using the parameters passed down to it by the Policy Server.

6.5.10 Termination sequence

When the PEP is shutting down its TCP connection to the PDP, it MAY first send a Delete-Request-State message (including the Handle object used in the initial Request message). The PEP MAY follow that with a Client-Close message. The PDP in response MUST automatically delete any state associated with the PEP when the TCP connection is terminated. When the PDP is going to shutdown, it SHOULD send a COPS Client-Close message to the PEP. In the COPS

Client-Close message, the PDP SHOULD NOT send the PDP redirect address object PDPRedirAddr. If the PEP receives a COPS Client-Close message from the PDP with a PDPRedirAddr object, the PDP MUST ignore the PDPRedirAddr while processing the COPS Client-Close message.

The PS and CMTS MUST NOT remove gates as a result of a failed COPS connection.

6.5.11 Procedures for State Synchronization

When a Policy Server wants to synchronize its state with that of a CMTS it MAY send a Synchronize-State-Request (SSQ) message. This SSQ MAY contain the Client-Handle of the Policy Server. If the optional Client-Handle is present, only state associated with this handle is synchronized. If the CMTS does not recognize the requested handle, it MUST immediately send a DRQ message to the Policy Server for the handle that was specified in the SSQ message. If no Client-Handle is specified in the SSQ message, all active state for clients with the IPCablecom Multimedia Client-Type MUST be synchronized with the PDP.

The CMTS performs state synchronization by issuing Request messages for Gates associated with the Client-Handle (if included in the SSQ) or for all known Gates (if no Client-Handle is provided). When synchronization is complete, the CMTS MUST issue a Synchronize-State-Complete (SSC) message to the PDP. If the initiating SSQ contained a Client-Handle, then the corresponding SSC MUST also contain the Client-Handle.

7 Event messaging interface description

7.1 Introduction

As in the IPCablecom-T architecture, event messages within IPCablecom Multimedia provide detailed information regarding QoS resource utilization, such as reservation, activation, and release. New for the IPCablecom Multimedia framework is the need to track status of policy decisions (requests, updates, deletions). Also, since use of network resources falls outside the profiles within IPCablecom-T (constant usage over time), there is the need to report volume-based and time-based usage information.

Event messages, as defined in this framework, are generated by network elements and stored on the Record Keeping Server (RKS). These EMs are then correlated by the RKS or other back-office system to record a single instance of a service. These records may be used to derive service billing information, network resource usage patterns, capacity planning, etc. EMs are not, however, intended for fault monitoring.

Currently, only the CMTS and Policy Server, which are part of the cable operator's network and considered trusted entities, generate EMs within the Multimedia framework. Other elements of the network, such as the various client types, are considered untrusted. In the case of the Application Manager, this element may or may not be part of the cable operator's network, and hence does not directly provide EMs to the RKS. The AM may, however, provide supplementary information as part of opaque data fields to the PS which would then be included in EMs generated by the PS.

IPCablecom event messages for Multimedia represent a simplification and modification of IPCablecom-T event messages. Telephony-specific events such as Call_Answer and Call_Disconnect are considered optional, as are telephony service-specific event messages (e.g., Service Instance y). The intent is to leverage existing EM implementations as much as possible while providing sufficient abstraction mechanisms to support general Multimedia services.

Specifically, of the fourteen EM message types defined in support of IPCablecom-T voice services, four will be required in IPCablecom Multimedia, including QoS_Reserve, QoS_Commit, QoS_Release, and Time_Change. Three new EM message types relating to policy decisions are

defined: Policy_Request, Policy_Delete, and Policy_Update. Table 5 below provides a summary overview of the IPCablecom Multimedia EM message types.

Table 5/J.179 – IPCablecom multimedia EM message types

Event message ID	Event message	Originating element	Description
7	QoS_Reserve	CMTS	Indicates the time at which the CMTS reserves bandwidth on the IPCablecom access network. The CMTS must also generate this event if the reserved bandwidth changes.
8	QoS_Release	CMTS	Indicates the time at which the CMTS released its bandwidth commitment on the IPCablecom access network.
17	Time_Change	PS, CMTS	Captures an instance of a time change. Whenever the (IPCablecom) clock on a trusted network element (PS, and CMTS) is changed by more than 200 milliseconds, the network element MUST generate a Time_Change message.
19	QoS_Commit	CMTS	Indicates the time at which the CMTS commits bandwidth on the IPCablecom access network. The CMTS must also generate this event if the committed bandwidth changes.
31	Policy_Request	PS	Indicates the time at which the Policy Server receives a new policy request from the AM.
32	Policy_Delete	PS	Indicates the time at which the Policy Server deletes a policy.
33	Policy_Update	PS	Indicates the time at which the Policy Server receives a request to update a policy.

Although IPCablecom Multimedia event messages are based upon IPCablecom-T, telephony specific events are optional for IPCablecom Multimedia and are listed below. For further details on these events and associated attributes, see the IPCablecom-T EM Recommendation [10].

Table 6/J.179 – IPCablecom-T telephony EM message types

Event message ID	Event message	Description
1	Signalling_Start	Indicates the time at which signalling starts.
2	Signalling_Stop	Indicates the time at which signalling terminates.
3	Database_Query	Indicates the time at which a one-time request/response transaction or database dip is completed by an intelligent peripheral (e.g., 800 number database, LNP database).
6	Service_Instance	Indicates the time at which the CMS provides an instance of a call control/feature service (e.g., call hold, call waiting).
9	Service_Activation	Indicates the time at which the CMS records an attempt to activate a service (e.g., call forwarding, call waiting).
10	Service_Deactivation	Indicates the time at which the CMS records an attempt to deactivate a service (e.g., call forwarding, call waiting).

Table 6/J.179 – IPCablecom-T telephony EM message types

Event message ID	Event message	Description
13	Interconnect_Start	Indicates the time at which the start of network interconnect signalling occurs.
14	Interconnect_Stop	Indicates the termination of bandwidth between the IPCablecom network and the PSTN.
15	Call_Answer	Indicates that the media connection is open because an answer event has occurred.
16	Call_Disconnect	Indicates the time at which the media connection is closed because the calling party has terminated the call by going on-hook, or that the destination party has gone on-hook and the called-party's call-continuation timer has expired.
20	Media_Alive	Indicates that service is active due to the continued existence of a bearer connection. This message may be generated by any trusted IPCablecom network element (CMS, MGC, and CMTS) as the vendor sees fit.

7.2 Record Keeping Server requirements

The Record Keeping Server (RKS) is a trusted network element function. The RKS is generally depicted in this Recommendation as a distinct standalone element, but this Recommendation does not preclude some other application from performing the functions of an RKS, providing that the application conforms to the requirements herein.

The RKS is the mediation layer between the IPCablecom Multimedia network and the back-office applications. The RKS is expected to process the data received from the IPCablecom Multimedia network and to present it to the back-office applications in the format and within the time constraints deemed necessary by the cable operator. The RKS therefore acts as a demarcation point between the IPCablecom network and the back-office applications.

The RKS MUST be capable of receiving and processing Event Messages formatted in accordance with this Recommendation.

The RADIUS messages inside which Event Messages are encapsulated are transported over UDP, which does not guarantee reliable delivery of messages; hence the request/response nature of the protocol defined herein. When an RKS receives and successfully records all the IPCablecom Event Messages contained in a RADIUS Accounting-Request message, it MUST transmit an Accounting-Response message to the client. The RKS MUST NOT transmit an Accounting-Response reply if it fails to record successfully all the Event Messages in a RADIUS Accounting-Request message.

The RKS SHOULD ignore Event Messages where the IPCablecom "Event Message type" is unrecognized. The RKS SHOULD also ignore IPCablecom event attributes where the Event Attribute ID is unrecognized.

7.3 General IPCablecom multimedia network element requirements

This clause lists requirements placed on the IPCablecom Multimedia network elements.

7.3.1 Element ID

Each IPCablecom network element that generates an Event Message MUST identify itself with a static, unique element ID. The Element ID is a statically configured element number, unique within an IPCablecom domain, which MUST be in the range 0 to 99,999.

7.3.2 Timing

It is important for elements that generate Event Messages to remain closely synchronized with one another and with a standard clock. The requirements in this clause ensure that such elements maintain this synchronization and report events with timestamps that are both accurate and precise.

Elements that generate Event Messages MUST use the Network Time Protocol as defined in [2]. Elements MUST operate in mode 3 (Client mode). The value of NTP.MAXPOLL MUST NOT exceed eleven, which corresponds to 2048 seconds.

Event Messages MUST include timestamps with a precision of one millisecond.

7.3.3 Primary and secondary RKS considerations

IPCablecom Multimedia supports an architecture that consists of a primary and secondary RKS. The secondary RKS is used as a fallback RKS when a network element (PS, CMTS) is unable to successfully send a message to the primary RKS. IPCablecom Multimedia network elements MUST support event message transport to a primary RKS and failover to a secondary RKS when communication with the primary RKS fails. Once a network element fails over to the secondary RKS, the secondary becomes the primary for the duration of that session or gate. The Policy Server is provisioned with primary and secondary RKSs as required for the applications it supports. The PS MUST provide the IP address and port of the primary RKS and optionally the secondary RKS to the CMTS in policy decision messages (Gate-Set). The PS MUST support multiple sets of primary and secondary RKSs.

In order to guarantee the reliable transfer of the data, the network elements should implement a user configurable retry time interval and the number of times the client needs to retransmit the event. The time interval should be configurable (suggested: 10 ms to 10 s), the number of retries should be configurable (suggested: 0 to 9). The number of retries should be attempted on both the primary RKS and secondary RKS. After exhausting the number of retries the event message should be written to an error file, and the event message can then be deleted from the network element.

If the IPCablecom network element does not receive an Accounting-Response within the configured retry interval, it MUST continue resending the Accounting-Request until it receives an Accounting-Response from an RKS or the maximum number of retries is reached. The IPCablecom network element MUST re-send the same Accounting Request to the primary RKS and if the retry limit is reached, re-send the same Accounting Request to the secondary RKS.

All Network Elements MUST store Event Messages until they have received an Acknowledgement (Accounting Response) from an RKS that the data was correctly received and stored, or until the maximum number of retries has been reached. Only when an Ack is received or the maximum retries reached are the network elements allowed to delete these Event Messages.

Once a Network Element succeeds in sending event messages to the secondary RKS, a failover to the secondary RKS should occur. This is a non-revertive failover, meaning that the secondary RKS becomes active, and is the new primary RKS. All subsequent event messages for the session should be sent to the now active secondary RKS. For all new sessions, the PS should instruct the CMTS to use the new active RKS as the primary (i.e., the previous secondary RKS becomes the new primary for subsequent session). Note that it is possible under certain circumstances that one element, PS or CMTS, may be able to communicate with the primary RKS while the other element may not for the same session. In cases such as this it is expected that the RKS be able to reconcile event messages between the primary and secondary RKS.

7.4 Event Messages for IPCablecom Multimedia

This clause provides a detailed description and definition of each of the Event Messages defined for IPCablecom Multimedia.

7.4.1 Policy Events

The Policy Event Messages are new for IPCablecom Multimedia. They indicate the time at which the Policy Server receives a request for a policy action and serve to bracket the ensuing set of Event Messages for any resource usage associated with the various instances of a service. Policy event messages are used to indicate the initial policy request, an update to the policy, and the deletion of a policy.

The PS MUST timestamp the Policy Event Messages upon receiving a policy request message from the AM. Immediately upon receiving an initial policy request, the PS MUST create a Billing Correlation ID (BCID). Each generated BCID MUST conform to the Billing Correlation ID (BCID) Attribute Structure format requirements in Table 17.

The PS MUST include the BCID in the EM header for all subsequently generated Policy Event Messages associated with this request. If the request is approved, the PS MUST include the BCID in the Gate-Set-Ack message sent to the AM (to acknowledge the request). Also, the PS MUST include the BCID in the Gate-Set message sent to the CMTS.

The PS MUST generate policy event messages immediately after determining the result of a policy request. The result may be based upon PS internal authorization and admission control mechanisms or, upon receiving a response to its Gate-Set and Gate-Delete messages from the CMTS. The PS creates a timestamp for an event message when it receives a request from the AM but does not generate the event until it knows the outcome of the request.

7.4.1.1 Policy_Request

The Policy Server MUST send a Policy_Request event message to the RKS if a request to create a new policy is received. The PS MUST set the Policy_Decision_Status to either Approved (1) or Declined (2), based upon the outcome of authorization and admission control.

NOTE – Because the PS does not send the Policy_Request event message until after the CMTS responds to the Gate-Set message, it is possible that QoS event messages from the CMTS may arrive at the RKS prior to a Policy-Request event message.

Table 7/J.179 – Policy_Request event message

Attribute name	Required or optional	Comment
Event_Message_Header	R	See Table 16
Application_Manager_ID	R	Contains the network wide unique identifier of the AM
Subscriber_ID	R	Subscriber IPv4 address
Policy_Decision_Status	R	1 – Policy Approved 2 – Policy Denied
Policy_Denied_Reason	O	Required when Policy_Decision_Status = 2 (Policy Denied) 1 – Policy Server admission control failure 2 – Insufficient resources 3 – Unknown subscriber 127 – Other
FEID	R	Financial Entity ID. Identifies the paying entity. Supplied by the PS.

Table 7/J.179 – Policy_Request event message

Attribute name	Required or optional	Comment
AM_Opaque_Data	O	If the Application Manager includes this object (ClientSI: Opaque-Data) in the "Policy Request" (COPS DEC), then the Policy Server MUST include this in the Policy-Event Event Message.
Volume_Usage_Limit	O	If the Application manager includes this object (ClientSI: Volume-Based-Usage-Limit) in the "Policy Request" (COPS DEC), then the Policy Server MUST include this in the Policy-Event Event Message.
Time_Usage_Limit	O	If the Application manager includes this object (ClientSI: Time-Based-Usage-Limit) in the "Policy Request" (COPS DEC), then the Policy Server MUST include this in the Policy-Event Event Message.

7.4.1.2 Policy_Delete

The Policy Server MUST send a Policy_Delete Event Message to the RKS when it receives a Gate-Delete from the AM indicating that the resources are no longer needed for a session, a Gate-Delete-Ack from the CMTS in response to a PS initiated Gate-Delete, or a Gate-State-Report from the CMTS indicating that the resources are no longer available for a session. The PS MUST always generate a Policy_Delete Event Message to close a session if it has previously generated a Policy-Request Event Message to open the session.

Table 8/J.179 – Policy_Delete event message

Attribute name	Required or optional	Comment
Event_Message_Header	R	See Table 16
Application_Manager_ID	R	Contains the network wide unique identifier of the AM
Subscriber_ID	R	Subscriber IPv4 address
Policy_Deleted_Reason	R	1 – Application Manager request 2 – CMTS decision 127 – Other
FEID	R	Financial Entity ID. Identifies the paying entity. Supplied by the PS.
AM_Opaque_Data	O	If the Application Manager includes this object (ClientSI: Opaque-Data) in the "Policy Request" (COPS DEC), then the Policy Server MUST include this in the Policy-Event Event Message.

7.4.1.3 Policy_Update

The Policy Server MUST send a PolicyUpdate Event Message to the RKS if a request to change the traffic profile, classifier, volume limit, time-limit, or opaque data of a gate is received from the AM.

Table 9/J.179 – Policy_Update event message

Attribute name	Required or optional	Comment
Event_Message_Header	R	See Table 16
Application_Manager_ID	R	Contains the network wide unique identifier of the AM
SubscriberID	R	SubscriberID
Policy_Decision_Status	R	1 – Policy Approved 2 – Policy Denied
Policy_Denied_Reason	O	Required when Policy_Decision_Status = 2 (Policy Denied) 1 – Policy Server admission control failure 2 – Insufficient resources 3 – Unknown subscriber 4 – Unauthorized AMID 5 – Undefined Service Class Name 6 – Incompatible Envelope 127 – Other
Policy_Update_Reason	R	1 – Traffic Profile 2 – Classifier 3 – Volume Limit 4 – Time-Limit 5 – Opaque data 6 – Multiple Updates (combination of 1-5) 127 – Other
FEID	R	Financial Entity ID. Identifies the paying entity. Supplied by the PS.
AM_Opaque_Data	O	If the Application Manager includes this object (ClientSI: Opaque-Data) in the "Policy Request" (COPS DEC), then the Policy Server MUST include this in the Policy-Event Event Message.
Volume_Usage_Limit	O	If the Application Manager includes this object (ClientSI: Volume-Based-Usage-Limit) in the "Policy Request" (COPS DEC), then the Policy Server MUST include this in the Policy-Event Event Message.
Time_Usage_Limit	O	If the Application Manager includes this object (ClientSI: Time-Based-Usage-Limit) in the "Policy Request" (COPS DEC), then the Policy Server MUST include this in the Policy-Event Event Message.

7.4.2 QoS_Reserve

This Event Message indicates the time at which the CMTS reserves bandwidth on the IPCablecom access Network. The CMTS MUST also generate this event if the Reserved bandwidth changes.

The CMTS MUST timestamp this message immediately upon transmission of a DSA-ACK or DSC-ACK acknowledging a successful DSA-RSP or DSC-RSP to the CM which completes a transaction reserving resources.

If the DSA-RSP or DSC-RSP confirmation code from the CM is not successful, the CMTS MUST NOT generate this message.

Table 10/J.179 – QoS_Reserve event message

Attribute name	Required or Optional	Comment
Event_Message_Header	R	See Table 16
QoS_Descriptor	R	None
SF_ID	R	None
Flow_Direction	R	None
Element_Requesting_QoS	R	0 = Client 1 = Policy Server 2 = Embedded Client

7.4.3 QoS_Commit

The QoS_Commit Event Message indicates the time at which the CMTS commits bandwidth on the IPCablecom access Network. The CMTS MUST also generate this event if the Committed bandwidth changes.

The CMTS MUST timestamp this message immediately upon transmission of a DSA-ACK or DSC-ACK acknowledging a successful DSA-RSP or DSC-RSP to the CM which completes a transaction committing resources.

If the DSA-RSP or DSC-RSP confirmation code from the CM is not successful, the CMTS MUST NOT generate this message.

Table 11/J.179 – QoS_Commit event message

Attribute name	Required or optional	Comment
Event_Message_Header	R	See Table 16
QoS_Descriptor	R	None
SF_ID	R	None
Flow_Direction	R	None

7.4.4 QoS_Release

The QoS_Release Event Message indicates the time at which the CMTS releases its reservation and/or bandwidth commitment on the IPCablecom access network.

The CMTS MUST timestamp this message immediately upon Transmission of a DSD-REQ that indicates the request to delete bandwidth.

Table 12/J.179 – QoS_Release event message

Attribute name	Required or optional	Comment
Event_Message_Header	R	See Table 16
SF_ID	R	None
Flow_Direction	R	None
QoS_Release_Reason	R	1 – Gate Closed by PS 2 – Inactivity timer expired 3 – CM Failure 4 – Pre-Empted 5 – RSVP PathTear request 6 – CM request 7 – Admitted (T2) timer expiration 127 – Other
Gate_Usage_Info	R	None
Gate_Time_Info	R	None

7.4.5 Time_Change

This event captures an instance of a time change. Whenever the (IPCablecom) clock on the network element (PS or CMTS) is changed by more than 200 milliseconds, the network element MUST generate a Time Change message. This includes time shift events (Daylight savings time), step adjustments to synchronize with the NTP reference clock and manual time setting changes. The Event_Time attribute in the Event Message header MUST reflect the new (adjusted) notion of time. Note that Time_Change message is not required for slew adjustments performed by NTP.

The network element (PS and CMTS) MUST send the Time_Change event message to the active (current primary) RKS. The Time_Change event message MUST be generated when a gate(s) is currently present in the CMTS. The Time_Change event message need not be generated when there are no gates in the CMTS. Only one Time_Change event message is sent to each primary RKS regardless of how many gates may exist on the CMTS. In other words, if the CMTS has several gates all of which point to the same RKS, then only one Time_Change event message should be sent to that RKS.

The BCID in the Event_Message_Header of the Time_Change event message MUST be generated locally by the network element at the time of the event. The BCID is not associated with any session related BCID, it is a unique BCID for this event.

Table 13/J.179 – Time_Change event message

Attribute name	Required or optional	Comment
Event_Message_Header	R	See Table 16
Time_Adjustment	R	None

7.5 Event messaging attributes for IPCablecom multimedia

This clause describes and defines the IPCablecom attributes included in the IPCablecom Event Messages.

It provides a mapping between each of the IPCablecom Event Messages and their associated attributes. Table 14 offers a detailed description of each of these attributes.

Table 14/J.179 – IPCablecom attributes mapped to IPCablecom MM event messages

EM attribute ID	EM attribute name	7 – QoS_Reserve	8 – QoS_Release	17 – Time_Change	19 – QoS_Commit	31 – Policy_Request	32 – Policy_Delete	33 – Policy_Update
1	Event_Message_Header	X	X	X	X	X	X	X
30	SF_ID	X	X		X			
32	QoS_Descriptor	X			X			
38	Time_Adjustment			X				
49	FEID					X	X	X
50	Flow_Direction	X	X		X			
51	AM_Opaque_Data					X	X	X
52	Subscriber_ID					X	X	X
53	Volume_Usage_Limit					X		X
54	Gate_Usage_Info		X					
55	Element_Requesting_QoS	X						
56	QoS_Release_Reason		X					
57	Policy_Denied_Reason					X		X
58	Policy_Deleted_Reason						X	
59	Policy_Update_Reason							X
60	Policy_Decision_Status					X		X
61	Application_Manager_ID					X	X	X
62	Time_Usage_Limit					X		X
63	Gate_Time_Info		X					

Table 15 provides a detailed definition of each of the IPCablecom Event Message attributes. A data value of an attribute may either be represented by a simple data format (one data field) or by a more complex data structure.

Table 15/J.179 – IPCablecom MM Event Message attributes

EM attribute ID	EM attribute length	EM attribute name	EM attribute value type	Attribute data description
1	76 bytes	EM_Header	Data structure See Table 16	Common data required on every IPCablecom Event Message
30	4 bytes	SF_ID	Unsigned integer	Service Flow ID, a 32-bit integer assigned by the CMTS to each DOCSIS Service Flow defined within a DOCSIS RF MAC domain. SFIDs are considered to be in either the upstream direction (USFID) or downstream direction (DSFID). USFIDs and DSFIDs are allocated from the same SFID number space.
32	Variable; Min 8 bytes	QoS_Descriptor	Data structure See Table 19	QoS parameters data
38	8 bytes	Time_Adjustment	Signed integer	Time adjustment of an element's (PS, CMTS) clock. This time is in milliseconds, detailing the amount of the time change.
49	Variable length, maximum of 247 bytes	FEID	ASCII character string.	Financial Entity ID. The first 8 bytes constitute cable operator defined data. By default, the first 8 bytes are zero filled. From the 9th byte on the field contains the cable operator's domain name which uniquely identifies the cable operator for billing and settlement purposes. The cable operator's domain name is limited to 239 bytes.
50	2 bytes	Flow Direction	Unsigned integer	Flow direction: 0 = Reserved 1 = Upstream 2 = Downstream
51	8 bytes	AM_Opaque_Data	Unsigned integer	Opaque data passed from Application Manager.
52	4 bytes	Subscriber_ID	Unsigned integer	Four concatenated byte values representing an IPv4 address
53	8 bytes	Volume_Usage_ Limit	Unsigned integer	Volume limit in octets set by the AM
54	8 bytes	Gate_Usage_Info	Unsigned integer	The number of octets transmitted on the DOCSIS RF network from the byte after the MAC header HCS to the end of the CRC

Table 15/J.179 – IPCablecom MM Event Message attributes

EM attribute ID	EM attribute length	EM attribute name	EM attribute value type	Attribute data description
55	2 bytes	Element_ Requesting_QoS	Unsigned integer	0 = Client 1 = Policy Server 2 = Embedded Client
56	2 bytes	QoS_Release_ Reason	Unsigned integer	1 – Gate Closed by PS 2 – Inactivity timer expired 3 – CM Failure 4 – Pre-Empted 5 – RSVP PathTear request 6 – CM request 7 – Admitted (T2) timer expiration 127 – Other
57	2 bytes	Policy_Denied_ Reason	Unsigned integer	1 – Policy Server admission control failure 2 – Insufficient resources 3 – Unknown subscriber 4 – Unauthorized AMID 5 – Undefined Service Class Name 6 – Incompatible Envelope 127 – Other
58	2 bytes	Policy_Deleted_ Reason	Unsigned integer	1 – Application Manager request 2 – CMTS decision 127 – Other
59	2 bytes	Policy_Update_ Reason	Unsigned integer	1 – Traffic Profile 2 – Classifier 3 – Volume Limit 4 – Time Limit 5 – Opaque data 6 – Multiple Updates (combination of 1-5) 127 – Other
60	2 bytes	Policy_Decision_ Status	Unsigned integer	1 – Policy Approved 2 – Policy Denied
61	4 bytes	Application_ Manager_ID	Unsigned integer	Network wide unique identifier assigned to the Application Manager.
62	4 bytes	Time_Usage_ Limit	Unsigned integer	Time limit in seconds set by the AM
63	4 bytes	Gate_Time_Info	Unsigned integer	The number of seconds a gate has been in the committed state

7.5.1 EM_Header attribute structure

Table 16 contains a detailed description of the fields in the EM_Header attribute structure. This Event Message Header attribute MUST be the first attribute in every IPCablecom Event Message.

Table 16/J.179 – EM_Header attribute structure

Field name	Semantics	Value type	Length
Version ID	Identifies version of this EM Header structure. 1 = IPCablecom 1.0 2 = IPCablecom 1.1 3 = IPCablecom Multimedia	Unsigned integer	2 bytes
BCID	Unique identifier for a transaction within a network.	Data Structure See Table 17	24 bytes
Event Message Type	Identifies the type of Event Message.	Unsigned integer	2 bytes
Element Type	Identifies Type of Originating Element: 0 = Reserved 1 = Reserved 2 = CMTS 3 = Reserved 4 = Policy Server	Unsigned integer	2 bytes
Element ID	Network wide unique identifier Five digits (statically configured element number unique within an IPCablecom domain in the range of 0-99,999)	Right justified, space padded ASCII Character String	8 bytes
Time Zone	Identifies daylight savings time and offset from universal time (UTC). Daylight Savings Time: 0 = Standard Time 1 = Daylight Savings UTC offset + HHMMSS The offset is reported from the network element (PS, CMTS) point of view; not based on the subscriber point of view.	ASCII character string	 1 byte 7 bytes
Sequence Number	Each network element MUST assign a unique and monotonically increasing unsigned integer for each Event Message sent to a given RKS. For the purpose of this Recommendation, monotonically increasing is to be interpreted as increasing by 1. This is used by the RKS to determine if Event Messages are missing from a given network element.	Unsigned integer	4 bytes
Event_time	Event generation time and date. Millisecond granularity. Format: yyyymmddhhmmss.mmm	ASCII character string	18 bytes
Status	Status indicators	See Table 18	4 bytes

Table 16/J.179 – EM_Header attribute structure

Field name	Semantics	Value type	Length
Priority	Indicates the importance to assign relative to other event messages. 255 = highest priority 0 = lowest priority 128 = default.	Unsigned integer	1 byte
Attribute Count	Indicates the number of attributes that follow (or are appended to) this header in the current Event Message	Unsigned integer	2 bytes
Event Object	This is a "place holder" for future IPCablecom releases to allow for a grouping of services. It may be IPCablecom Voice, IPCablecom Video, etc. or it could be IPCablecom, DOCSIS, etc. It MUST have a value of zero for IPCablecom Release 1.0 and IPCablecom Multimedia.	Unsigned integer	1 byte

7.5.2 Billing Correlation ID (BCID) Attribute Structure

Table 17 describes the Billing Correlation ID (BCID). The RKS, or some other back-office application, uses the BCID to correlate Event Messages that are generated for a single transaction. It is one of the fields in the Event Message Header. The BCID is unique for each Transaction in the network. All Event Messages from the same network element with the same BCID MUST be sent to the same primary RKS except in failover circumstances in which case the Event Messages MUST be sent to secondary RKS.

Table 17/J.179 – BCID description

Field name	Semantics	Value type	Length
Timestamp	High-order 32 bits of NTP time reference	Unsigned integer	4 bytes
Element_ID	Network wide unique identifier Five digits (statically configured element number unique within an IPCablecom domain in the range of 0-99,999)	Right justified, space padded ASCII character string	8 bytes
Time Zone	Identifies daylight savings time and offset from universal time (UTC). Daylight Savings Time: 0 = Standard Time 1 = Daylight Savings UTC offset: ± HHMMSS The offset is reported from the network element (PS, CMTS) point of view; not based on the subscriber point of view.	ASCII character string	1 byte 7 bytes
Event Counter	Monotonically increasing for each Transaction	Unsigned integer	4 bytes

7.5.3 Status field attribute structure

The Status field of the Event Message Header is a 32-bit mask. Bit 0 is the low-order bit; the field is treated as a 4 byte unsigned integer. Table 18 presents Status Field description.

Table 18/J.179 – Status field description

Start bit	Semantics	Bit count
0	Error Indicator: 0 = No Error 1 = Possible Error 2 = Known Error 3 = Reserved	2
2	Event Origin: 0 = Trusted Element 1 = Untrusted Element	1
3	Event Message Proxied: 0 = Not proxied, all data known by sending element 1 = proxied, data sent by a trusted element on behalf of an untrusted element	1
4	Reserved. IPCablecom 1.0 value MUST be 0.	28

7.5.4 QoS descriptor attribute structure

Table 19 describes the QoS Descriptor Data Structure.

Table 19/J.179 – QoS descriptor data structure

Field name	Semantics	Value type	Length
Status_Bitmask	Bitmask describing structure contents. (See Table 20.)	Bit map	4 bytes
Service_Class_Name	Service profile name	Right justified, space padded ASCII character string	16 bytes
QoS_Parameter_Array	QoS Parameters. Contents determined by Status Bitmask.	Unsigned integer array	Variable length array of 32-bit unsigned integers

Table 20 describes the QoS Status Bitmask field of the QoS Descriptor attribute. Bits 2-17 describe the contents of the QoS_Parameter_Array. Each of these bits indicates the presence (bit = 1) or absence (bit = 0) of the named QoS parameter in the array. The location of a particular QoS parameter in the array matches the order in which that parameter's bit is encountered in the bitmask, starting from the low-order bit.

Each QoS parameter present in the QoS_Parameter_Array must occupy four bytes. The definition and encoding of the QoS parameters can be found in Appendix C of the DOCSIS RFI Recommendation [1]. QoS parameters whose definition specifies less than four bytes must be right-justified (where the 4 bytes are to be treated as an unsigned integer) in the four bytes allocated for the array element.

Table 20/J.179 – QoS status bitmask

Start bit	Semantics	Bit count
0	State Indication: 0 = Illegal Value 1 = Resource Reserved but not Activated 2 = Illegal Value 3 = Resource Reserved & Activated	2
2	Service Flow Scheduling Type	1
3	Nominal Grant Interval	1
4	Tolerated Grant Jitter	1
5	Grants Per Interval	1
6	Unsolicited Grant Size	1
7	Traffic Priority	1
8	Maximum Sustained Rate	1
9	Maximum Traffic Burst	1
10	Minimum Reserved Traffic Rate	1
11	Minimum Packet Size	1
12	Maximum Concatenated Burst	1
13	Request/Transmission Policy	1
14	Nominal Polling Interval	1
15	Tolerated Poll Jitter	1
16	IP Type of Service Override	1
17	Maximum Downstream Latency	1

7.6 RADIUS accounting protocol

This clause specifies the protocol used between the IPCablecom network elements that generate Event Messages (PS, CMTS) and the Record Keeping Server (RKS). These network elements MUST support RADIUS Accounting (RFC 2866) [8] with IPCablecom extensions as defined in this Recommendation.

The RADIUS Accounting protocol is a client/server protocol that consists of two message types: Accounting-Request and Accounting-Response. IPCablecom network elements that generate Event Messages are RADIUS clients that send Accounting-Request messages to the RKS. The RKS is a RADIUS server that sends Accounting-Response messages back to the IPCablecom network elements indicating that it has successfully received and stored the Event Message.

The Event Messages are formatted as RADIUS Accounting-Request and Accounting-Response packets as specified in [8].

7.6.1 Authentication and confidentiality

Refer to clause 8 for details concerning the use of IPsec to provide both authentication and confidentiality of the RADIUS messages, and the details of the correct usage of the RADIUS shared secret.

7.6.2 Standard RADIUS attributes

Each RADIUS message starts with the standard RADIUS header shown in Table 21.

Table 21/J.179 – RADIUS message header

Field name	Semantics	Field length
Code	Accounting-Request = 4 Accounting-Response = 5	1 byte
Identifier	Used to match accounting-request and accounting-response messages.	1 byte
Length	Total length of RADIUS message min value = 20 max value = 4096	2 bytes
Authenticator	Computed as per RADIUS Specification	16 bytes

Two standard RADIUS attributes MUST follow the RADIUS Message Header: NAS-IP-Address and Acct_Status_Type. These two fields are included to improve interoperability with existing RADIUS server implementations since they are mandatory attributes in a RADIUS Accounting-Request packet.

The NAS-IP-Address indicates the originator of the Accounting-Request message and MUST contain the IP address of the originating IPCablecom network element.

The Acct-Status-Type attribute typically indicates whether the Accounting-Request marks the beginning of the user service (Start) or the end (Stop). An IPCablecom Accounting-Request message may contain the beginning, end, or update of the user service. For this reason, an Acct-Status-Type value of Interim-Update is used to represent IPCablecom Event Messages.

Table 22/J.179 – Mandatory RADIUS attributes

Name	Type	Length	Value
NAS-IP-Address	4	6	IP address of originating IPCablecom network element
Acct-Status-Type	40	6	Interim-Update = 3

Table 23/J.179 – RADIUS Acct_Status_Type

Type	Length	Value
40	6 bytes	Interim-Update = 3

IPCablecom attributes are encoded in the RADIUS Vendor-Specific Attributes (VSA) structure as described in this clause. Additional IPCablecom or vendor-specific attributes can be added to existing Event Messages by adding additional RADIUS VSAs to the message.

The Vendor-Specific attribute includes a field to identify the vendor, and the Internet Assigned Numbers Authority (IANA) has assigned IPCablecom an SMI Network Management Private Enterprise Number of 4491 for the encoding of these attributes.

Table 24/J.179 – Radius VSA structure for IPCablecom attributes

Field name	Semantics	Field length
Type	Vendor Specific = 26	1 byte
Length	Total Attribute Length NOTE – Value is Vendor Length + 8	1 byte
Vendor ID	CableLabs = 4491	4 bytes
Vendor Attribute Type	IPCablecom Attribute Type	1 byte (see Table 15)
Vendor Attribute Length	IPCablecom Attribute Length	1 byte (see Table 15) NOTE – Value is Vendor Length + 2
Vendor Attribute Value	IPCablecom Attribute Value	Vendor Length bytes

7.6.3 IPCablecom RADIUS Accounting-Request Packet Syntax

```
<<RADIUS Accounting-Request> ::==
                  <RADIUS message Header>
                  <RADIUS NAS-IP-Address Attribute>
                  <RADIUS Acct-Status-Type Attribute>
                  <Packet Cable EM>

<Packet Cable EM> ::==
                  <RADIUS VSA for IPCablecom EM Header Attribute>
                  <IPCablecom EM Attribute List>

<IPCablecom EM Attribute List> ::==
                  <RADIUS VSA for IPCablecom EM Attribute> |
                  <IPCablecom EM Attribute List>
                  <RADIUS VSA for Packet Cable EM Attribute>>
```

The Event Message Header is the first attribute within a given Event Message. The order of the Event Message attributes which follow the Event Message Header is arbitrary.

IPCablecom extends RADIUS Accounting, by introducing new attributes and new values for existing attributes. Since the RADIUS protocol is extendable in this manner, it is expected that existing RADIUS server implementations will require minimal modifications to support the batch collection of IPCablecom Event Messages.

8 Security requirements

Security for IPCablecom Multimedia interfaces utilizes security mechanisms defined in [11] as well as in [1]. Table 25 provides a summary of security mechanisms for each of the IPCablecom Multimedia interfaces.

Table 25/J.179 – Multimedia security interfaces

Interface	Description	Security mechanisms
pkt-mm-1	CMTS – CM	HMAC-based authentication defined by the J.112 Annex B RFI Recommendation.
pkt-mm-2	PS – CMTS	IPsec ESP using IKE or Kerberos-based key management.
pkt-mm-3	AM – PS	IPsec ESP using IKE or Kerberos-based key management.
pkt-mm-4	PS – RKS	IPsec ESP using IKE or Kerberos-based key management.
pkt-mm-5	CMTS – RKS	IPsec ESP using IKE or Kerberos-based key management.
pkt-mm-6	Client – CMTS	Out of scope for this version of this Recommendation.

Table 25/J.179 – Multimedia security interfaces

Interface	Description	Security mechanisms
pkt-mm-7	Client – AM	Out of scope for this version of this Recommendation.
pkt-mm-8	AM – Peer	Out of scope for this version of this Recommendation.
pkt-mm-9	CMTS – cable operator-Managed IP Network	Out of scope for this version of this Recommendation.
pkt-mm-10	Client – Peer	Out of scope for this version of this Recommendation.

The following clauses describe security that is applied to each IPCablecom Multimedia interface and specify additional requirements or extensions whenever necessary.

8.1 CMTS – CM QoS Interface (pkt-mm-1)

J.112 Annex B QoS messages are authenticated using an HMAC (Hash Message Authentication Code), which is a keyed cryptographic hash. Calculation of the HMAC attribute that must be included in J.112 Annex B QoS messages is specified in B.C.1.4.1 of [1].

8.2 Policy server – CMTS COPS Interface (pkt-mm-2)

The Policy Server – CMTS COPS interface MUST be secured using the IPsec ESP protocol, as specified in 7.2.1.3.2 of [11]. The key management requirements for this interface MUST comply with 7.2.1.4.1 of [11]. For this interface, Policy Server MUST comply with all the Gate Controller requirements listed in 7.2.1.3.2 and 7.2.1.4.1 of [11]. IKE with pre-shared keys is required to implement.

8.3 Application Manager – Policy Server COPS Interface (pkt-mm-3)

The Application Manager – Policy Server COPS interface MUST be secured using the IPsec ESP protocol, as specified in 7.2.1.3.2 of [11]. The key management requirements for this interface MUST comply with 7.2.1.4.1 of [11]. For this interface, Application Manager MUST comply with all the Gate Controller requirements listed in 7.2.1.3.2 and 7.2.1.4.1 of [11]. IKE with pre-shared keys is required to implement.

8.4 Policy Server – RKS Event Message Interface (pkt-mm-4)

The Policy Server – RKS Event Message interface MUST be secured using the IPsec ESP protocol, as specified in 7.3.2 of [11]. The key management for this interface MUST be identical to the one specified for a CMTS-RKS interface in 7.3.3.2 of [11]. IKE with pre-shared keys is required to implement.

8.5 CMTS – RKS Event Message Interface (pkt-mm-5)

The CMTS – RKS Event Message interface MUST be secured using the IPsec ESP protocol, as specified in 7.3.2 of [11]. The key management for this interface is specified in 7.3.3.2 of [11]. IKE with pre-shared keys is required to implement, while IKE with certificates and Kerberized IPsec are both optional to implement.

9 Mapping a FlowSpec Traffic Profile to DOCSIS

A Traffic Profile defines the QoS attributes of the IP flow or the J.112 Annex B Service Flow to be used in performing authorization, reservation and commit operations. A Traffic Profile can be defined via one of the following methods:

- FlowSpec;
- DOCSIS Service Class Name;
- DOCSIS Specific Parameterization.

This clause describes the mapping procedures for deriving the DOCSIS-specific QoS parameters from the various Traffic Profile representations. A Traffic Profile may include the authorization, reservation or commit envelopes. As defined in [3], a FlowSpec consists of a TSpec and an optional RSpec.

9.1 Mapping FlowSpecs to DOCSIS scheduling types

FlowSpecs support two types of services: Controlled Load and Guaranteed. Controlled Load services provide minimum bandwidth guarantees, but not latency/delay guarantees. Guaranteed services provide both bandwidth and latency/delay guarantees. Guaranteed service may be closely approximated through DOCSIS Real-Time Polling and UGS scheduling types. Controlled Load service may be closely approximated through the DOCSIS Best-Effort scheduling type. The FlowSpec service number in the FlowSpec definition distinguishes between Controlled Load and Guaranteed services. Service number 5 indicates the definition is for Controlled Load service, and service number 2 indicates the definition is for Guaranteed service. Further, Controlled Load service contains only the TSpec token bucket parameters, but not the RSpec. Guaranteed service MUST contain both the TSpec and the RSpec.

For latency and jitter-sensitive applications such as voice, MPEG video or gaming, one could request Guaranteed service. The CMTS can then use the traffic profile parameters specified in the FlowSpec to select one of the two types of DOCSIS scheduling types that could provide Guaranteed service: rtPS and UGS. For other applications that are not latency sensitive one could request Controlled Load service, which can be used to provide minimum bandwidth guarantees. Table 26 below summarizes the choices.

Table 26/J.179 – Mapping FlowSpecs types

DOCSIS scheduling type	FlowSpec service number	Application example
Unsolicited Grant Service (UGS)	2 (Guaranteed)	Voice over IP
Real-Time Polling Service (rtPS)	2 (Guaranteed)	VPN
Best Effort (BE)	5 (Controlled Load)	Best Effort Internet Data

The general FlowSpec-to-DOCSIS mapping procedure for upstream service flows is as follows:

- Upon receipt of a Gate-Set message with a FlowSpec the CMTS MUST analyse the TSpec Service Header to determine whether Controlled load or Guaranteed service is being requested.
- If Controlled Load Service, then the CMTS MUST use only the TSpec parameters to resolve the DOCSIS scheduling parameters to define the DOCSIS Traffic Parameters for a DOCSIS Best-Effort Scheduling type.
- If Guaranteed Service, the CMTS MUST examine the TSpec parameter's values for Reserved Rate (R) and Bucket Rate (r). If the two values are equal, then the CMTS MUST use the TSpec and the RSpec to define the DOCSIS Traffic Parameters for a DOCSIS UGS scheduling type.

- If the Reserved Rate (R) and Bucket Rate (r) are not equal, then the CMTS MUST use the TSpec and the RSpec to define the DOCSIS Traffic Parameters for a DOCSIS Real-Time Polling scheduling type.

Note that two other types of DOCSIS scheduling types are not mentioned above. These are:

- Unsolicited Grant Service with Activity Detection.
- Non-Real-Time Polling Service.

If the Application Manager wishes to request either one of these services, it can only do so by using either the Service Class Name or the DOCSIS-specific parameterization method of defining the Traffic Profile.

9.2 Mapping FlowSpecs to DOCSIS traffic parameters

The FlowSpec is made up of two parts, the TSpec and the RSpec. The TSpec describes the traffic for the flow, and the RSpec describes the desired service; note that for a controlled load service, the RSpec is not used. The RSpec parameters MUST be specified for a guaranteed service. The CMTS MUST ignore the RSpec parameters for a controlled load service. The RSpec is used to provide latency guarantees for guaranteed services. Please refer to RFCs 2210 [3], 1305 [2], 2211 [4], and 2212 [5] for more information on how these parameters should be used by Application Managers to specify the traffic profile. Note that the IPCablecom Multimedia interpretation of Flowspecs differs from the RFCs in the following respects:

- Guaranteed Service as defined in [5] controls Layer 3 queuing delay (i.e., the delays associated with packet scheduling), whereas in IPCablecom Multimedia we are primarily concerned with controlling the access delay of the DOCSIS MAC layer. Consequently we reserve bandwidth resources according to the TSpec's r parameter rather than the RSpec's R.
- As defined in [4], Controlled Load service defines only a guaranteed minimum rate for a flow. IPCablecom Multimedia's Controlled Load service facilitates definition of the maximum rate for a flow, as well as definition of flows without a guaranteed minimum rate.
- The Guaranteed Service Slack Term parameter is not needed in IPCablecom Multimedia, so the field is redefined to enable control of DOCSIS polling jitter.

TSpec Parameters:

- Bucket Depth (b), bytes;
- Bucket Rate (r), bytes/second;
- Maximum Datagram Size (M), bytes;
- Minimum Policed Unit (m), bytes;
- Peak Rate (p), bytes/second.

RSpec Parameters:

- Reserved Rate (R), bytes/second;
- Slack Term (S), microseconds.

The parameter mapping, roughly approximated, involves the following associations for DOCSIS upstream BE (Best-Effort) and downstream Controlled Load Service Flows. The actual mapping procedure would involve normalizing these parameters to account for Layer 2 and Layer 3 header considerations.

- TSpec Bucket Depth (b) ~= DOCSIS Maximum Traffic Burst;
- TSpec Maximum Datagram Size (M) ~= <not required by DOCSIS >;

- TSpec Minimum Policed Unit (m) ~= DOCSIS Assumed Minimum Reserved Rate Packet Size;
- TSpec Bucket Rate (r) ~= DOCSIS Minimum Reserved Rate;
- TSpec Peak Rate (p) ~= DOCSIS Maximum Sustained Rate for Controlled Load service.

For downstream Guaranteed service flows, the RSpec parameters are added to provide latency and reservation guarantees.

- TSpec Bucket Depth (b) ~= DOCSIS Maximum Traffic Burst;
- TSpec Maximum Datagram Size (M) ~= <not required by DOCSIS >;
- TSpec Minimum Policed Unit (m) ~= DOCSIS Assumed Minimum Reserved Rate Packet Size;
- TSpec Bucket Rate (r) ~= DOCSIS Minimum Reserved Rate;
- RSpec Reserved Rate (R) ~= DOCSIS Maximum Sustained Rate for Guaranteed service;
- RSpec Slack Term ~= DOCSIS Downstream Latency.

The parameter mapping, roughly approximated, involves the following associations for DOCSIS UGS Service Flows.

- TSpec Bucket Depth (b) = TSpec Maximum Datagram Size (M) = TSpec Minimum Policed Unit (m) ~= DOCSIS Unsolicited Grant Size;
- TSpec Bucket Rate (r) = TSpec Peak Rate (p) = RSpec Reserved Rate (R) ~= <not required by DOCSIS>;
- RSpec Slack Term ~= DOCSIS Tolerated Grant Jitter.

Similarly, the following associations apply for DOCSIS Real-Time Polling Service Flows.

- TSpec Bucket Depth (b) ~= DOCSIS Maximum Traffic Burst;
- TSpec Maximum Datagram Size (M) ~= <not required by DOCSIS >;
- TSpec Bucket Rate (r) ~= DOCSIS Maximum Sustained Rate for Guaranteed Service;
- RSpec Reserved Rate (R) ~= used to calculate the Polling Interval;
- RSpec Slack Term ~= Tolerated Polling Jitter.

This abstraction model allows standards-based RSVP implementations (as anticipated in Scenarios 2 and 3) to request and receive controlled load or guaranteed service from the network without necessarily requiring DOCSIS-specific info.

In some situations, where the Application Manager and the Policy Server is acutely aware of DOCSIS, it MAY specify the Traffic Profile for the Gate using the DOCSIS Service Class Name or the DOCSIS-specific parameterization format.

Note that there are several DOCSIS Service Flow parameters that cannot be directly resolved from the FlowSpecs; in these cases, the IPCablecom Multimedia Recommendation defines defaults for those Service Flow parameters. If the Application Manager/Policy Server wishes to set those Service Flow parameters to something other than the defaults specified by this Recommendation, the Application Manager/Policy Server MUST use either the Service Class Names or the DOCSIS-specific parameterization formats to define the traffic profile.

For Guaranteed Service, the Minimum Reserved Rate and the Maximum Sustained Rates are set to the same value, and are based on the Bucket Rate, 'r'. This is because Guaranteed Service provides latency guarantees, and this means a flow cannot be sustained at a rate greater than the rate at which the source has agreed to generate (when the reservation was initially made). A reservation made with a Traffic Profile specifying a Bucket Rate 'r' means that the source will not sustain a traffic flow greater than 'r'. Thus, it would be incorrect to use the Reserved Rate 'R' to represent any DOCSIS sustained rate (either minimum or maximum), in the Guaranteed service case.

For real-time polling Scheduling however, the CMTS uses the Reserved Rate R to calculate the polling interval, so that traffic sources can burst at the rate of R without increasing the delay that the packets experience waiting for a DOCSIS upstream transmission opportunity. Although the traffic source may generate traffic at rate 'R' in this case, the CMTS will ensure the sustained rate does not violate 'r' over time.

For Controlled Load Service, because there are no latency guarantees and because we want to enable one to use the DOCSIS-specific concepts of guaranteed minimum as well as maximum sustained rates, the TSpec Bucket Rate 'r' is mapped to the DOCSIS minimum rate, and the TSpec Peak Rate 'p' is mapped to the DOCSIS Maximum Sustained Rate. If a zero or infinite value is indicated for 'r', then the DOCSIS Minimum Reserved Rate parameter MUST be omitted. If a zero or infinite value is indicated for 'p', then the DOCSIS Maximum Sustained Rate parameter MUST be omitted.

9.3 DOCSIS upstream parameters

For all of the upstream packet size calculations following formula must be used: The DOCSIS size MUST be calculated from the DOCSIS MAC header FC to the end of the CRC. This value includes the Ethernet header overhead of 18 bytes (6 bytes for source address, 6 bytes for destination address, 2 bytes for length, and 4 bytes for CRC). The value also incorporates DOCSIS MAC layer overhead, including the DOCSIS base header (6 bytes), the UGS extended header (3 bytes), and the BPI+ extended header (5 bytes).

DOCSIS Unsolicited Grant Size = M + 32

The example above assumes that BPI+ [12] is enabled.

9.3.1 Unsolicited Grant Scheduling (UGS and UGS/AD)

Unsolicited Grant Scheduling MUST be used when the Service Number is 2 (Guaranteed), the Peak Rate, Bucket Rate, and the Reserved Rate are all equal, and Maximum Datagram Size is equal to Minimum Datagram Size.

The DOCSIS upstream objects MUST be set as stated below. All service flow quality of service TLV encodings that are not here defined MUST be given their default values as indicated by DOCSIS.

The DOCSIS Maximum Sustained Traffic Rate and DOCSIS Assumed Minimum Reserved Rate Packet Size parameters MUST NOT be used for upstream flows.

The DOCSIS Grants per Interval parameter MUST be set to 1.

The DOCSIS Nominal Grant Interval parameter MUST be set to the Maximum Datagram Size divided by the Reserved Rate.

DOCSIS Nominal Grant Interval = M/R

The DOCSIS Tolerated Grant Jitter parameter MUST be set to Slack Term. If the value is less than the duration of a DOCSIS minislot, then the minislot duration MUST be used instead. If a value of zero is specified, then the default value of 800 μs MUST be used.

The DOCSIS Nominal Polling Interval parameter MUST NOT be specified in the Traffic Profile for UGS service flows. For UGS/AD service flows, the DOCSIS Nominal Polling Interval MUST be explicitly specified in the DOCSIS-specific parameter. The CMTS MUST use the parameter supplied in the Traffic Profile for handling UGS/AD flows.

The DOCSIS Tolerated Polling Jitter parameter MUST NOT be specified in the Traffic Profile for UGS service flows. For UGS/AD service flows, the DOCSIS Tolerated Polling Jitter MUST be explicitly specified in the DOCSIS-specific parameter. The CMTS MUST use the parameter supplied in the Traffic Profile for handling UGS/AD flows.

The DOCSIS Request/Transmission Policy parameter is a bitmask; bits 0-6 and 8 MUST be set for UGS and UGS/AD service flows.

The DOCSIS Unsolicited Grant Size parameter MUST be derived from the Maximum Datagram Size by adding Layer 2 DOCSIS overhead.

9.3.2 Real-Time Polling Scheduling

Real-Time Polling Scheduling MUST be used when the Service Number is 2 (Guaranteed Service) and Peak Rate is not equal to Bucket Rate or Maximum Datagram Size is not equal to Minimum Datagram Size.

The DOCSIS upstream objects MUST be set as stated below. All service flow quality of service TLV encodings that are not here defined MUST be given their default values as indicated by DOCSIS.

The DOCSIS Maximum Sustained Traffic Rate parameter is given in bits per second, and includes MAC layer overhead. The conversion from IP-specific parameters involves first determining the packetization rate by dividing the Bucket Rate by the Minimum Datagram Size. This value is then multiplied by the packet size, Minimum Datagram Size, including MAC layer overhead, and the entire product is scaled from bytes to bits.

$$\text{DOCSIS Maximum Sustained Traffic Rate} = r/m \times (m + 29) \times 8$$

The DOCSIS Maximum Traffic Burst parameter MUST be set to the greater of:

1) Bucket Depth including the DOCSIS overhead calculated using the Minimum Datagram Size; or
2) the DOCSIS specified minimum value of 1522.

$$\text{DOCSIS Maximum Traffic Burst} = \max((\text{Bucket Depth}/m) \times (m + 29), 1522)$$

The DOCSIS Minimum Reserved Traffic Rate parameter is the same as the DOCSIS Maximum Sustained Traffic Rate.

$$\text{DOCSIS Minimum Reserved Traffic Rate} = r/m \times (m + 29) \times 8$$

The DOCSIS Request/Transmission Policy parameter is a bitmask; all bits should be set to 0.

The DOCSIS Nominal Polling Interval parameter MUST be set to Reserved Rate divided by Minimum Datagram Size.

$$\text{DOCSIS Nominal Polling Interval} = R/m$$

The DOCSIS Tolerated Polling Jitter parameter MUST be set to Slack Term. If the value is non-zero but less than the duration of a minislot, then it MUST be set to the duration of a minislot. If a value of zero is specified, then the DOCSIS Tolerated Polling Jitter must default to a value of 800 μs.

$$\text{DOCSIS Nominal Polling Jitter} = S$$

9.3.3 Best Effort Scheduling

Best Effort Scheduling MUST be used when the Service Number is 5 (Controlled-Load).

The DOCSIS upstream objects MUST be set as stated below. All service flow quality of service TLV encodings that are not here defined MUST be given their default values as indicated by DOCSIS.

The DOCSIS Traffic Priority MUST be set to 5.

The DOCSIS Maximum Sustained Traffic Rate parameter is given in bits per second, including MAC layer overhead. The conversion from IP-specific parameters involves first determining the packetization rate by dividing the Peak Rate by the Minimum Datagram Size. This value is then

multiplied by the packet size, Minimum Datagram Size, amended to include MAC layer overhead, and the entire product is scaled from bytes to bits. The DOCSIS Maximum Sustained Traffic Rate MUST be converted from the Minimum Datagram Size so long as a non-zero value is provided. If the Minimum Datagram Size object is zero, this parameter MUST be omitted.

DOCSIS Maximum Sustained Traffic Rate = $p/m \times (m + 29) \times 8$

The DOCSIS Maximum Traffic Burst parameter MUST be set to the greater of:

1) Bucket Depth including the DOCSIS overhead calculated using the Minimum Datagram Size; or

2) the DOCSIS specified minimum value of 1522.

DOCSIS Maximum Traffic Burst = max ((Bucket Depth/m) × (m + 29), 1522)

The DOCSIS Minimum Reserved Traffic Rate parameter is calculated in a manner similar to the DOCSIS Maximum Sustained Traffic Rate, except that instead of using the Peak Rate parameter, the Bucket Rate is used.

DOCSIS Minimum Reserved Traffic Rate = $r/m \times (m + 29) \times 8$

9.3.4 Upstream packet classification encodings

9.3.4.1 DOCSIS upstream packet classification requests

The DOCSIS upstream objects MUST be set as stated below. All classification TLV encodings that are not here defined MUST be given their default values as indicated by DOCSIS.

If defined by the CMTS, the DOCSIS Classifier Identifier parameter MUST be used.

The DOCSIS Rule Priority parameter MUST be set to the Priority value in the Classifier object.

The DOCSIS Classification Activation State parameter MUST be set to active (1) when the Gate utilizing the service flow is committed, and for all the other cases it MUST be set to inactive (0).

The DOCSIS Dynamic Service Change Action MAY use the DSC Add Classifier (0), DSC Replace Classifier (1) and DSC Delete Classifier (2) operations per the DOCSIS RFI Recommendation.

The DOCSIS IP Protocol parameter MUST be set to the same meaning as Protocol.

The DOCSIS IP Source Address parameter MUST be set to the same address as that in the Classifier object, so long as a non-zero value is provided. If the address specified in the Classifier object is zero, this parameter MUST be omitted.

The DOCSIS IP Source Mask parameter MUST be omitted.

The DOCSIS IP Source Port Start and DOCSIS IP Source Port End parameters MUST be set to the same transport port value as the Classifier object.

The DOCSIS IP Destination Address parameter MUST be set to the same address as that in the Classifier object, so long as a non-zero value is provided. If the address specified in the Classifier object is zero, this parameter MUST be omitted.

The DOCSIS IP Destination Mask parameter MUST be omitted.

The DOCSIS IP Destination Port Start and DOCSIS IP Destination Port End parameters MUST be set to the same transport port as the Classifier object, so long as a non-zero value is provided. If the Destination IP Port is specified as a value of zero in the Classifier object, then the DOCSIS IP Destination Port End TLV MUST be omitted.

The DOCSIS Ethernet LLC Packet Classification Encodings parameters MUST be omitted.

The DOCSIS 802.1P/Q Packet Classification Encodings parameters MUST be omitted.

9.4 DOCSIS downstream parameters

9.4.1 Downstream QoS encodings for guaranteed service

The DOCSIS downstream service flow quality-of-service TLV encodings MUST be set as stated below. All service flow quality of service TLV encodings that are not here defined MUST be given their default values as indicated by DOCSIS.

The downstream DOCSIS parameters are calculated using DOCSIS MAC header from the byte following the HCS to the end of the CRC. The MAC layer (i.e., Ethernet) overhead is 18 bytes (6 bytes for source address, 6 bytes for destination address, 2 bytes for length, and 4 bytes for CRC).

Based on this overhead, the DOCSIS Assumed Minimum Reserved Rate Packet Size parameter MUST be calculated as:

DOCSIS Assumed Minimum Reserved Rate Packet Size = m + 18

The DOCSIS Maximum Sustained Traffic Rate parameter is given in bits per second, including MAC layer overhead. The conversion from IP-specific parameters involves first determining the packetization rate by dividing the Bucket Rate by the Minimum Datagram Size. This value is then multiplied by the packet size, Minimum Datagram Size, amended to include MAC layer overhead, and the entire product is scaled from bytes to bits. The DOCSIS Maximum Sustained Traffic Rate MUST be calculated as:

DOCSIS Maximum Sustained Traffic Rate = $r/m \times (m + 18) \times 8$

The DOCSIS Minimum Reserved Traffic Rate is equal to the DOCSIS Maximum Sustained Traffic Rate.

Note that the DOCSIS Maximum Sustained Traffic Rate and the DOCSIS Minimum Reserved Traffic Rate are calculated slightly differently in IPCablecom Multimedia and IPCablecom DQoS. IPCablecom Multimedia is based on r and IPCablecom DQoS is based on p. This is due to the fact that in DQoS r = p, while in Multimedia these values differ (in which case r is the proper rate value to use).

The DOCSIS Maximum Traffic Burst parameter MUST be set to the greater of:

1) Bucket Depth including the DOCSIS overhead calculated using the Minimum Datagram Size; or
2) the DOCSIS specified minimum value of 1522.

DOCSIS Maximum Traffic Burst = max ((Bucket Depth/m) × (m + 18), 1522)

The DOCSIS Traffic Priority parameter MUST be set to 5.

The DOCSIS Downstream Latency parameter MUST be set to Slack Term, if Slack Term is non-zero. If Slack Term is zero, this parameter MUST NOT be populated.

9.4.2 Downstream QoS encodings for controlled load service

The DOCSIS downstream service flow quality of service TLV encodings MUST be set as stated below. All service flow quality of service TLV encodings that are not here defined MUST be given their default values as indicated by DOCSIS.

The downstream DOCSIS parameters are calculated using the DOCSIS MAC header from the byte following the HCS to the end of the CRC. The MAC layer (i.e., Ethernet) overhead is 18 bytes (6 bytes for source address, 6 bytes for destination address, 2 bytes for length, and 4 bytes for CRC).

Based on this overhead, the DOCSIS Assumed Minimum Reserved Rate Packet Size parameter MUST be calculated as:

DOCSIS Assumed Minimum Reserved Rate Packet Size = m + 18

The DOCSIS Maximum Sustained Traffic Rate parameter is given in bits per second, including MAC layer overhead. The conversion from IP-specific parameters involves first determining the packetization rate by dividing the Peak Rate by the Minimum Datagram Size. This value is then multiplied by the packet size, Minimum Datagram Size, amended to include MAC layer overhead, and the entire product is scaled from bytes to bits. The DOCSIS Maximum Sustained Traffic Rate MUST be calculated as:

DOCSIS Maximum Sustained Traffic Rate = $p/m \times (m + 18) \times 8$

The DOCSIS Minimum Reserved Traffic Rate parameter is calculated in a manner similar to the DOCSIS Maximum Sustained Traffic Rate, except that instead of using the Peak Rate, the Bucket Rate is used.

DOCSIS Minimum Reserved Traffic Rate= $r/m \times (m + 18) \times 8$

The DOCSIS Maximum Traffic Burst parameter MUST be set to the greater of:

1) Bucket Depth including the DOCSIS overhead calculated using the Maximum Datagram Size; or
2) the DOCSIS specified minimum value of 1522.

DOCSIS Maximum Traffic Burst = max ((Bucket Depth/M) × (M + 18), 1522)

The DOCSIS Traffic Priority parameter MUST be set to 5.

The DOCSIS Downstream Latency parameter MUST NOT be populated.

9.4.3 Downstream packet classification encodings

9.4.3.1 DOCSIS downstream packet classification requests

The DOCSIS downstream classification objects MUST be set as stated below. All classification TLV encodings that are not here defined MUST be given their default values as indicated by DOCSIS.

If defined by the CMTS, the DOCSIS Classifier Identifier parameter MUST be used.

If defined by the CMTS, the DOCSIS Service Flow Identifier parameter MUST be used.

The DOCSIS Rule Priority parameter MUST be set to Priority value specified in the Classifier object.

The DOCSIS Classification Activation State parameter MUST be set to active (1) when the Gate utilizing the service flow is committed, and for all the other cases it MUST be set to inactive (0).

The DOCSIS Dynamic Service Change Action MAY use the DSC Add Classifier (0), DSC Replace Classifier (1) and DSC Delete Classifier (2) operations per the DOCSIS RFI Recommendation.

The DOCSIS IP TOS and mask fields MUST NOT be used.

The DOCSIS IP Protocol parameter MUST be set to the Protocol ID value specified in the Classifier object.

The DOCSIS IP Source Address parameter MUST be set to the source address provided in the Classifier object, so long as a non-zero value is provided. If the address specified in the Classifier object is zero, this parameter MUST be omitted.

The DOCSIS IP Source Mask parameter MUST be omitted.

The DOCSIS IP Source Port Start and DOCSIS IP Source Port End parameters MUST be set to the same transport port value as indicated in the Classifier, so long as a non-zero value is provided. If the Source IP Port is specified as a value of zero in the Classifier, then the DOCSIS IP Source Port End TLV MUST be omitted.

The DOCSIS IP Destination Address parameter MUST be set to the same address as indicated in the Classifier object.

The DOCSIS IP Destination Mask parameter MUST be omitted.

The DOCSIS IP Destination Port Start and DOCSIS IP Destination Port End parameters MUST be set to the same port as indicated in the Classifier object.

The DOCSIS Ethernet LLC Packet Classification Encodings MUST be omitted.

The DOCSIS 802.1P/Q Packet Classification Encodings MUST be omitted.

10 Message flows

This clause provides two interaction scenarios between the various network elements previously introduced in this Recommendation. The first interaction outlines at a relatively high-level the basic message exchanges that take place within the IPCablecom Multimedia framework in order to authorize, reserve and commit access network resources under Scenario 1. The second interaction provides a very detailed breakout of each message and attribute involved in the IPCablecom Multimedia QoS and EM interfaces.

10.1 Basic message sequence

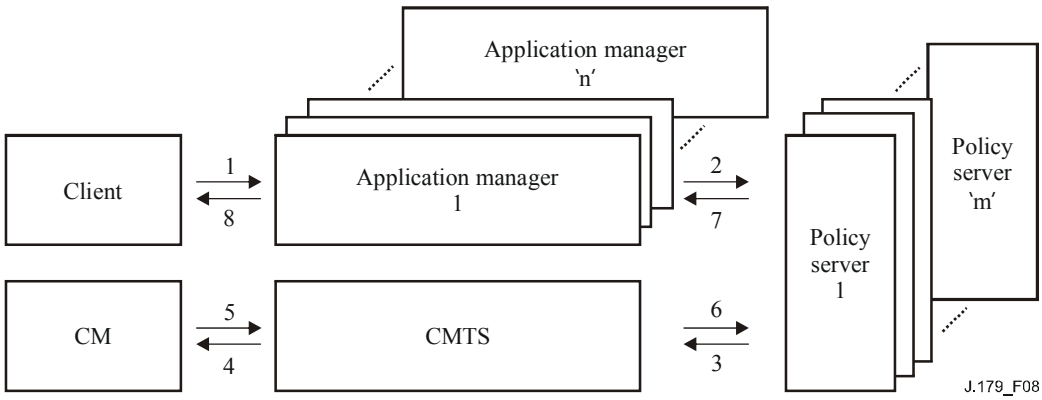

Figure 8/J.179 – Basic message sequence

1) Client issues a session setup request to the Application Manager via Application Layer signalling. The client may authenticate itself to the Application Manager during this step.
2) Before the Application Manager activates the session, the Application Manager issues a Gate Set (in a COPS DECISION message) and sends it to Policy Server in order to determine if the session setup request should be allowed to proceed. Message includes:
   a) AMID;
   b) SubscriberID;
   c) TransactionID;
   d) Classifier;
   e) Traffic Profile for Flow.

3) Upon receiving the request, the Policy Server checks the request against the policy rules and if the request is approved, sends a Gate-Set to the CMTS. Message includes:
   a) AMID;
   b) SubscriberID;
   c) TransactionID;
   d) Classifier;
   e) Traffic Profile for Flow (Authorized, Reserved and Committed).
4) CMTS uses the classifier and Traffic Profile information to trigger the activation of the flow by issuing the appropriate DOCSIS DSx messages.
5) CM acknowledges with the appropriate DSx messaging.
6) CMTS issues a Gate-Set-Ack to the Policy Server in response to the Gate-Set message received in Step 3. Message includes:
   a) AMID;
   b) TransactionID;
   c) GateID.
7) In response the Policy Server will generate a Gate-Set-Ack to the AM; this tells the AM that the Policy Request has been admitted and the client's request can proceed, and the necessary resources in the underlying network have been reserved. Message includes:
   a) AMID;
   b) TransactionID;
   c) GateID.
8) Application Manager, upon receiving the Gate-Set-Ack will inform the client that the session establishment can proceed.

10.2 Detailed message sequence

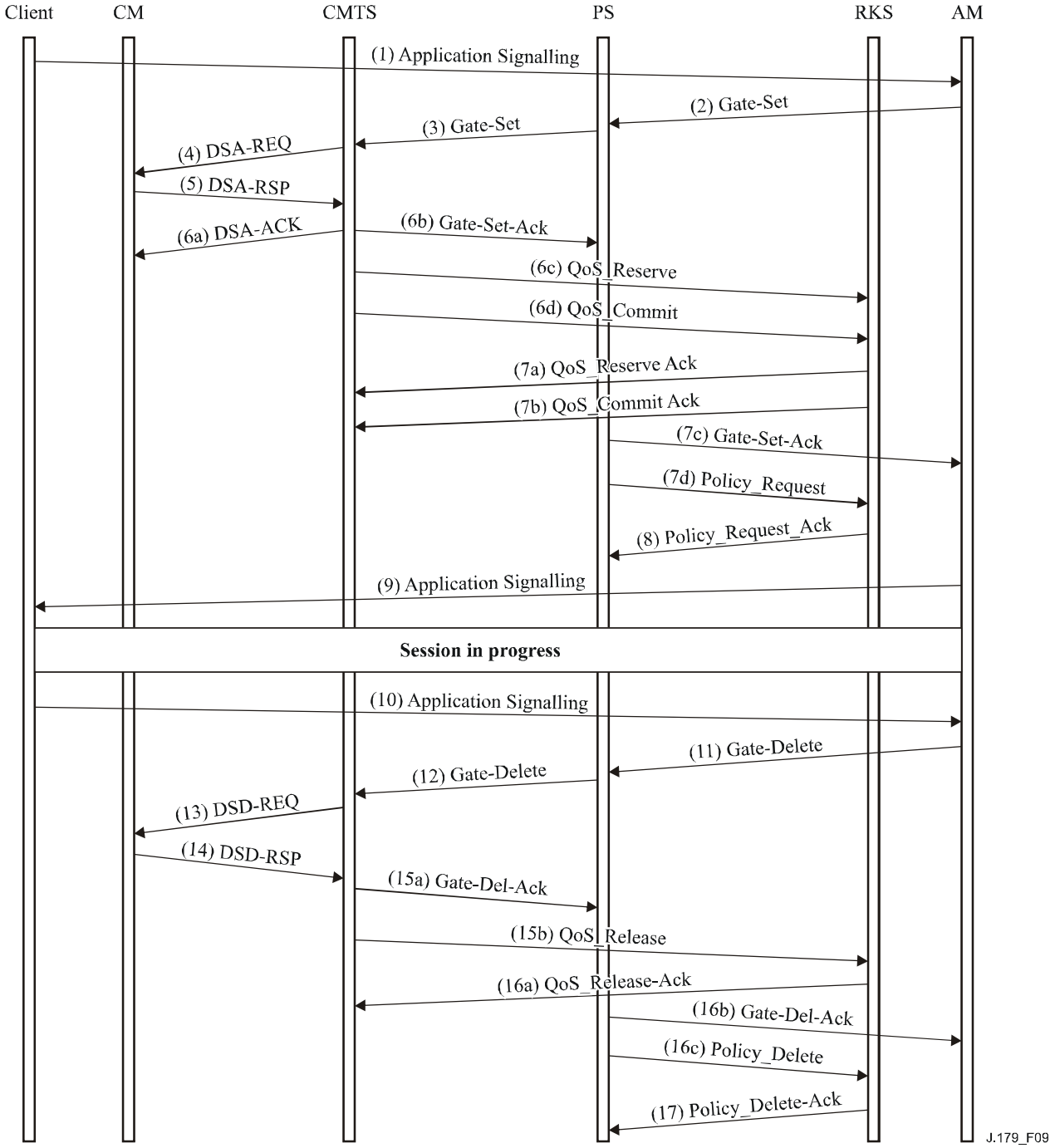

Figure 9/J.179 – Detailed message sequence

The pages that follow describe in detail the messages that are being exchanged in an example IPCablecom Multimedia session. The bandwidth numbers are purely examples, and do not correlate to any particular service. Only the upstream access network resources are being reserved and committed for clarity. Also, BPI related TLVs have been left out of the DOCSIS messages for clarity.

1) The client initializes the session by querying an Application Manager for the necessary resources to use the application. An example of this would be a software based video game, asking for resources to play an online game. This signalling is out of scope for this Recommendation.

2) After receiving the application signalling from the client, the Application Manager issues a Gate-Set to the Policy Server, requesting the needed resources for this session.

0	1	2	3
COPS Header			
Version 0x1	Flags 0x0	Op-Code 0x02	Client-Type 0x800A
Message Length 0x000000C0			
COPS Handle Object			
Length 0x0008		C-Num 0x01	C-Type 0x01
Handle 0x00001234			
COPS Context Object			
Length 0x0008		C-Num 0x02	C-Type 0x01
Request Type (R-Type) 0x0008 (Configuration Request)		Message Type (M-Type) 0x0000	
COPS Decision Object			
Length 0x0008		C-Num 0x06	C-Type 0x01
Command Code 0x0001 (Install Configuration)		Flags 0x0000	
COPS ClientSI Object Header			
Length 0x00A0		C-Num 0x09	C-Type 0x01
Multimedia TransactionID Object			
Length 0x0008		S-Num 0x01	S-Type 0x01
TransactionID 0x9999		Gate Command 0x0004 (Gate-Set)	
Multimedia AMID Object			
Length 0x0008		S-Num 0x02	S-Type 0x01
AMID 0x00005678			
Multimedia SubscriberID Object			
Length 0x0008		S-Num 0x03	S-Type 0x01
SubscriberID 0x01010101			

Multimedia GateSpec Object			
Length 0x0010		S-Num 0x05	S-Type 0x01
Flags 0x01	DSCP/TOS Field 0x00	DSCP/TOS Mask 0x00	SessionClassID 0x00
Timer T1 0x00C8 (200 seconds)		Timer T2 0x012C (300 seconds)	
Timer T3 0x003C (60 seconds)		Reserved	
Multimedia FlowSpec Object			
Length 0x005C		S-Num 0x07	S-Type 0x01
Envelope 0x07	Service Number 0x02	Reserved	Reserved
Authorized Envelope			
Token Bucket Rate [r] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Token Bucket Size [b] (encoded as IEEE floating point) 0x43480000 (200 bytes)			
Peak Data Rate [p] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Minimum Policed Unit [m] 0x000000C8 (200 bytes)			
Maximum Packet Size [M] 0x000000C8 (200 bytes)			
Rate [R] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Slack Term [S] 0x00000320 (800 μs)			
Reserved Envelope			
Token Bucket Rate [r] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Token Bucket Size [b] (encoded as IEEE floating point) 0x43480000 (200 bytes)			
Peak Data Rate [p] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Minimum Policed Unit [m] 0x000000C8 (200 bytes)			
Maximum Packet Size [M] 0x000000C8 (200 bytes)			
Rate [R] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			

Slack Term [S] 0x00000320 (800 µs)
Committed Envelope
Token Bucket Rate [r] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)
Token Bucket Size [b] (encoded as IEEE floating point) 0x43480000 (200 bytes)
Peak Data Rate [p] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)
Minimum Policed Unit [m] 0x000000C8 (200 bytes)
Maximum Packet Size [M] 0x000000C8 (200 bytes)
Rate [R] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)
Slack Term [S] 0x00000320 (800 µs)

Multimedia Classifier Object			
Length 0x0018		S-Num 0x06	S-Type 0x01
Reserved	ProtocolID 0x11 (17 UDP)	DSCP/TOS Field 0x00	DSCP/TOS Mask 0x00
Source IP Address 0x01010101			
Destination IP Address 0x02020202			
Source Port 0x1234		Destination Port 0x9876	
Priority 0x0040 (64)		Reserved	

3) After the PS receives the Gate-Set from the Application Manager, it checks to see if the request is authorized, and if so, sends a Gate-Set to the CMTS.

0	1	2	3
COPS Header			
Version 0x1	Flags 0x0	Op-Code 0x02	Client-Type 0x800A
Message Length 0x000000EC			

COPS Handle Object			
Length 0x0008		C-Num 0x01	C-Type 0x01
Handle 0x00005678			
COPS Context Object			
Length 0x0008		C-Num 0x02	C-Type 0x01
Request Type (R-Type) 0x0008 (Configuration Request)		Message Type (M-Type) 0x0000	
COPS Decision Object			
Length 0x0008		C-Num 0x06	C-Type 0x01
Command Code 0x0001 (Install Configuration)		Flags 0x0000	
COPS ClientSI Object Header			
Length 0x00CC		C-Num 0x09	C-Type 0x01
Multimedia TransactionID Object			
Length 0x0008		S-Num 0x01	S-Type 0x01
TransactionID 0x0001		Gate Command 0x0004 (Gate-Set)	
Multimedia AMID Object			
Length 0x0008		S-Num 0x02	S-Type 0x01
AMID 0x00005678			
Multimedia SubscriberID Object			
Length 0x0008		S-Num 0x03	S-Type 0x01
SubscriberID 0x01010101			
Multimedia GateSpec Object			
Length 0x0010		S-Num 0x05	S-Type 0x01
Direction 0x01	DSCP/TOS Field 0x00	DSCP/TOS Mask 0x00	SessionClassID 0x00
Timer T1 0x00C8 (200 seconds)		Timer T2 0x012C (300 seconds)	
Timer T3 0x003C (60 seconds)		Reserved	

Multimedia FlowSpec Object			
Length 0x005C		S-Num 0x07	S-Type 0x01
Envelope 0x07	Service Number 0x02	Reserved	Reserved
Authorized Envelope			
Token Bucket Rate [r] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Token Bucket Size [b] (encoded as IEEE floating point) 0x43480000 (200 bytes)			
Peak Data Rate [p] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Minimum Policed Unit [m] 0x000000C8 (200 bytes)			
Maximum Packet Size [M] 0x000000C8 (200 bytes)			
Rate [R] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Slack Term [S] 0x00000320 (800 µs)			
Reserved Envelope			
Token Bucket Rate [r] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Token Bucket Size [b] (encoded as IEEE floating point) 0x43480000 (200 bytes)			
Peak Data Rate [p] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Minimum Policed Unit [m] 0x000000C8 (200 bytes)			
Maximum Packet Size [M] 0x000000C8 (200 bytes)			
Rate [R] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Slack Term [S] 0x00000320 (800 µs)			
Committed Envelope			
Token Bucket Rate [r] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Token Bucket Size [b] (encoded as IEEE floating point) 0x43480000 (200 bytes)			
Peak Data Rate [p] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			

Minimum Policed Unit [m] 0x000000C8 (200 bytes)			
Maximum Packet Size [M] 0x000000C8 (200 bytes)			
Rate [R] (encoded as IEEE floating point) 0x461C4000 (10,000 Bit/s)			
Slack Term [S] 0x00000320 (800 µs)			
Multimedia Classifier Object			
Length 0x0018		S-Num 0x06	S-Type 0x01
Reserved	ProtocolID 0x11	DSCP/TOS Field 0x00	DSCP/TOS Mask 0x00
Source IP Address 0x01010101			
Destination IP Address 0x02020202			
Source Port 0x1234		Destination Port 0x9876	
Priority 0x0040 (64)		Reserved	
Multimedia Event Generation Info Object			
Length 0x002C		S-Num 0x08	S-Type 0x01
Primary RKS Address 0x03030303			
Primary RKS Port 0x1111		Reserved	
Secondary RKS Address 0x04040404			
Secondary RKS Port 0x1111		Reserved	
BCID 0x3E4812082020202020313436302D3035303030300003DB77			

4) If CMTS admission control succeeds, the CMTS will initiate reserving and committing the access network resources by issuing a DSA to the Cable Modem.

0	1	2	3
MAC Management Header			
TransactionID 0x0007		US Service Flow 0x18	Length 0x29
Service Flow ID 0x02	Length 0x04	Value 0x0000	
Value (cont.) 0001		Service ID 0x03	Length 0x02
Value 0x0001		QoS Param Set 0x06	Length 0x01
Value 0x06 (Ad.+Act.)	Scheduling Type 0x0F	Length 0x01	Value 0x06
UGS Size 0x13	Length 0x02	Value 0x00E8 (232 bytes)	
Nom. Grant Int. 0x14	Length 0x04	Value 0x0000	
Value (cont.) 4E20 (20,000 µs)		Grants Per Interval 0x16	Length 0x01
Value 0x01	RX/TX Policy 0x10	Length 0x04	Value 0x00
Value (cont.) 00017F			Tol. Grant Jitter 0x15
Length 0x04	Value 0x000003		
Value (cont.) 20 (800 µs)	US Pkt. Clfr. 0x16	Length 0x13	Clfr. ID 0x02
Length 0x02	Value 0x0001		Service Flow ID 0x04
Length 0x04	Value 0x000000		
Value (cont.) 01	Rule Priority 0x05	Length 0x01	Value 0x40
Clfr Act. State 0x06	Length 0x01	Value 0x01 (Active)	Change Action 0x07 (Add)
Length 0x01	Value 0x00	IP Pkt. Clfr. 0x09	Length 0x001A

IP Protocol 0x02	Length 0x02	Value 0x0011 (17 UDP)	
IP Src. Addr 0x03	Length 0x04	Value 0x0101	
Value (cont.) 0101		IP Src Port Start 0x07	Length 0x02
Value 0x1234		IP Src Port End 0x08	Length 0x02
Value 0x1234		IP Dest Port Start 0x09	Length 0x02
Value 0x9876		IP Dest Port End 0x0A	Length 0x02
Value 0x9876			

5) The CM responds to the CMTS with a DSA-RSP.

0	1	2	3
MAC Management Header			
TransactionID 0x0007		Confirm. Code 0x00	

6a) The CMTS completes the transaction with a DSA-ACK.

0	1	2	3
MAC Management Header			
TransactionID 0x0007		Confirm. Code 0x00	

6b) Once a DSA-RSP is received by the CMTS from the CM confirming a successful transaction, the CMTS will send a Gate-Set-Ack to the Policy Server.

0		1	2	3
COPS Header				
Version 0x1	Flags 0x1	Op-Code 0x03	Client-Type 0x800A	
Message Length 0x0000003C				
COPS Handle Object				
Length 0x0008			C-Num 0x01	C-Type 0x01
Handle 0x00005678				
COPS Report-Type Object				
Length 0x0008			C-Num 0x12	C-Type 0x01
Report Type (R-Type) 0x0001 (Success)			Reserved	
COPS ClientSI Object Header				
Length 0x0024			C-Num 0x09	C-Type 0x01
Multimedia TransactionID Object				
Length 0x0008			S-Num 0x01	S-Type 0x01
TransactionID 0x0001			Gate Command 0x0005 (Gate-Set-Ack)	
Multimedia AMID Object				
Length 0x0008			S-Num 0x02	S-Type 0x01
AMID 0x00005678				
Multimedia SubscriberID Object				
Length 0x0008			S-Num 0x03	S-Type 0x01
SubscriberID 0x01010101				
Multimedia GateID Object				
Length 0x0008			S-Num 0x04	S-Type 0x01
GateID 0x12345678				

6c) The CMTS will also send a QoS_Reserve Event message to signal to the RKS that the access network resources have been reserved.

0	1	2	3
Accounting-Request Radius Header			
Radius Ven. Spec. 0x1A	Length 0x54	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type (EM Header) 0x01	Length 0x4E
Version 0x0003		BCID 0x3D48	
BCID (cont.) 120820202020203134363 02D3035303030300003DB77			
		Event Message Type 0x0007 (QoS-Reserve)	
Element Type 0x0002 (CMTS)		Element ID 0x2020202031323334	
		Time Zone 0x302D303530303030	
		Sequence Number 0x0000	
Sequence Number (cont.) 0001		Event Time 0x3230	
Event Time (cont.) 303331323036303030303030 2E303030			
Status 0x00000000			
Priority 0x80 (128)	Attribute Count 0x0004		Event Object 0x00

Radius Ven. Spec. 0x1A	Length 0x5C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x20	Length 0x56
QoS_Descriptor 0x0003FFFD0000000000000000000000000000000000000600004E2000003200000000 1000000E8 00000000000000000000BE40000004000000000000005F20000017F000186A0000007D000FF0000 00000000			
Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x1E	Length 0x06
SF_ID 0x00000001			
Radius Ven. Spec. 0x1A	Length 0x0A	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x32	Length 0x04
Flow Direction 0x0001 (Upstream)		Radius Ven. Spec. 0x1A	Length 0x0A
Vendor ID 0x0000118B			
Type 0x37	Length 0x04	Element_Requesting_QoS 0x0001 (Policy Server)	

6d) Immediately after sending the QoS_Reserve Event Message to the RKS, the CMTS will send the QoS_Commit Event Message to the RKS. This is due to the fact that the access network resources are being reserved and committed in one step.

0	1	2	3
Accounting-Request Radius Header			
Radius Ven. Spec. 0x1A	Length 0x54	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type (EM Header) 0x01	Length 0x4E
Version 0x0003		BCID 0x3E48	
BCID (cont.) 1208202020202031343630 2D3035303030300003DB77			
		Event Message Type 0x0013 (QoS-Commit)	
Element Type 0x0002 (CMTS)		Element ID 0x2020202031323334	
		Time Zone 0x302D303530303030	
		Sequence Number 0x0000	
Sequence Number (cont.) 0002		Event Time 0x3230	
Event Time (cont.) 303331323036303030303030 2E303030			
Status 0x00000000			
Priority 0x80 (128)	Attribute Count 0x0003		Event Object 0x00

Radius Ven. Spec. 0x1A	Length 0x5C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x20	Length 0x56
QoS_Descriptor 0x0003FFFD0000000000000000000000000000000000000000600004E2000000320000000001000000E8 00000000000000000000BE40000004000000000000005F20000017F000186A0000007D000FF0000 00000000			
Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x1E	Length 0x06
SF_ID 0x00000001			
Radius Ven. Spec. 0x1A	Length 0x0A	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x32	Length 0x04
Flow Direction 0x0001 (Upstream)			

7a) After receiving and recording the QoS_Reserve Event Message, the RKS acknowledges the message.

0	1	2	3
Accounting-Response Radius Header			

7b) After receiving and recording the QoS_Commit Event Message, the RKS acknowledges the message.

0	1	2	3
Accounting-Response Radius Header			

7c) As a result of receiving a Gate-Set-Ack from the CMTS, the Policy Server will send a Gate-Set-Ack to the Application Manager to complete the transaction.

0		1	2	3
COPS Header				
Version 0x1	Flags 0x1	Op-Code 0x03	Client-Type 0x800A	
Message Length 0x0000003C				
COPS Handle Object				
Length 0x0008			C-Num 0x01	C-Type 0x01
Handle 0x00001234				
COPS Report-Type Object				
Length 0x0008			C-Num 0x12	C-Type 0x01
Report Type (R-Type) 0x0001 (Success)			Reserved	
COPS ClientSI Object Header				
Length 0x0024			C-Num 0x09	C-Type 0x01
Multimedia TransactionID Object				
Length 0x0008			S-Num 0x01	S-Type 0x01
TransactionID 0x9999			Gate Command 0x0005	

Multimedia AMID Object		
Length 0x0008	S-Num 0x02	S-Type 0x01
AMID 0x00005678		
Multimedia SubscriberID Object		
Length 0x0008	S-Num 0x03	S-Type 0x01
SubscriberID 0x01010101		
Multimedia GateID Object		
Length 0x0008	S-Num 0x04	S-Type 0x01
GateID 0x12345678		

7d) The Policy Server will also send a Policy_Request Event Message to the RKS to track the Policy Request and associated outcome.

0	1	2	3
Accounting-Request Radius Header			
Radius Ven. Spec. 0x1A	Length 0x54	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type (EM Header) 0x01	Length 0x4E
Version 0x0001		BCID 0x3E48	
BCID (cont.) 120820202020203134363 02D3035303030300003DB77			
		Event Message Type 0x0015 (Policy_Request)	

Element Type 0x0004 (Policy Server)		Element ID 0x2020202035363738	
		Time Zone 0x302E303530303030	
		Sequence Number 0x0000	
Sequence Number (cont.) 0001		Event Time 0x3230	
Event Time (cont.) 3033313230363030303030302E323130			
Status 0x00000000			
Priority 0x80 (128)	Attribute Count 0x0004		Event Object 0x00
Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x3D	Length 0x06
Application_Manager_ID 0x00005678			
Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x34	Length 0x06
Subscriber_ID 0x01010101			
Radius Ven. Spec. 0x1A	Length 0x0A	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x3C	Length 0x04
Policy_Decision_Status 0x0001 (Policy Approved)		Radius Ven. Spec. 0x1A	Length 0x1C
Vendor ID 0x0000118B			

Type 0x31	Length 0x16	FEID 0x0000
FEID (cont.) 00000000000005061636B65744361626C65		

8) After receiving and recording the Policy_Request Event Message, the RKS acknowledges the message.

0	1	2	3
Accounting-Response Radius Header			

9) The Application Manager will reply to the client to inform the client that it can now play the game. This signalling is out of scope for this Recommendation.

10) When the client is finished with the application, it will notify the Application Manager. This signalling is out of scope for this Recommendation.

11) The Application Manager will terminate the session by sending a Gate-Delete to the Policy Server.

0	1	2	3
COPS Header			
Version 0x1	Flags 0x0	Op-Code 0x02	Client-Type 0x800A
Message Length 0x00000044			
COPS Handle Object			
Length 0x0008		C-Num 0x01	C-Type 0x01
Handle 0x00001234			
COPS Context Object			
Length 0x0008		C-Num 0x02	C-Type 0x01
Request Type (R-Type) 0x0008 (Configuration Request)		Message Type (M-Type) 0x0000	

COPS Decision Object		
Length 0x0008	C-Num 0x06	C-Type 0x01
Command Code 0x0001 (Install Configuration)	Flags 0x0000	
COPS ClientSI Object Header		
Length 0x0014	C-Num 0x09	C-Type 0x01
Multimedia TransactionID Object		
Length 0x0008	S-Num 0x01	S-Type 0x01
TransactionID 0x9998	Gate Command 0x000A (Gate-Delete)	
Multimedia AMID Object		
Length 0x0008	S-Num 0x02	S-Type 0x01
AMID 0x00005678		
Multimedia SubscriberID Object		
Length 0x0008	S-Num 0x03	S-Type 0x01
SubscriberID 0x01010101		
Multimedia GateID Object		
Length 0x0008	S-Num 0x04	S-Type 0x01
GateID 0x12345678		

12) The Policy Server will instruct the CMTS to tear down the session by sending a Gate-Delete.

0		1	2	3
COPS Header				
Version 0x1	Flags 0x0	Op-Code 0x02	Client-Type 0x800A	
Message Length 0x00000044				
COPS Handle Object				
Length 0x0008			C-Num 0x01	C-Type 0x01
Handle 0x00005678				

COPS Context Object		
Length 0x0008	C-Num 0x02	C-Type 0x01
Request Type (R-Type) 0x0008 (Configuration Request)	Message Type (M-Type) 0x0000	
COPS Decision Object		
Length 0x0008	C-Num 0x06	C-Type 0x01
Command Code 0x0001 (Install Configuration)	Flags 0x0000	
COPS ClientSI Object Header		
Length 0x0014	C-Num 0x09	C-Type 0x01
Multimedia TransactionID Object		
Length 0x0008	S-Num 0x01	S-Type 0x01
TransactionID 0x0002	Gate Command 0x000A (Gate-Delete)	
Multimedia TransactionID Object		
Length 0x0008	S-Num 0x02	S-Type 0x01
AMID 0x00005678		
Multimedia SubscriberID Object		
Length 0x0008	S-Num 0x03	S-Type 0x01
SubscriberID 0x01010101		
Multimedia GateID Object		
Length 0x0008	S-Num 0x04	S-Type 0x01
GateID 0x12345678		

13) The CMTS will tear down the access network resources by sending a DSD-REQ to the CM.

0	1	2	3
MAC Management Header			
TransactionID 0x0008		Reserved	
SFID 0x00000001			

14) The CM will acknowledge the session deletion with a DSD-RSP.

0	1	2	3
MAC Management Header			
TransactionID 0x0008		Confirm. Code 0x00	Reserved

15a) The CMTS will complete the Gate-Control transaction with a Gate-Delete-Ack.

0	1	2	3
COPS Header			
Version 0x1	Flags 0x1	Op-Code 0x03	Client-Type 0x800A
Message Length 0x00000034			
COPS Handle Object			
Length 0x0008		C-Num 0x01	C-Type 0x01
Handle 0x00005678			
COPS Report Type Object			
Length 0x0008		C-Num 0x12	C-Type 0x01
Report Type (R-Type) 0x0001		Reserved	
COPS ClientSI Object Header			
Length 0x001C		C-Num 0x09	C-Type 0x01

Multimedia TransactionID Object		
Length 0x0008	S-Num 0x01	S-Type 0x01
TransactionID 0x0002	Gate Command 0x000B (Gate-Delete-Ack)	
Multimedia AMID Object		
Length 0x0008	S-Num 0x02	S-Type 0x01
AMID 0x00005678		
Multimedia GateID Object		
Length 0x0008	S-Num 0x04	S-Type 0x01
GateID 0x12345678		

15b) Also upon the receipt of the DSD-RSP, the CMTS will inform the RKS that the network access resources have been freed by sending a QoS_Release.

0	1	2	3
Accounting-Request Radius Header			
Radius Ven. Spec. 0x1A	Length 0x54	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type (EM Header) 0x01	Length 0x4E
Version 0x0001		BCID 0x3E48	
BCID (cont.) 120820202020203134363 02D3035303030300003DB77			
		Event Message Type 0x0008 (QoS_Release)	

Element Type 0x0002 (CMTS)		Element ID 0x2020202031323334	
		Time Zone 0x302D303530303030	
		Sequence Number 0x0000	
Sequence Number (cont.) 0003		Event Time 0x3230	
Event Time (cont.) 3032313230363030303030302E333030			
Status 0x00000000			
Priority 0x80 (128)	Attribute Count 0x0005		Event Object 0x00
Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x1E	Length 0x06
SF_ID 0x00000001			
Radius Ven. Spec. 0x1A	Length 0x0A	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x32	Length 0x04
Flow_Direction 0x0001 (Upstream)		Radius Ven. Spec. 0x1A	Length 0x0A
Vendor ID 0x0000118B			
Type 0x38	Length 0x04	QoS_Release_Reason 0x0001 (Gate Closed by PS)	
Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x36	Length 0x06
QoS_Usage_Info 0x77777777 (bytes)			

Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x3F	Length 0x06
QoS_Time_Info 0x77777777 (seconds)			

16a) After receiving and recording the QoS_Release Event Message, the RKS acknowledges the message.

0	1	2	3
Accounting-Response Radius Header			

16b) After receiving the Gate-Delete-Ack from the CMTS, the Policy Server will send a Gate-Delete-Ack to complete the Gate-Control transaction.

0		1	2	3
COPS Header				
Version 0x1	Flags 0x1	Op-Code 0x03	Client-Type 0x800A	
Message Length 0x00000034				
COPS Handle Object				
Length 0x0008			C-Num 0x01	C-Type 0x01
Handle 0x00001234				
COPS Report-Type Object				
Length 0x0008			C-Num 0x12	C-Type 0x01
Report Type (R-Type) 0x0001 (Success)			Reserved	
COPS ClientSI Object Header				
Length 0x001C			C-Num 0x09	C-Type 0x01
Multimedia TransactionID Object				
Length 0x0008			S-Num 0x01	S-Type 0x01
TransactionID 0x9998			Gate Command 0x000B (Gate-Delete-Ack)	

Multimedia AMID Object		
Length 0x0008	S-Num 0x02	S-Type 0x01
AMID 0x00005678		
Multimedia GateID Object		
Length 0x0008	S-Num 0x04	S-Type 0x01
GateID 0x12345678		

16c) The Policy Server sends a Policy_Delete Event Message to the RKS to complete the entire process.

0	1	2	3
Accounting-Request Radius Header			
Radius Ven. Spec. 0x1A	Length 0x54	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type (EM Header) 0x01	Length 0x4E
Version 0x0001		BCID 0x3E48	
BCID (cont.) 120820202020203134363 02D303530303030 0003DB77			
		Event Message Type 0x0016 (Policy_Delete)	
Element Type 0x0004 (Policy Server)		Element ID 0x2020202035363738	
		Time Zone 0x302D303530303030	
		Sequence Number 0x0000	

Sequence Number (cont.) 0002		Event Time 0x3230	
Event Time (cont.) 3032313230363030303030302E343030			
Status 0x00000000			
Priority 0x80	Attribute Count 0x0004		Event Object 0x00
Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x3D	Length 0x06
Application_Manager_ID 0x00005678			
Radius Ven. Spec. 0x1A	Length 0x0C	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x34	Length 0x06
Subscriber_ID 0x01010101			
Radius Ven. Spec. 0x1A	Length 0x0A	Vendor ID 0x0000	
Vendor ID (cont.) 118B		Type 0x3A	Length 0x04
Policy_Deleted_Reason 0x0001 (Application Manager Request)		Radius Ven. Spec. 0x1A	Length 0x1C
Vendor ID 0x0000118B			
Type 0x31	Length 0x16	FEID 0x0000	
FEID (cont.) 00000000000005061636B65744361626C65			

17) After receiving and recording the Policy_Delete Event Message, the RKS acknowledges the message.

0	1	2	3
Accounting-Response Radius Header			

11 Issues for future study

The following issues have been identified as topics for future study.

- Error-handling requirements (i.e., mandate specific error codes for specific conditions).
- Gate Control message routing within the Multimedia framework.
- State synchronization requirements (i.e., granularity, scope, frequency, etc.) and protocol mechanism.
- Protocol support for failover and redundancy strategies. Also, how gates should be handled in the event of a failed COPS connection.
- Policy Server rule format and provisioning mechanism: CMS provisioning with Multimedia-specific XML DTD.
- Support for Payload Header Suppression (PHS).
- Gate Control message delivery for failed connections (currently these messages are suppressed).

Appendix I

Background information

I.1 Introduction

This appendix describes an architecture which provides an IP-based platform to support a variety of multimedia applications and services requiring QoS treatment over CableModem access networks. This architecture defines functional components and protocol interfaces that will enable each cable operator to deliver the QoS-enhanced multimedia services that meet their unique business requirements.

Because the architecture is agnostic of the application-level details of particular multimedia offerings, specific provisioning, signalling and operations support system (OSS) functions required to provide a particular service are out of scope. Rather, the focus of IPCablecom Multimedia centers on the delivery of reliable QoS over the access network, specifically addressing the technical issues of policy authorization, QoS signalling, resource accounting, and security.

I.1.1 IPCablecom overview

The IPCablecom project is aimed at defining interface specifications used by the vendor community to develop interoperable equipment capable of providing IP-based voice, video and other high-speed multimedia services over hybrid fibre coax (HFC) cable systems which conform to the CableModem broadband access network Recommendations.

Voice over IP (VoIP) was the first such service identified for delivery over the IPCablecom platform. The current set of IPCablecom Recommendations, known collectively as IPCablecom-T,

define an IPCablecom architecture optimized for the delivery of residential VoIP services. See ITU-T Rec. J.160.

I.1.2 IPCablecom multimedia motivation

Like VoIP, most popular multimedia applications (e.g., online gaming, streaming media, real-time video communication) are sensitive to transmission delay within the network. Further, as new applications emerge that are designed to take advantage of broadband networks, they, too, will present unique bandwidth and latency requirements.

Currently, broadband customers receive multimedia services via best-effort data delivery. This results in an inconsistent online experience that varies in quality based on bandwidth availability and congestion in the network. A network that is able to reserve resources and deliver bandwidth on demand as service requirements dictate will be positioned to provide a wide variety of new services for their customers.

In order to address these needs for VoIP services, IPCablecom currently defines dynamic Quality of Service (DQoS) signalling mechanisms that allow voice applications to request and obtain bandwidth from the CableModem data link layer. The current DQoS framework also supports secure session establishment through endpoint authentication and authorization and a QoS-based usage tracking model. Based on these core capabilities, the IPCablecom architecture is well positioned to support existing and future QoS-enhanced applications and services beyond telephony.

The primary objective of IPCablecom Multimedia is to define the core architectural framework required to support QoS-based multimedia applications. At the heart of this framework are the Quality of Service mechanisms defined in the CableModem and IPCablecom DQoS Specifications. Successful completion of this initiative will provide a strong technical foundation to support specific multimedia service offerings going forward.

I.2 IPCablecom Multimedia objectives and scope

The main objective of IPCablecom Multimedia is to develop a general-purpose architecture that:

- supports a wide range of QoS-enabled services, beyond-voice;
- is based on existing mechanisms defined in IPCablecom-T and CableModem Recommendations;
- requires a minimal set of extensions beyond IPCablecom-T;
- reduces development complexity by eliminating telephony-specific requirements where not applicable (e.g., PSTN interconnect, electronic surveillance, telephony billing models, etc.);
- coexists with the IPCablecom-T architecture in such a way that:
  - IPCablecom Multimedia requirements are sufficient to support a QoS-based multimedia service delivery platform;
  - IPCablecom Multimedia requirements may be added to relevant existing IPCablecom-T functional components;
  - IPCablecom-T requirements may be added to relevant IPCablecom Multimedia functional components;
- supports IPCablecom-T MTAs as "Client Type 2" devices (defined within) in the IPCablecom Multimedia architecture;
- interoperates with the IPCable2Home (ITU-T Rec. J.191) and CableModem (ITU-T Recs J.112 and J.122) architectures.

This clause describes the requirements which have been identified in order to satisfy the above objectives and outlines the scope of the work that will be addressed by the architecture.

I.2.1 Requirements

This architecture outlines the interaction of a variety of network elements, including Client Devices, Application Managers, Policy Servers, CMTSs and Cable Modems. These network elements are formally defined in the Multimedia Framework clause of this Recommendation. However, specific assumptions regarding management authority and trust relationships have made about some of these network elements, and these assumptions are captured below as IPCablecom Multimedia requirements. High-level requirements addressing QoS signalling, resource management, event messaging and security are also included in this clause.

IPCablecom Multimedia is application signalling protocol-agnostic concerning interaction between the Client Device and Application Manager. It is understood that the Client Device and Application Manager may support a variety of application and signalling protocols (e.g., HTTP, SIP, H.323, DCS, NCS, etc.).

Client Devices in the IPCablecom Multimedia architecture:

1) reside directly on the operator access network, or within the home;
2) may be standalone devices or may contain an embedded CableModem; and
3) are considered untrusted network elements and, as such, some form of authentication of the user, application, or application messaging may be required by the operator network.

Application Managers in the IPCablecom Multimedia architecture:

1) reside on the operator managed network;
2) are managed by the operator; and
3) are responsible for ensuring that clients requesting service from the operator network are authorized to receive that service.

Policy Servers in the IPCablecom Multimedia architecture:

1) reside in the operator managed network;
2) are managed by the operator; and
3) are responsible for making QoS-related policy decisions based on operator-defined policy rules.

CMTSs in the IPCablecom Multimedia architecture are responsible for enforcing QoS-related policy decisions.

QoS signalling and resource management requirements

- Dynamic resource request mechanisms must be defined, including:
  - access to all CableModem QoS scheduling models;
  - time-restricted resource requests;
  - volume-restricted resource requests.
- Single-phase and two-phase resource reservation models must be supported.
- Unidirectional reservations must be supported; support for bidirectional reservations should be allowed.
- Application Managers may initiate QoS reservation requests on behalf of Client Devices.
- The architecture must provide a means to detect client and/or server failures and reclaim associated resources.

Event message information collection requirements

- A comprehensive set of event messages must be defined to track per-flow resource usage, including:
  - policy event denoting a request for access-network resources, subject to operator-defined policy rules;
  - policy event denoting the release of access-network resources;
  - QoS events denoting reservation, commitment, and release of QoS resources;
  - additional event(s) supporting per-flow resource usage based on volume (metered packet counts).
- The following information should be contained in the messages:
  - source of request (e.g., subscriber or service provider);
  - characteristics of the requested resources;
  - policy authorization decision.

Security requirements

- Security is required and must be defined for relevant interfaces.
- Clients that initiate QoS signalling may require some form of authentication of the user or the application.

I.2.2 Scope

The following items outline the current, initial phase scope of the IPCablecom Multimedia initiative:

- The architecture will address network elements that reside:
  1) on the access network; or
  2) within a single operator's managed IP network.
- The architecture will define the protocols and interfaces necessary to support policy authorization, QoS admission control, resource accounting, and security mechanisms.
- The architecture will not address application-specific issues (e.g., service provisioning, signalling, billing, etc.).
- The architecture will not address provisioning and OSS requirements for IPCablecom Multimedia network elements.
- The architecture will focus on QoS management between the CMTS and CM.
- The architecture will not preclude the delivery of multicast services, even though it will not explicitly address multicast considerations.
- The architecture will not address Network Address Translation (NAT) traversal and interoperability requirements at the present time.
- The architecture will not define end-to-end QoS requirements in the present phase.
- The architecture will provide support for "Client Type 1" and "Scenario 1" (as defined within) in the present phase. For completeness and in anticipation of future elaboration, this appendix describes all three client types and all three service scenarios.
- The architecture will not provide dynamic topology discovery (i.e., relationships among Application Managers, Policy Servers, CMTSs, RKSs, etc.) in the present phase.
- The architecture will not address Client authentication by the Application Manager.
- The architecture will not address specific mechanisms by which the Policy Server obtains and manages policy rules.

- The architecture will not support the collection of application or service-specific events for incorporation into the resource usage audit trail.

I.3 IPCablecom multimedia framework

To facilitate the delivery of quality broadband multimedia applications requiring QoS guarantees, the multimedia framework offers general-purpose QoS functionality based on mechanisms defined in the core IPCablecom-T specifications. In support of this objective, several key network elements have been identified and profiled. Figure I.1 presents the IPCablecom Multimedia components which reside within the operator's managed IP network.

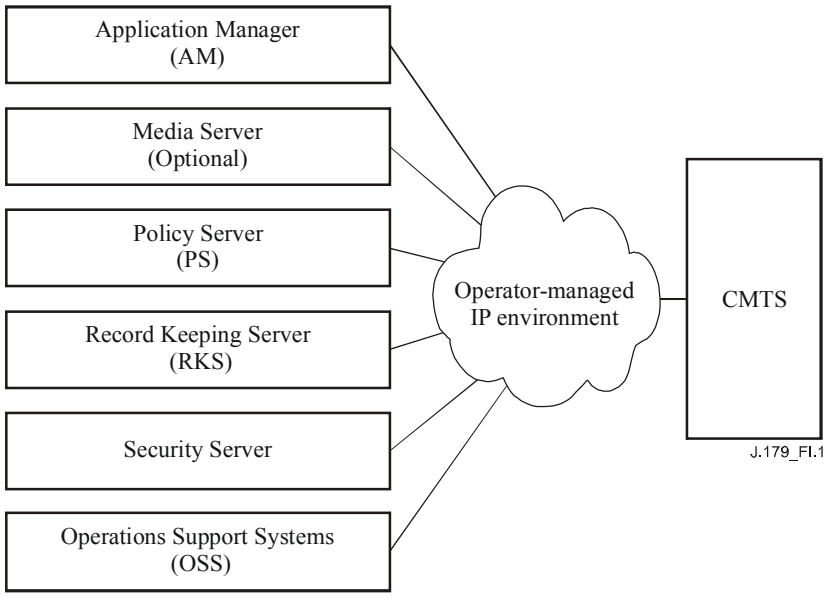

Figure I.1/J.179 – Operator multimedia network elements

In addition to a CMTS facilitating parameter-based QoS capabilities, the operator multimedia network architecture consists of a server farm which may be further divided into the following areas:

- An Application Manager and (optional) Media Server hosting a QoS-enabled application.
- A policy administration framework providing QoS authorization and admission control in support of per-flow network resource management.
- An event messaging subsystem used to monitor and record resource usage information.

Operations support systems to perform provisioning, network management, and monitoring functions may also be included in the operator multimedia network configuration, though these elements fall outside the scope of the current architecture.

I.3.1 IPCablecom multimedia architecture reference model

In addition to the elements residing within the operator head-end network, a number of client devices located on the customer premises have also been defined in order to complete the model. Figure I.2 shows the IPCablecom Multimedia architectural framework and identifies key interfaces between the components. These interfaces have been tagged with identifiers which will be referenced in the discussion that follows.

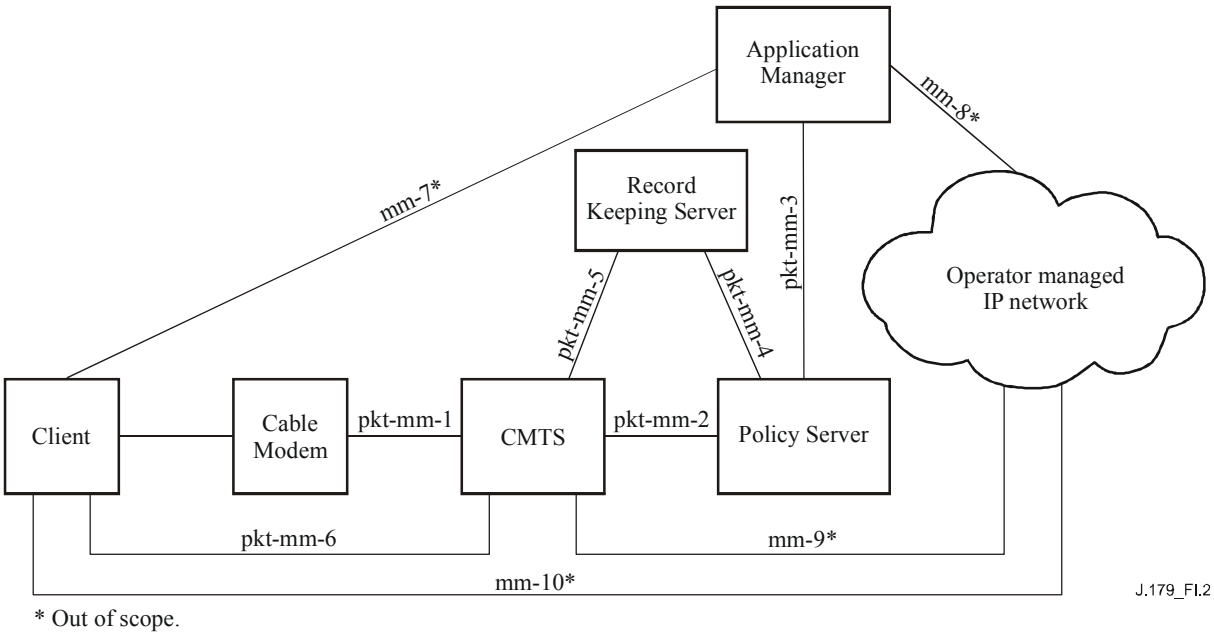

Figure I.2/J.179 – IPCablecom multimedia architectural framework

In this architecture, Clients may or may not support the IPCablecom Multimedia framework. Clients that support the framework and its QoS signalling mechanisms explicitly issue requests for network resources on their own behalf, which are authorized at the head-end by the Policy Server. Clients that do not support the QoS signalling mechanisms have network resource requests proxied on their behalf by an Application Manager, with which they interact.

Regardless of the QoS signalling method, access-network resource requests are always subject to policy control, which is enforced at the Cable Modem Termination System (CMTS) serving as a Policy Enforcement Point (PEP) and defined at the Policy Server (PS), serving as a Policy Decision Point (PDP).

- Policy decisions may be pulled from the Policy Server by the CMTS. In this case, the CMTS typically issues a policy request as the result of a currently unauthorized, yet conformant QoS resource request. Based on the resulting decision, the original QoS request is serviced or rejected.
- Alternatively, policy decisions may be pushed to the CMTS by the Policy Server. In this case, the Policy Server shall install a policy decision in advance of a QoS resource request based on a policy request originating from an Application Manager. An Application Manager generates such a request, based on Client interaction (through some unspecified signalling mechanism).

Both the Policy Server and the CMTS generate event messages to track QoS requests and usage. These event messages are sent to a Record Keeping Server (RKS) where they may be used for billing or other accounting purposes.

Table I.1 summarizes the interfaces presented in Figure I.2. Interfaces that are defined by this Recommendation are labelled "pkt-mm-x", while other interfaces, which are included for completeness, are labelled "mm-x".

Table I.1/J.179 – IPCablecom multimedia interfaces

Interface	Description	Notes
pkt-mm-1	CMTS – CM	The CM may request QoS from the CMTS via CableModem DSx signalling. Alternatively, the CMTS may instruct the Cable Modem (CM) to set up, tear down or change a CableModem service flow in order to satisfy a QoS request, again via DSx signalling.
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. Additionally, the AM may also request that the PS proxy QoS requests to 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 adds, changes, deletes, 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. This interface is out of scope for this Recommendation.
mm-7	Client – AM	This interface may be used by the client to interact with the AM and to indirectly request and manage QoS resources. This interface is out of scope for this Recommendation.
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 Recommendation.
mm-9	CMTS – operator-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 Recommendation.
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 Recommendation.

I.3.2 Multimedia components

In this clause, we expand on the previous discussion regarding the architectural framework by providing additional detail for each of the network elements.

I.3.2.1 Client

A multimedia client is a logical entity that may send or receive data. IPCablecom Multimedia defines three different client types, which differ in how the client signals QoS and how the policy decisions associated with the QoS are installed in the CMTS.

Client Type 1 represents existing "legacy" endpoints (e.g., PC applications, gaming consoles) which lack specific QoS awareness or signalling capabilities. This Client knows nothing about CableModem, IPCable2Home, or IPCablecom messaging, and hence no related requirements can be placed upon it. Such clients may range from simple analog audio and video presentation devices to complex networked peripherals and consumer electronics, such as set-top boxes or gaming consoles. This Client communicates with an Application Manager to request service, and does not request QoS resources directly from the operator access network.

Client Type 2 is similar to an IPCablecom-T telephony MTA in that it supports QoS signalling based on IPCablecom DQoS. This Client is aware of IPCablecom Multimedia QoS, and communicates with an Application Manager to request service and obtain a token for access-network resources. The client then presents this token when requesting QoS resources from the access network (pkt-mm-1, pkt-mm-6).

Client Type 3 requests QoS based on RSVP without Application Manager interaction. This Client is aware of IETF standards-based RSVP and uses this protocol to request QoS resources from the access network directly from the CMTS.

I.3.2.2 Policy Server

The policy-management framework for the IPCablecom Multimedia initiative is based upon the work of the IETF's Resource Allocation Protocol (RAP) working group. As defined and described in RFC 2753, the Policy Server (PS) network element implements operator-defined authorization and resource-management procedures. In addition to the requested resource parameters and the status of available resources, policy decisions may involve client identity and associated profile information, application parameters, security considerations, time-of-day, etc. Also, particular operators may elect to deploy multiple Policy Servers and delegate certain policy decisions among these servers in order to satisfy scalability and fault-tolerance requirements.

The main functions of the Policy Server include:

- A policy decision request mechanism, invoked by Application Managers (pkt-mm-3, push model) or CMTSs (pkt-mm-2, pull model).
- A policy decision delivery mechanism, used to install policy decisions on the CMTS (pkt-mm-2).
- A mechanism to allow for the proxying of QoS management messages to the CMTS on behalf of the Application Manager (for clients who do not have native QoS signalling capabilities).
- An event recording interface to a Record Keeping Server (pkt-mm-4) used to log policy requests, which may also be correlated with network resource usage records.

The Policy Server may support two different models for installing policy decisions on the CMTS:

- The Policy Server may install (push) a policy decision on the CMTS before a QoS reservation request arrives at the CMTS.
- The CMTS may request (pull) a policy decision from the Policy Server when a QoS reservation request arrives at the CMTS.

Policy rules may contain the following information:

- Rules defining resources authorized by the policy server:
  - per-service;
  - per-subscriber;
  - bandwidth (specified using token-bucket parameters);
  - latency guarantees;
  - policy expiration times;
  - policy volume limits.
- Rules defining scarcity/value of bandwidth based on time of day.
- Pre-emption rules.

At a minimum, under the "push" scenario the policy server must perform the following functions:

- Authenticate and verify policy messages from Application Managers.
- Process policy messages based on operator-defined rules.
- Resolve the correct identity of the CMTS to which policy is to be pushed.
- Communicate policy decisions and other messages securely with the CMTS.
- Send event messages tracking these requests to the RKS.

At a minimum, under the "pull" scenario the policy server must perform the following functions:

- If an Application Manager is involved in the service, authenticate and verify policy messages from the Application Manager.
- Communicate policy decisions and other messages securely with the CMTS.
- Process policy messages based on operator-defined rules.
- Send event messages tracking these requests to the RKS.

The policy server may perform the following additional functions:

- Track resource usage based on internally-maintained state information (e.g., timers).
- Track authorized resources on per-user, per-service, or aggregate basis.

I.3.2.3 Cable Modem Termination System

IPCablecom Multimedia provides access to the full set of CMTS upstream scheduling algorithms as defined in the CableModem Recommendations. Specifically, the architecture defines an IPCablecom Multimedia "Traffic Profile" that provides a layer of abstraction from associated CableModem Scheduling Types (UGS, UGS/AD, etc.). Further, the telephony-specific features and assumptions found in the IPCablecom-T DQoS specification will be generalized to provide a QoS infrastructure that can be used by multiple types of clients and applications.

The CMTS supports both single and two-phase reservation models for managing access-network resources. In the two-phase model, access-network resources are initially reserved, then committed for use as they are required at a later time. The CMTS also supports a single-phase reservation model in which access-network resources are simultaneously reserved and committed for immediate use.

The CMTS sets up the relevant service flow(s) on the CableModem access network via pkt-mm-1. The CMTS sends event messages for QoS resource reservations and usage to a Record Keeping Server via pkt-mm-5 interface identifier. Finally, the CMTS monitors QoS-based service flows and accounts for them as defined in the (optional) Account Management subsystem in the CableModem Recommendations.

I.3.2.4 Application Manager

The Application Manager plays a coordinating role involving application signalling and semantics as well as interaction with the IPCablecom Multimedia policy framework, as outlined during the previous discussion of the Policy Server element. Note that an Application Manager may be co-hosted with a Media Server or, under a divided model, the two elements may exist separately.

The Application Manager interfaces with a client via mm-7. Based on its knowledge of particular service offerings, the Application Manager must infer or define the particular QoS parameters necessary to deliver the service to Client Type 1. Once this information has been ascertained, the Application Manager sends a policy request to the Policy Server via pkt-mm-3. If necessary, the Application Manager may use mm-8 to synchronize with a Media Server.

Client Type 2 also interacts with the Application Manager and communicates service request information via mm-7. Again, the Application Manager must infer the QoS parameters necessary to deliver the service to Client Type 2. The Application Manager sends a policy request to the Policy Server via pkt-mm-3. Upon successful authorization, the Application Manager receives a token from the Policy Server and sends the token to the Client via mm-7. If necessary, the Application Manager may use mm-8 to synchronize with a Media Server.

Client Type 3 does not require an Application Manager, although the presence of an Application Server in sophisticated service delivery scenarios is quite likely.

I.3.2.5 Record Keeping Server

The Record Keeping Server (RKS) receives event messages indicating the usage of access-network QoS resources. The RKS interfaces with the Policy Server (pkt-mm-4) and the CMTS (pkt-mm-5). The RKS does not receive application-specific information directly from the Application Manager. Instead, application-specific information may be included in an event message as opaque data sent from the Application Manager to the Policy Server and embedded in the policy request event message to the RKS.

I.4 Proxied QoS with Policy Push (Scenario 1)

As noted above, three architectural scenarios have been identified in support of the three client types. The "proxied-QoS with policy-push" authorization model (Scenario 1) supports Client Type 1, which does not itself support native QoS signalling mechanisms. A high-level overview of the element interaction involved in this scenario is shown in Figure I.3.

The Client requests an application-specific service by sending a "Service Request" to the Application Manager. Upon receipt of this request, the Application Manager determines the QoS needs of the requested service and sends a "Policy Request" to the Policy Server. The Policy Server in turn validates the "Policy Request" against the operator-defined policy rules and, if the decision is affirmative, sends a "Policy Set" message to the CMTS. The CMTS performs admission control on the requested QoS envelope (verifying that adequate resources are available to satisfy this request), installs the policy decision, and (eventually) establishes the service flow(s) with the requested QoS levels.

It should be noted that the actual management of the service flow(s) (i.e., add, change, delete requests) may be closely controlled and monitored by the Application Manager through extensions to the basic signalling mechanisms outlined here for the installation of the policy decision. In Scenario 1, there is no direct communication between the Client and the CMTS.

Note that the interface between the Client and Application Manager, including the details of the "Service Request", are out of scope for this Recommendation. It is possible that the Client has no knowledge of QoS and simply requests service (e.g., the user wants to play a multi-player game with a friend) from the Application Manager in the "Service Request" message. It is also possible that the Client has full knowledge of its QoS requirements (e.g., the user requests guaranteed

128 kbit/s service for access to his corporate VPN, secured by IPSec) and communicates this additional information in the "Service Request". The mechanism by which the Application Manager determines the QoS requirements for the requested service is out of scope for this architecture.

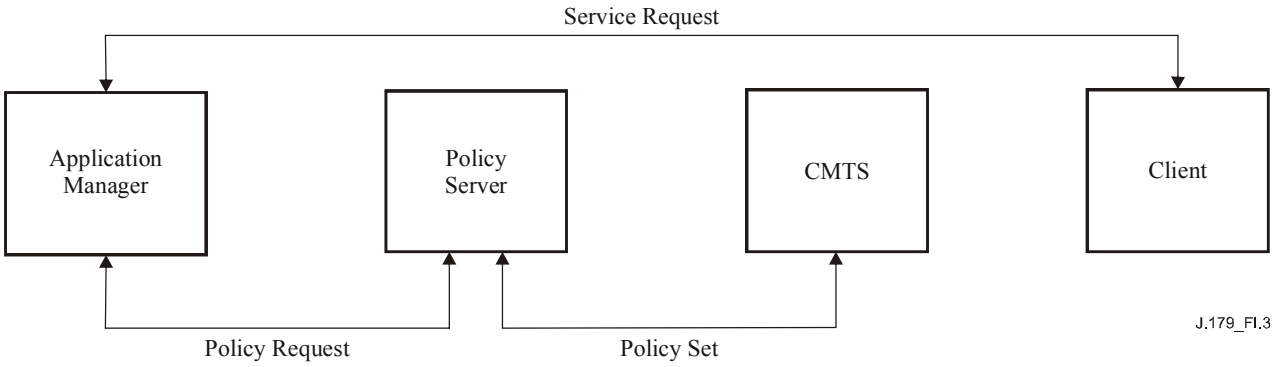

Figure I.3/J.179 – Authorization framework for Scenario 1

Under Scenario 1, the CMTS supports a single-phase resource reservation model, as shown below in Figure I.4, to enable immediate activation and usage of access-network resources by the Client. (A two-phase resource reservation model is also supported under this scenario as outlined later in this clause.)

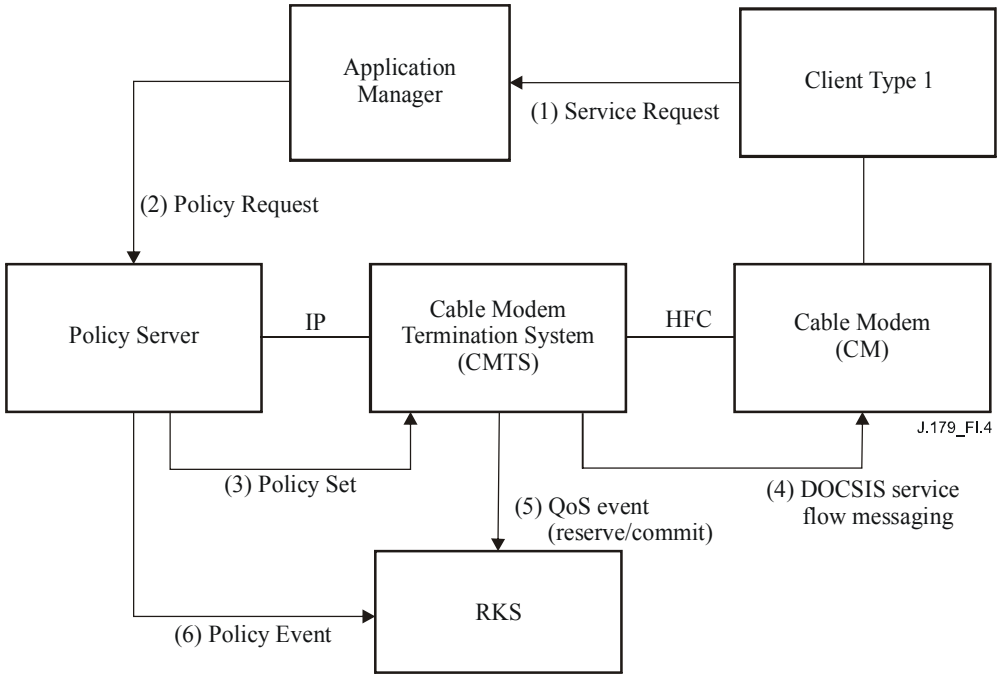

Figure I.4/J.179 – Single-phase resource reservation model for Scenario 1

Based on this single-phase messaging sequence, Table I.2 provides a high-level summary of each of these messages. Details specific to protocol messages and objects have been deferred to the respective IPCablecom Multimedia specifications.

Table I.2/J.179 – Single-phase resource reservation message details for Scenario 1

Message	Function	Fields	Protocol candidate	Comments
(1) Service Request	The Client requests service from the Application Manager.	<none>	Out of scope for IPCablecom Multimedia	This protocol should support the authentication of both Client and Application Manager. Also, the protocol should provide sufficient information for the Application Manager to convey the QoS needs of the requested service.
(2) Policy Request	The Application Manager requests QoS setup on behalf of the client.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint	Gate-Control (COPS)	The Policy Server uses operator-managed policy rules to allow or disallow the request.
(3) Policy Set	The Policy Server sends a message to the CMTS, installing its policy decision and requesting service flow establishment.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS)	Gate-Control (COPS)	Under the single-phase model, this request is for authorization, reservation and commitment of the QoS resources.
(4) CableModem Messaging	The CMTS establishes QoS-enhanced service flows.	CableModem Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier	CableModem DSx Messaging	The QoS functions here are based on the mechanisms defined in the CableModem RFI specification.
(5) QoS Event	The CMTS generates the proper event message, indicating QoS usage and other billing parameters	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS) Policy Decision, Service Usage Data, Time-of-Day	Event-Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.

Table I.2/J.179 – Single-phase resource reservation message details for Scenario 1

Message	Function	Fields	Protocol candidate	Comments
(6) Policy Event	The Policy Server generates the proper event message, indicating the Policy Request and action taken.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS), Policy Decision	Event-Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.

The information summarized in the Fields column in Table I.2 is intended to provide an example of the type of information carried by each message. The details of each protocol message have been deferred to the appropriate specification documents.

Scenario 1 also supports a two-phase resource reservation model, as shown below in Figure I.5. Here, the Application Manager first asks for access-network QoS resources to be authorized and reserved. Once these resources have been reserved, the Application Manager may continue its dialogue with the Client concerning the service. When appropriate, the Application Manager asks for access-network QoS resources to be committed. This two-phase reserve/commit model guarantees that access-network resources are available before offering service to the Client.

Note that acknowledgements for each of the messages shown are not explicitly included, but are implied. Each acknowledgement message can only be sent once the final outcome of the corresponding request is known. This is particularly important in the sequencing of acknowledgements of messages 5 (DOCSIS Reserve), 3 (Policy Set), and 2 (Policy Request) since the Application manager will likely wait for successful confirmation of the reservation phase before continuing its dialogue with the Client and eventually committing the resources.

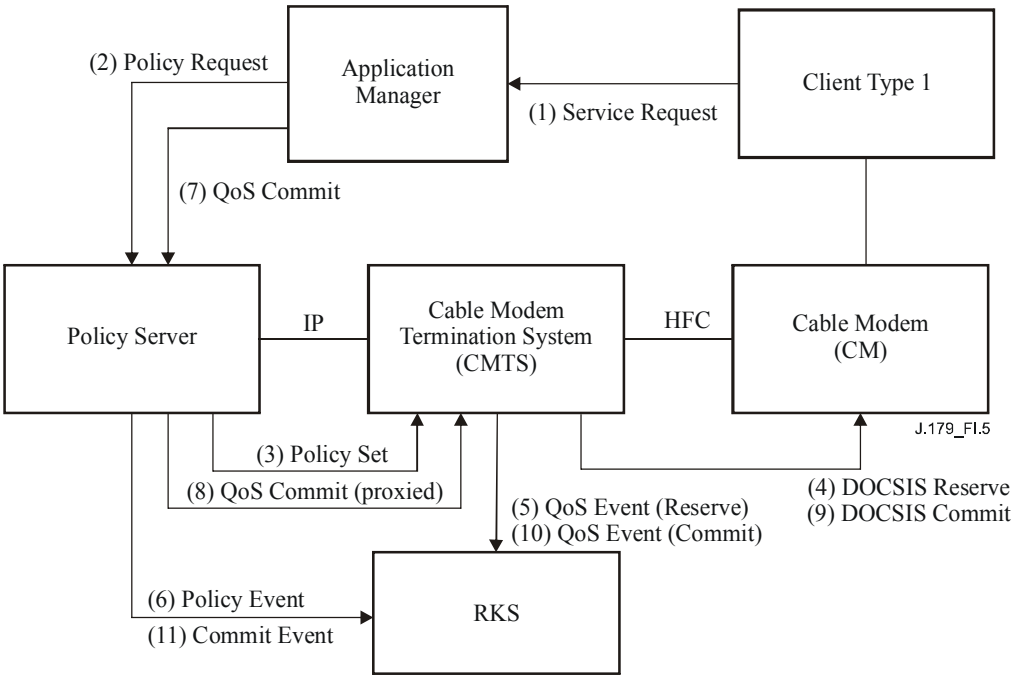

Figure I.5/J.179 – Two-phase resource reservation model for Scenario 1

A summary of the messages outlined in Figure I.5 is provided in Table I.3 below. Note that messages (7-10) are added in support of the commit signalling phase.

Table I.3/J.179 – Two-phase resource reservation message details for Scenario 1

Message	Function	Fields	Protocol candidate	Comments
(1) Service Request	The Client requests service from the Application Manager.	<none>	Out of scope for IPCablecom Multimedia	This protocol should support the authentication of Client and Application Manager. Also, the protocol should provide sufficient information for the Application Manager to convey the QoS needs of the requested service.
(2) Policy Request	The Application Manager requests QoS setup on behalf of the client.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint	Gate-Control (COPS)	The Policy Server uses operator-managed policy rules to allow or disallow the request.
(3) Policy Set	The Policy Server sends a message to the CMTS, installing its policy decision and requesting service flow reservation.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS)	Gate-Control (COPS)	Under the two-phase model, this request is for authorization and reservation of the QoS resources.
(4) DOCSIS Reserve	The CMTS establishes QoS-enhanced service flows and places them in an "admitted" state.	CableModem Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier	CableModem DSx Messaging	The QoS functions here are based on the mechanisms defined in the CableModem RFI specification. Reserved resources remain inactive and may be used by best-effort traffic on other flows until committed.
(5) QoS Event	The CMTS generates the proper event message, indicating the QoS reservation and other billing parameters.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS) Policy Decision, Service usage data, Time-of-day	Event-Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.

Table I.3/J.179 – Two-phase resource reservation message details for Scenario 1

Message	Function	Fields	Protocol candidate	Comments
(6) Policy Event	The Policy Server generates the proper event message, indicating the Policy Request and action taken.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS) Policy Decision	Event-Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.
(7) QoS Commit	The AM signals to commit the QoS resources.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Policy Identifier	Gate-Control (COPS)	The AM commitment may depend on further messaging with the client.
(8) QoS Commit (Proxied)	The Policy Server receives the AM request and proxies to the CMTS.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Policy Identifier	Gate-Control (COPS)	Even though the PS may apply policy rules during the commit phase, it is generally assumed that the reserved bandwidth may be committed at any time by the AM.
(9) DOCSIS Commit	The CMTS places the service flow in the "active" state.	CableModem Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier, Service Flow ID	CableModem DSx Messaging	The QoS functions here are based on the mechanisms defined in the CableModem RFI specification.
(10) QoS Event (Commit)	The CMTS generates the proper event message, indicating the QoS usage and other billing parameters.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS) Policy Decision, Service usage data, Time-of-day	Event Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.

Table I.3/J.179 – Two-phase resource reservation message details for Scenario 1

Message	Function	Fields	Protocol candidate	Comments
(11) Commit Event	The Policy Server generates the proper event message, indicating the QoS commit and action taken.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Policy Identifier	Event Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.

Once QoS resources have been successfully authorized, reserved, and committed on the access network, these resources are monitored for activity at the CMTS. In general, a soft-state model is used in which periodic refresh messages are required during periods of inactivity on reserved and committed service flows. If activity timers expire without a refresh, the associated resources may be recovered by the CMTS. This provides for network resilience in the case of a failed endpoint.

A more standard resource recovery sequence is also provided under this scenario in which the Application Manager signals the Policy Server when the service session has ended. The Policy Server responds by generating an event message, which is sent to the RKS, and issuing a directive to the CMTS to tear down the associated service flow(s) and recover associated resources. Regardless of whether a service flow times out due to inactivity or is explicitly deleted, a robust audit trail is maintained, tracking actual resource usage via event messages produced at the CMTS and sent to the RKS.

I.4.1 Example: Web-based bandwidth on demand

One example of how the mechanisms in Scenario 1 may be applied in a service delivery context is the case of an operator-hosted secure website, which would allow subscribers to request bandwidth reservations on-demand.

Assume, for example, that a subscriber's normal service is rate limited to 128 kbit/s downstream and 128 kbit/s upstream. While this level of service may be adequate for most usage, there may be times when the application the subscriber is using requires more bandwidth or has different QoS needs. If the user decides to use the bandwidth on-demand service to make temporary changes to their normal service level, they would simply login to the operator's website (Application Manager) and request a temporary service upgrade.

One possible motivation for such a request would be the desire to stream high bit rate media files from a content provider. In this case, the subscriber might explicitly request 512 kbit/s downstream minimum reserved rate service for the next three hours. Alternatively, the exact QoS needs of the application might be opaque to the subscriber, who might simply request a given three-hour video clip (which, unbeknownst to the subscriber, happens to be encoded at 512 kbit/s). Either way, this exchange represents the subscriber's "Service Request" to the Application Manager.

In either case, the Application Manager would present a "Policy Request" for 512 kbit/s minimum reserved rate service for three hours to the Policy Server on behalf of the subscriber. The Policy Server would then apply its own authorization criteria and, if the request was approved, ask the CMTS (through a "Policy Set") to provide the bandwidth for the subscriber. The CMTS would, in turn, perform internal admission control and establish the QoS using the CableModem messaging, tracking this process through a QoS event message.

I.4.2 Example: Online gaming via networked consoles

Alternatively, consider a case in which two gaming consoles wish to engage one another via a network tunnel. In this example, two users may typically network their consoles only if they are co-located. However, special software installed on each user's Personal Computer, collocated on a local network and serving as a proxy for the remote console, enables networking such that two gaming consoles no longer need to be co-located. The only problem with this novel approach is that the resulting tunnel requires sufficient QoS so that the gaming consoles can be played as if they were co-located on a high-speed network.

In this scenario, the user(s) would connect to the Application Manager via the PC(s) that tunnel their packets. Through application-specific messaging, they authenticate themselves and indicate their request to play a game with one another. The Application Manager grants the request, and generates the "Policy Request(s)" on behalf of the user(s). The Policy Server makes its decision and relays the message as a "Policy Set" to the CMTS. The CMTS performs admission control and enables access network QoS between the PCs for the gaming tunnel using CableModem messaging. From this point on, the gaming consoles may exchange packets without knowing that they are not co-located. Note that Event Messaging has been omitted from this example for simplicity.

In this hypothetical example, if the users reside on separate HFC nodes, it is the operator's responsibility to ensure that backbone QoS to and from the CMTS is handled properly at the level that their policy and service agreements require. Figure I.6 provides a graphical illustration of this example for the simplified case in which both users receive service from on a single CMTS.

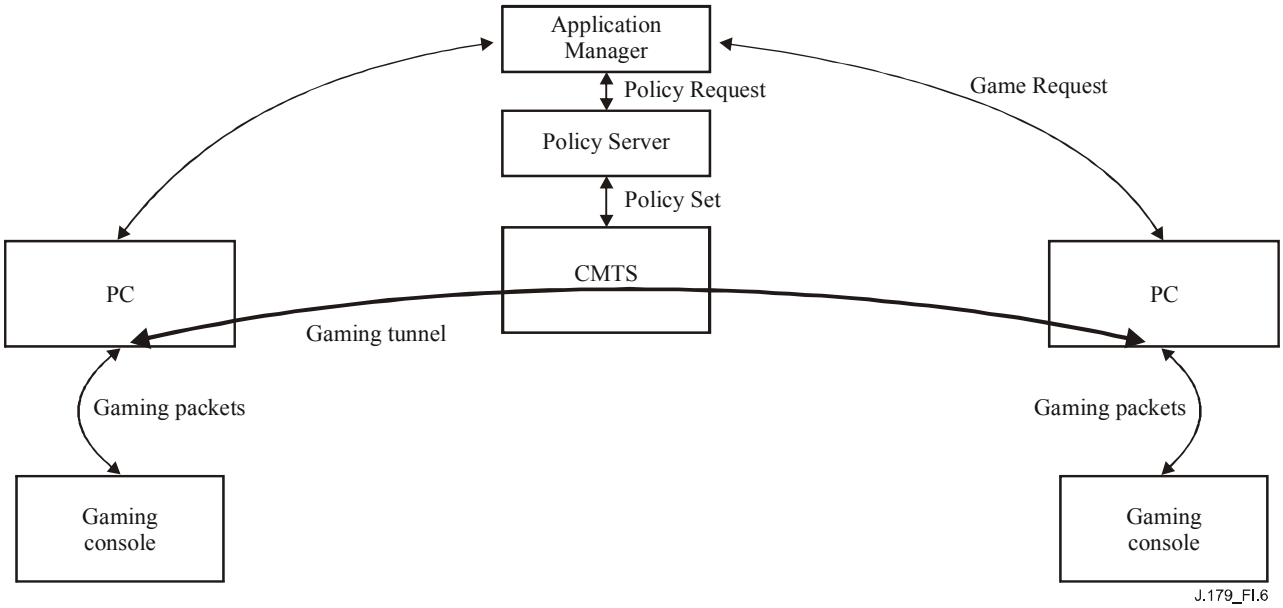

Figure I.6/J.179 – Gaming consoles networked via a QoS-enhanced IP tunnel

I.5 Client-requested QoS with policy-push (Scenario 2)

Scenario 2's "client-requested QoS with policy-push" model supports Client Type 2, which is capable of signalling and managing its own QoS resources but requires prior authorization of these requests via an Application Manager. In this scenario, the policy authorization and QoS reservation model closely resembles the IPCablecom-T telephony model defined in the DQoS specification. The Policy Server pushes policy to the CMTS in a manner similar to that in which the Gate Controller sends policy to the CMTS via COPS. Client Type 2 uses either CableModem DSx or RSVP+ messaging similar to MTA devices in IPCablecom-T.

A high-level overview of Scenario 2 is shown in Figure I.7. Note the similarities to the authorization framework outlined for Scenario 1. Here again, the client requests an application-specific service by sending a "Service Request" to the Application Manager. The Application Manager then determines the QoS needs of the requested service and sends a "Policy Request" to the Policy Server. The "Policy Request" contains the "authorized envelope" or maximum QoS allowed for the client. The Policy Server in turn validates the "Policy Request" against the operator-defined policy rules and, if the decision is affirmative, sends a "Policy Set" to the CMTS. The CMTS performs admission control over the requested QoS and installs the policy authorization. As in Scenario 1, event messages are generated by the Policy Server and the CMTS and sent to the RKS. The Policy Server records an event any time it makes a decision, or updates its state, and the CMTS tracks QoS resource maintenance and usage.

In Scenario 2 and unlike Scenario 1, there is direct communication between the Client and the CMTS in order to add, change and delete resource reservations. After the CMTS receives the "Policy Set" message from the Policy Server, the Client may request QoS directly from the CMTS using the previously noted QoS signalling mechanisms. The Client may also change the QoS dynamically as long as the requested QoS is within the "authorized envelope" approved by the Policy Server. The advantage of this method is that the Application Manager does not have to negotiate Client bandwidth utilization, which is a very useful factor when the Client's QoS needs change dynamically.

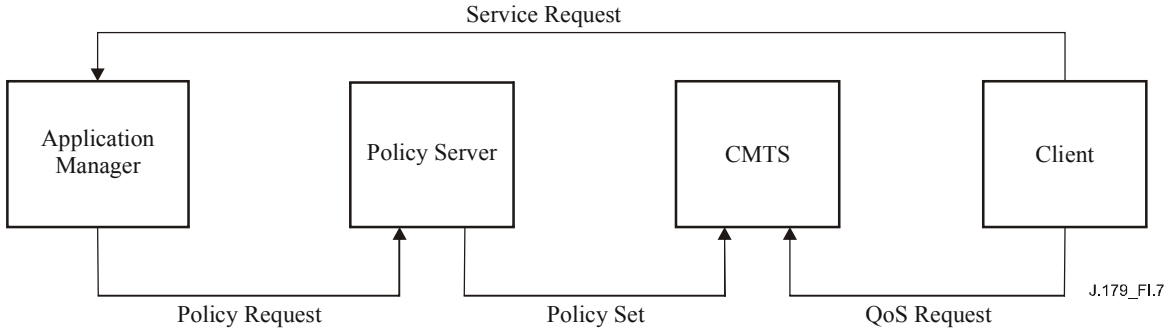

Figure I.7/J.179 – Authorization framework for Scenario 2

As in the previous scenario, Scenario 2 (as shown in Figure I.8) supports a single-phase resource reservation model to enable immediate activation and usage of access-network resources by the client.

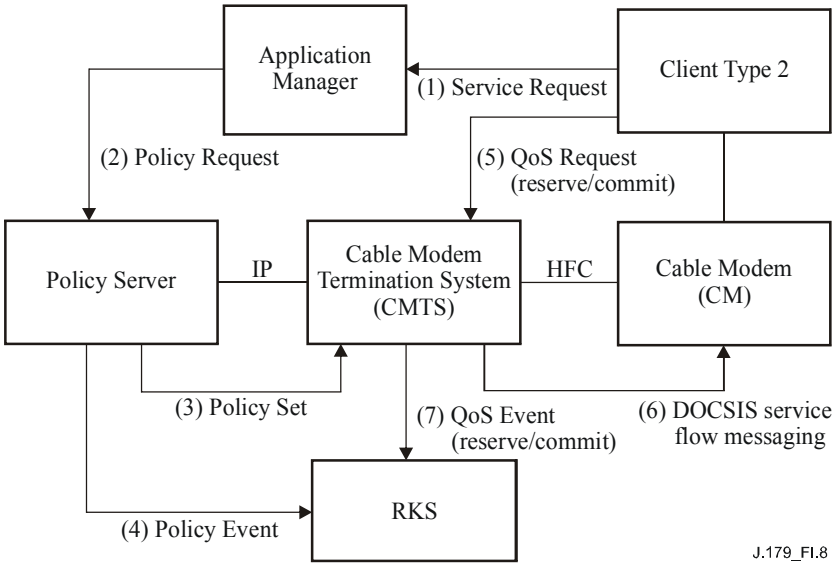

Figure I.8/J.179 – Single-phase resource reservation model for Scenario 2

Based on this single-phase messaging sequence, Table I.4 provides a high-level summary of each of these messages.

Table I.4/J.179 – Single-phase resource reservation message details for Scenario 2

Message	Function	Fields	Protocol candidate	Comments
(1) Service Request	The Client requests service from the Application Manager.	<none>	Out of scope for IPCablecom Multimedia	This protocol should support the authentication of Client and Application Manager. Also, the protocol should provide sufficient information for the Application Manager to convey the QoS needs of the requested service.
(2) Policy Request	The Application Manager requests QoS authorization on behalf of the Client.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint	Gate-Control (COPS)	The Policy Server uses operator-managed policy rules to allow or disallow the request.

Table I.4/J.179 – Single-phase resource reservation message details for Scenario 2

Message	Function	Fields	Protocol candidate	Comments
(3) Policy Set	The Policy Server sends a message to the CMTS, installing its policy decision.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS)	Gate-Control (COPS)	In this scenario, this request is for authorization only.
(4) Policy Event	The Policy Server generates the proper event message, indicating the Policy Request and action taken.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS) Policy Decision	Event-Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.
(5) QoS Request (Reserve/ Commit)	The Client requests that QoS resources are reserved and immediately committed for use.	Bandwidth and Latency Parameters, Traffic Classifier	CableModem DSx or RSVP+	The Client may directly establish CableModem service flows via DSx messaging or may issue RSVP+ messages to establish these flows.
(6) DOCSIS Messaging	The CMTS establishes QoS-enhanced service flows and places them in an "active" state.	CableModem Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier	CableModem DSx Messaging	This step is only necessary if RSVP+ signalling was provided to the CMTS in the previous message, otherwise service flows have already been set up and activated via CableModem DSx messaging. The QoS functions here are based on the mechanisms defined in the CableModem Recommendations.

Table I.4/J.179 – Single-phase resource reservation message details for Scenario 2

Message	Function	Fields	Protocol candidate	Comments
(7) QoS Event	The CMTS generates the proper event message, indicating the QoS usage and other billing parameters.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS) Policy Decision, Service usage data, Time-of-day	Event-Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.

The CMTS also supports a two-phase resource reservation model, as shown in Figure I.9. In this model, the Client first asks for access-network QoS resources to be reserved. Once these resources have been reserved, the Client then signals for these QoS resources to be committed. The two-phase reserve/commit model guarantees that access-network resources are available before offering services to the client.

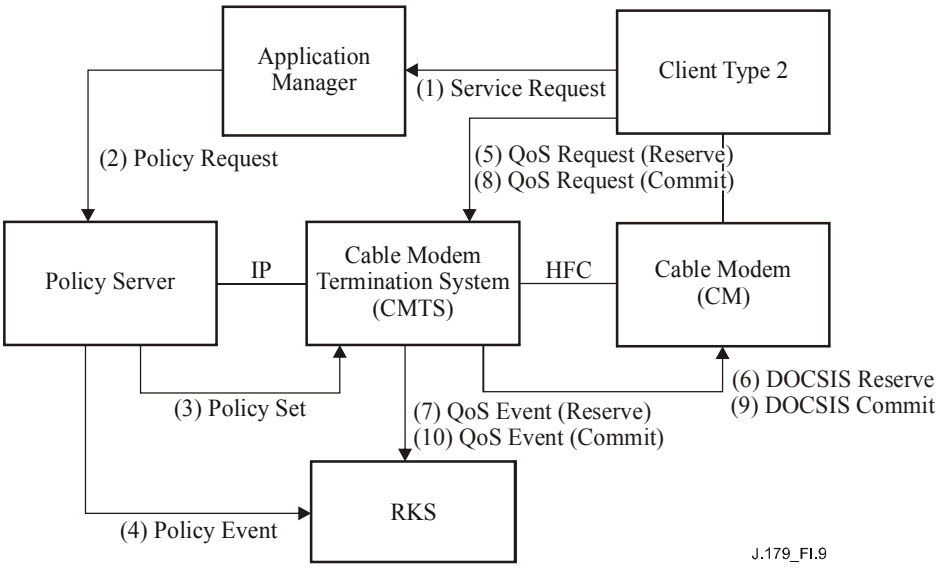

Figure I.9/J.179 – Two-phase resource reservation model for Scenario 2

Table I.5/J.179 – Two-phase resource reservation message details for Scenario 2

Message	Function	Fields	Protocol candidate	Comments
(1) Service Request	The Client requests service from the Application Manager.	<none>	Out of scope for IPCablecom Multimedia	This protocol should support the authentication of Client and Application Manager. Also, the protocol should provide sufficient information for the Application Manager to convey the QoS needs of the requested service.
(2) Policy Request	The Application Manager requests QoS authorization on behalf of the Client.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint	Gate-Control (COPS)	The Policy Server uses operator-managed policy rules to allow or disallow the request.
(3) Policy Set	The Policy Server sends a message to the CMTS, installing its policy decision.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS)	Gate-Control (COPS)	In this scenario, this request is for authorization only.
(4) Policy Event	The Policy Server generates the proper event message, indicating the Policy Request and action taken.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS) Policy Decision	Event-Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.
(5) QoS Request (Reserve)	The Client requests that QoS resources are reserved.	Bandwidth and Latency Parameters, Traffic Classifier	CableModem DSx or RSVP+	The Client may directly establish CableModem service flows via DSx messaging or may issue RSVP+ messages to establish these flows.

Table I.5/J.179 – Two-phase resource reservation message details for Scenario 2

Message	Function	Fields	Protocol candidate	Comments
(6) DOCSIS Reserve	The CMTS establishes QoS-enhanced service flows and places them in an "admitted" state.	CableModem Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier	CableModem DSx Messaging	This step is only necessary if RSVP+ signalling was provided to the CMTS in the previous message, otherwise service flows have already been set up and activated via CableModem DSx messaging. The QoS functions here are based on the mechanisms defined in the CableModem Recommendations.
(7) QoS Event	The CMTS generates the proper event message, indicating the QoS usage and other billing parameters.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS), Policy Decision, Service usage data, Time-of-day	Event-Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.
(8) QoS Request (Commit)	The Client requests that QoS resources are committed.	Bandwidth and Latency Parameters, Traffic Classifier	CableModem DSx or RSVP+	The Client may directly establish CableModem service flows via DSx messaging or may issue RSVP+ messages to establish these flows.

Table I.5/J.179 – Two-phase resource reservation message details for Scenario 2

Message	Function	Fields	Protocol candidate	Comments
(9) DOCSIS Commit	The CMTS places the service flows in an "active" state.	CableModem Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier, Service Flow ID	CableModem DSx Messaging	This step is only necessary if RSVP+ signalling was provided to the CMTS in the previous message, otherwise service flows have already been set up and activated via CableModem DSx messaging. The QoS functions here are based on the mechanisms defined in the CableModem Recommendations.
(10) QoS Event	The CMTS generates the proper event message, indicating the QoS usage and other billing parameters.	IPCablecom MM QoS Type, IPCablecom MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint (for both AM and PS), Policy Decision, Service usage data, Time-of-day	Event Messaging (RADIUS)	This message should contain sufficient constructs to allow for a reconstruction of the event(s) and decision(s) made with regard to a particular service for the purposes of support and/or reconciliation.

As in the previous scenario, two alternatives are possible with regard to QoS resource teardown and recovery. The resources may timeout (as detected at the CMTS) due to inactivity without a signalled timer refresh, or they may be explicitly deleted by the Client at the conclusion of a service session. The mechanism provided to explicitly signal service flow deletion is a component of the QoS protocol defined for Client Device 2. The only variation between the resource recovery sequence defined for Scenario 1 and that of Scenario 2 is that service flow deletion is signalled directly via the Client versus being proxied through the Application Manager in the second scenario.

I.5.1 Example: Online gaming via networked consoles

The networked gaming console example outlined for Scenario 1 in I.4.2 may easily be altered to conform to the QoS resource management model presented in Scenario 2. In this case, the consoles would still coordinate with an Application Manager in order to locate one another and establish application-specific signalling. In addition, the Application Manager would submit a Resource Request to the Policy Server requesting authorization for necessary QoS resources. However, upon successful installation of this authorization decision on the CMTS, the Application Manager would simply return an affirmative acknowledgement containing an authorization token to each PC proxy. This token could then be used by the PCs in their QoS signalling to the CMTSs in order to reserve, commit, and delete the service flows required by the gaming tunnel.

I.6 Client-requested QoS with policy-pull (Scenario 3)

The third scenario, with its "client-requested QoS with policy-pull" authorization model, supports Client Type 3. Scenario 3 defines a model in which policy authorization decisions are not pre-established and pushed to the CMTS via the Application Manager and Policy Server mechanisms outlined in the previous scenarios, but are requested on demand by the CMTS from the Policy Server as incoming reservation requests dictate. This allows for a very flexible and dynamic resource reservation model stimulated by the Client, while maintaining authoritative operator control at the head-end for all resource requests.

In this scenario, the CMTS receives a QoS request from the Client prior to a policy decision being installed by the Policy Server. Included with this QoS request are credentials that enable the client to be authenticated. The CMTS constructs a policy request which it sends to the Policy Server. At the Policy Server, the request is authenticated and an authorization decision is made based upon operator-specified criteria (e.g., resource availability, customer profile, credit rating, service class, interaction with other network elements, etc.). If the policy authorization is successful the resource reservation is allowed to proceed on the CMTS and the appropriate CableModem service flows are established based on the QoS requested. IPCablecom Multimedia interfaces (defined in I.3.1) involved in this interaction include: pkt-mm1, pkt-mm-2, pkt-mm-4, pkt-mm-5, pkt-mm-6, and mm-9. The pkt-mm-3 interface may be used as well as specific application signalling requirements dictate, but it is not assumed to be in use.

Figure I.10 illustrates the information flow between the core access-network elements for Scenario 3. Table I.6 following Figure I.10 provides further description of each message. In the example shown below, QoS is only established in the upstream direction between the CM and CMTS. A similar flow would be required in order to establish symmetric downstream QoS.

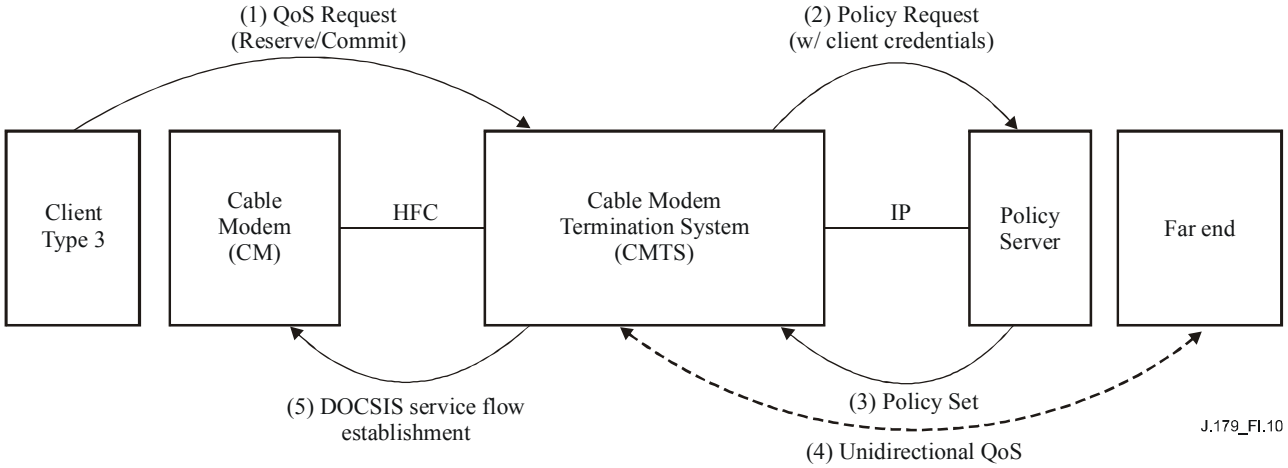

Figure I.10/J.179 – Authorization framework for Scenario 3

Table I.6/J.179 – Message details for Scenario 3

Message	Function	Fields	Protocol candidate	Comments
(1) QoS Request (Reserve/ Commit)	Client requests resource reservation from CMTS.	Bandwidth and Latency Parameters, Traffic Classifier, Authentication Credentials	RSVP	This scenario assumes RFC 2205 capabilities exist on the client.
(2) Policy Request	CMTS solicits policy authorization decision from Policy Server.	Bandwidth and Latency Parameters, Traffic Classifier, Authentication Credentials	COPS	RFC 2748
(3) Policy Set	Policy Server installs authorization on CMTS.	Bandwidth and Latency Parameters, Traffic Classifier	COPS	RFC 2748
(4) Unidirectional QoS	CMTS forwards far-end RSVP signalling.	Bandwidth and Latency Parameters, Traffic Classifier, Authentication Credentials	RSVP	RFC 2205
(5) CableModem Service Flow Establishment	CMTS negotiates CableModem scheduled service flow establishment with CM.	CableModem Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier, Service Flow ID	CableModem DSx Messaging	The QoS functions here are based on the mechanisms defined in the CableModem Recommendations.

One of the primary distinguishing characteristics of this scenario is its support for RSVP, a standards-based QoS signalling mechanism. Whereas Scenario 1 addresses Clients with no native QoS signalling capabilities, and Scenario 2 defines an IPCablecom-specific QoS signalling mechanism (based upon RSVP, but including proprietary extensions), this scenario is based on an IETF standard itself. This will provide for interoperability with standards-based Clients which have subscribed to operator QoS services and have a means of securely authenticating themselves on the access network. It also does not require applications to push policy decisions ahead of time, and hence does not impose architectural constraints on application signalling.

Scenario 3 assumes that RSVP messaging is exchanged between the Client and the far end. Note, however, that this does not require that all of the network elements between the Client and the far end need to support RSVP, nor does it imply the use of an integrated services (IntServ [18]) end-to-end QoS strategy. For example, differentiated services (DiffServ 19) or another QoS scheme may be used beyond the CMTS. Also, intermediate routers that do not wish to support RSVP may simply pass the RSVP messages without processing. Alternatively, if QoS guarantees can be obtained by other means, such routers can be defined as aggregation regions and hence pass RSVP messages transparently, as defined in RFC 3175 [20].

NOTE – RFC 3175 requires implementation of this aggregation function on both the near- and far-end edge router.

Furthermore, it should be noted that the use of RSVP in this scenario conforms closely to standard (i.e., RFC 2205 [14]) RSVP operation, and, hence, resource reservations on the access network are unidirectional. Thus, the Client reserves upstream resources, and the far-end is responsible for reserving downstream resources.

Successful resource reservations are maintained similarly to reservations in the other scenarios via soft-state refreshes. RSVP clients must periodically send messages to maintain their reservations or they will timeout and be reclaimed at the CMTS.

Finally, specific mechanisms are included in the RSVP protocol to allow for either the transmitting or receiving endpoint to signal the termination and tear-down of a service flow. Based on the unidirectional nature of RSVP reservations, an endpoint maintaining multiple service flows is responsible for explicitly deleting each of these flows at the conclusion of a service session.

Given this model, authentication of far-end request in order to enable downstream resource reservation needs special consideration. One solution is to require that the Policy Server can authenticate both the near-end and the far-end Clients. Other solutions are possible as well, but security implications, and in particular the potential for theft of service, must be considered carefully.

I.6.1 Example: Online gaming via native QoS signalling

One potential service which may take advantage of Scenario 3 is online gaming. In this example, all that would be required is integrated, standards-based RSVP support on the Client. That is, the online game could be designed to work either with or without an Application Server.

When a Client wishes to join a game, they would simply send an application-specific message to the far-end, and then proceed in requesting network QoS by sending an RSVP message, again addressed to the remote endpoint. When the CMTS receives this message, it would send a request to the Policy Server in order to authenticate the Client and decide if the QoS should be granted or not. Successful authorization would result in a unidirectional QoS reservation.

Similarly, the far-end would send an RSVP message addressed to the Client. Again, when received at the CMTS, this message would be sent to the Policy Server to determine if the QoS should be granted. Upon successful authorization and servicing, the Client would then have QoS in both directions and could proceed with the online game.

I.7 Comparison of IPCablecom-T and IPCablecom Multimedia

This clause describes, at a high level, the main differences between the IPCablecom-T and IPCablecom Multimedia architectures. Consider that most of the specific protocol characteristics and functional details of IPCablecom Multimedia have yet to be defined as of this writing. See Table I.7, summarizing known differences for quick reference.

Table I.7/J.179 – Contrast of IPCablecom-T and IPCablecom multimedia

	IPCablecom-T	IPCablecom multimedia
Services supported	Residential telephony: • Basic residential telephony features; • Extended telephony features.	Multimedia services: • Client-based (Peer-to-Peer); • Server-based.
Event messaging	Robust audit trail for all policy and QoS events Supports PSTN billing model	Robust audit trail for all policy and QoS events Supports QoS-based accounting Supports time- and volume-based accounting

Table I.7/J.179 – Contrast of IPCablecom-T and IPCablecom multimedia

	IPCablecom-T	IPCablecom multimedia
QoS capabilities	CableModem QoS scheduling algorithms: • Unsolicited Grant Service; • Unsolicited Grant Service with Activity Detection. Bandwidth characteristics: • constant Bit Rate; • symmetric upstream/downstream. Level of QoS Guaranteed: Client-to-Client (i.e., end-to-end via segmented model)	CableModem QoS scheduling algorithms: • Unsolicited Grant Service; • Unsolicited Grant Service with Activity Detection; • Real-Time Polling; • Non-Real-Time Polling; • Best-Effort with or without Priority; Bandwidth characteristics: • constant bit rate; • variable bit rate; • symmetric upstream/downstream; • asymmetric upstream/downstream. Level of QoS Guaranteed: CMTS-to-CM (i.e., access network)
Security	Secure signalling and media Secure device provisioning and configuration management	COPS and RADIUS secured via IPsec; key management via IKE with pre-shared key authentication (IKE with certificates or Kerberized Key Management are optional). Client signalling is out of scope, thus there is no security defined for the client signalling interface.

I.7.1 DQoS

The primary focus of IPCablecom-T is residential telephony services. As a part of this effort, the Dynamic Quality of Service (DQoS) specification was developed, defining the mechanisms necessary to deliver QoS on the CableModem-based access portion of the IP network. That is, IPCablecom-T adopts a segmented approach (dividing the end-to-end media and signalling path into near and far access networks joined by a backbone network) under which DQoS specifically addresses resource reservations on the access segment, not backbone or the end-to-end QoS.

IPCablecom Multimedia is targeted toward more general multimedia applications, which transcend voice support. However, it builds on some fundamental IPCablecom-T DQoS mechanisms in order to provide QoS-enhanced services for these applications.

I.7.1.1 Access-network elements

IPCablecom-T supports the following network elements: MTA, CM, CMTS, CMS (logically composed of a Call Agent and Gate Controller) and RKS. In the IPCablecom Multimedia architecture, the Call Agent may functionally be mapped to an Application Manager, and the Gate Controller may functionally be mapped to the Policy Server. In the IPCablecom Multimedia architecture, additional network elements may be introduced, including, for example, a Media Server. The Application Manager and Media Server may physically reside in the same equipment, or they may be deployed separately.

I.7.1.2 DQoS architecture

The IPCablecom DQoS [14] architecture is based on CableModem, RSVP+, and QoS policies installed on the CMTS by the CMS (Gate Controller).

As described throughout this appendix, the IPCablecom Multimedia architecture is also based on these technologies. In addition, the multimedia effort has an objective to support a more standards-based RSVP signalling model (Scenario 3) with the intent that this capability will make QoS-enhanced services available to a larger consumer base.

The CMTS in the IPCablecom-T DQoS architecture serves as the policy enforcement point for QoS policies. The CMTS will perform a similar function in the IPCablecom Multimedia architecture. In addition to servicing client-originated QoS requests, the CMTS may also receive proxied QoS requests from the Policy Server (Scenario 1). This differs from the IPCablecom-T DQoS architecture, where only the standalone MTA or the embedded MTA may initiate the activation of QoS.

I.7.1.3 QoS interfaces

In the IPCablecom-T architecture, signalling interfaces have been defined between all of the network elements, as well as between CMTSs for on-net to on-net calls supporting Gate Coordination. In summary, the primary signalling protocol between the MTA and the Call Agent is NCS, between the embedded MTA and the CMTS is CableModem, and between a standalone MTA and the CMTS is RSVP+. Signalling from the GC to CMTS is COPS-based Gate Control messaging.

IPCablecom Multimedia builds on these signalling interfaces and additionally supports signalling interfaces between the Application Manager and the Policy Server. Recall that any application-specific signalling occurring between the Application Manager and its clients is out of scope for this architecture.

I.7.1.4 Framework for IPCablecom QoS

In the IPCablecom-T QoS architecture, "a QoS defined construct called a Gate provides the control point for the connection of access networks to high quality backbone service". (See DQoS specification [14].) The Gate represents a QoS authorization that is installed on the CMTS for policy enforcement purposes. IPCablecom Multimedia defines a similar QoS policy construct, and it is anticipated that the IPCablecom-T DQoS Gate construct will be leveraged to provide the policy function in IPCablecom Multimedia. Changes to the existing IPCablecom-T Gate Control mechanisms may be required to provide attenuated QoS control (e.g., in support of Scenario 1).

I.7.1.5 Requirements of access-network resource management

The IPCablecom-T architecture "aims to provide a high degree of generality with the intention of enabling new services and future evolution of network architectures". The goal leads to several requirements for a viable QoS architecture in the following areas (note that each of these QoS-related capabilities is clearly defined and discussed in the IPCablecom DQoS specification):

- resource changes during a session;
- dynamic binding of resources;
- session class (priority designation);
- two-phase resource commitment;
- segmented resource assignment;
- backbone QoS support;
- preventing theft of service.

The IPCablecom Multimedia architecture will also support a single phase resource reservation model. Initially, the multimedia architecture will not address backbone QoS support, although this functionality may be formally addressed as operator needs dictate. For more information on the existing IPCablecom-T DQoS requirements, please refer to the IPCablecom-T DQoS specification [14].

I.7.1.6 Theory of operation

IPCablecom-T DQoS involves distinct reserve and commit phases for obtaining access-network resources. At the end of the reserve phase, resources are set aside but not yet active or available to the MTA. At the end of the second phase, the resources are committed and made available for use. Under the traditional telephony model, billing begins during the commit phase.

In the embedded MTA model, RSVP+ is not required between the MTA and the CMTS. Instead, the E-MTA may signal resource reservation and commitment via CableModem DSx messaging. In the standalone MTA model, RSVP+ messaging is used to perform these steps. The CM and the CMTS then coordinate via CableModem DSx messaging to schedule required service flows on the access network.

As outlined in this appendix, IPCablecom Multimedia supports a model similar to IPCablecom-T, and additionally supports a more standard usage of RSVP. It also provides a proxied QoS request model, where the Application Manager manages QoS on behalf of the Client. These models are detailed in the scenarios section of this appendix. The existing IPCablecom-T model maps to Scenario 2. The other two models are supported in the IPCablecom Multimedia architecture to provide more flexibility in the way multimedia services may be deployed in the operator's network.

I.7.2 Event Messages for billing

IPCablecom Event Messages are designed to be flexible and extensible in order to carry information about network usage for a wide variety of services delivered over the IPCablecom architecture. The IPCablecom-T Event Message specification defines the general Event Message architecture as well as the specific requirements to support IPCablecom-T voice service. The IPCablecom Event Message specification [15] details a transport protocol-independent Event Message TLV format, an Event Message file format, and mandatory and optional transport protocols.

These messages contain sufficient per-session information to support customer billing for service. The information contained in the Event Messages supports a wide variety of billing and settlement models. IPCablecom does not mandate the use of specific billing or settlement models as these models are defined by and based on the specific business requirements of the individual cable operator. IPCablecom neither mandates nor precludes the use of a clearinghouse for settlements.

IPCablecom Event Messages are based on a model where a session or service is divided into an originating half and a terminating half. The originating CMS or MGC must generate a unique Billing Correlation ID (BCID) to identify all Event Messages associated with the originating half of the session. The terminating CMS or MGC must generate a unique BCID to identify all Event messages associated with the terminating half of the session. For each half of the session or service, the set of IPCablecom network elements that generate Event Messages (CMS, MGC, CMTS) must provide all necessary information required for billing and/or settlements as appropriate based on the service. The information generated by the originating half must be sent to the RKS supporting the originating half. The information generated by the terminating half must be sent to the RKS supporting the terminating half.

A limited set of Event Messages are required for IPCablecom Multimedia services. These messages include:

- Signal_Start for "enhanced QoS service" generated by the Policy Server indicating the time at which the Policy Server receives a request for access-network QoS;
- Signal_Stop for "enhanced QoS service" generated by the Policy Server indicating the time at which the Policy Server receives notification that network QoS usage has terminated;
- QoS_Reserve, QoS_Commit, QoS_Stop generated by the CMTS. These messages indicate the time at which the CMTS reserves, commits or releases access-network QoS.

I.7.3 Security

The IPCablecom-T security architecture defines the mechanisms, algorithms and protocols that meet security service requirements. The IPCablecom Multimedia interfaces are secured using identical mechanisms for the corresponding interfaces.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Cable networks and transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure, Internet protocol aspects and Next Generation Networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2004