

PacketCable™ Technical Report

Multimedia Architecture Framework

PKT-TR-MM-ARCH-V03-091029

ISSUED

Notice

This PacketCable technical report is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document.

Neither CableLabs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein.

© Copyright 2003-2009 Cable Television Laboratories, Inc.

All rights reserved.

Document Status Sheet

Document Control Number: PKT-TR-MM-ARCH-V03-091029

Document Title: Multimedia Architecture Framework

Revision History: V01 - June 27, 2003
 V02 - December 21, 2005
 V03 - October 29, 2009

Date: October 29, 2009

Status:	Work in Progress	Draft	Released	Closed
Distribution Restrictions:	Author Only	GL/Member	GL/Member/Vendor	Public

Trademarks:

CableLabs®, DOCSIS®, EuroDOCSIS™, eDOCSIS™, M-CMTS™, PacketCable™, EuroPacketCable™, PCMM™, CableHome®, CableOffice™, OpenCable™, OCAP™, CableCARD™, M-Card™, DCAS™, tru2way™, and CablePC™ are trademarks of Cable Television Laboratories, Inc.

Abstract

The PacketCable Multimedia initiative defines an IP-based platform for delivering QoS-enhanced multimedia services over DOCSIS[®] 1.1 and greater access networks. This platform expands on the core capabilities of PacketCable 1.x (e.g., QoS authorization and admission control, event messages for billing and other back-office functions, and security) to support a wide range of IP-based services beyond telephony. That is, while the PacketCable 1.x architecture is customized for the delivery of residential telephony services, the PacketCable Multimedia architecture offers a general-purpose platform for cable operators to deliver a variety of IP-based multimedia services that require QoS treatment. For this reason, specific services are not defined or addressed in this report.

Although the PacketCable Multimedia platform is based upon PacketCable 1.x work, the full voice infrastructure defined in PacketCable 1.x is not a prerequisite to the deployment of multimedia services. Rather, it is intended that a particular cable operator may choose to initially deploy either voice or multimedia services, with the assurance that these platforms will seamlessly integrate and interoperate if and when they are deployed in parallel.

The current scope of the PacketCable Multimedia architecture is limited to the DOCSIS 1.1-based and greater access portion of a cable operator's network. Therefore, the architecture and service delivery scenarios described in this technical report assume that all application managers and clients reside within a single MSO-administered network, the details of which are defined by and based on a particular MSO's needs and requirements.

In order to provide architectural flexibility while standardizing the QoS and policy interfaces of this platform, several classes of clients have been identified and profiled. These client classes may be distinguished by noting the distribution of QoS signaling capabilities and control across the client-server topology.

The first client class represents existing "legacy" endpoints (e.g., PC applications, gaming consoles) which lack specific QoS awareness or signaling capabilities. To service this class of clients, QoS signaling must be managed entirely at the head-end on behalf of the client. Only the operation of this first class of clients is presently defined.

The second client class is similar to the PacketCable 1.x MTA in that it provides explicit support for QoS signaling but relies upon policy authorization mechanisms initiated and managed at the head-end. This class of clients represents a transition from the previous class in that a portion of the signaling has migrated to a PacketCable client platform.

In the third class of clients, both policy requests and QoS signaling initiate at the client, supporting a distributed end-point control model. This client has been identified and profiled in anticipation of sophisticated endpoint devices which may, for example, require precise management of QoS resources in order to provide converged voice, data and video services. Regardless of the intended usage, resource requests are securely authenticated, authorized and tracked at the MSO head-end through the mechanisms described in this document.

Finally, a recent addition to the PacketCable Multimedia architecture is a SOAP/XML web services interface that allows for rapid integration of application services into a MSO's multimedia environment.

This technical report provides a thorough description of the network elements and interfaces making up the PacketCable Multimedia service delivery platform and demonstrates the dynamic interaction of these elements through several service delivery scenarios. The PacketCable Multimedia architecture is also contrasted and compared with the PacketCable 1.x architecture. Technical and protocol-specific discussion has been deferred to the PacketCable Multimedia specification.

Table of Contents

1	INTRODUCTION	1
1.1	PACKETCABLE OVERVIEW	1
1.2	PACKETCABLE MULTIMEDIA MOTIVATION	1
2	REFERENCES	3
2.1	INFORMATIVE REFERENCES	3
2.2	REFERENCE ACQUISITION	3
3	TERMS AND DEFINITIONS	4
4	ABBREVIATIONS	5
5	PACKETCABLE MULTIMEDIA REQUIREMENTS AND SCOPE	7
5.1	REQUIREMENTS	7
5.2	SCOPE	9
6	PACKETCABLE MULTIMEDIA FRAMEWORK	10
6.1	PACKETCABLE MULTIMEDIA ARCHITECTURE REFERENCE MODEL	11
6.1.1	<i>PacketCable Multimedia Gates</i>	13
6.2	MULTIMEDIA COMPONENTS	14
6.2.1	<i>Client</i>	14
6.2.2	<i>Policy Server</i>	14
6.2.3	<i>Cable Modem Termination System</i>	15
6.2.4	<i>Application Manager</i>	16
6.2.5	<i>Application Server</i>	16
6.2.6	<i>Record Keeping Server</i>	17
7	PROXIED QOS WITH POLICY PUSH (SCENARIO 1)	18
7.1	EXAMPLE: WEB-BASED BANDWIDTH ON DEMAND	24
7.2	EXAMPLE: ONLINE GAMING VIA NETWORKED CONSOLES	24
8	CLIENT-REQUESTED QOS WITH POLICY-PUSH (SCENARIO 2)	26
8.1	EXAMPLE: ONLINE GAMING VIA NETWORKED CONSOLES	31
9	CLIENT-REQUESTED QOS WITH POLICY-PULL (SCENARIO 3)	32
9.1	EXAMPLE: ONLINE GAMING VIA NATIVE QoS SIGNALING	34
10	COMPARISON OF PACKETCABLE 1.X AND PACKETCABLE MULTIMEDIA	35
10.1	DQoS	35
10.1.1	<i>Access-Network Elements</i>	36
10.1.2	<i>DQoS Architecture</i>	36
10.1.3	<i>QoS Interfaces</i>	36
10.1.4	<i>Framework for PacketCable QoS</i>	36
10.1.5	<i>Requirements of Access-Network Resource Management</i>	37
10.1.6	<i>Theory of Operation</i>	37
10.2	EVENT MESSAGES FOR BILLING	37
10.3	SECURITY	38

List of Figures

FIGURE 1 - MSO MULTIMEDIA NETWORK ELEMENTS.....	10
FIGURE 2 - PACKETCABLE MULTIMEDIA ARCHITECTURAL FRAMEWORK.....	11
FIGURE 3 - AUTHORIZATION FRAMEWORK FOR SCENARIO 1.....	18
FIGURE 4 - SINGLE-PHASE RESOURCE RESERVATION MODEL FOR SCENARIO 1	19
FIGURE 5 - TWO-PHASE RESOURCE RESERVATION MODEL FOR SCENARIO 1	21
FIGURE 6 - GAMING CONSOLES NETWORKED VIA A QOS-ENHANCED IP TUNNEL.....	25
FIGURE 7 - AUTHORIZATION FRAMEWORK FOR SCENARIO 2.....	26
FIGURE 8 - SINGLE-PHASE RESOURCE RESERVATION MODEL FOR SCENARIO 2	27
FIGURE 9 - TWO-PHASE RESOURCE RESERVATION MODEL FOR SCENARIO 2	29
FIGURE 10 - AUTHORIZATION FRAMEWORK FOR SCENARIO 3.....	32

List of Tables

TABLE 1 - PACKETCABLE MULTIMEDIA INTERFACES	12
TABLE 2 - SINGLE-PHASE RESOURCE RESERVATION MESSAGE DETAILS FOR SCENARIO 1	20
TABLE 3 - TWO-PHASE RESOURCE RESERVATION MESSAGE DETAILS FOR SCENARIO 1	22
TABLE 4 - SINGLE-PHASE RESOURCE RESERVATION MESSAGE DETAILS FOR SCENARIO 2	27
TABLE 5 - TWO-PHASE RESOURCE RESERVATION MESSAGE DETAILS FOR SCENARIO 2	29
TABLE 6 - MESSAGE DETAILS FOR SCENARIO 3	33
TABLE 7 - CONTRAST OF PACKETCABLE 1.X AND PACKETCABLE MULTIMEDIA.....	35

This page left blank intentionally.

1 INTRODUCTION

This technical report describes an architecture which provides an IP-based platform to support a variety of multimedia applications and services requiring network resource conditioning over DOCSIS 1.1 and greater access networks. Throughout this document, the term DOCSIS will be used to refer to any DOCSIS version equal to or greater than 1.1. 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, signaling and operations support system (OSS) functions required to provide a particular service are out of scope. Rather, the initial focus of PacketCable Multimedia centered on the delivery of reliable QoS over the access network, specifically addressing the technical issues of policy authorization, QoS signaling, resource accounting, and security. A later addition to the PacketCable Multimedia architecture is a SOAP/XML interface to the Applications Manager that allows for rapid integration of application services into an MSO's multimedia environment [MM].

1.1 PacketCable Overview

PacketCable is a project conducted by Cable Television Laboratories, Inc. (CableLabs®) and its member companies. The PacketCable 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 fiber coax (HFC) cable systems which conform to the DOCSIS 1.1 and greater broadband access network specifications.

Voice over IP (VoIP) was the first such service identified for delivery over the PacketCable platform. The current set of PacketCable specifications, known collectively as PacketCable 1.x, define a PacketCable architecture optimized for the delivery of residential VoIP services. These specifications were developed using a phased approach as outlined below:

- PacketCable 1.0 defines the core PacketCable VoIP architecture facilitating basic call functionality within a single operator domain.
- PacketCable 1.5 includes the functionality defined by PacketCable 1.0 and defines additional capabilities for enhanced call functionality, end-to-end VoIP service between different cable operators and electronic surveillance.

1.2 PacketCable 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 network capabilities, they, too, will present unique network conditioning requirements.

Historically broadband customers had received multimedia services via best-effort data delivery, resulting in an inconsistent online experience that varied in quality based on the present network condition. A network that is able to reserve resources and deliver bandwidth on demand as service requirements dictate will be positioned to provide maximal return on MSO investment.

In order to address these needs for VoIP services, PacketCable currently defines dynamic Quality of Service (DQoS) signaling mechanisms that allow voice applications to request and obtain bandwidth from the DOCSIS 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 PacketCable architecture is well positioned to support existing and future QoS-enhanced applications and services beyond telephony.

The primary objective of PacketCable 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 DOCSIS and PacketCable DQoS Specifications. A PacketCable Multimedia web services interface (described herein) defined in [MM] was developed to facilitate rapid integration of application services into an MSO's multimedia environment. Interface requirements allowing for third-party application services were considered in the development of this interface.

2 REFERENCES

2.1 Informative References

The following are informative references for this technical report.

- [DOCSIS] DOCSIS MAC and Upper Layer Protocols Interface Specification v3.0, CM-SP-MULPIv3.0-I11-091002, October 2, 2009, Cable Television Laboratories, Inc.
- [RFC1633] IETF RFC 1633, Braden, R., Integrated Services in the Internet Architecture: An Overview, June 1994.
- [RFC2138] IETF RFC 2138, Rigley, C., Remote Authentication Dial In User Service (RADIUS), April 1997.
- [RFC2139] IETF RFC 2139, Rigley, C., RADIUS Accounting, April 1997.
- [RFC2205] IETF RFC 2205, Braden, R., Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, September 1997.
- [RFC2475] IETF RFC 2475, Blake, S., An Architecture for Differentiated Services, December 1998.
- [RFC2748] IETF RFC 2748, Boyle, J., The COPS (Common Open Policy Service) Protocol, January 2000.
- [RFC2753] IETF RFC 2753, Yavatkar, R., A Framework for Policy Based Admission Control, January 2000.
- [RFC3175] IETF RFC 3175, Baker, F., Aggregation of RSVP for IPv4 and IPv6 Reservations, September 2001.
- [ARCH 1.0] PacketCable 1.0 Architecture Framework Technical Report, PKT-TR-ARCH-C01-071129, November 29, 2007, Cable Television Laboratories, Inc.
- [ARCH 1.5] PacketCable 1.5 Architecture Technical Report, PKT-TR-ARCH1.5-V02-090528, May 28, 2009, Cable Television Laboratories, Inc.
- [DQOS1.5] PacketCable 1.5 Dynamic Quality of Service Specification, PKT-SP-DQOS1.5-I04-090624, June 24, 2009, Cable Television Laboratories, Inc.
- [EM1.5] PacketCable 1.5 Event Messages, PKT-SP-EM1.5-I03-070412, April 12, 2007, Cable Television Laboratories, Inc.
- [SEC1.5] PacketCable 1.5 Security, PKT-SP-SEC1.5-I03-090624, June 24, 2009, Cable Television Laboratories, Inc.
- [MM] PacketCable Multimedia Specification, PKT-SP-MM-I05-091029, October 29, 2009, Cable Television Laboratories, Inc.

2.2 Reference Acquisition

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

3 TERMS AND DEFINITIONS

This document and associated PacketCable Multimedia specifications use the following terms:

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

4 ABBREVIATIONS

This document and associated PacketCable Multimedia specifications use 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.
AS	Applications Server. A server that interfaces to the PacketCable Multimedia Applications Manager that requests PacketCable Multimedia services on behalf of clients.
BCID	Billing Correlation ID. Defined in the PacketCable Event Messaging specification [EM1.5].
CM	DOCSIS [®] Cable Modem.
CMS	Call Management Server. Network element defined in the PacketCable 1.0 Architecture technical report.
CMTS	Cable Modem Termination System. Device at a cable head-end which implements the DOCSIS MAC protocol [DOCSIS] and connects to CMs over an HFC network.
COPS	Common Open Policy Service. Defined in [RFC2748].
DOCSIS	Data-Over-Cable Service Interface Specifications. A set of interface specifications for transmitting data over cable television systems in a standard fashion.
DQoS	Dynamic Quality of Service.
DSx (Messaging)	DOCSIS QoS signaling mechanism providing Dynamic Service Add, Change and Delete semantics.
HFC	Hybrid Fiber/Coax. An HFC system is a broadband bi-directional shared media transmission system using fiber trunks between the head-end and the fiber nodes, and coaxial distribution from the fiber nodes to the customer locations.
IETF	Internet Engineering Task Force. A body responsible for, among other things, developing standards used on the Internet. (See http://www.ietf.org .)
IP	Internet Protocol.
MGC	Media Gateway Controller. Network element defined in the PacketCable 1.0 Architecture technical report.
MSO	Multiple Service Operator.
MTA	Multimedia Terminal Adapter. Network element defined in the PacketCable 1.0 Architecture technical report. Contains the interface to a physical voice device, a network interface, CODECs, and all signaling and encapsulation functions required for VoIP transport, class features signaling, and QoS signaling. May be implemented as a standalone (S-MTA) or embedded (E-MTA) device, depending upon whether a DOCSIS CM is integrated.
MULPI	MAC and Upper Layer Protocols Interface, part of the DOCSIS specification, defining MAC and Upper Layer Protocol interfaces between CMTS and CM network elements.
NAT	Network Address Translation. Function performed to convert IP addresses, and potentially transport ports (PAT), from one network address numbering convention to another.
PDP	Policy Decision Point. Defined in [RFC2753].
PEP	Policy Enforcement Point. Defined in [RFC2753].

PS	Policy Server. A system that primarily acts as an intermediary between Application Manager(s) and CMTS(s). It applies network policies to Application Manager requests and proxies messages between the Application Manager and CMTS.
PSTN	Public Switch Telephone Network.
QoS	Quality of Service. Method used to reserve network resources and guarantee availability for applications.
RADIUS	Remote Authentication Dial-In User Service. Defined in [RFC2138] and [RFC2139]. An Internet protocol originally designed for allowing users dial-in access to the Internet through remote servers. Its flexible design has allowed it to be extended well beyond its original intended use.
RAP	Resource Allocation Protocol Working Group in the IETF. Responsible for the definition and maintenance of the COPS protocol.
RFC	Request for Comments. Technical policy documents approved by the IETF which are available at http://www.ietf.org/rfc.html
RKS	Record Keeping Server. Network element defined in PacketCable 1.0 Architecture technical report. Also defined in the PacketCable Event Messaging specification. The device which collects and correlates the various Event Messages.
RSVP	Resource Reservation Protocol. Defined in [RFC2205].
TLV	Type-Length-Value. Technique used in formatting protocol elements.
UGS	Unsolicited Grant Service. DOCSIS QoS scheduling type used for constant bit rate services (e.g., voice codecs).
UGS/AD	Unsolicited Grant Service with Activity Detection. DOCSIS QoS scheduling type used for constant bit rate services with periodic inactivity (e.g., voice codecs implementing silence suppression).
VoIP	Voice over IP.
VPN	Virtual Private Network.

5 PACKETCABLE MULTIMEDIA REQUIREMENTS AND SCOPE

The main objective of PacketCable 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 PacketCable 1.x ([ARCH 1.0] and [ARCH 1.5]) and [DOCSIS].
- Requires a minimal set of extensions beyond PacketCable 1.x
- Reduces development complexity by eliminating telephony-specific requirements where not applicable (e.g., PSTN interconnect, electronic surveillance, telephony billing models, etc.)
- Co-exists with the PacketCable 1.x architecture in such a way that:
 - PacketCable Multimedia requirements are sufficient to support a QoS-based multimedia service delivery platform
 - PacketCable Multimedia requirements may be added to relevant existing PacketCable 1.x functional components
 - PacketCable 1.x requirements may be added to relevant PacketCable Multimedia functional components
- Supports PacketCable 1.x MTAs as "Client Type 2" devices (defined within) in the PacketCable Multimedia architecture
- Interoperates with CableHome™ and DOCSIS 1.1 and greater architectures
- Supports Application Server requests to PacketCable Multimedia infrastructure on behalf of client applications requiring use of PacketCable Multimedia resources or services.

This section 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.

5.1 Requirements

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

Client Devices may communicate their PacketCable Multimedia resource or service needs directly to a MSO-owned Applications Manager or indirectly through an Applications Server. Application Servers may or may not be owned by the MSO. This model allows for third-party Application Servers to request PacketCable Multimedia resource or services needs on behalf of client applications that may not be PacketCable Multimedia aware.

Client Devices in the PacketCable Multimedia architecture: (1) reside directly on the MSO access network, or within the home, (2) may be standalone devices or may contain an embedded DOCSIS cable modem, and (3) are considered untrusted network elements. If Client Devices communicate directly to an Applications Manager the PacketCable Multimedia architecture is application signaling protocol-agnostic concerning interaction between the Client Device and Application Manager. The Client Device and Application Manager may support a variety of application and signaling protocols (e.g., HTTP, SIP, H.323, DCS, NCS, etc.). For this case, some form of authentication of the user, application, or application messaging may be required by the MSO network. When Client Devices communicate their needs indirectly to the MSO access network via an Application Server (AS), the AS will communicate PacketCable Multimedia policy via the SOAP/XML-based web services interface (described within).

For this case, some form of user, Client Device, client and/or AS authentication may be required by the AS and/or the MSO network.

Application Servers (ASs) in the PacketCable Multimedia architecture: (1) may or may not be owned by the MSO, and (2) may or may not be trusted network elements. Owing to this, the web services interface was developed such that untrusted ASs could be supported (e.g., third-party ASs). If the AS is untrusted, some form of AS authentication may be required by the MSO network.

Application Managers in the PacketCable Multimedia architecture: (1) reside on the MSO managed network, (2) are managed by the MSO, and (3) are responsible for ensuring that clients requesting service from the MSO network are authorized to receive that service.

Policy Servers in the PacketCable Multimedia architecture: (1) reside in the MSO managed network, (2) are managed by the MSO, and (3) are responsible for making QoS-related policy decisions based on MSO-defined policy rules.

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

QoS Signaling and Resource Management Requirements:

- Dynamic resource request mechanisms must be defined, including:
 - Access to all DOCSIS 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 bi-directional reservations should be allowed.
- Application Managers may initiate QoS reservation requests on behalf of Client Devices.
- Applications Servers may request PacketCable Multimedia resource or service needs on behalf of Client Devices or client applications.
- The architecture must provide a means to detect client or server failures (e.g., lack of media for VoIP) and reclaim PacketCable Multimedia resources.
- Support QoS Signaling and Management for both Unicast and Multicast Flows.

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 MSO-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 signaling may require some form of authentication of the user or the application.

5.2 Scope

The following items outline the current scope of the PacketCable Multimedia initiative:

- The architecture will address network elements that reside: (1) on the access network, or (2) within a single MSO's managed IP network.
- The architecture will define the protocols and interfaces necessary to support policy such as authorization, QoS admission control, resource accounting, and security mechanisms.
- The architecture will not address application-specific issues (e.g., service provisioning, signaling, billing, etc).
- The architecture will not address provisioning and OSS requirements for PacketCable Multimedia network elements.
- The architecture continues to have major focus on QoS management between the CMTS and CM. However, extensions to the initial architecture address general applications services-oriented communication mechanisms between Applications Servers (third-party or MSO-owned) and the PacketCable Multimedia infrastructure.
- The architecture will support delivery of downstream Multicast Services through the use of Multicast Gates. The architecture will not preclude the delivery of upstream multicast services, even though it will not explicitly address any upstream multicast considerations.
- The architecture will not address Network Address Translation (NAT) traversal and interoperability requirements.
- The architecture will not define end-to-end QoS requirements in the present phase.
- The architecture will support all client types and service scenarios (described herein). The PacketCable Multimedia Specifications presently provide specification detail only for "Client Type 1" and "Scenario 1" (defined herein).
- 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 mechanisms employed by the Application Manager or Application Servers.
- 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.
- This architecture will support event notification of state changes of existing PacketCable Multimedia resources.

6 PACKETCABLE 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 PacketCable 1.x specifications. In support of this objective, several key network elements have been identified and profiled. The following diagram presents the PacketCable Multimedia components which reside within the MSO's managed IP network.

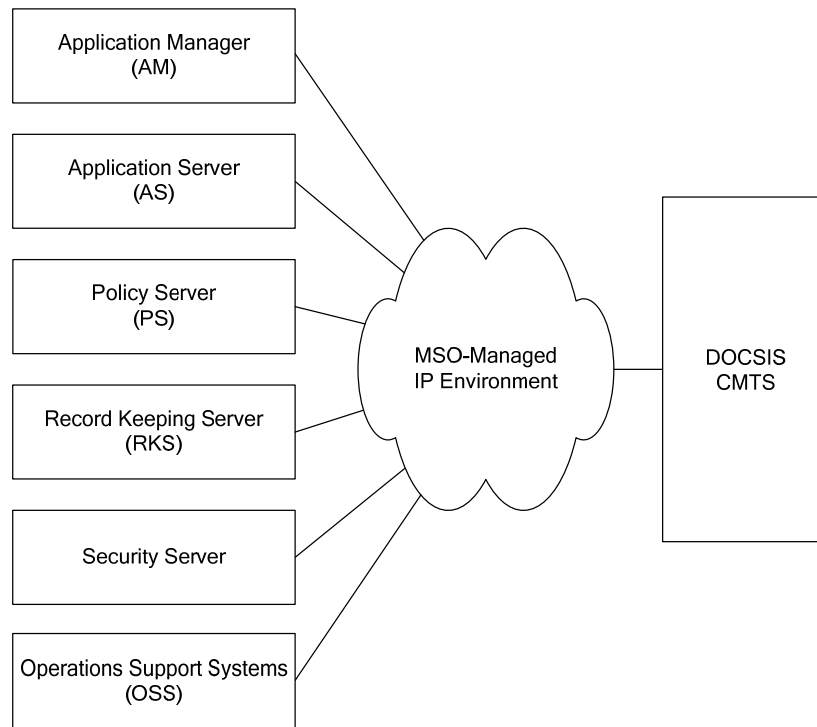


Figure 1 - MSO Multimedia Network Elements

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

- An Application Manager and (optional) Application Server hosting a QoS-enabled application. The Application Server may or may not reside within the MSO's managed IP network. In case, the Application Server is not part of the MSO's managed IP network, it interacts with the MSO network through the Application Manager via a trust relationship between the MSO and the owner of the Application Server.
- 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 MSO multimedia network configuration, though these elements fall outside the scope of the current architecture.

6.1 PacketCable Multimedia Architecture Reference Model

In addition to the elements residing within the MSO head-end network, a number of client devices located on the customer premise have also been defined in order to complete the model. The following diagram shows the PacketCable 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.

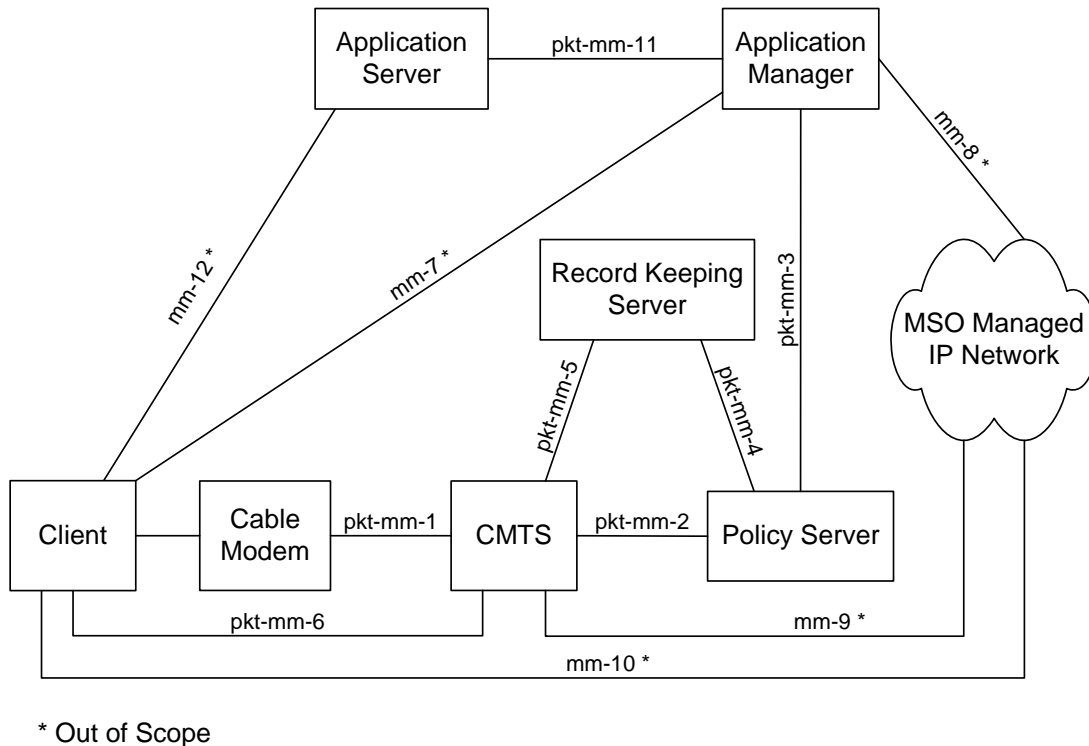


Figure 2 - PacketCable Multimedia Architectural Framework

In this architecture, Clients may or may not support the PacketCable Multimedia framework. Clients that support the framework and its QoS signaling 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 signaling mechanisms have network resource requests either issued directly to the Application Manager or issued to an Application Server. This Application Server sends the network resource request on the behalf of the client to an Application Manager, which proxies these requests to the Policy Server, with which it interacts.

Regardless of the QoS signaling 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 received from an Application Manager. An Application Manager proxies a service request, originating from an Application Server. An Application Server may generate such a request, based on Client interaction (through some unspecified signaling 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.

The table below summarizes the interfaces presented in Figure 2. Interfaces that are subject to the PacketCable Multimedia specification are labeled "pkt-mm-x", while other interfaces, which are included for completeness, are labeled "mm-x".

Table 1 - PacketCable Multimedia Interfaces

Interface	Description	Notes
pkt-mm-1	CMTS – CM	The CM may request QoS from the CMTS via DOCSIS DSx signaling. Alternatively, the CMTS may instruct the Cable Modem (CM) to setup, teardown or change a DOCSIS service flow in order to satisfy a QoS request, again via DSx signaling.
pkt-mm-2	PS – CMTS	This interface is fundamental to the policy-management framework. It controls policy decisions, which may be: (a) pushed by the Policy Server (PS) onto the CMTS, or (b) pulled from the PS by the CMTS. The interface also allows for proxied QoS requests on behalf of a client. In some scenarios, this interface may also be used to inform the PS when QoS resources have become inactive.
pkt-mm-3	AM – PS	The Application Manager (AM) may request that the PS install a policy decision on the CMTS. 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.
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 the current effort.
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 the current effort.
mm-9	CMTS – MSO-Managed IP Network	This interface on the CMTS may be used in support of end-to-end QoS requests beyond the access network. This interface is out of scope for the current effort.
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 the current effort.

Interface	Description	Notes
pkt-mm-11	AS - AM	The Application Server uses this interface to send network resource requests on behalf of the client to the AM. This interface may also be used to notify the AS of changes in the status of the network resources.
mm-12	Client – AS	This interface may be used by the client to interact with the AS and to request applications, which indirectly requests network resources. This interface is out of scope for the current effort.

6.1.1 PacketCable Multimedia Gates

A PacketCable 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 the underlying DOCSIS cable network. Gates are unidirectional; a single Gate controls access to a flow in either the upstream or the downstream direction, but not both. For a bi-directional IP session, two Gates are required, one for upstream and one for downstream, each identified by a unique GateID.

The Gate defines the authorization, reservation and committed envelopes to be used by the CMTS to perform authorization, reservation and commit operations.

Gates can be further qualified as either unicast or multicast, PacketCable Multimedia Gates can be tied to Unicast Flows or Multicast flows. Fundamentally the two Gates are as described above with one key differentiator between them – the destination IP address indicated in the classifier object. In a Unicast Gate the destination IP address (v4 or v6) contained in the classifier identifies a unicast IP address (or a range of unicast IP addresses). A Multicast Gate contains a destination IP address (v4 or v6) in the classifier identifying a multicast IP address.

The CMTS performs admission control checks of the envelopes of the Gate 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.

6.2 Multimedia Components

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

6.2.1 Client

A multimedia client is a logical entity that may send or receive data. PacketCable 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 signaling capabilities. This Client knows nothing about DOCSIS, CableHome, or PacketCable 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. There are two ways how the Client interacts with the PacketCable Multimedia environment. The Client may communicate directly with an Application Manager to request services via the mm-7 interface. In addition, the Client may communicate with an Application Server via the mm-12 interface to request service. In both cases the Client does not request QoS resources directly from the MSO access network.

Client Type 2 is similar to a PacketCable 1.x telephony MTA in that it supports QoS signaling based on PacketCable DQoS. This Client is aware of PacketCable Multimedia QoS, and communicates either directly with an Application Manager (mm-7) or indirectly via the Application Server (mm-12) 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.

6.2.2 Policy Server

The policy-management framework for the PacketCable 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 (PDP) network element implements MSO-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 MSOs 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 signaling 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 MSO-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 MSO-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.

6.2.3 Cable Modem Termination System

PacketCable Multimedia provides access to the full set of CMTS upstream scheduling algorithms as defined in DOCSIS. Specifically, the architecture defines a PacketCable Multimedia "Traffic Profile" that provides a layer of abstraction from associated DOCSIS Scheduling Types (UGS, UGS/AD, etc.). Further, the telephony-specific features and assumptions found in the PacketCable 1.x 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 DOCSIS 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 [DOCSIS].

6.2.4 Application Manager

The Application Manager plays a coordinating role involving application signaling and semantics and as well as interaction with the PacketCable Multimedia policy framework, as outlined during the previous discussion of the Policy Server element.

The Application Manager may receive service requests either from an Application Server or from a Client. The following depicts the interaction between the Application Manager and those entities:

- The Application Manager interfaces with an Application Server via pkt-mm-11. Once the Application Server is authenticated and authorized, the Application Manager sends the policy request to the Policy Server via pkt-mm-3. Upon successful completion, the Application Manager sends an acknowledgement to the Application Server.
- The Application Manager interfaces with a Client Type 1 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.
- Client Type 2 may also interact 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.
- Client Type 3 does not require an Application Manager, although the presence of an Application Manager in sophisticated service delivery scenarios is quite likely.
- Finally, the Application Manager may also receive event registrations from the Application Server via the pkt-mm-11 interface. Depending on those registrations, the Application Manager sends notification messages to the Application Server, in case the specific events have occurred.

6.2.5 Application Server

The Application Server receives application requests from a client via the mm-12 interface. It generates a service request, based on the original request, and sends the request message via the pkt-mm-11 interface to the Application Manager in order to deliver the requested service to Client Type 1.

Client Type 2 may also interface with the Application Server and request services via mm-12. Again, the Application Server communicates the service request to the Application Manager via pkt-mm-11. Upon successful authorization, the Application Server receives a token from the Application Manager and forwards it to the Client via mm-12.

The Application Server may also register via the pkt-mm-11 interface to specific events. Depending on the event registration, the Application Server receives notification from the Application Manager.

6.2.6 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.

7 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 signaling mechanisms. A high-level overview of the element interaction involved in this scenario is shown in Figure 3.

The Client may request an application-specific service by sending a "Service Request" to the Application Manager or the Client may request an application-specific service from an Application Server, which, in turn, will make a "Service Request" to the Application Manager on its behalf. 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 MSO-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 signaling 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 and that between the Client and the Application Server are out of scope for the current discussion. 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 kbps 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.

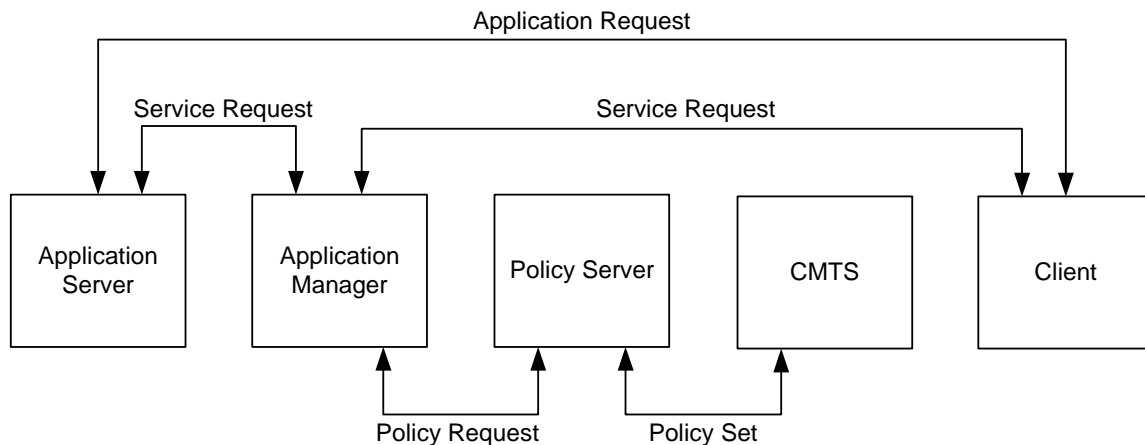


Figure 3 - Authorization Framework for Scenario 1

Under Scenario 1, the CMTS supports a single-phase resource reservation model, as shown below in Figure 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 section.)

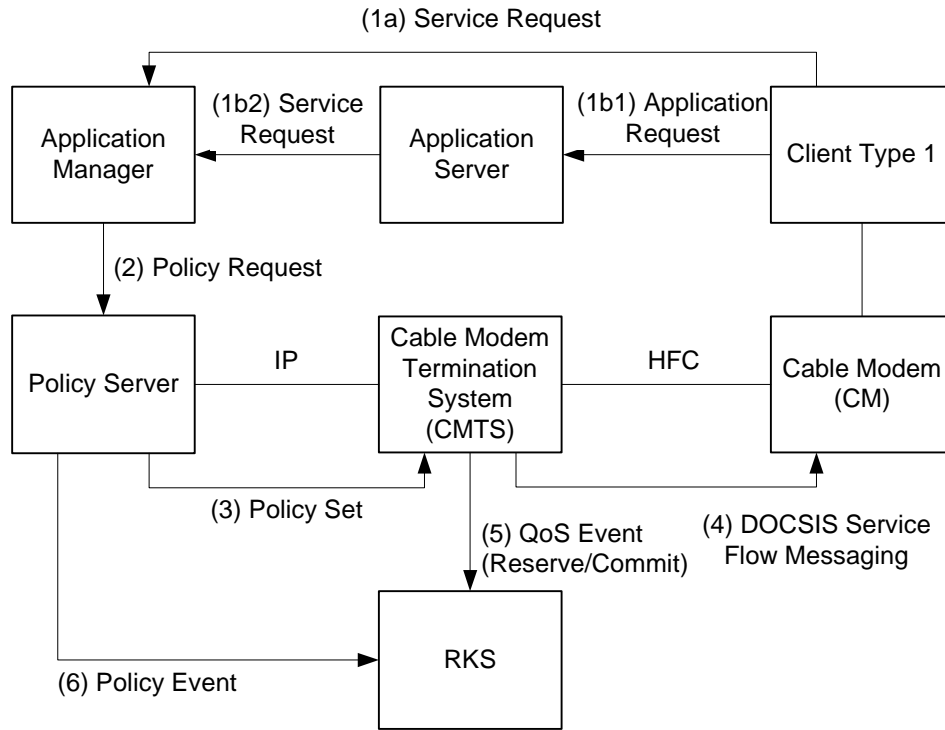


Figure 4 - Single-phase Resource Reservation Model for Scenario 1

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

Table 2 - Single-phase Resource Reservation Message Details for Scenario 1

Message	Function	Fields	Protocol Candidate	Comments
(1a) Service Request	The Client requests service from the Application Manager.	<none>	Out of scope for PacketCable Multimedia	Client may interact directly with an Application Manager to request a service (alternative a). 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.
(1b1) Application Request	The Client requests service from the Application Server.	<none>	Out of scope for PacketCable Multimedia	Client may instead interact with an Application Server which in turn makes service requests on its behalf. (alternative b) This protocol should support the authentication of both Client and Application Server. Also, the protocol should provide sufficient information for the Application Manager to convey the QoS needs of the requested service.
(1b2) Service Request	Application Server requests service from the Application Manager.	ServiceName, SubscriberIdentity	(SOAP/XML)	PacketCable Multimedia Web Services interface
(2) Policy Request	The Application Manager requests QoS setup on behalf of the client.	PacketCable MM QoS Type, PacketCable MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint	Gate-Control (COPS)	The Policy Server uses MSO-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.	PacketCable MM QoS Type, PacketCable 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) DOCSIS Messaging	The CMTS establishes QoS-enhanced service flows.	DOCSIS Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier	DOCSIS DSx Messaging	The QoS functions here are based on the mechanisms defined in [DOCSIS].
(5) QoS Event	The CMTS generates the proper event message, indicating QoS usage and other billing parameters	PacketCable MM QoS Type, PacketCable 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.

Message	Function	Fields	Protocol Candidate	Comments
(6) Policy Event	The Policy Server generates the proper event message, indicating the Policy Request and action taken.	PacketCable MM QoS Type, PacketCable 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 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 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 dialog with the Client and eventually committing the resources.

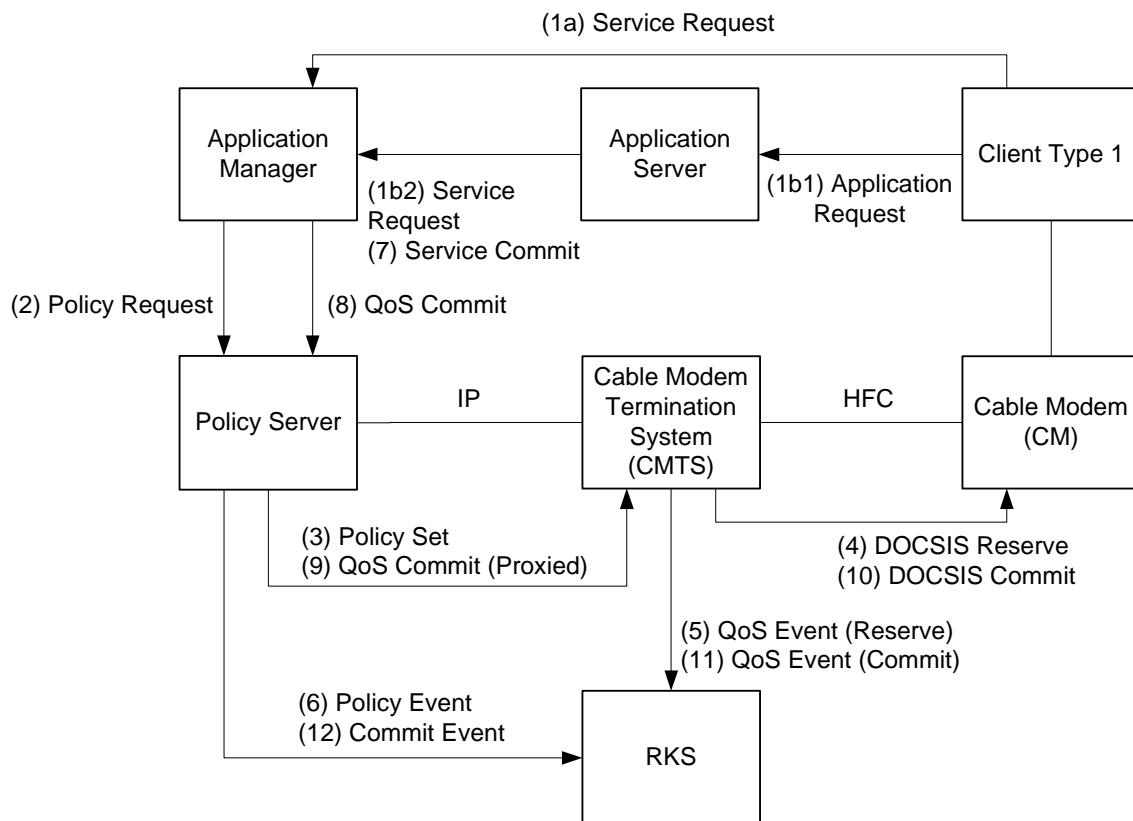


Figure 5 - Two-phase Resource Reservation Model for Scenario 1

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

Table 3 - Two-phase Resource Reservation Message Details for Scenario 1

Message	Function	Fields	Protocol Candidate	Comments
(1a) Service Request	The Client requests service from the Application Manager.	<none>	Out of scope for PacketCable Multimedia	Client may interact directly with an Application Manager to request a service (alternative a). 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.
(1b1) Application Request	The Client requests service from the Application Server.	<none>	Out of scope for PacketCable Multimedia	Client may instead interact with an Application Server which in turn makes service requests on its behalf (alternative b). This protocol should support the authentication of both Client and Application Server. Also, the protocol should provide sufficient information for the Application Manager to convey the QoS needs of the requested service.
(1b2) Service Request	Application Server requests service from the Application Manager.	ServiceName, SubscriberIdentity	(SOAP/XML)	PacketCable Multimedia Web Services interface
(2) Policy Request	The Application Manager requests QoS setup on behalf of the client.	PacketCable MM QoS Type, PacketCable MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint	Gate-Control (COPS)	The Policy Server uses MSO-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.	PacketCable MM QoS Type, PacketCable 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.	DOCSIS Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier	DOCSIS DSx Messaging	The QoS functions here are based on the mechanisms defined in [DOCSIS]. 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.	PacketCable MM QoS Type, PacketCable 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.

Message	Function	Fields	Protocol Candidate	Comments
(6) Policy Event	The Policy Server generates the proper event message, indicating the Policy Request and action taken.	PacketCable MM QoS Type, PacketCable 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) Service Commit	Application Server request the Application Manager to commit the service.	ServiceName, SubscriberIdentity	(SOAP/XML)	PacketCable Multimedia Web Services interface
(8) QoS Commit	The AM signals to commit the QoS resources.	PacketCable MM QoS Type, PacketCable 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.
(9) QoS Commit (Proxied)	The Policy Server receives the AM request and proxies to the CMTS.	PacketCable MM QoS Type, PacketCable 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.
(10) DOCSIS Commit	The CMTS places the service flow in the "active" state.	DOCSIS Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier, Service Flow ID	DOCSIS DSx Messaging	The QoS functions here are based on the mechanisms defined in [DOCSIS].
(11) QoS Event (Commit)	The CMTS generates the proper event message, indicating the QoS usage and other billing parameters.	PacketCable MM QoS Type, PacketCable 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.
(12) Commit Event	The Policy Server generates the proper event message, indicating the QoS commit and action taken.	PacketCable MM QoS Type, PacketCable 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 Server signals the Application Manager when the service session has ended. The Application Manager would then indicate

the ending of the service to the Policy Server. 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.

7.1 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 MSO-hosted secure web site, which would allow subscribers to request bandwidth reservations on-demand.

Assume, for example, that a subscriber's normal service is rate limited to 128 kbps downstream and 128 kbps 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 MSO's web site (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 kbps 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 kbps). Either way, this exchange represents the subscriber's "Service Request" to the Application Manager. Alternatively, such a request could have been made by the user to an Application Server which would in turn formulate a request to the Application Manager.

In the either case, the Application Manager would present a "Policy Request" for 512 kbps 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 DOCSIS messaging, tracking this process through a QoS event message.

7.2 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 Server 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 Server makes a "Service Request" of the Application Manager using the Packet Cable Multimedia Web Services interface. 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 DOCSIS 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 MSO'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 6 provides a graphical illustration of this example for the simplified case in which both users receive service from on a single CMTS.

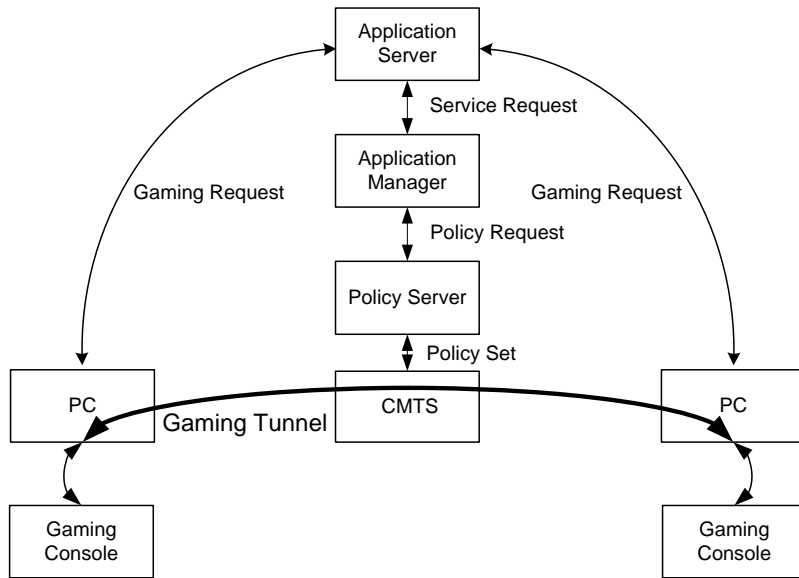


Figure 6 - Gaming Consoles Networked via a QoS-enhanced IP Tunnel

8 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 signaling 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 PacketCable 1.x 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 DOCSIS DSx or RSVP+ messaging similar to MTA devices in PacketCable 1.x.

Client Type 2 has the ability to indirectly request authorization from the Application Manager via the Application Server. The Application Server performs a "Service Request" to the Application Manager. The Application Manager requests for policy authorization as stated in the above paragraph. The interface between the client and the Application Server is out of scope for the current discussion.

A high-level overview of Scenario 2 is shown in Figure 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 or sending an application request to the Application Server, which sends 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 MSO-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 signaling 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.

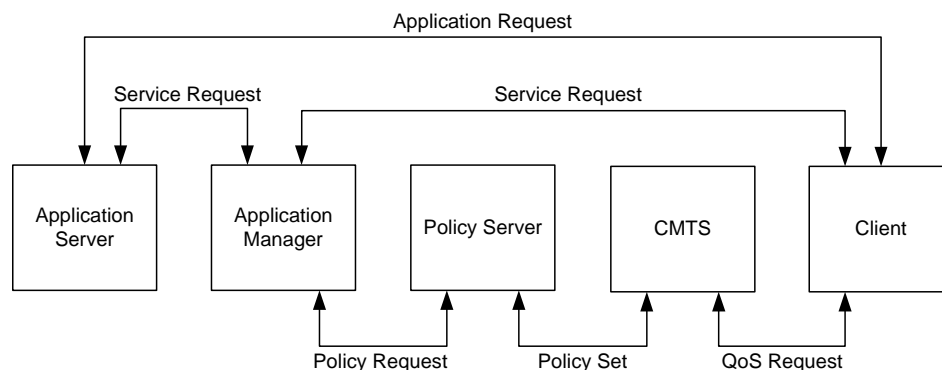


Figure 7 - Authorization Framework for Scenario 2

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

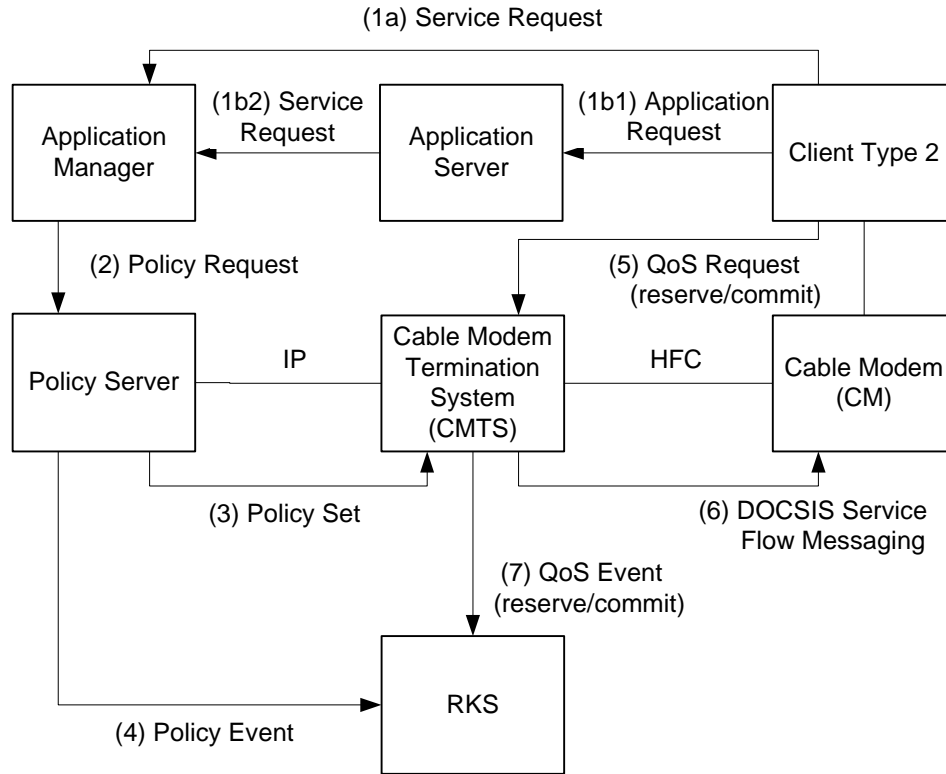


Figure 8 - Single-phase Resource Reservation Model for Scenario 2

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

Table 4 - Single-phase Resource Reservation Message Details for Scenario 2

Message	Function	Fields	Protocol Candidate	Comments
(1a) Service Request	The Client requests service from the Application Manager.	<none>	Out of scope for PacketCable Multimedia	Client may interact directly with an Application Manager to request a service (alternative a). 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.
(1b1) Application Request	The Client requests service from the Application Server.	<none>	Out of scope for PacketCable Multimedia	Client may instead interact with an Application Server which in turn makes service requests on its behalf. (alternative b) This protocol should support the authentication of both Client and Application Server. Also, the protocol should provide sufficient information for the Application Manager to convey the QoS needs of the requested service.
(1b2) Service Request	Application Server requests service from the Application Manager.	ServiceName, SubscriberIdentity	(SOAP/XML)	PacketCable Multimedia Web Services interface

Message	Function	Fields	Protocol Candidate	Comments
(2) Policy Request	The Application Manager requests QoS authorization on behalf of the Client.	PacketCable MM QoS Type, PacketCable MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint	Gate-Control (COPS)	The Policy Server uses MSO-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.	PacketCable MM QoS Type, PacketCable 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.	PacketCable MM QoS Type, PacketCable 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	DOCSIS DSx or RSVP+	The Client may directly establish DOCSIS 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.	DOCSIS Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier	DOCSIS DSx Messaging	This step is only necessary if RSVP+ signaling was provided to the CMTS in the previous message, otherwise service flows have already been setup and activated via DOCSIS DSx messaging. The QoS functions here are based on the mechanisms defined in [DOCSIS].
(7) QoS Event	The CMTS generates the proper event message, indicating the QoS usage and other billing parameters.	PacketCable MM QoS Type, PacketCable 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 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.

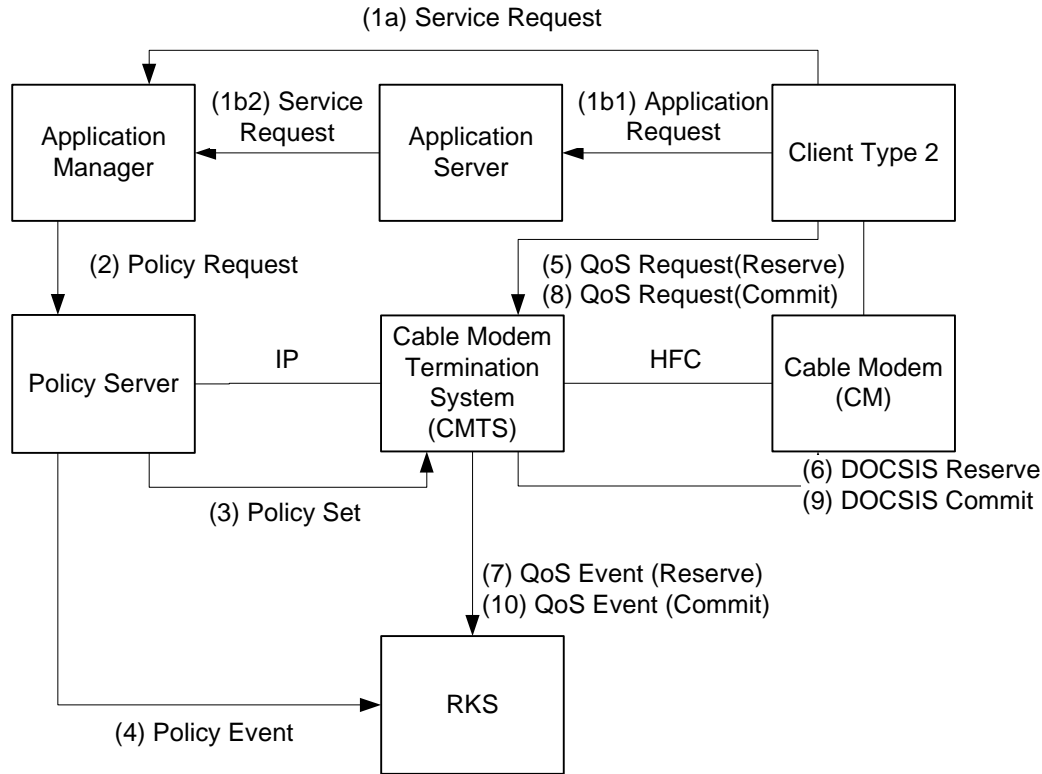


Figure 9 - Two-phase Resource Reservation Model for Scenario 2

Table 5 - Two-phase Resource Reservation Message Details for Scenario 2

Message	Function	Fields	Protocol Candidate	Comments
(1a) Service Request	The Client requests service from the Application Manager.	<none>	Out of scope for PacketCable Multimedia	Client may interact directly with an Application Manager to request a service (alternative a). 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.
(1b1) Application Request	The Client requests service from the Application Server.	<none>	Out of scope for PacketCable Multimedia	Client may instead interact with an Application Server which in turn makes service requests on its behalf. (alternative b) This protocol should support the authentication of both Client and Application Server. Also, the protocol should provide sufficient information for the Application Manager to convey the QoS needs of the requested service.
(1b2) Service Request	Application Server requests service from the Application Manager.	ServiceName, SubscriberIdentity	(SOAP/XML)	PacketCable Multimedia Web Services interface

Message	Function	Fields	Protocol Candidate	Comments
(2) Policy Request	The Application Manager requests QoS authorization on behalf of the Client.	PacketCable MM QoS Type, PacketCable MM Session Class, Bandwidth and Latency Parameters, Traffic Classifier, Opaque Billing Hint	Gate-Control (COPS)	The Policy Server uses MSO-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.	PacketCable MM QoS Type, PacketCable 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.	PacketCable MM QoS Type, PacketCable 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	DOCSIS DSx or RSVP+	The Client may directly establish DOCSIS service flows via DSx messaging or may issue RSVP+ messages to establish these flows.
(6) DOCSIS Reserve	The CMTS establishes QoS-enhanced service flows and places them in an "admitted" state.	DOCSIS Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier	DOCSIS DSx Messaging	This step is only necessary if RSVP+ signaling was provided to the CMTS in the previous message, otherwise service flows have already been setup and activated via DOCSIS DSx messaging. The QoS functions here are based on the mechanisms defined in [DOCSIS].
(7) QoS Event	The CMTS generates the proper event message, indicating the QoS usage and other billing parameters.	PacketCable MM QoS Type, PacketCable 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	DOCSIS DSx or RSVP+	The Client may directly establish DOCSIS service flows via DSx messaging or may issue RSVP+ messages to establish these flows.
(9) DOCSIS Commit	The CMTS places the service flows in an "active" state.	DOCSIS Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier, Service Flow ID	DOCSIS DSx Messaging	This step is only necessary if RSVP+ signaling was provided to the CMTS in the previous message, otherwise service flows have already been setup and activated via DOCSIS DSx messaging. The QoS functions here are based on the mechanisms defined in [DOCSIS].
(10) QoS Event	The CMTS generates the proper event message, indicating the QoS usage and other billing parameters.	PacketCable MM QoS Type, PacketCable 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 signaled 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 signaled directly via the Client versus being proxied through the Application Manager in the second scenario.

8.1 Example: Online Gaming via Networked Consoles

The networked gaming console example outlined for Scenario 1 in Section 7.2, 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 signaling. 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 signaling to the CMTSs in order to reserve, commit, and delete the service flows required by the gaming tunnel.

9 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 MSO 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 MSO-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 DOCSIS service flow are established based on the QoS requested. PacketCable Multimedia interfaces (defined in Section 6.1) 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 signaling requirements dictate, but it is not assumed to be in use.

Figure 10 illustrates the information flow between the core access-network elements for Scenario 3. Table 6, following Figure 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.

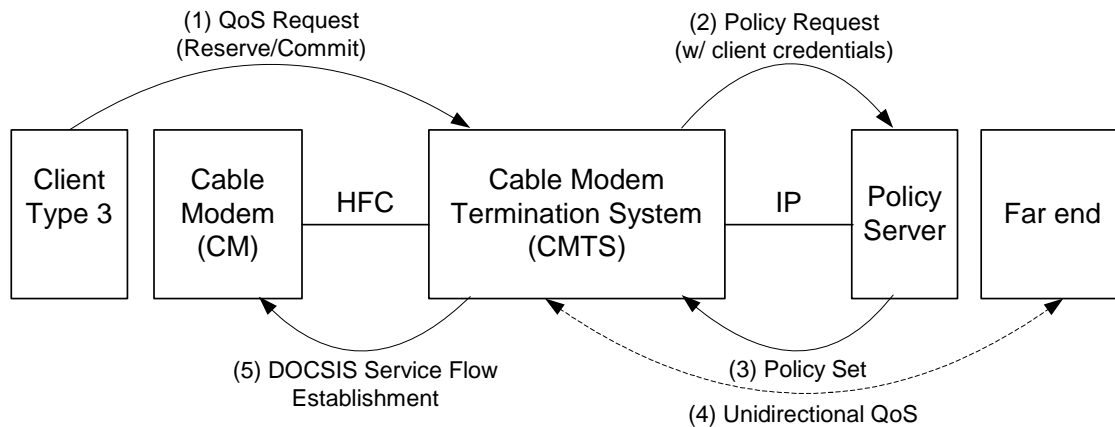


Figure 10 - Authorization Framework for Scenario 3

Table 6 - 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 signaling.	Bandwidth and Latency Parameters, Traffic Classifier, Authentication Credentials	RSVP	RFC 2205
(5) DOCSIS Service Flow Establishment	CMTS negotiates DOCSIS scheduled service flow establishment with CM.	DOCSIS Scheduling Type, Bandwidth and Latency Parameters, Traffic Classifier, Service Flow ID	DOCSIS DSx Messaging	The QoS functions here are based on the mechanisms defined in [DOCSIS].

One of the primary distinguishing characteristics of this scenario is its support for RSVP, a standards-based QoS signaling mechanism. Whereas Scenario 1 addresses Clients with no native QoS signaling capabilities, and Scenario 2 defines a PacketCable-specific QoS signaling 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 MSO 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 signaling.

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 [RFC1633]) end-to-end QoS strategy. For example, differentiated services (DiffServ [RFC2475]) 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 [RFC3175]. 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., [RFC2205]) 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.

9.1 Example: Online Gaming via Native QoS Signaling

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 receive 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.

10 COMPARISON OF PACKETCABLE 1.X AND PACKETCABLE MULTIMEDIA

This section describes, at a high level, the main differences between the PacketCable 1.x and PacketCable Multimedia architectures. Consider that most of the specific protocol characteristics and functional details of PacketCable Multimedia have yet to be defined as of this writing. See Table 7, summarizing known differences for quick reference.

Table 7 - Contrast of PacketCable 1.x and PacketCable Multimedia

	PacketCable 1.x	PacketCable Multimedia
Services Supported	Residential Telephony <ul style="list-style-type: none"> Basic residential telephony features Extended telephony features 	Multimedia Services <ul style="list-style-type: none"> 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
QoS Capabilities	DOCSIS QoS scheduling algorithms <ul style="list-style-type: none"> Unsolicited Grant Service Unsolicited Grant Service with Activity Detection Bandwidth Characteristics <ul style="list-style-type: none"> Constant Bit Rate Symmetric upstream/downstream Level of QoS Guaranteed <ul style="list-style-type: none"> Client-to-Client (i.e., end-to-end via segmented model) 	DOCSIS QoS scheduling algorithms <ul style="list-style-type: none"> Unsolicited Grant Service Unsolicited Grant Service with Activity Detection Real-Time Polling Non-Real-Time Polling Best-Effort with or without Priority Bandwidth Characteristics <ul style="list-style-type: none"> Constant bit rate Variable bit rate Symmetric upstream/downstream Asymmetric upstream/downstream Level of QoS Guaranteed <ul style="list-style-type: none"> CMTS-to-CM (i.e., access network)
Security	Secure signaling 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 signaling is out of scope, thus there is no security defined for the client signaling interface.

10.1 DQoS

The primary focus of PacketCable 1.x is residential telephony services. As a part of this effort, the Dynamic Quality of Service [DQOS1.5] specification was developed, defining the mechanisms necessary to deliver QoS on the DOCSIS-based access portion of the IP network. That is, PacketCable1.x adopts a segmented approach (dividing the

end-to-end media and signaling 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.

PacketCable Multimedia is targeted toward more general multimedia applications, which transcend voice support. However, it builds on some fundamental PacketCable 1.x DQoS mechanisms in order to provide QoS-enhanced services for these applications.

10.1.1 Access-Network Elements

PacketCable 1.x supports the following network elements: MTA, CM, CMTS, CMS (logically composed of a Call Agent and Gate Controller) and RKS. In the PacketCable 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 PacketCable 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.

10.1.2 DQoS Architecture

The PacketCable DQoS [DQOS1.5] architecture is based on DOCSIS, RSVP+, and QoS policies installed on the CMTS by the CMS (Gate Controller).

As described throughout the course of this report, the PacketCable Multimedia architecture is also based on these technologies. In addition, the multimedia effort has an objective to support a more standards-based RSVP signaling model (Scenario 3) with the intent that this capability will make QoS-enhanced services available to a larger consumer base.

The CMTS in the PacketCable 1.x DQoS architecture serves as the policy enforcement point for QoS policies. The CMTS will perform a similar function in the PacketCable 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 PacketCable 1.x DQoS architecture, where only the standalone MTA or the embedded MTA may initiate the activation of QoS.

10.1.3 QoS Interfaces

In the PacketCable 1.x architecture, signaling 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 signaling protocol between the MTA and the Call Agent is NCS, between the embedded MTA and the CMTS is DOCSIS, and between a standalone MTA and the CMTS is RSVP+. Signaling from the GC to CMTS is COPS-based Gate Control messaging.

PacketCable Multimedia builds on these signaling interfaces and additionally supports signaling interfaces between the Application Server and the Application Manager and between the Application Manager and the Policy Server. Recall that any application-specific signaling occurring between the Application Manager and its clients or between the Application Server and its clients are out of scope for this architecture.

10.1.4 Framework for PacketCable QoS

In the PacketCable 1.x QoS architecture, "a QoS defined construct called a Gate provides the control point for the connection of access networks to high quality backbone service." The Gate represents a QoS authorization that is installed on the CMTS for policy enforcement purposes (see [DQOS1.5]). PacketCable Multimedia defines a similar QoS policy construct, and it is anticipated that the PacketCable 1.x DQoS Gate construct will be leveraged to provide the policy function in PacketCable Multimedia. Changes to the existing PacketCable 1.x Gate Control mechanisms may be required to provide attenuated QoS control (e.g., in support of Scenario 1).

10.1.5 Requirements of Access-Network Resource Management

The PacketCable 1.x 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 PacketCable 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 PacketCable 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 MSO needs dictate. For more information on the existing PacketCable 1.x DQoS requirements please refer to the PacketCable 1.x DQoS specification [DQOS1.5].

10.1.6 Theory of Operation

PacketCable 1.x 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 DOCSIS DSx messaging. In the standalone MTA model, RSVP+ messaging is used to perform these steps. The CM and the CMTS then coordinate via DOCSIS DSx messaging to schedule required service flows on the access network.

As outlined in this report, PacketCable Multimedia supports a model similar to PacketCable 1.x, 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 Application Server or the Client. These models are detailed in the scenarios section of this document. The existing PacketCable 1.x model maps to Scenario 2. The other two models are supported in the PacketCable Multimedia architecture to provide more flexibility in the way multimedia services may be deployed in the MSO's network.

10.2 Event Messages for Billing

PacketCable 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 PacketCable architecture. The PacketCable 1.x Event Message specification defines the general Event Message architecture as well as the specific requirements to support PacketCable 1.x voice service. The PacketCable Event Message specification [EM1.5] 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. PacketCable 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. PacketCable neither mandates nor precludes the use of a clearinghouse for settlements.

PacketCable 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 PacketCable 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 PacketCable 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

10.3 Security

The PacketCable 1.x security architecture defines the mechanisms, algorithms and protocols that meet security service requirements. The PacketCable Multimedia interfaces are secured using identical mechanisms for the corresponding interfaces.

