INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.192

(03/2004)

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

A Residential Gateway to support the delivery of cable data services

CAUTION !

PREPUBLISHED RECOMMENDATION

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

ITU-T Recommendation J.192

A Residential Gateway to support the delivery of cable data services

Summary

This Recommendation provides a set of IP based features, , typically associated with a Residential Gateway, that may be embedded within or connect to a Cable Modem (e.g. ITU-T Recommendations J.122, J.112), that will enable cable operators to provide a set of enhanced home network based services (relative to recommendation J.191) to their customers. This includes support for Quality of Service (QoS), device and service discovery, enhanced security, firewall management, home network focused management and provisioning features, managed network address translation, improved addressing and packet handling, and LAN device diagnostics.

TABLE OF CONTENTS

List of Figures

1 SCOPE

This Recommendation creates a Residential Gateway by providing a set of IP based features that may be added to a Cable Modem or incorporated into a stand alone device. This will enable cable operators to provide an additional set of enhanced home network based services to their customers including support for Quality of Service (QoS), device and service discovery, enhanced security, firewall management, home network focused management and provisioning features, managed network address translation, improved addressing and packet handling and LAN device diagnostics. This Recommendation is based upon the architectural frameworks defined in recommendation J.190.

This Recommendation represents an enhancement to J.191, retaining a majority of J.191 functionality as a foundation, and building upon this base to provide additional advanced features. A key design goal for equipment conforming to this Recommendation is interoperability with equipment conforming to J.191. For example, common MIBS are used for the foundational functionality. As a result, a J.192 based headend may manage a mixed J.191 and J.192 deployment.

The key functionality that this Recommendation defines in addition to that defined by J.191 includes:

- Device and service discovery for applications and services on the LAN
- NAT support for IPSec VPN clients and home based servers
- Standardized firewall configuration language and reporting
- Standardized baseline firewall functionality
- Simple parental control
- Quality of Service for the LAN, managed at the Residential Gateway

2 REFERENCES

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

2.1 References (normative)

ITU-T Recommendation J.191 IP Feature Package to enhance cable modems

ITU-T Recommendation J.125 Link privacy for cable modem implementations.

ITU-T Recommendation J.112 Annex B Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification.

ANSI/SCTE 23-3 2003 DOCSIS 1.1 Part 3: Operations Support System Interface .

FIPS 140-2 *Security Requirements for Cryptographic Modules*, Department of Commerce, NIST, May 25, 2001.

FIPS 180-1, *Secure Hash Algorithm*, Department of Commerce, NIST, April, 1995

IANAifType MIB Definitions, http://www.iana.org/assignments/ianaiftype-mib

ISO 8025 (December 1987) – Information processing systems - Open Systems Interconnection - Specification of the Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

ITU-T Recommendation X.25 (10/96), Interface between data terminal equipment and data circuit-terminating equipment for terminals operating in the packet mode and connected to public data networks by dedicated circuit.

ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1977.

IETF RFC 2315, PKCS #7, Cryptographic Message Syntax March 1998

ITU-T Recommendation J.175 Audio server protocol.

ITU-T Recommendation J.178 IPCablecom CMS to CMS signalling .

ITU-T Recommendation J.161 Audio codec requirements for the provision of bidirectional audio service over cable television networks using cable modems.

ITU-T Recommendation J.163 Dynamic quality of service for the provision of real time services over cable television networks using cable modems.

ITU-T Recommendation J.164 Event message requirements for the support of real-time services over cable television networks using cable modem.

ITU-T Recommendation J.162 Network call signalling protocol for the delivery of time critical services over cable television networks using cable modems .

ITU-T Recommendation J.167 Media Terminal Adapter (MTA) device provisioning requirements for the delivery of real time services over cable television networks using cable modems

ITU-T Recommendation J.170 IPCablecom security specification.

IETF RFC-0347 *Echo Process*, May 1972.

IETF RFC-0768 *User Datagram Protocol (UDP)* , August 1980.

IETF RFC-0791 (MIL STD 1777), *Internet Protocol*, September, 1981.

IETF RFC-0792, *Internet Control Message Protocol (ICMP),* September 1981.

IETF RFC-0868, *Time Protocol* May 1983.

IETF RFC 919, *Broadcasting Internet Datagrams*, Oct-01-1984.

IETF RFC 922, *Broadcasting Internet datagrams in the presence of subnets*, Oct-01-1984.

IETF RFC-1034, *Domain Names - Concepts and Facilities*, November 1987.

IETF RFC-1035, *Domain Names - Implementation and Specification,* November 1987.

IETF RFC-1122, *Requirements for Internet Hosts -- Communication Layers*, October 1989.

IETF RFC-1123 *Requirements for Internet Hosts – Application and Support*, October 1989.

IETF RFC-1213, *Management Information Base for Network Management of TCP/IP-based Internets*, March 1991.

IETF RFC-1157, *A Simple Network Management Protocol (SNMP)*, May, 1990.

IETF RFC 1350 *The TFTP Protocol (Revision 2)*, July, 1992.

IETF RFC 1510, *The Kerberos Network Authentication Service (V5)* September, 1993.

IETF RFC-1633, *Integrated Services in the Internet Architecture: An Overview*, June, 1994.

IETF RFC-1812, *Requirements for IP Version 4 Routers*, June, 1995.

IETF RFC-1889, *RTP: A Transport Protocol for Real-Time Applications*, January 1996.

IETF RFC-1901, *Introduction to Community-based SNMPv2*, January 1996.

IETF RFC-2104, *HMAC: Keyed-Hashing for Message Authentication,* February 1997.

IETF RFC-2011, *SNMPv2 Management Information Base for the Internet Protocol using SMIv2*, November 1996.

IETF RFC 2013, *SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2*, November 1996.

IETF RFC-2131, *Dynamic Host Configuration Protocol*, March 1997.

IETF RFC-2132, *DHCP Options and BOOTP Vendor Extensions*, March 1997.

IETF RFC-2211, *Specification of the Controlled-Load Network Element Service*, September 1997.

IETF RFC-2212, S*pecification of Guaranteed Quality of Service*, September 1997.

IETF RFC-2233, *The Interfaces Group MIB using SMIv2*, November 1997.

IETF RFC-2236, *Internet Group Management Protocol, Version 2*, November 1997.

IETF RFC-2246, *The TLS Protocol Version 1.0*, January 1999.

IETF RFC-2349, *TFTP Timeout Interval and Transfer Size Options*, May 1998.

IETF RFC-2401, *Security Architecture for the Internet Protocol*, November 1998

IETF RFC 2402, *IP Authentication Header* , November 1998

IETF RFC 2406, *IP Encapsulating Security Payload (ESP)*, November 1998

IETF RFC-2409, *The Internet Key Exchange (IKE)*, November 1998

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

IETF RFC-2576, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*, March 2000.

IETF RFC 2578, *Structure of Management Information Version 2 (SMIv2)*, April 1999.

IETF RFC 2579, *Textual Conventions for SMIv2*, April 1999.

IETF RFC 2580, *Conformance Statements for SMIv2*, April 1999.

IETF RFC-2616, *Hypertext Transfer Protocol -- HTTP/1.1*, June 1999.

IETF RFC-2663, *IP Network Address Translator (NAT) Terminology and Considerations*, August 1999.

IETF RFC-2665, *Definitions of Managed Objects for Ethernet-like Interface Types*, August 1999.

IETF RFC-2669, *DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems*, August 1999.

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

IETF RFC-2786, *Diffie-Hellman USM Key Management Information Base and Textual Convention*, March, 2000.

IETF RFC-2863, *The Interfaces Group MIB*, June 2000.

IETF RFC-3022, *Traditional IP Network Address Translator (Traditional NAT)*, January 2001.

IETF RFC-3046, *DHCP Relay Agent Information Option*, January 2001.

IETF RFC-3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,* April 2002.

IETF RFC-3291, *Textual Conventions for Internet Network Addresses,* May 2002.

IETF RFC-3410, *Introduction and Applicability Statements for Internet-Standard Management Framework,* December 2002.

IETF RFC-3411, *An Architecture for Describing SNMP Management Frameworks*, December 2002.

IETF RFC-3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*, December 2002.

IETF RFC-3413, *SNMP Applications*, December 2002.

IETF RFC-3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*, December 2002.

IETF RFC-3415, *View-based Access Control Model (VACM) for the Simple Network Control Model (SNMP)*, December 2002.

IETF RFC-3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*, December 2002.

IETF-RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP)*, December 2002.

IETF RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*, December 2002.

ANSI/SCTE 22-1 2002, DOCSIS 1.0, Radio Frequency Interface Standard.

SOAP Version 1.2, W3C Working Draft, World Wide Web Consortium (W3C), December 19, 2002, http://www.w3.org/2000/xp/Group/#drafts.

XML Protocol (XMLP) Requirements, W3C Working Draft, World Wide Web Consortium (W3C), June 26, 2002, http://www.w3.org/TR/2002/WD-xmlp-reqs-20020626.

2.2 References (informative)

[draft-ietf-ipcdn-bpiplus-mib-05]
DOCSIS Baseline Privacy Plus MIB - Management Information Base for DOCSIS Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus, IETF Internet Draft, http://www.watersprings.org/pub/id/draft-ietf-ipcdn-bpiplus-mib-05.txt

SCTE 22-3 2003 DOCSIS 1.0 Part 3: Operations Support System Interface .

Federal Information Processing Standards Publications (FIPS PUB) 186, Digital Signature Standard, 18 May 1994

Fenner W., et al., IGMP-based Multicast Forwarding ("IGMP Proxying"), IETF Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-magma-igmp-proxy-01.txt

IANA Port Numbers, http://www.iana.org/assignments/port-numbers

RSA Laboratories, PKCS #1, v2.0: RSA Cryptography Standard, October 1, 1999.

IETF RFC-2644, *Changing the Default for Directed Broadcasts in Routers*, August 1999.

IETF RFC 3164, *The BSD Syslog Protocol*, August 2001.

IETF RFC-3235, *Network Address Translator (NAT)-Friendly Application Design Guidelines*, January 2002.

IETF RFC 3435, *Media Gateway Control Protocol (MGCP) Version 1.0*, January 2003

3 DEFINITIONS

IPCable2Home Security Portal (CSP) A functional element that provides security management and translation functions between the HFC and Home network.

Embedded PS A Portal Services element that does not use a standalone interface to connect to a CM.

Home Access (HA) Device A grouping of logical elements used to achieve HFC access for IPCable2Home network(s), referred to as a Residential Gateway in this Recommendation.

Home Client (HC) Device A group of logical elements used to provide functionality to client applications, referred to as an IPCable2Home Host in this Recommendation.

LAN IP Device A LAN IP Device is representative of a typical IP device expected to reside on home networks, and is assumed to contain a TCP/IP stack as well as a DHCP client.

Portal Services (PS) A functional element that provides management and translation functions between the HFC and Home network.

Standalone PS A Portal Services element that connects to the CM using only a standalone interface.

4 ABBREVIATIONS AND CONVENTIONS

4.1 Abbreviations

A/V	Audio/Video
ALG	Application Layer Gateway
APP	Application
ASP	Application Specific Proxy
BP	Boundary Point
BPSC	Bulk Portal Services Configuration
CA	Certificate Authority
CAP	IPCable2Home Address Portal
CAT	IPCable2Home Address Translation
CDC	IPCable2Home DHCP Client
CDP	IPCable2Home DHCP Portal
CDS	IPCable2Home DHCP Server
CH	IPCable2Home Host
CTL	Certification Testing Laboratory
CM	Cable Modem
CMP	IPCable2Home Management Portal
CMS	Call Management Server
CMTS	Cable Modem Termination System
C-NAT	IPCable2Home Network Address Translation
C-NAPT	IPCable2Home Network Address and Port Translation
CNP	IPCable2Home Naming Portal
CPU	Central Processing Unit
CqoS	IPCable2Home Quality of Service
CQP	IPCable2Home QoS Portal
CRG	IPCable2Home Residential Gateway
CRL	Certificate Revocation List
CSP	IPCable2Home Security Portal
CTP	IPCable2Home Testing Portal
CVC	Code Verification Certificate
CVS	Code Verification Signature
CxP	IPCable2Home Portal Services Sub-function
DER	Distinguished Encoding Rules
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DOCSIS	Data-Over-Cable Service Interface Specification
DOS	Denial of Service
DQoS	Dynamic Quality-of-Service (PacketCable)
E-MTA	Embedded Multimedia Terminal Adapter
FTP	File Transfer Protocol
FW	Firewall
GMT	Greenwich Mean Time
HA	Home Access
HE	Headend
HEX	Hexidecimal
HFC	Hybrid Fiber Coax
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPCDN	IP over Cable Data Network - a working group of the IETF
IPF	Inbound Packet Filter
IPSec	Internet Protocol Security

KDC	Key Distribution Center
LAN	Local Area Network
LAN-Pass	Pass-through Local Area Network address
LAN-Trans	Translated Local Area Network address
MAC	Media Access Control
MBP	Management Boundary Point
MCF	Management Client Function
MGCP	Media Gateway Control Protocol
MIB	Management Information Base
MPLS	Multiprotocol Label Switching
MSF	Management Server Function
MTA	Multimedia Terminal Adapter
NAPT	Network Address and Portal Translation
NAT	Network Address Translation
NCS	Network-based Call Signaling
NMS	Network Management System
NS	Authoritative Name Server
OID	Object Identifier
OPF	Outbound Packet Filter
OSI	Open System Interconnection
OSS	Operations Support System
PDU	Protocol Data Unit
PF	Packet Filter
PING	Packet Inter-Network Groper
PKI	Public Key Infrastructure
PKINIT	Public-Key Cryptography for Initial Authentication
PS	Portal Services
PS WAN-Man	CableHome Portal Services element WAN management interface
PS WAN-Data	CableHome Portal Services element WAN data interface
QBP	Quality of Service Boundary Point
QCC	Quality of Service Characteristics Client
QCS	Quality of Service Characteristics Server
QFM	Quality of Service Forwarding & Media Access
QoS	Quality of Service
RAM	Random Access Memory
RDN	Relative Distinguished Name
RFC	Request for Comments
RG	Residential Gateway
ROM	Read Only Memory
RSA	Rivest, Shamir, Adleman (See Glossary)
RSVP	Resource ReSerVation Protocol
RTCP	Real-Time Control Protocol
RTP	Real-Time Transport Protocol
SDP	Session Description Protocol
SHA – 1	Secure Hash Algorithm 1
S-MTA	Standalone Multimedia Terminal Adapter
SNMP	Simple Network Management Protocol
SOA	Start of Authority
SPF	Stateful Packet Filtering
SYSLOG	System Log
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TLV	Type-Length-Value
ToD	Time of Day
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USFS	Upstream Selective Forwarding Switch

USM	User Security Model
UTC	Coordinated Universal Time
VACM	View-based Access Control Model
VoIP	Voice over Internet Protocol
WAN	Wide Area Network
WAN-Data	Wide Area Network Data Address Realm
WAN-Man	Wide Area Network Management Address Realm

4.2 Conventions:

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

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

5 REFERENCE ARCHITECTURE

The goal of IPCable2Home is to enable the delivery of new cable-based services to devices within the home, complementing the CableModem and IPCablecom infrastructures and enabling the delivery of these services. Specifically, IPCable2Homeprovides an infrastructure, by specifying a home networking environment, over which IPCablecom and other related application services can be delivered, managed, and supported.

This Recommendation facilitates the development of an interoperable Residential Gateway (CRG) and compliant hosts (CH). The goal is the creation of a cable operator-configurable Residential Gateway centric environment that will interact meaningfully with IP based home devices (LAN IP Devices) whether they are compliant or not. This brings cable operator driven management, provisioning, QoS, and Security to the Residential Gateway. In addition, LAN messaging, prioritized QoS, and simple remote diagnostics for home devices is specified. Quality of Service for applications running on IPCable2Home compliant LAN hosts is also specified. A summary of the capabilities provided by this Recommendation follows:

Management, Discovery, and Provisioning

- Remote management and configuration of the Residential Gateway device
- Simple Residential Gateway diagnostics proxy for IP based home devices
- Hands off provisioning for Residential Gateway devices
- Discovery of IP based home devices and associated applications
- Management of the Residential Gateway from the LAN

Addressing and Packet Handling

- One to many addressing translation for home devices

- One to one addressing translation for home devices
- Non-translated addressing for home devices (for translated address phobic applications)
- HFC traffic protection from in-home device intra-communications
- Home addressing support during HFC outage
- Simple DNS server in the Residential Gateway
- NAT support for IPsec VPN clients
- NAT support for IP based servers in the home using address translation

Quality of Service (QoS)

- Residential Gateway device transparent bridging functionality for IPCablecom QoS messaging from/to IPCablecom compliant applications
- Ability to assign traffic priorities (differentiated media access) to specific applications
- Ability to prioritize queuing in the Residential Gateway device in conjunction with the packet handling functionality.

Security

- Residential Gateway device authentication
- Secure management messages between the cable data network and the Residential Gateway
- Secure download of configuration and software files
- Optional configuration file security
- Remote Residential Gateway firewall management
- Standardized firewall configuration and reporting
- Simple parental control

IPCable2Home communication across the WAN and LAN is IPv4 based, leveraging specific protocols defined throughout the remainder of this document. IPCable2Home compliant devices MUST implement version 4 of the Internet Protocol suite (IPv4) [RFC 791], [RFC 3280].

The remainder of this section examines the IPCable2Home Reference Architecture from six perspectives:

- Logical view (Section 5.1)
- Functional view (Section 5.2)
- Messaging Interface view (Section 5.3)
- Informational view (Section 5.4)
- Operational view (Section 5.5)
- Physical Interface view (Section 5.6)

5.1 Logical Reference Architecture

As shown in Figure 5-1, this section introduces the logical concepts of the IPCable2Home domain, logical elements, and the IPCable2Home devices.

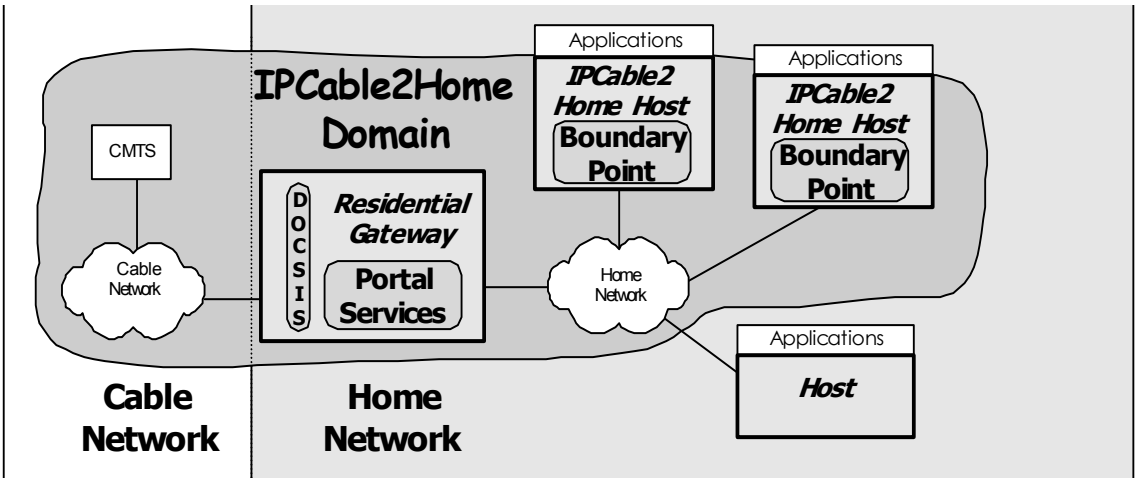

Figure 5-1 IPCable2Home Key Logical Concepts

5.1.1 IPCable2Home Domains

The IPCable2Home Domain represents the set of network elements that are compliant with this Recommendation, and is diagrammatically represented as a shaded region in Figure 5-1. This region serves as a visual tool to clearly identify those elements within the home network that are IPCable2Home compliant. Elements that reside within the IPCable2Home domain (i.e., compliant elements) are directly or indirectly manageable by cable operators. The IPCable2Home Domain exists on a per-home basis.

5.1.2 IPCable2Home Devices

The IPCable2Home architecture identifies devices in order to lend tangible context to the logical elements described in Section 5.1.3. Device definitions provide an informative way of depicting home network topology as well as logical elements located within the home network, but are not considered definitive or restrictive. IPCable2Home devices include the Residential Gateway and the IPCable2Home Host.

The Residential Gateway device (HA in J.190) represents the physical location of the Portal Services (PS) logical element, which is described in Section 5.1.3.1. The Residential Gateway has a single WAN interface, a single PS logical element, and may have one or more LAN interfaces.

The term LAN IP Device is used to refer to any LAN Device that has an IP Interface. A LAN IP Device that implements IPCable2Home functionality, making it compliant with the IPCable2Home specification, is referred to as a *IPCable2Home Host device* (HC in J.190). A LAN IP Device without IPCable2Home functionality is referred to as a *Host*.

The IPCable2Home Host device represents the physical location of the Boundary Point (BP). The BP, defined in Section 5.1.3.2, enables IPCable2Home Hosts to interact with IPCable2Home Residential Gateways. The IPCable2Home Host has only one LAN interface in the IPCable2Home Domain.

IPCable2Home assumes a home networking topology with only one DOCSIS cable modem (CM) and one IPCable2Home Residential Gateway on the home LAN. It is assumed that the DOCSIS CM is the only direct connection to the HFC. Ideally, the IPCable2Home Residential Gateway, will be directly connected to the CM with no other devices attached between the CM and IPCable2Home Residential Gateway in order for the IPCable2Home Residential Gateway to provide the specified protection to the home network. All LAN Hosts are connected to the LAN behind the IPCable2Home Residential Gateway.

5.1.3 Logical Elements

The architectural framework introduces the concept of logical elements. IPCable2Home logical elements are logically bounded functional entities that can generate and respond to specified messages. IPCable2Home logical elements operate at the IP protocol layer and above, thus remaining independent of any particular physical network technology. They also include the ability to gather and communicate information as needed to discover, manage, and deliver services over IPCable2Home networks. IPCable2Home defines a logical entity specific to each

IPCable2Home Device: The PS logical entity encapsulates IPCable2Home functionality defined for Residential Gateways and the BP logical entity encapsulates functionality defined for IPCable2Home Hosts (see Section 5.1.2 for a description of the IPCable2Home Devices).

5.1.3.1 Portal Services (PS)

The Portal Services is a logical element that provides in-premise and aggregated security, management, provisioning, addressing, and QoS services. The term "portal" is used to indicate services that interface the WAN to the LAN. This section describes features of the Portal Services logical element.

5.1.3.1.1 Standalone PS and PS with Embedded Cable Modem

The two primary functional entities possible within a Residential Gateway, the Cable Modem (CM) and the Portal Services (PS) element may use shared or independent hardware and software resources. It is the lack of resource sharing between the CM and PS functions that distinguishes the Standalone PS from an Embedded PS.

A Standalone PS MUST NOT share hardware or software components with a CM. The separation of the CM from the standalone PS MUST appear to the PS as a simple disconnection of its WAN – i.e., the PS will continue fully functional as if it had the WAN disconnected. Otherwise the PS will be considered Embedded. Given these definitions, it is possible that a PS might reside within the same physical enclosure as a CM, yet still be considered a Standalone PS.

The CM and the PS are considered to be separate elements in the Standalone and Embedded cases, and respond to unique management addresses. In the Embedded case, the CM and PS share hardware or software components, but from the management perspective, they are separate entities.

Figure 5-2 illustrates the Standalone and Embedded PS.

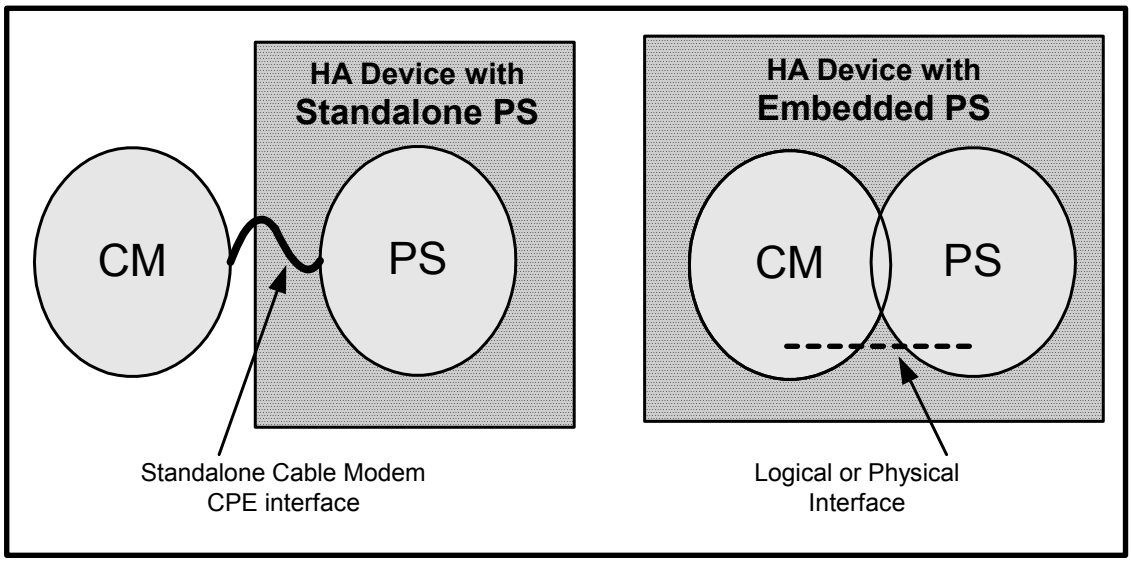

Figure 5-2 Standalone PS and PS with Embedded CM

5.1.3.2 Boundary Point (BP)

A Boundary Point (BP) is a logical element which encapsulates all of the IPCable2Home functionality defined for a IPCable2Home Host. This functionality includes messaging and behavior required for device and application discovery by the cable operator, as well as for enabling prioritized QoS on the home network. The BP interacts with the PS in order to convey device and application information and to query cable operator-provisioned preferences for application priorities.

5.1.4 Address Realms

An Address Realm is defined as "a network domain in which the network addresses are uniquely assigned to entities such that datagrams can be routed to them" [RFC 2663]. Within this Recommendation, address realms are categorized as WAN address realms and LAN address realms. (See Figure 5-3).

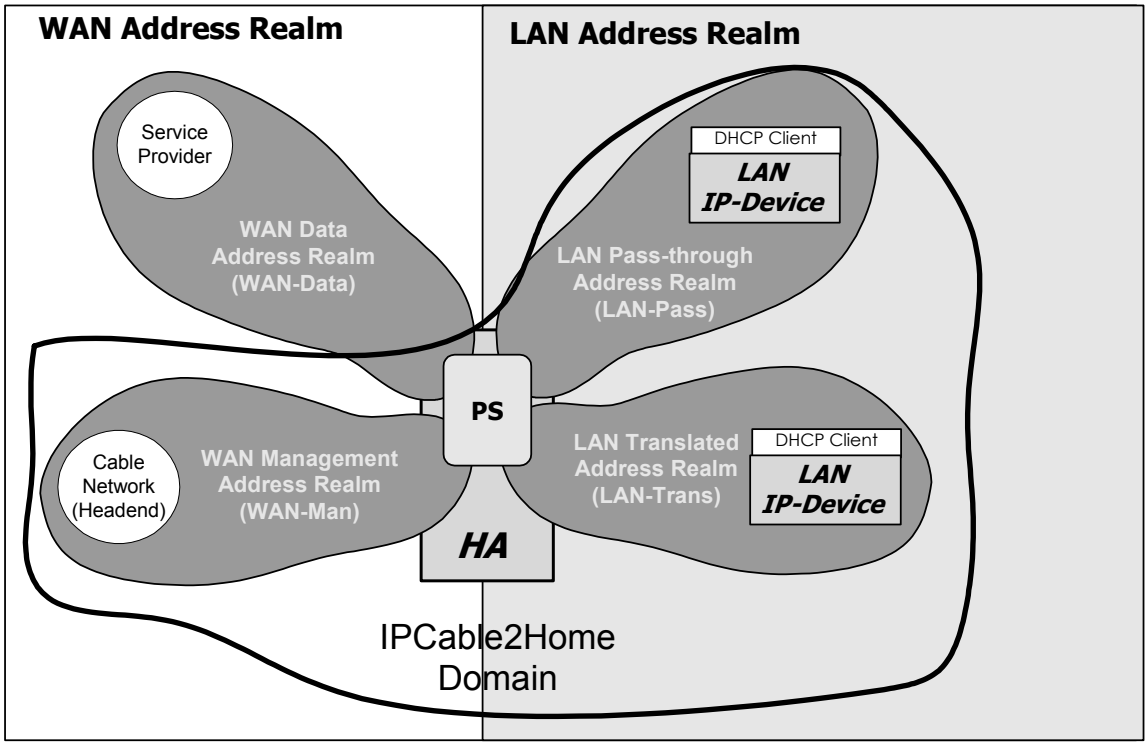

Figure 5-3 IPCable2Home Address Realms

WAN addresses reside in one of two realms: the WAN Management Address Realm (WAN-Man) or the WAN Data Address Realm (WAN-Data). LAN addresses also reside in one of two realms: LAN Passthrough Address Realm (LAN-Pass) or LAN Translated Address Realm (LAN-Trans). The properties of these addressing realms are as follows:

- The WAN Management Address Realm (WAN-Man) is intended to carry network management traffic on the cable network between the network management system and the PS element. Typically, addresses in this realm will reside in private IP address space.
- The WAN Data Address Realm (WAN-Data) is intended to carry subscriber application traffic on the cable network and beyond, such as traffic between IPCable2Home Hosts and Internet hosts. Typically, addresses in this realm will reside in public IP address space.
- The LAN Translated Address Realm (LAN-Trans) is intended to carry subscriber application and management traffic on the home network between IPCable2Home Hosts, LAN IP Devices, and the PS element. Typically, addresses in this realm will reside in private IP address space, and can typically be reused across subscribers.
- The LAN Passthrough Address Realm (LAN-Pass) is intended to carry subscriber application traffic, such as traffic between IPCable2Home Hosts, LAN IP Devices and Internet hosts, on the home network, cable network, and beyond. Typically, addresses in this realm will reside in public IP address space.

On the LAN side, the addresses in the LAN Passthrough Address Realm (LAN-Pass) are directly extracted from the addresses in WAN Data Address Realm. These are used by LAN IP Devices and applications such as IPCablecom services that are intolerant of address translation and require a globally routable IP address. Additionally on the LAN side, LAN IP Devices may be assigned translated addresses from the LAN Translated Address Realm (LAN-Trans). The LAN-Pass and LAN-Trans Address Realms exist on a per-home basis.

Physical LAN interfaces in the PS are assigned an index in accordance with the Interfaces Group MIB [RFC 2233]

as described in Section 6.3.3.1.4.8 Interfaces Group MIB. A virtual LAN interface aggregating the physical LAN interfaces is also defined for the PS in Section 6.3.3.1.4.8. The LAN-side IP address defined for the PS is "bound" to this virtual interface. PS DHCP and domain name server functions, and the PS router function, are applications implemented in the PS addressed using the LAN-side IP address bound to the virtual LAN interface.

5.2 IPCable2Home Functional Reference Model

IPCable2Home Functions are IP-based services to be implemented by the PS, the BP, or the cable operator's data network, and support the delivery of cable-based services. IPCable2Home functions are defined for each of the major specification areas: Provisioning, Management, Security, and Quality of Service.

Sub-elements are defined for both the PS and the BP. Sub-elements represent groupings of related functionality within the PS and BP. The PS and BP logical elements can contain any number of sub-elements, and sub-elements may themselves contain sub-groupings of functions (i.e., sub-elements within sub-elements).

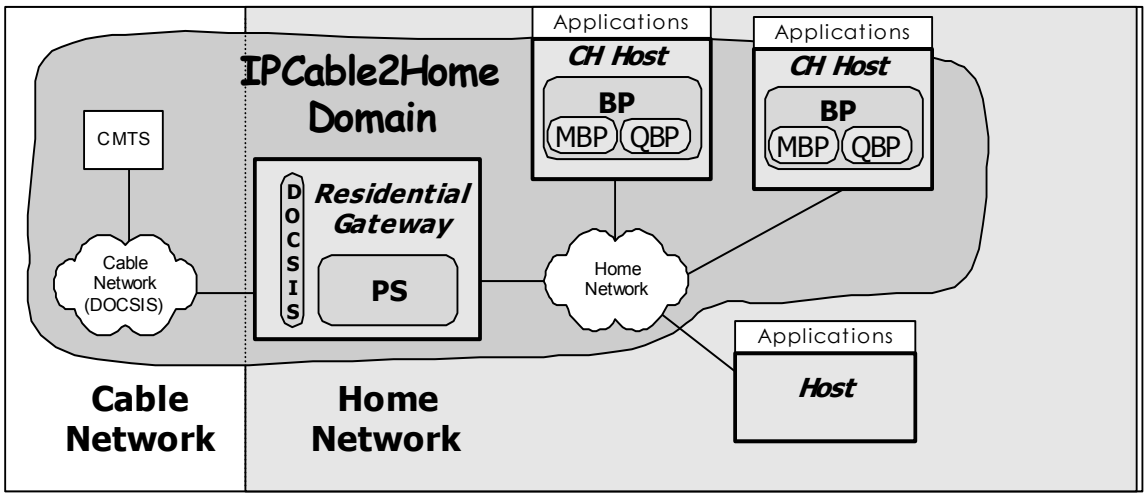

Figure 5-4 IPCable2Home Sub-elements

The PS contains a number of sub-elements, which are introduced below. Within the Boundary Point there are two primary sub elements, the Management Boundary Point (MBP) and the Quality of Service Boundary Point (QBP), which define discovery and management, and QoS functionality, respectively. The QBP contains additional sub-elements of its own.

5.2.1 IPCable2Home Management and Provisioning Functions

To support the requirements during the provisioning and management of IPCable2Home Hosts within the home, IPCable2Home uses management and provisioning functions that reside in the cable data network, and defines functions for the PS and for the BP. Cable network-based management and provisioning functions include a number of services used by IPCable2Home-defined management and provisioning processes. Portal Services management and provisioning functions are located within the Residential Gateway and include server-like, client-like, and other types of functionality. Boundary Point functions are found within IPCable2Home Hosts and typically include client as well as other types of functionality. Examples of Cable Network, PS, and BP functions are introduced in Table 5-1, Table 5-2, and Table 5-3 and are illustrated in Figure 5-5.

Table 5-1 Cable Network Management Functions

Cable Network Management Functions	Description
Cable Network DHCP Server	The DHCP server is a cable network component that provides address information for the WAN-Man and WAN-Data address realms to the PS
Cable Network Management Servers	The IPCable2Home management messaging, download, event notification servers including protocols such as SNMP, SYSLOG, and TFTP [RFC 2349]
Cable Network Time of Day Server	The time of day (ToD) server provides clients with the current time of day.

Table 5-2 PS Management and Provisioning Functions

Management Portal Functions	Description
IPCable2Home Address Portal (CAP)	Within the PS, the CAP interconnects the WAN and LAN address realms for data traffic. (See CAT/Passthrough)
IPCable2Home Address Translation (CAT)	A sub-function of the CAP, a CAT translates public IP network addresses on the WAN-Data side of the CAP to private IP network addresses within a single logical subnet on the LAN-Trans side.
Passthrough	A sub-function of the CAP, the Passthrough function bridges packets on the WAN-Data side of the CAP to the LAN-Pass side unchanged.
IPCable2Home Management Portal (CMP)	The function that provides an interfaces between the MSO and the PS -database.
IPCable2Home DHCP Portal (CDP)	Address information functions (e.g. those transmitted via DHCP) including a server for the LAN realm and a client for the WAN realms
IPCable2Home Naming Portal (CNP)	The CNP provides a simple DNS service for LAN IP Devices requiring naming services.
IPCable2Home Testing Portal (CTP)	The CTP provides a remote means to initiate pings and loopbacks within the LAN.
HTTP Server	HTTP is the transport protocol used to convey SOAP messaging on the LAN. The PS contains an HTTP server which serves data upon BP requests
XML and SOAP Parsers	SOAP and XML are used for messaging on the LAN. The PS contains parsers for both.

Table 5-3 BP Management and Provisioning Functions

Management Client Functions	Description
Cable Home Host DHCP Client	The IPCable2Home DHCP client function is a in-home component used during the LAN IP Device provisioning process to dynamically request IP addresses and other logical element configuration information.
IPCable2Home Host Loopback responder	Within LAN IP Device, the loopback responder loops data sourced from the CTP loopback function back to the CTP loopback function.
HTTP Client	HTTP is the transport protocol used to convey SOAP messaging on the LAN. The BP contains an HTTP client which requests data from the PS housed HTTP server
XML and SOAP Parsers	SOAP and XML are used for messaging on the LAN. The BP contains parsers for both.

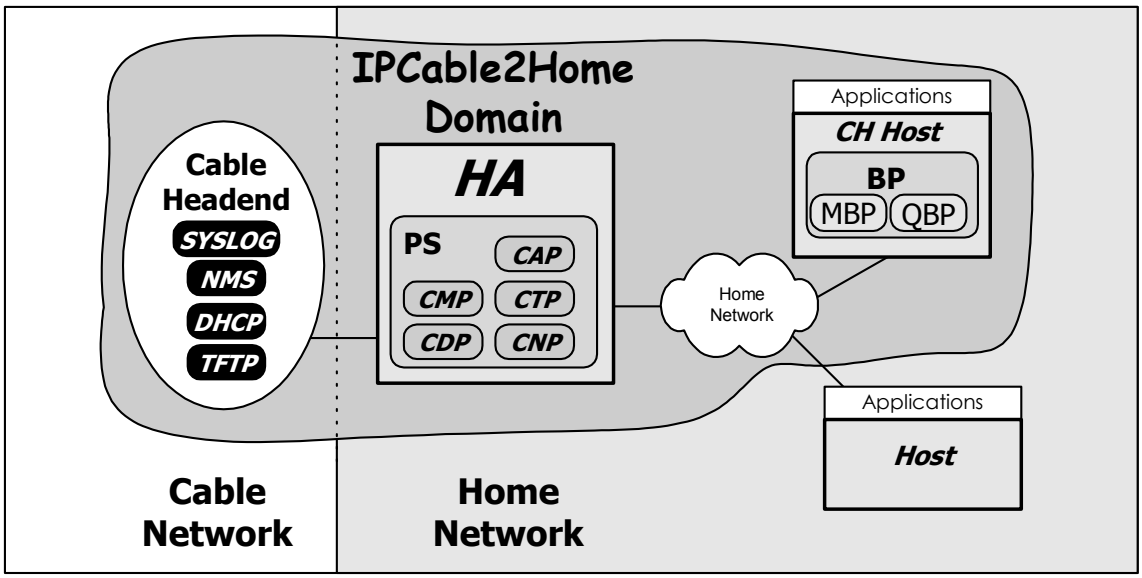

Figure 5-5 IPCable2Home Management Elements

5.2.2 IPCable2Home Security Functions

To support the IPCable2Home security requirements (see Section 11.2.1), IPCable2Home uses security functions that reside in the cable data network and defines functions for the PS. These functions reside within the IPCable2Home Security Domain, which exists on a per-home basis. Cable network-based security functions include servers used for key distribution, encryption, and authentication. Portal Services security functions are located within the Residential Gateway includes client functions and other types of functions. Examples of cable network-based and PS security functions are introduced in Table 5-4 and Table 5-5 and are illustrated in Figure 5-6.

Table 5-4 Portal Services Security Functions

Portal Service Security Functions	Description
IPCable2Home Security Portal (CSP)	The CSP communicates with Headend security servers, and includes functions that provide client side participation in the authentication, key exchange and certificate management processes. Other security functions include management message security, participation in secure download processes, and remote firewall management.
Firewall (FW)	The Firewall provides functionality that protects the home network from malicious attack.

Table 5-5 Cable Network Security Function

Cable Network Security Functions	Description
Key Distribution Center (KDC) Servers	The key distribution center (KDC) servers provide security services to the CSP and include functions that participate in the authentication and key exchange processes.

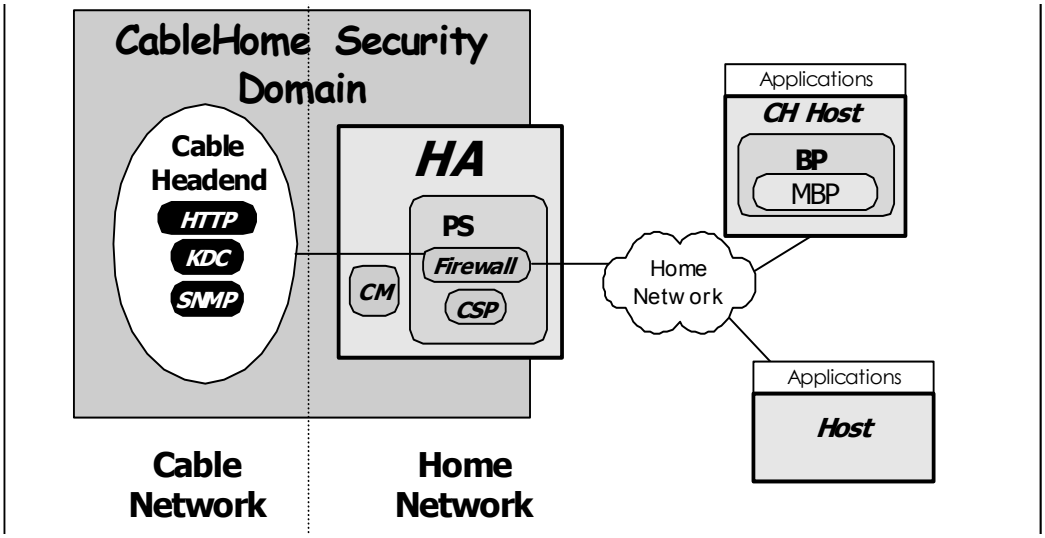

Figure 5-6 IPCable2Home Security Elements

5.2.3 IPCable2Home QoS Functions

To support the Quality of Service requirements (see Section 10.2.1), IPCable2Home defines functions for the PS and the BP. Portal Services QoS functions are located within the Residential Gateway and include a server function and other types of functions. BP QoS functions are located within IPCable2Home Host devices and include a client and other types of functions. Examples of PS and BP QoS functions are introduced in Table 5-6 and Table 5-7 and are illustrated in Figure 5-7.

Table 5-6 Portal Services QoS Functions

Portal Service QoS Functions	Description
QoS Characteristics Server (QCS)	Acquires QoS priority information for applications from the cable network management system. Acquires BP application list from the BP. Provides information about application priorities to the BP, as established by the cable operator.
QoS Forwarding and Media access (QFM)	Orders the packets arriving from multiple LAN interfaces to the PS and forwards them to a destination LAN interface according to their priorities. Also provides prioritized access to the shared media during the packet transmission based on the packet priority.

Table 5-7 BP QoS Function

Boundary Point QoS Functions	Description
QoS Characteristics Client (QCC)	Provides information to the PS about applications residing on the IPCable2Home Host and also requests information about application priorities established by the MSO. Also provides prioritized access to the shared media during the packet transmission based on the packet priority.

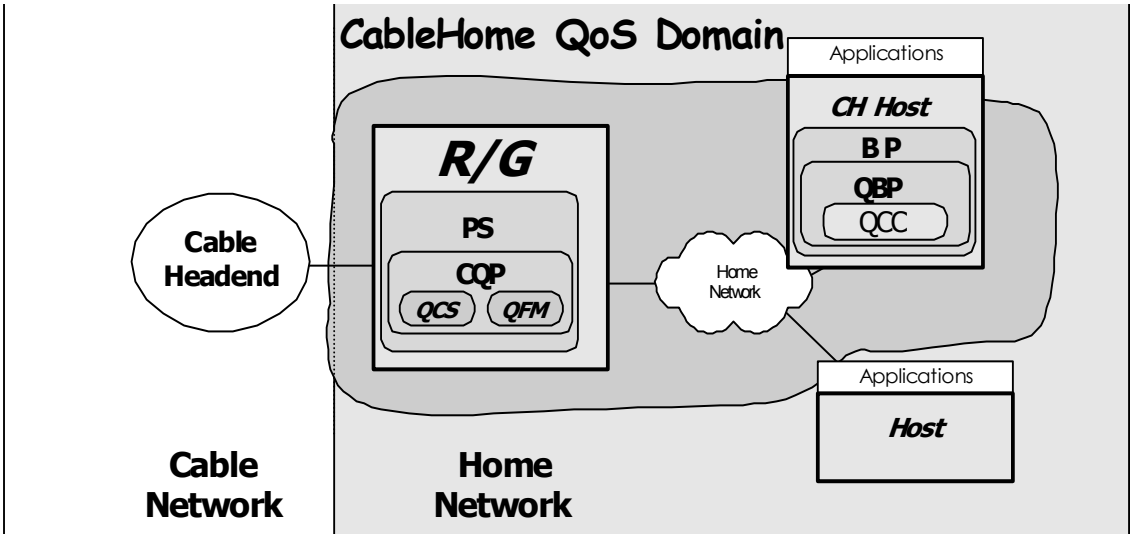

Figure 5-7 IPCable2Home QoS Elements

5.3 IPCable2Home Messaging Interface Model

Communication between the functions in the cable data network, Residential Gateway, and LAN IP Devices occur on messaging interfaces identified and labeled in Figure 5-8. The types of messaging interfaces are differentiated by the elements that are involved in the communication.

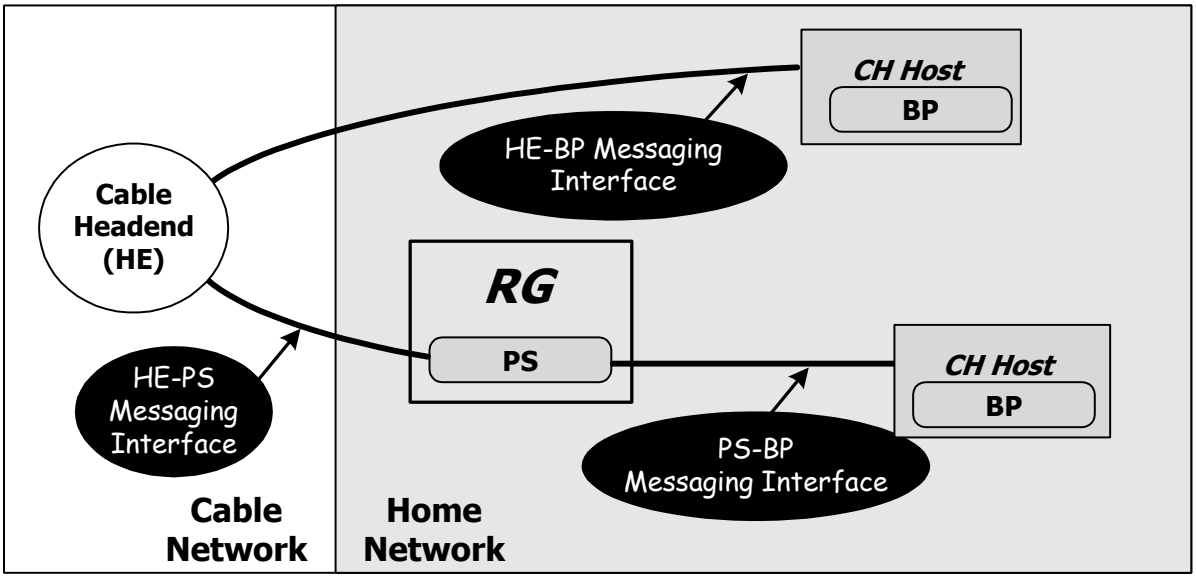

Figure 5-8 IPCable2Home Reference Interfaces

Table 5-8 identifies interfaces for which IPCable2Home specifies messaging.

Table 5-8 Valid Interface Paths for Each Functionality

		Interface		
Functionality	**Protocol**	**HE-PS**	**HE-BP**	**RG-BP**
Name service	DNS	Unspecified	Unspecified	This Recommendation
Software Download	TFTP	This Recommendation	Unspecified	Unspecified
Address Acquisition	DHCP	This Recommendation	Unspecified	This Recommendation
Management (single) (bulk)	 SNMP TFTP or HTTP	 This Recommendation This Recommendation	 Unspecified Unspecified	 Unspecified Unspecified
Event Notification	SNMP SYSLOG	This Recommendation This Recommendation	Unspecified	Unspecified
QoS	IPCablecom QoS Protocols, IPCable2Home Priorities SOAP/XML	Unspecified	IPCablecom	This Recommendation
Security (key distribution)	Kerberos	This Recommendation	Unspecified	Unspecified
Security (authentication)	Kerberos or TLS	This Recommendation	Unspecified	Unspecified
Ping	ICMP	This Recommendation	Unspecified	This Recommendation
Loopback/Echo	UDP/TCP	Unspecified	Unspecified	This Recommendation
Application Discovery	SNMP SOAP/XML	This Recommendation	Unspecified	 This Recommendation

5.4 IPCable2Home Information Reference Model

The operation of the management model is based upon a store of information maintained in the PS by the various sub-elements of the PS (CAP, CDP, CMP, etc.). These sub-elements need a means of interacting via information exchange, and the PS Database is a conceptual entity that represents a store for this information. The PS Database is not an actual specified database per se, but rather a tool to aid in the understanding of the information that is exchanged between the various IPCable2Home elements.

Figure 5-9 shows the relationship between the database and the PS functions. Table 5-9 describes the typical information associated with each of these functions. Figure 5-10 shows a detailed example implementation indicating the set of information, the functions that derive the information, and the relationships between the functions and the information.

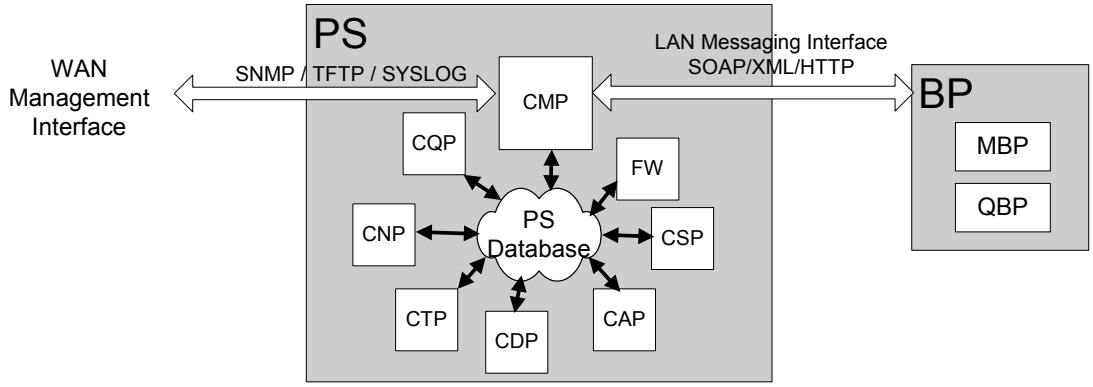

Figure 5-9 PS Function and Database Relationship

The PS Database stores a myriad of data relationships. The CMP provides the WAN management interface (SNMP) to the PS database. The functions within the PS enter and revise data relationships in the PS Database. Additionally, the Functions within the PS may retrieve information from the PS Database that is maintained by other Functions within the PS.

Table 5-9 Typical PS Database information examples

Name	Usage (in general)
CDP Information	Information associated with addresses acquired and allocated via DHCP.
CAP information	Information associated with IPCable2Home address translation mappings.
CMP information	Information associated with the state of the PS functions. Information about IPCable2Home Host devices.
CTP information	Information associated with results of LAN test performed by the CMP.
CNP information	Information associated with LAN IP Device name resolution.
USFS information	Information associated with the Upstream Selective Forwarding Switch function.
CSP information	Information associated with authentication, key exchange, etc.
Firewall information	Information associated with the behavior of the Firewall (ruleset), firewall events and logging.
Event information	Information associated with the local log for all general events, traps, etc.
IPCable2Home Host Device Information	BP Device Profile information collected through BP_Init messaging from IPCable2Home hosts.
IPCable2Home Host QoS Characteristics Information	QoS Characteristics received from cable operator and QoS Profile information received from the IPCable2Home hosts via BP_Init Messaging.

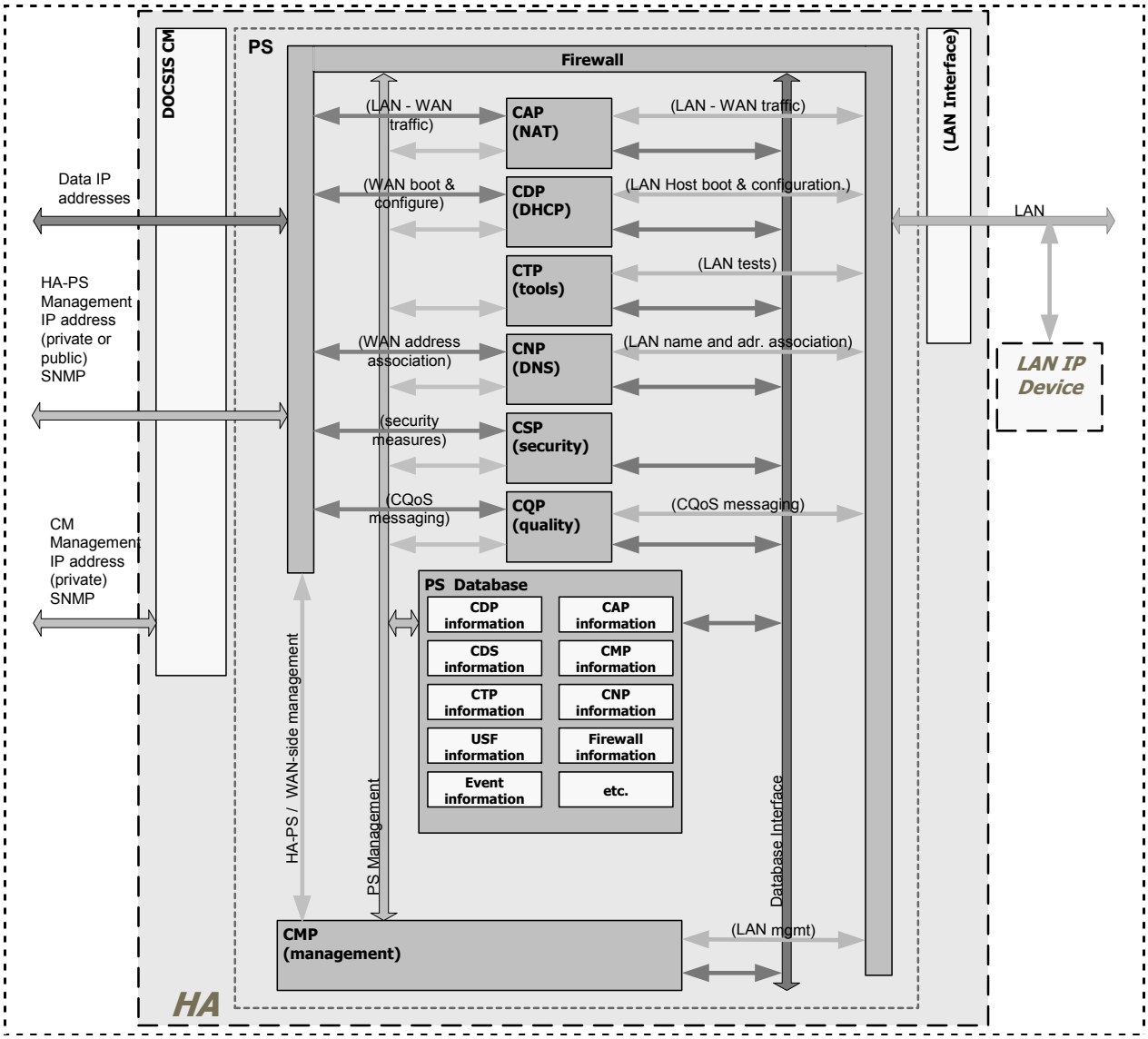

Figure 5-10 PS Database Detailed Example Implementation

The PS is primarily managed from the WAN via the CMP, and to a large degree this involves access to the information in the PS Database. Management is used for initialization and provisioning of the PS functions, and remote diagnostics or status of the LAN. The diagnostics may rely on the CTP to get better visibility into the current state of the LAN. Connectivity and rudimentary network performance can be measured.

The CNP is the LAN Domain Name Server (DNS). All LAN-Trans LAN IP Devices are configured by the CDP to use the CNP as the primary Name Server. The CNP resolves textual host names of LAN IP Devices, returning their corresponding IP addresses and in addition, refers LAN IP Devices to external DNS servers for requests that cannot be answered from local information.

The CDP contains the address functions to act as the DHCP server in the LAN-Trans realm and implements a DHCP client in the WAN realms.

The CAP creates address translation mappings between the WAN-Data and LAN-Trans address realms. The CAP is also responsible for Upstream Selective Forwarding Switch decisions to preserve HFC upstream channel (WAN) bandwidth from the local LAN only traffic. Finally, the CAP contains the Passthrough function, which bridges traffic between the LAN and WAN address realms.

The CSP provides PS authentication capabilities as well as key exchange activities.

The CQP is part of a system that enables IPCable2Home QoS. The CQP provides IPCable2Home traffic priorities

as well as differentiated media access functions.

5.5 IPCable2Home Operational Modes

The functionality of the Portal Services element is compatible with a variety of cable network infrastructures, which are accommodated by a number of different PS operational modes. These various operating modes enable the PS to function properly within a CableModem (J.112 or J.122) only provisioning infrastructure, as well as within an CableModem plus IPCablecom provisioning infrastructure. The CableModem plus IPCablecom provisioning IPCable2Home infrastructure builds upon the CableModem infrastructures to enable additional services, and incorporates a number of capabilities that are similar to those within a IPCablecom provisioning system.

For the purpose of configuration, the PS may operate within one of two provisioning modes:

- The DHCP Provisioning Mode
- The SNMP Provisioning Mode

If the PS is not configured to operate in either DHCP Provisioning Mode or SNMP Provisioning Mode, it assumes that the back office support is not currently available, and will default to operate in Dormant CableHome Mode. In Dormant CableHome Mode, the Residential Gateway will be fully operational from the user perspective, but it will not be operator configured or managed.

When the PS is configured to operate in DHCP Provisioning Mode, it can be configured to begin a Transport Layer Security (TLS) session over HTTP in order provide secure download of PS and Firewall configuration files.

When the PS is operating within the DHCP Provisioning Mode, it can operate in one of two Network Management sub-modes:

- NmAccess Mode
- SNMP v3 Coexistence Mode

When the PS is configured to operate in SNMP Provisioning Mode, it operates in SNMPv3 Coexistence Network Management Mode only.

Figure 5-11 illustrates the various PS operational modes along with the associated triggers for each. See Section 7.3.3.2.4 (CDC Requirements) for a full description of provision mode determination.

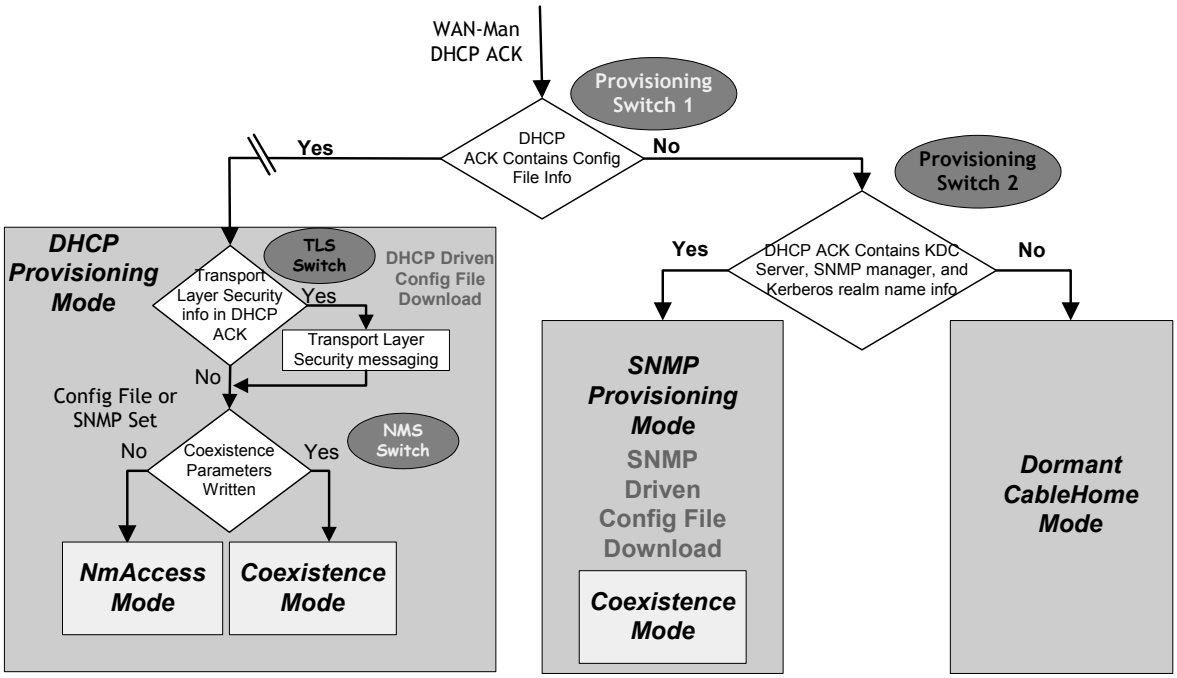

Figure 5-11 PS Operational Modes

Table 5-10 describes the infrastructures within which each PS mode is intended to operate.

Table 5-10 PS Infrastructures

Mode	Capability Directly Effected	Intended Infrastructure
SNMP Provisioning Mode	Configuration file download.	The CableModem plus IPCablecom provisioning Infrastructure
DHCP Provisioning Mode	Configuration file download.	CableModem infrastructures with IPCable2Home support
DHCP Provisioning Mode: with TLS/HTTP	Secure configuration file download	CableModem infrastructures with IPCable2Home and TLS support
DHCP Provisioning Mode: NmAccess Network Management Mode	SNMP version used between NMS and PS	J.112 (1999?) Infrastructure (SNMP v1/v2) with IPCable2Home support
DHCP Provisioning Mode: SNMP Coexistence Network Management Mode	SNMP version used between NMS and PS	J.112 & J.122, and The CableModem plus IPCablecom provisioning Infrastructures (SNMP v3) with IPCable2Home support
Dormant IPCable2Home Mode	Configuration and Management	No IPCable2Home support

5.6 Physical Interfaces on the Residential Gateway

There are many types of physical interfaces that may be implemented on a device containing PS functionality. Several are described in the following list:

- WAN Networking Interfaces, to the cable network via the cable modem acting as a transparent bridge for a PS with an embedded cable modem, and other WAN Networking Interfaces, intended for WAN connection, in the Standalone PS case.
- LAN Networking Interfaces for connection to LAN IP Devices and IPCable2Home hosts.
- Hardware Test Interfaces, such as JTAG and other proprietary approaches, which are part of the silicon and don't always have software controls to turn the interfaces off. These interfaces are hardware state machines that sit passively until their input lines are clocked with data. Though these interfaces can be used to read and write data, they require an intimate knowledge of the chips and the board layout and are therefore difficult to "attack". Hardware test interfaces MAY be present on a device implementing PS functionality. Hardware test interfaces MUST NOT be either labeled or documented for customer use.
- Management Access Interfaces, also called console ports, which are communications paths (usually RS-232, but could be Ethernet, etc.) and debugging software that interact with a user. The software prompts the user for input and accepts commands to read and write data to the PS. If the software for this interface is disabled, the physical communications path is disabled. A PS MUST NOT allow access to PS functions via a Management Access Interface. (PS functions are defined by this Recommendation.) Access to PS functions MUST only be allowed via interfaces specifically prescribed by the this Recommendation, e.g., operator-controlled access via SNMP.
- Read-only Diagnostic Interfaces can be implemented in many ways and are used to provide useful debug, trouble-shooting, and PS status information to users. A PS MAY have Read-only Diagnostic Interfaces.
- Some products might choose to implement higher layer functions (such as customer premise data network functions) that could require configuration by a user. A PS MAY provide the ability to configure non-IPCable2Home functions. Management interface (read/write) access to PS functions MUST NOT be allowed through the mechanism used for configuring non-IPCable2Home functions.

6 MANAGEMENT TOOLS

6.1 Introduction/Overview

The IPCable2Home Management Tools provide the cable operator with functionality to monitor and configure the Portal Services (PS) element, to discover LAN IP Devices and the applications they offer, to remotely check connectivity between the PS and LAN IP Devices, to provide Quality of Service policy to BPs in support of prioritized QoS between IPCable2Home Host devices, and to report on status and exception events in the PS. This section describes and specifies requirements for these capabilities.

Differences between Management Tools defined in J.191 and those defined in this Recommendation are listed below:

- This Recommendation adds the requirement for the PS to support SNMP management from any LAN Interface
- This Recommendation adds the requirement for both the PS and the BP to support PS-BP messaging for the exchange of QoS priorities
- This Recommendation adds the requirement for the BP to implement a device profile in XML format
- This Recommendation adds the following MIB objects to the PS:

  - objects needed to support prioritized Quality of Service on the LAN

  - objects supporting enhanced firewall functionality

  - objects enabling the cable operator to discover attributes of IPCable2Home Host devices

6.1.1 Goals

The goals for the IPCable2Home Management Tools include:

- Provide a means for the cable operator to discover LAN IP Devices.
- Provide cable operators with visibility to LAN IP Devices.
- Provide cable operators with visibility to applications on IPCable2Home Host devices.
- Define a method for passing QoS priorities to the applications on IPCable2Home Host devices.
- Define a minimum set of remote diagnostic tools that will allow the cable operator to verify connectivity between the Portal Services element and any LAN IP Device.
- Provide cable operators with access, via the MIBs, to internal data in the PS element and enable the cable operator to monitor IPCable2Home-specified parameters and to configure or re-configure IPCable2Home-specified capabilities as necessary.
- Provide a means for reporting exceptions and other events in the form of SNMP traps, messages to a local log, or messages to a system log (SYSLOG) in the cable network.

6.1.2 Assumptions

The assumptions for the IPCable2Home network management environment include the following:

- IPCable2Home-compliant devices implement the Internet Protocol (IPv4) suite of protocols.
- IPCable2Home Host Devices implement a Device Profile and a Quality of Service Profile in XML format.
- SNMP is used for the exchange of management messages between the cable network NMS and the PS in the IPCable2Home Residential Gateway device. SNMP provides visibility for the NMS to interfaces on the PS, via access to internal PS data, through required MIBs.
- Any of SNMPv1/v2c/v3 can be used as a management protocol between the NMS and the IPCable2Home Portal Services element.
- LAN IP Devices implement a DHCP client.

- The IPCable2Home Residential Gateway and LAN IP Devices support ICMP.
- The PING utility supplies functionality sufficient to provide the cable operator with the desired information about connectivity between the PS element and LAN IP Devices.

6.2 Management Architecture

6.2.1 System Design Guidelines

The Management Tools system design guidelines are listed in Table 6-1. This list provided guidance for the development of the IPCable2Home management tools specifications.

Table 6-1 Management Tools System Design Guidelines

Reference	Management Tools System Design Guidelines
Mgmt 1	The PS will implement SNMPv1/v2c/v3 protocols to provide access to internal Portal Services data.
Mgmt 2	The PS will be capable of issuing a an ICMP Request (Ping) command to any LAN IP Device specified by the cable operator and store results in the PS Database. Remote Ping test results will be accessible through CTP MIB objects.
Mgmt 3	The PS will be capable of executing a Connection Speed Test with a specified LAN IP Device specified by the cable operator and store results in the PS Database. Remote Connection Speed test results will be accessible through CTP MIB objects.
Mgmt 4	The PS element will be capable of reporting events.
Mgmt 5	The PS element will be capable of communicating with IPCable2Home Host devices in the LAN-Pass and LAN-Trans realms for the exchange of device attributes, QoS priorities, and IPCable2Home Host application information.
Mgmt 6	In the event that the PS loses connectivity with the cable data network and its applications, the Discovery function and LAN Messaging function will continue to operate.

6.2.2 Management Tools System Description

As shown in Figure 6-1, IPCable2Home Management Tools architecture consists of the following components: (1) the IPCable2Home Management Portal (CMP), (2) the IPCable2Home Test Portal (CTP), (3) a Management Information Base (MIB), (4) an SNMP Network Management System (NMS) that is part of the cable network, and (5) a Device Profile in XML format implemented by each IPCable2Home Host device (BP logical element).

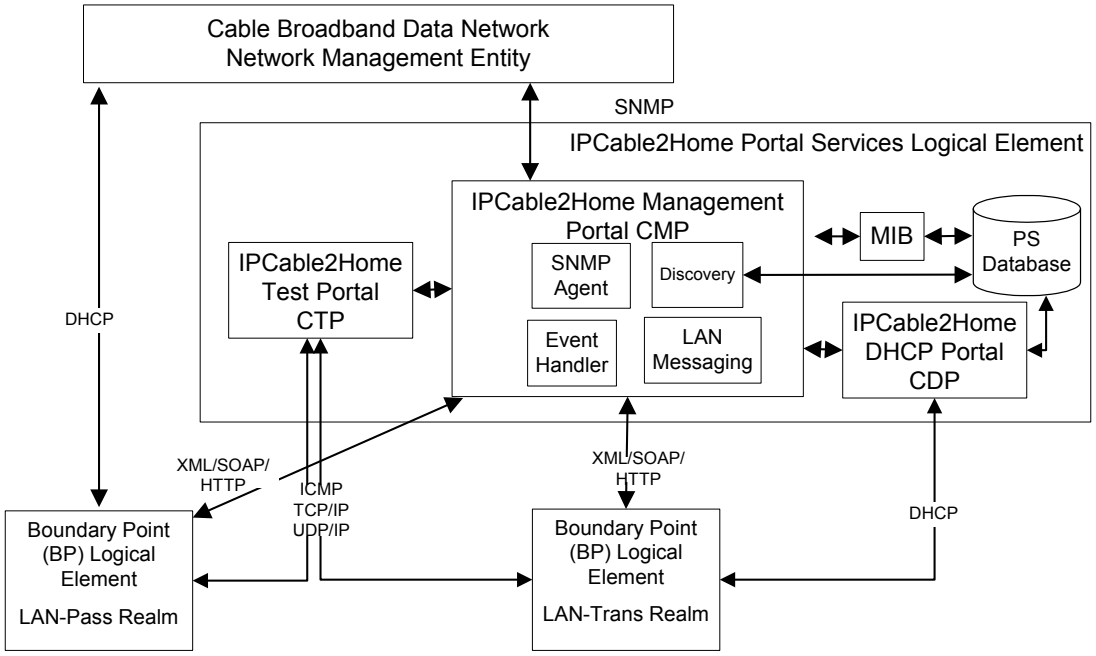

Figure 6-1 IPCable2Home Management Architecture

The cable data network NMS monitors and configures the PS by accessing the PS Database through MIBs specified in Section 6.3.3.1.4.7. The cable operator accesses IPCable2Home Host Device and IPCable2Home Residential Gateway attributes through the PSDev MIB [Annex E.4] and through the QoS MIB [Annex E.7], and configures IPCable2Home Host devices with QoS policy (in the form of QoS priorities) using the PS as a proxy.

Upon receiving DHCP ACKNOWLEDGE (DHCPACK) [RFC 2131] from its DHCP server the BP logical element in each IPCable2Home Host device initiates communication with the PS via a LAN messaging interface. This messaging, in the form of Simple Object Access Protocol (SOAP) on Hypertext Transfer Protocol (HTTP) transport, is done to inform the PS of device attribute information (Device Profile) and a list of applications (QoS Profile) implemented in the IPCable2Home Host. When the PS receives the Device Profile and QoS Profile it does the following:

- Stores the BP Device Profile information in a BP device profile MIB table (cabhPsDevBpProfileTable).

  The BP Device Profile enables the cable operator to discover information about IPCable2Home Host devices in the LAN-Pass realm, and provides the cable operator with information about IPCable2Home Host devices in the LAN-Trans realm in addition to the information obtained via DHCP messaging between the PS and the LAN-Trans BP.

- Stores the BP QoS Profile information in a BP Application Priority MIB table (cabhPriorityQosBpTable).

  The BP QoS Profile enables the cable operator to discover applications implemented on IPCable2Home Host devices. The applications are identified by the IANA "well-known" port number registered to them.

If the cable operator has provisioned the PS with QoS policy by populating the Application Priority Master Table (cabhPriorityQosMasterTable), the PS will also provide QoS priorities from the table to the BP via the same LAN messaging interface. This procedure is described in Section 10.3.2.4.2 LAN Information Exchange.

The NMS can also directly communicate with LAN IP Devices in the IPCable2Home LAN-Pass realm.

The IPCable2Home DHCP Portal, described in the Provisioning Tools section (Section 7), plays a role in basic LAN IP Device discovery. Through DHCP communication between LAN IP Devices and the CDP, the LAN IP Device provides its hardware address and may provide configuration information to the CMP through DHCP Option codes. The CMP will use the information to populate CDP MIB LAN Address Table (cabhCdpLanAddrTable) objects.

The CMP and CTP functional elements reside within the PS. The PS logical element may be co-resident with an embedded cable modem or stand alone, without embedded cable modem functionality, as described in Section 5.1.3.1.1.

The CM and PS are separate and independent management entities. In the case of a PS with an embedded cable modem, no data sharing between CM and PS is implied, with the following exceptions:

1. the software image download is controlled via the cable modem's MIB,

2. the MIB for SNMP [RFC 3418], the SNMP Group of MIB-2 (mib-2 11) [RFC 1213], the IP Group and the ICMP Group of the SNMPv2 MIB for IP [RFC 2011], and the SNMPv2 MIB for UDP [RFC 2013] are allowed to be shared between the PS and CM.

In a PS with an embedded cable modem, the cable modem's docsDevSoftware objects are accessed to set up, initiate, and monitor the download of a single combined software image. This process is described in Section 11.8, Secure Software Download for the PS.

Because of this management independence, the CM and PS respond to different and independent management IP addresses. CM MIB Objects are only visible when the manager accesses them through the CM management IP address, and are not visible via the PS management IP address (and vice-versa). The SNMP access rights to the PS and CM entities MUST be set independently. IPCable2Home does not preclude the use of a single SNMP agent for a PS with an embedded CM.

The Portal Services element supports SNMPv1, SNMPv2c, and SNMPv3 protocols. Section 5.5 introduced the provisioning modes supported by a IPCable2Home Portal Services element, and Section 7 provides additional detail about these modes. The provisioning mode in which the PS is operating partially determines which version of SNMP the PS uses. Additional detail is provided in Section 6.3.3.

6.3 PS Logical Element - IPCable2Home Management Portal (CMP)

The IPCable2Home Management Portal (CMP) is a sub-element of the PS logical element. It serves as the hub of management control of the PS and for discovery of devices present on the LAN.

The CMP aggregates and interconnects management information in the WAN-Man and LAN-Trans realms, since they are not directly accessible to each other.

6.3.1 CMP Goals

The goals for the IPCable2Home Management Portal include:

- Enable the NMS to remotely view and update IPCable2Home Address Portal (CAP) configuration information
- Enable the NMS to remotely view and update Firewall configuration information
- Enable remote testing of connectivity between the CableHome Residential Gateway and LAN IP Devices in the LAN-Trans realm, via the IPCable2Home Test Portal (CTP)
- Enable remote configuration of LAN IP Device Addressing parameters
- Enable viewing of LAN IP Device information obtained via the IPCable2Home DHCP Portal (CDP)
- Provide cable operator access to the attributes of IPCable2Home Host devices and applications implemented by IPCable2Home Host devices, acquired through the IPCable2Home discovery process
- Support the exchange of device attributes, application list, and QoS priorities for applications between the IPCable2Home Residential Gateway and IPCable2Home Host devices
- Enable viewing of the results of LAN IP Device performance monitoring done by the IPCable2Home Test Portal (CTP)
- Enable the NMS to access other PS configuration parameters
- Facilitate security by providing access to security parameters, and the use of SNMPv1/v2c/v3 in the appropriate network management mode

- Provide the capability to disable LAN segments

6.3.2 CMP Design Guidelines

The CMP design guidelines are listed in Table 6-2. This list provides guidance for the specification of CMP functionality.

Table 6-2 CMP System Design Guidelines

Reference	CMP System Design Guidelines
CMP 1	Interfaces will support the management and diagnosis features and functions required to support cable-based services provisioned across the home network.
CMP 2	Loss of connection between broadband service provider(s) and the home network will not disable or degrade the operation of internal home networking functions
CMP 3	The IPCable2Home Hosts in the home network should recover from a power outage and return to a reasonable operational state upon the reinstatement of power.
CMP 4	Home network devices will be easy to install and configure for operation, much like a home appliance.
CMP 5	The PS and LAN IP Devices will support a protocol for discovering LAN IP Devices connected to the home LAN.
CMP 6	The PS will provide to the cable operator, upon request, information about devices added to the home LAN.
CMP 7	The PS and BP will support a protocol for exchanging IPCable2Home Host Device attributes and applications implemented by IPCable2Home Host Devices, and the QoS priorities for those applications.
CMP 8	The PS will provide to the cable operator, upon request, information about IPCable2Home Host Device attributes and applications implemented by IPCable2Home Host Devices.
CMP 9	Discovery protocol message exchange within the home LAN will not noticeably degrade performance of the home LAN.
CMP 10	Discovery messaging will not propagate onto the WAN.

6.3.3 CMP System Description

The CMP is responsible for the following important IPCable2Home capabilities:

- Enable management of the Portal Services functions from the cable operator's data network Network Management System (NMS) by providing access to the PS Database and its state variables through IPCable2Home-specified Management Information Base (MIB) objects
- Enable visibility to the PS Database for the subscriber through IPCable2Home-specified MIB objects
- Enable exchange of QoS priorities between the PS and BP
- Enable the manager to remotely discover devices connected to the home LAN and the applications running on them
- Process and log event messages

The CMP is comprised of the following four functions to support the management and discovery responsibilities listed above. These functions are also shown in Figure 6-1:

1. SNMP agent function:

   The SNMP agent function receives and processes SNMP messages from the WAN Interface through the WAN-Man IP address and from the LAN Interface, through the PS Server Router IP address. It provides access to MIB objects for the purpose of monitoring and/or configuring PS and LAN IP Device functionality.

2. Event handling function:

   The CMP reports events according to the settings of the docsDevEvent table settings. The list of supported events appears in Annex B.

3. Discovery function:

   The CMP, through its discovery functionality, acquires information about each IPCable2Home Host device and the applications on it. The CMP stores this information in the PS database and makes it available to an SNMP management entity, via the PSDev MIB [Annex E.4] and QoS MIB [Annex E.7].

4. LAN Messaging function:

   The CMP exchanges QoS parameters and Device Profile attributes in XML format, with IPCable2Home Hosts across the LAN using Simple Object Access Protocol.

These functions are described in Section 6.3.3.1 - Section 6.3.3.4.

6.3.3.1 CMP SNMP Agent Function

6.3.3.1.1 SNMP Agent Function Goals

Goals of the SNMP Agent function of the CMP are listed below:

- Receive and process SNMP messages received through PS WAN-Man and PS Server Router (LAN) Interfaces
- Provide SNMP manager access to the PS Database through IPCable2Home-specified MIBs
- Enforce PS Database access rules defined by the docsDevNmAccessTable and VACM views
- Support authentication and encryption/decryption processes for SNMP defined by IETF RFCs
- Adhere to SNMP implementation rules and guidelines defined by IETF RFCs

6.3.3.1.2 SNMP Agent Function System Design Guidelines

The system design guidelines listed in Table 6-3 guided development of SNMP Agent Function requirements.

Table 6-3 System Design Guidelines

Reference	SNMP Agent Function System Design Guidelines
SNMP Agent 1	The PS will provide remote access to manageable parameters in the PS database via specified MIBs.
SNMP Agent 2	The PS will implement an SNMP agent compatible with existing cable data network management systems.
SNMP Agent 3	The PS will support access control methods enabling the cable operator to configure control of PS Database access.

6.3.3.1.3 SNMP Agent Function System Description

The CMP SNMP Agent function serves as the hub of Management control for WAN side management accesses and it gathers information for, and interconnects management of, WAN Management and LAN network elements. It also supports management messaging, via SNMP through any LAN Interface.

The CMP works in any of three network management modes:

- SNMP Provisioning Mode/SNMPv3 Coexistence Management Mode
- DHCP Provisioning Mode/NmAccess Table Management Mode
- DHCP Provisioning Mode/SNMPv3 Coexistence Management Mode

SNMP Provisioning Mode/SNMP Coexistence Management Mode

As described in Section 5.5, when in SNMP Provisioning Mode, the PS defaults to operating in SNMPv3 Coexistence Mode with SNMPv1 and SNMPv2 not enabled, and uses Kerberos to distribute keying material. User-based Security Model (USM) [RFC 3414] and View-based Access Control Model (VACM) [RFC 3415] are supported to allow the cable operator to implement management policy for access to specified MIBs.

DHCP Provisioning Mode/NmAccessTable Management Mode

As described in Section 5.5, when in DHCP Provisioning Mode, the PS defaults to operate in NmAccess Table mode. In NmAccessTable mode, management access is controlled by the NmAccessTable of the DOCSIS Device MIB [RFC 2669] and the SNMPv1/v2c protocols are supported.

DHCP Provisioning Mode/SNMPv3 Coexistence Management Mode

When the PS is operating in DHCP Provisioning Mode, the cable operator can populate the Coexistence Table via

SNMP set-request messages or via PS configuration file, thereby configuring the PS to operate in SNMPv3 Coexistence Management Mode. For a PS configured to operate in SNMPv3 Coexistence Mode, management access is controlled as described in [RFC 2576], the SNMPv1/v2c/v3 protocols are supported, USM and VACM are supported, and SNMPv3 keying material is distributed using [RFC 2786] and TLVs in the PS Configuration File.

Table 6-4 contains definitions for terms that are specific to the CMP.

Table 6-4 Definition of Terms

Management-control	**Read or write access to a set of parameters that control or monitor the behavior of the PS.**
PS Database	A set of parameters that controls or monitors the behavior of the PS element readable by the WAN management system. It can be thought of as a repository of information describing the current state of the PS.
User	As defined in SNMP (section 2.1 of [RFC 3414]), a User has a name associated with it, associated security definitions and access to a View.
View	A View is a set of MIB objects and the access rights to those objects. Each View has a name and it is associated with a User (section 2.4 of [RFC 3415]).
Ultimate Authorization	The single authority that establishes, modifies, or deletes User IDs, authentication keys, encryption keys, and access rights to the PS Database. This User is entrusted with all security management operations.
Maintenance User	A User that typically performs only read-only operations on the PS database. This is typically used for performance monitoring and accounting.
Administrator User	A User that typically performs both read and write operations on the PS database. These operations are used for Configuration and Fault Management.

Examples of the types of information that can be read or manipulated via IPCable2Home Management-control include the firewall policy settings, NMS-configured NAT mappings, remote diagnostic tool initiation and results access, PS status, discovered device and applications information, and LAN address range configuration. As will be illustrated later, the various management messaging interfaces may have access rights to different sets of parameters. A compliant PS supports access to the PS database through the MIB hierarchy from both the WAN and LAN using SNMP. Compliant IPCable2Home Host devices can also exchange messages with the Residential Gateway using XML-formatted data transported, via HTTP. Figure 6-2 indicates management messaging interfaces:

- NMS - CMP: management message exchange between the cable network NMS and the CMP.
- CMP - IPCable2Home Host/LAN-Trans: message exchange between the CMP and IPCable2Home Hosts in the LAN-Trans realm.
- CMP - IPCable2Home Host/LAN-Pass: message exchange between the CMP and IPCable2Home Hosts in the LAN-Pass realm.
- NMS - LAN IP Device: management message exchange between the cable network NMS and LAN IP Devices in the LAN-Pass realm. This management messaging is outside the scope of this Recommendation.

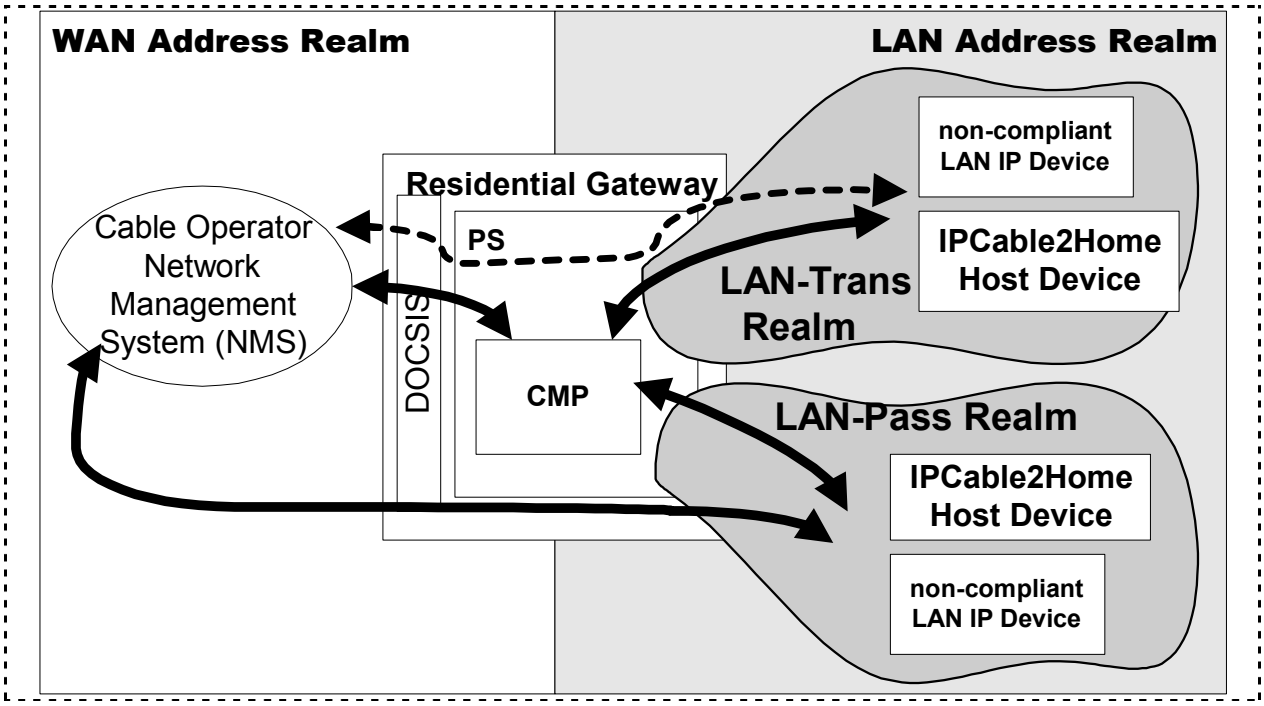

Figure 6-2 CableHome Management Message Interfaces

The CMP is primarily a WAN (NMS) accessed and WAN controlled entity, but also supports access from the PS LAN interface (Server Router address - usually the default gateway for LAN IP Devices in the LAN-Trans realm). Additionally the CMP may be called upon to inform the cable network NMS of events or transfer system log files as required. An example of a CMP implementation is illustrated in Figure 6-3, to convey concepts for CMP functionality.

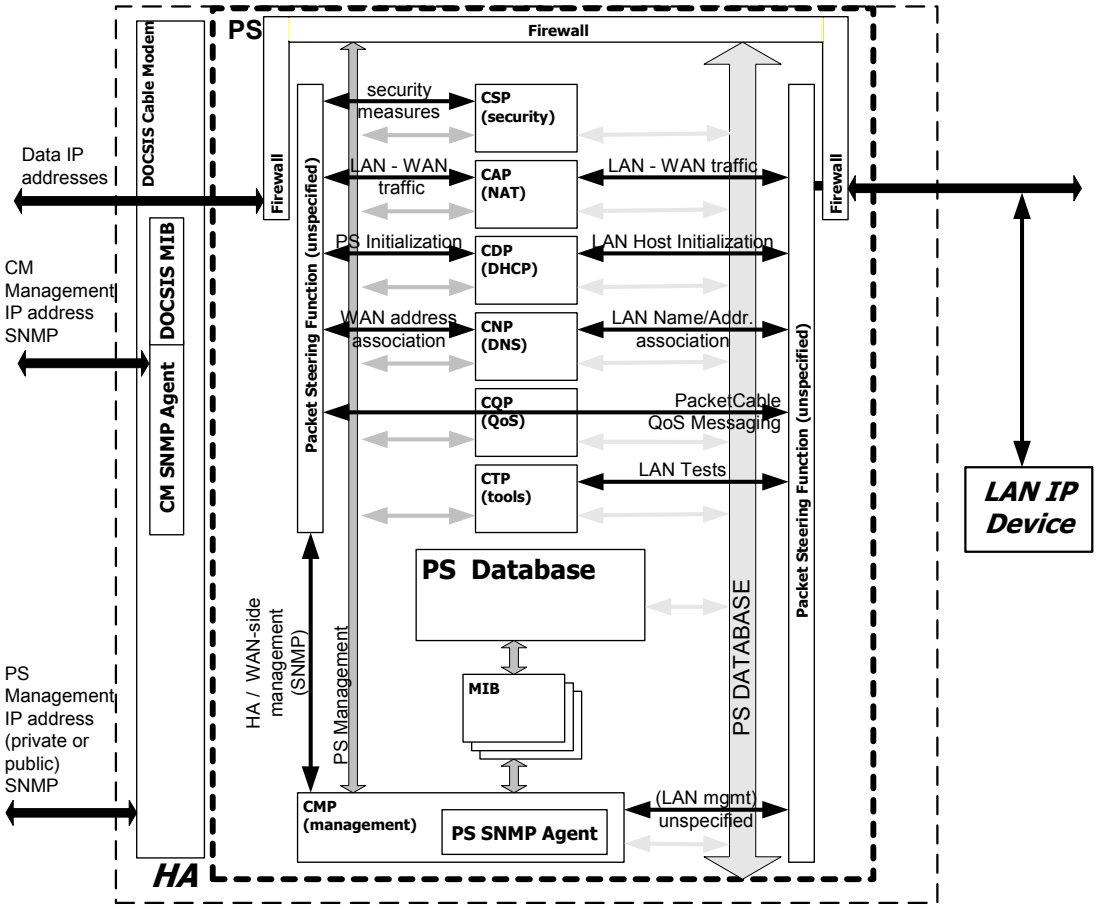

Figure 6-3 PS Block Diagram

The NMS management tools use SNMP to access and manage objects in the PS. If the PS is operating in SNMPv3 Coexistence Mode, SNMPv3 provides NMS operator User authentication to the PS, view-based access to the management information base (MIB) objects in the PS, and encryption of management messages if requested.

The CMP SNMP agent function has the task of mapping the Object ID (OID) and the instance of the OID for all the leaves within the functional blocks in the PS, such as the CAP or local storage such as the PS Database.

A cable data network NMS operator may access or "manage" CableHome Hosts in one of two ways. The cable operator can directly access CableHome Hosts using pass-through addressing between the cable network and the LAN device element (BP) to be managed. The cable operator can also access BP Device profile attributes through the PSDev MIB in the PS and a list of BP applications and their priorities through the QoS MIB in the PS. The cable operator accesses these MIBs via SNMP set-request or SNMP get-request messages issued to the PS WAN-Man IP address and the PS, acting as a management proxy, accesses a BP using SOAP/HTTP. The cable operator can provision QoS Policy, in the form of QoS priorities for CableHome Host applications, in the PS via SNMP.

6.3.3.1.4 SNMP Agent Function Requirements

The PS MUST implement an SNMP agent compliant with IETF RFCs as indicated in Section 6.3.3.1.4.1, "SNMP Protocol Requirements," on page 48.

The SNMP agent in the PS MUST only receive and process SNMP messages addressed to its WAN-Man IP address or addressed to its LAN Server Router address (cabhCdpServerRouter), when operating in DHCP Provisioning Mode or SNMP Provisioning Mode (cabhPsDevProvMode = dhcpmode(1) or snmpmode(2)).

The SNMP agent in the PS MUST receive and process all SNMP messages addressed to the PS LAN Server Router address (cabhCdpServerRouter) if the PS has never been provisioned.

The PS MUST ignore SNMP messages received through any LAN interface addressed to the PS WAN-Man IP address.

In the case of a PS co-resident with an embedded cable modem, i.e., an embedded PS, the PS and cable modem MUST respond to different and independent management IP addresses.

The PS MUST implement ICMP Echo and Echo Reply Message types (Type 8 and Type 0) and ICMP Timestamp and Timestamp Reply Message types (Type 13 and Type 14) as described in [RFC 792], and reply appropriately to Ping requests received on any interface.

If the PS is operating in DHCP Provisioning Mode (indicated by a value of '1' in cabhPsDevProvMode) the PS MUST default to using SNMPv1/v2c for management messaging with the NMS and follow rules for NmAccess mode and Coexistence Mode, described in Section 6.3.3.1.4.2.1, "Network Management Modes for a PS Operating in DHCP Provisioning Mode," on page 50.

If the PS is operating in SNMP Provisioning Mode (indicated by a value of '2' in MIB object cabhPsDevProvMode), the PS MUST use SNMPv3 for management messaging with the NMS, following rules described in Section 6.3.3.1.4.3, "Network Management Mode for a PS Operating in SNMP Provisioning Mode," on page 52.

When the PS is operating in SNMP Coexistence Mode, the default Ultimate Authorization setting MUST be WAN Administrator (CHAdministrator).

The PS MUST include - in the following specified order - the hardware version, vendor name, boot ROM image version, software version, and model number in the sysDescr object (from [RFC 3418]). The format of the specific information contained in the sysDescr MUST be as shown in Table 6-5:

Table 6-5 Format of sysDescr Fields

To Report	Format of Each Field
Hardware Version	HW_REV: <hardware version>
Vendor Name	VENDOR: <vendor name>
Boot ROM	BOOTR: <boot ROM version>
Software Version	SW_REV: <software version>
Model Number	MODEL: <model number>

The sysDescr MUST be composed of a list of five Type/Value pairs enclosed in double angle brackets. The separation between the Type and Value is ": " - a colon and a blank space. For instance, a sysDescr of a PS of vendor X, hardware version 5.2, Boot ROM version 1.4, software version 2.2, and model number X would appear as follows:

anytext<<HW_REV: 5.2, VENDOR: X; BOOTR: 1.4; SW_REV: 2.2; MODEL: X>>any text

The PS MUST report in the sysDescr at least all of the information necessary to determine what software and firewall policy versions the PS is capable of loading. If any fields of the sysDescr object are not applicable, the PS MUST report "NONE" as the value. For example, a PS with no BOOTR will report "BOOTR: NONE".

The value of the docsDevSwCurrentVers MIB object MUST contain the same software version information as that contained in the software version information included in the sysDescr object.

When a PS and a CM are embedded in the same device, the sysDescr and docsDevSwCurrentVers objects of the PS MUST report the same values as those of the CM.

The sysObjectID object of the MIB-2 System group [RFC 3418] MUST be implemented and MUST be persistent across device reset and power cycles.

The sysUpTime object of the MIB-2 System group [RFC 3418] MUST be implemented. SysUpTime is the amount of time that has elapsed since the system reset.

The sysContact object of the MIB-2 System group [RFC 3418] MUST be implemented and MUST be persistent across device reset and power cycles. SysContact returns the name of the user or system administrator if known.

The sysLocation object of the MIB-2 System group [RFC 3418] MUST be implemented and MUST be persistent across device reset and power cycles.

The sysServices object of the MIB-2 System group [RFC 3418] MUST be implemented and MUST be persistent across device reset and power cycles.

The sysName object of the MIB-2 System group [RFC 3418] MUST be implemented and MUST be persistent across device reset and power cycles. Querying sysName returns the system name.

The Interfaces Group MIB [RFC 2863] MUST be implemented in accordance with Annex A and requirements in Section 6.3.3.1.4.8

The MIB-2 SNMP group [RFC 3418] MUST be implemented.

The snmpSetSerialNo object of the snmpSet group [RFC 3418] MUST be implemented. SnmpSetSerialNo is an advisory lock used to allow several cooperating SNMPv2 entities, all acting in a manager role, to coordinate their use of the SNMPv2 set operation.

The PS MUST count LAN-to-WAN and WAN-to-LAN octets as defined by cabhPsDevLanIpTrafficTable [Annex E.4], according to the value of cabhPsDevLanIpTrafficEnabled [Annex E.4].

When the PS element MIB objects are set to their factory defaults using the cabhCapSetToFactory, cabhCdpSetToFactory, cabhCtpSetToFactory, or cabhPsDevSetToFactory MIB objects the corresponding PS functionality MUST use these factory default settings for operation without having to re-provision the PS element.

6.3.3.1.4.1 SNMP Protocol Requirements

The PS MUST adhere to or implement, as appropriate, the following IETF RFCs:

- "A Simple Network Management Protocol" [RFC 1157]
  NOTE: This RFC has been declared "historic" by [RFC 3410]. The PS is required to support SNMPv1.
- "Introduction to Community-based SNMPv2" [RFC 1901]
  NOTE: This RFC has been declared "historic" by [RFC 3410]. The PS is required to support SNMPv2c.
- "Introduction and Applicability Statements for Internet Standard Management Framework" [RFC 3410]
- "An Architecture for Describing Simple Network Management Protocol Management Frameworks" [RFC 3411]
- "Message Processing and Dispatching for SNMP" [RFC 3412]
- "Simple Network Management Applications" [RFC 3413]
- "User-based Security Model (USM) for the Simple Network Management Protocol" [RFC 3414]
- "View-based Access Control Model (VACM) for the Simple Network Management Protocol" [RFC 3415]
- "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)" [RFC 3416]
- "Transport Mappings for the Simple Network Management Protocol" [RFC 3417]
- "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)" [RFC 3418]
- "Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework" [RFC 2576]

In support of SMIv2, the PS MUST implement the following IETF RFCs:

- "Structure of Management Information Version 2 (SMIv2)" [RFC 2578]
- "Textual Conventions for SMIv2" [RFC 2579]
- "Conformance Statements for SMIv2" [RFC 2580]

6.3.3.1.4.2 Network Management Mode Requirements

Section 5.5 introduced two provisioning modes, (DHCP Provisioning Mode and SNMP Provisioning Mode) and two network management modes (NmAccessTable Mode and SNMPv3 Coexistence Mode) that the PS is required to support. Section 7.3.3.1 and Section 7.3.3.2 provide additional detail about PS operation in each of the two provisioning modes, in addition to Dormant CableHome Mode of operation.

This section describes rules for the network management modes the PS is required to support. Section 6.3.3.1.4.2.1 and its sub-sections describe network management modes for a PS operating in DHCP Provisioning Mode. Section 6.3.3.1.4.3 and its sub-sections describe network management modes for a PS operating in SNMP Provisioning Mode.

The PS can operate in SNMPv3 Coexistence network management mode, regardless of whether it is configured to operate in DHCP Provisioning Mode or SNMP Provisioning Mode. It defaults to operation in SNMPv3 Coexistence mode when operating in SNMP Provisioning Mode. When operating in DHCP Provisioning Mode the PS defaults to operating in NmAccessTable network management mode, but can be configured to operate in SNMPv3 Coexistence Mode.

Control of access to the MIBs implemented by the PS depends upon the network management mode in which the PS is configured to operate. When the PS is configured to operate in NmAccessTable network management mode, MIB access is controlled by writing to the docsDevNmAccessTable [RFC 2669]. When operating in SNMPv3 Coexistence Mode, access to the MIBs is controlled by the SNMPv3 tables ([RFC 2576], [RFC 3413], [RFC 3414], and [RFC 3415]). The SNMPv3 tables can be configured by the NMS through SNMP Set commands, or via the PS Configuration File. Section 6.3.3.1.4.6 Mapping TLV Fields Into Created SNMPv3 Table Rows describes how PS Configuration File configuration parameters are mapped into these SNMPv3 tables.

6.3.3.1.4.2.1 Network Management Modes for a PS Operating in DHCP Provisioning Mode

The PS MUST support SNMPv1, SNMPv2c, and SNMPv3 and SNMP Coexistence as described by [RFC 3411] through [RFC 3415] and [RFC 2576]. The PS MUST also support NmAccessTable mode as defined by [RFC 2669]. Support for the network management modes for a PS operating in DHCP Provisioning Mode is subject to the guidelines described in Section 6.3.3.1.4.2.2, Section 6.3.3.1.4.3, and Section 6.3.3.1.4.4.

6.3.3.1.4.2.2 Basic Operation for a PS Operating in DHCP Provisioning Mode

Initial operation of the PS configured for DHCP Provisioning Mode can be thought of as having three steps: (1) behavior of the PS after it has been configured for DHCP Provisioning Mode, but before its network management mode has been configured via the PS Configuration File; (2) determination of the network management mode, and; (3) behavior of the PS after its network management mode has been configured. Rules of operation for each of these steps follow:

1. Once the PS has been configured to operate in DHCP Provisioning Mode (indicated by a cabhPsDevProvMode value of '1' (DHCPmode)), but before it has been configured for a network management mode, the PS MUST operate as follows:
    - All SNMP packets are dropped.
    - None of the SNMPv3 MIBs (Community MIB, TARGET-MIB, VACM-MIB, USM-MIB, NOTIFICATION-MIB) are accessible to the SNMP manager in the NMS.
    - None of the elements in the SNMP-USM-DH-OBJECTS-MIB is accessible to the SNMP manager in the NMS.
    - The PS Configuration File specified in the DHCP OFFER is downloaded and processed.
    - Successful processing of all MIB elements in the PS Configuration File MUST be completed before beginning the calculation of the public values in the USMDHKickstart Table.
2. If a PS is operating in DHCP Provisioning Mode, the content of the PS Configuration File determines the network management mode, as described below:
    - The PS is in SNMPv1/v2c docsDevNmAccess mode if the PS Configuration File contains ONLY docsDevNmAccess Table setting for SNMP access control.
    - If the PS Configuration File does not contain SNMP access control items (docsDevNmAccessTable or snmpCommunityTable or TLV 34.1/34.2 or TLV38), then the PS is in NmAccess mode.
    - If the PS Configuration File contains snmpCommunityTable setting and/or TLV type 34.1 and 34.2 and/or TLV type 38, then the PS is in SNMP Coexistence Mode. In this case, any entries made to the docsDevNmAccessTable are ignored.

3. After completion of the provisioning process described in Section 13.2 (indicated by the value 'pass' (1) in cabhPsDevProvState), the PS operates in one of two network management modes. The network management mode is determined by the contents of the PS Configuration File as described above. Rules for PS operation for each of the two network management modes follow:

   NmAccess Mode using SNMPv1/v2c

   - The PS MUST process SNMPv1/v2c packets and drop SNMPv3 packets.
   - docsDevNmAccessTable controls access and trap destinations as described in [RFC 2669]. The PS MUST enforce the management access policy, as defined by the NmAccess Table, for any access to the CableHome-specified MIB objects, regardless of the interface or access protocol used.
   - None of the SNMPv3 MIBs (Community MIB, TARGET-MIB, VACM-MIB, USM-MIB, NOTIFICATION-MIB) is accessible.

   When the PS is operating in SNMP v1/v2c NmAccess mode it MUST support the capability of sending traps as specified by the following MIB object (proposed MIB extension to the docsDevNmAccess table):

   DocsDevNmAccessTrapVersion OBJECT-TYPE

   SYNTAX INTEGER {

   DisableSNMPv2trap(1),

   EnableSNMPv2trap(2),

   }

   MAX-ACCESS read-create

   STATUS current

   DESCRIPTION

   "Specifies the TRAP version that is sent to this NMS. Setting this object to disableSNMPv2trap

   (1) causes the trap in SNMPv1 format to be sent to particular NMS. Setting this object to

   EnableSNMPv2trap(2) causes the trap in SNMPv2 format be sent to particular NMS"

   DEFVAL { DisableSNMPv2trap }

   ::={docsDevNmAcessEntry 8}

   Coexistence Mode using SNMPv1/v2c/v3

   When in SNMPv3 Coexistence Mode, the PS MUST support the "SNMPv3 Initialization" and "DH Key Changes" requirements specified in Section 11.4.4.1.3 and Section 11.4.4.1.4. These requirements include calculation of USM Diffie-Hellman Kickstart Table public parameters. The following rules for PS operation apply during and after calculation of the public parameters (values) as indicated:

   During calculation of USMDHKickstartTable public values:

   - The PS MUST NOT allow any SNMP access from the WAN.
   - The PS MAY continue to allow access from the LAN with the limited access as configured by USM MIB, community MIB and VACM-MIB.

   After calculation of USMDHKickstartTable public values:

- The PS MUST send the cold start or warm start trap to indicate that the PS is now fully SNMPv3 manageable.
- SNMPv1/v2c/v3 Packets are processed as described by [RFC 3411], [RFC 3412], [RFC 3413], [RFC 3414], [RFC 3415], and [RFC 2576].
- docsDevNmAccessTable is not accessible.
- Access control and trap destinations are determined by the snmpCommunityTable, Notification MIB, Target MIB, VACM-MIB, and USM-MIB. The PS MUST enforce the management access policy, as defined by the VACM View configured by the cable operator, for any access to the CableHome-specified MIB objects, regardless of the interface or access protocol used.
- Community MIB controls the translation of SNMPv1/v2c packet community string into security name which select entries in the USM MIB. Access control is provided by the VACM MIB.
- USM MIB and VACM MIB controls SNMPv3 packets.
- Trap destinations are specified in the Target MIB and Notification MIB.

In case of failure to complete SNMPv3 initialization for a User (i.e., NMS cannot access the PS via SNMPv3 PDU), the USM User Table for that User MUST be deleted, the PS is in Coexistence Mode, and the PS will allow SNMPv1/v2c access if and only if the community MIB entries (and related entries) are configured.

6.3.3.1.4.3 Network Management Mode for a PS Operating in SNMP Provisioning Mode

If the PS is operating in SNMP Provisioning Mode following DHCP ACK (as indicated by a value '2' (SNMPmode) for cabhPsDevProvMode), it operates in SNMPv3 Coexistence Mode using SNMPv3 by default for exchanging management messages with the NMS, and uses Kerberos for exchanging key material with the KDC, following rules described in this section. Just as when the PS is operating in DHCP Provisioning Mode and has been configured for SNMPv3 Coexistence network management mode, when the PS is operating in SNMP Provisioning Mode and SNMPv3 Coexistence network management mode it is required to ignore attempts to configure the docsDevNmAccessTable.

6.3.3.1.4.4 Management Views

The management controls defined for CableHome are in the CMP function of the PS. Settings, based on management mode, define the access rights that are granted to a User for access to the Portal Services database, through CableHome-specified MIBs, via SNMP from the PS WAN-Man or LAN Server Router interfaces. A single User is defined by the This Recommendation.

The concept of Management Views was introduced with SNMPv3, and is defined in [RFC 3410] through [RFC 3415] and [RFC 2576]. It is a method for specifying what user(s) is/are allowed to access which MIB object(s).

Figure 6-4 illustrates some possible management Views for the PS. A WAN Administrator View (CHAdministrator view) and a WAN Administrator User (CHAdministrator user) are defined by This Recommendation. Other Views and Users, such as the WAN Maintenance View, the LAN Administrator View, or the LAN User View can be established by the Ultimate Authorization (CHAdministrator), following rules defined in [RFC 3414] and [RFC 3415].

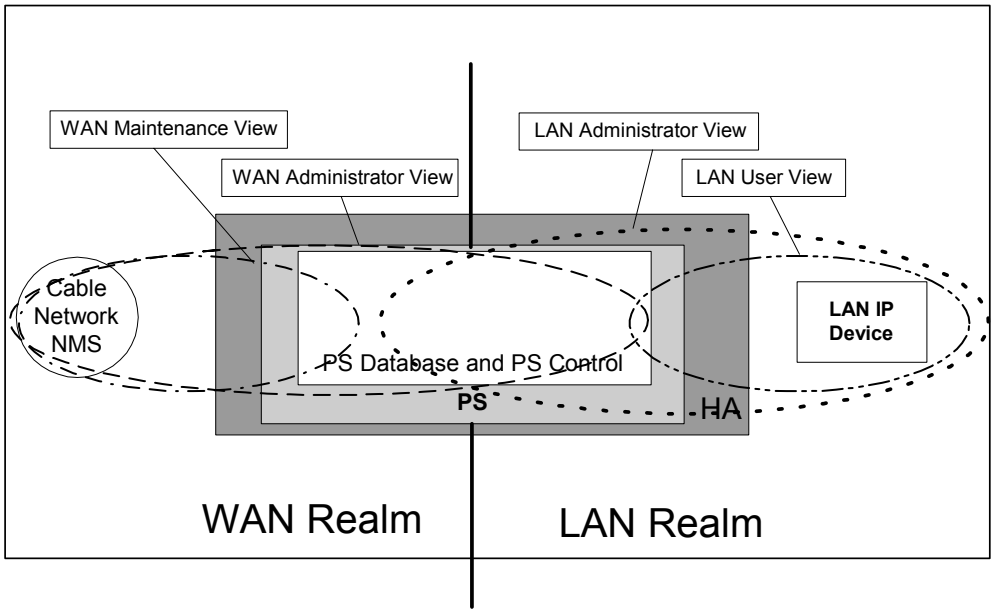

Figure 6-4 Management Views

Managed parameters defined by CableHome are stored in the PS Database. As shown in Figure 6-4, there is a concept of Access Views into the PS Database and PS Control, which allows simultaneous management from both the LAN and WAN by defining Management Views into the PS Database and PS Control. The Views are a mechanism to provide privacy and security, and the policy can be set separately by the CHAdministrator User.

The Ultimate Authorization (CHAdministrator User) has its own User ID and keys, and has the following responsibilities:

- Responsible for setting up all access Views on both the LAN and WAN management interface.
- Responsible for creating and managing all User profiles including user IDs, Keys, and PS database access privileges.
- Responsible for setting policy for both LAN and WAN side access.

Descriptions for how View-based Access Control Model and User-based Security Model work are provided in [RFC 3414] and [RFC 3415].

The CHAdministrator View provides full read and write access to all MIBs specified by CableHome.

Management View requirements are specified in Section 6.3.3.1.4.5 of this specification.

6.3.3.1.4.4.1 WAN-Access Control

SNMP Access Control, per [RFC 3415], will be used to control access to CableHome-specified MIB objects, regardless of the interface through which the request arrives. The View-based Access Control Model (VACM) [RFC 3415] defines a set of services that can be used for checking access rights. VACM Groups define the rights to access the CMP.

As defined in [RFC 3415] section 2.4, a "MIB View" is a specific set of managed object types that can be defined, and this concept is used in CableHome to support WAN Management of the PS. The CHAdministrator User access and View are specified in Section 11.4.4.1.3 and Section 6.3.3.1.4.5. An example sequence of PS Database access from the WAN interface is provided in Section 12.3.1.

6.3.3.1.4.4.2 Security

Security of management messages is provided by SNMPv3. Refer to Section 11 for a detailed description of how

SNMPv3 is used. The CMP may use SNMP v3 to counter threats identified in Annex C.

To protect against replay attacks, a time of day clock is utilized to provide timestamps for messaging. Management messaging security requirements are specified in Section 11.4.

6.3.3.1.4.5 View-based Access Control Model (VACM) Requirements

To provide controlled access to management information and the creation of distinct management realms for a PS operating in SNMP v3 Coexistence Mode, View-based Access Control Model (VACM) MUST be employed as defined by [RFC 3415].

The WAN Administrator View MUST be implemented in a compliant Portal Services element. Default Views other than the WAN Administrator View MUST NOT be available on the PS. Other Views MAY be created by the Ultimate Authorization through the cable network NMS by configuring the VACM MIB.

The User specification for the WAN Administrator View MUST be implemented as follows:

vacmSecurityModel	3 (USM)
vacmSecurityName	'CHAdministrator'
vacmGroupName	'CHAdministrator'
vacmSecurityToGroupStorageType	permanent
vacmSecurityToGroupStatus	active

The Group specification for the CHAdministrator View MUST be implemented as follows:

CHAdministrator Group	
vacmGroupName	'CHAdministrator'
vacmAccessContextPrefix	''
vacmAccessSecurityModel	3 (USM)
vacmAccessSecurityLevel	AuthPriv
vacmAccessContextMatch	exact
vacmAccessReadViewName	'CHAdministratorView'
vacmAccessWriteViewName	'CHAdministratorView'
vacmAccessNotifyViewName	'CHAdministratorView'
vacmAccessStorageType	permanent
vacmAccessStatus	active

The VACM View for the CHAdministrator view MUST be implemented as follows:

CHAdministratorView subtree 1.3.6.1 (Entire MIB)

6.3.3.1.4.6 Mapping TLV Fields Into Created SNMPv3 Table Rows

This section details how the *SNMP Notification Receiver* Configuration File Element (TLV Type 38) is mapped into SNMPv3 functional tables. Refer to Section 7.4.4.1.9 SNMP Notification Receiver for a description of configuration parameter TLV Type 38. Details of how the encryption keys are exchanged for SNMP v3 operation

are provided in Section 11.4.4.2.2.

Upon receiving one Type 38 configuration file element, the PS MUST make MIB table entries following the procedure described in Table 6-12 snmpNotifyTable through Table 6-15 vacmSecurityToGroupTable, using values passed in the TLV as described below. The MIB tables the PS is required to populate when it receives a Type 38 configuration file element are listed below for convenience:

- snmpNotifyTable
- snmpTargetAddrTable
- snmpTargetAddrExtTable
- snmpTargetParamsTable
- snmpNotifyFilterProfileTable
- snmpNotifyFilterTable
- snmpCommunityTable
- usmUserTable
- vacmSecurityToGroupTable
- vacmAccessTable
- vacmViewTreeFamilyTable

A PS configuration file is allowed to contain TLV MIB elements (Type 28) that make entries to any of the 11 tables listed above.

The tables in this section show how the fields from the PS Configuration file TLV element (the tags in angle brackets <>) are placed into the SNMP V3 tables.

The correspondence between TLV fields and table tags <TAG> is shown below:

PS<IP Address> TLV 38.1

<Port> - TLV 38.2

<Trap type> TLV 38.3

<Timeout> TLV 38.4

<Retries> TLV 38.5

<Filter OID> TLV 38.6

<Security Name> TLV 38.7

These tables are shown in the order that the agent will search down through them when a notification is generated in order to determine who to send the notification to and how to fill out the contents of the notification packet.

snmpNotifyTable
Create two rows with fixed values, if one or more TLV elements are present.

Table 6-6 snmpNotifyTable

snmpNotifyTable [RFC 3413] SNMP-NOTIFICATION-MIB	First Row	Second Row
Column Name (* = Part of Index)	Column Value	Column Value
* snmpNotifyName	"@PSconfig_inform"	"@PSconfig_trap"
snmpNotifyTag	"@PSconfig_inform"	"@PSconfig_trap"
snmpNotifyType	inform(2)	trap(1)
snmpNotifyStorageType	volatile	volatile
snmpNotifyRowStatus	Active(1)	Active(1)

snmpTargetAddrTable

Create one row for each TLV element in the PS configuration file.

Table 6-7 snmpTargetAddrTable

snmpTargetAddrTable [RFC 3413] SNMP-TARGET-MIB	New Row
Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@PSconfig_n", where n ranges from 0 to m-1, and m is the number of notification receiver TLV elements in the PS configuration file
snmpTargetAddrTDomain	snmpUDPDomain – snmpDomains
snmpTargetAddrTAddress (IP Address and UDP Port of the Notification Receiver)	OCTET STRING (6) Octets 1 – 4: <IP Address> Octets 5 – 6: <Port>
snmpTargetAddrTimeout	<Timeout> from the TLV
snmpTargetAddrRetryCount	<Retries> from the TLV
snmpTargetAddrTagList	If <Trap type> == 1,2, or 4 "@PSconfig_trap" Else If <Trap type> = 3 or 5 "@PSconfig_inform"
snmpTargetAddrParams	"@PSconfig_n" (same as snmpTargetAddrName value)
snmpTargetAddrStorageType	volatile
snmpTargetAddrRowStatus	active(1)

snmpTargetAddrExtTable

Create one row for each TLV element in the PS configuration file.

Table 6-8 snmpTargetAddrExtTable

snmpTargetAddrExtTable [RFC 2576] SNMP-COMMUNITY MIB	New Row
Column Name (* = part of index)	Column Value
* snmpTargetAddrName	"@PSconfig_n", where n ranges from 0 to m – 1, and m is the number of notification receiver TLV elements in the PS configuration file
snmpTargetAddrMask	<zero length octet string>
snmpTargetAddrMMS	0

snmpTargetParamsTable

Create 1 row for each TLV element in the config file. If <Trap type> is 1, 2, or 3, or if the <Security Name> Field is zero-length, create the table as follows:

Table 6-9 snmpTargetParamsTable for <Trap Type> 1, 2, or 3

snmpTargetParamsTable [RFC 3413] SNMP-TARGET-MIB	New Row
Column Name (* = part of index)	Column Value
* snmpTargetParamsName	"@PSconfig_n", where n ranges from 0 to m – 1, and m is the number of notification receiver TLV elements in the PS configuration file
snmpTargetParamsMPModel SYNTAX: SnmpMessageProcessingModel	If <Trap type> = 1 SNMPv1(0) Else if <Trap type> = 2 or 3 SNMPv2c(1) Else if <Trap type> = 4 or 5 SNMPv3(3)
snmpTargetParamsSecurityModel SYNTAX: SnmpSecurityModel	If <Trap type> = 1 SNMPv1(1) Else if <Trap type> = 2 or 3 SNMPv2c(2) Else if <Trap type> = 4 or 5 USM(3) NOTE: The mapping of SNMP protocol types to value here are different from snmpTargetParamsMPModel
snmpTargetParamsSecurityName	"@PSconfig"
snmpTargetParamsSecurityLevel	noAuthNoPriv
snmpTargetParamsStorageType	volatile
snmpTargetParamsRowStatus	active(1)

If <Trap type> is 4 or 5, and the <Security Name field is non-zero length, create the table as follows:

Table 6-10 snmpTargetParamsTable for <Trap Type> 4 or 5

snmpTargetParamsTable [RFC 3413] SNMP-TARGET-MIB	New Row
Column Name (* = part of index)	Column Value
* snmpTargetParamsName	"@PSconfig_n", where n ranges from 0 to m – 1, and m is the number of notification receiver TLV elements in the PS configuration file
snmpTargetParamsMPModel SYNTAX: SnmpMessageProcessingModel	If <Trap type> = 1 SNMPv1(0) Else if <Trap type> = 2 or 3 SNMPv2c(1) Else if <Trap type> = 4 or 5 SNMPv3(3)

snmpTargetParamsSecurityModel SYNTAX: SnmpSecurityModel	If <Trap type> = 1 SNMPv1(1) Else if <Trap type> = 2 or 3 SNMPv2c(2) Else if <Trap type> = 4 or 5 USM(3) NOTE: The mapping of SNMP protocol types to value here are different from snmpTargetParamsMPModel
snmpTargetParamsSecurityName	<Security Name>
snmpTargetParamsSecurityLevel	The security level of <Security Name>
snmpTargetParamsStorageType	volatile
snmpTargetParamsRowStatus	active(1)

snmpNotifyFilterProfileTable

Create one row for each TLV that has a non-zero <Filter Length>.

Table 6-11 snmpNotifyFilterProfileTable

snmpNotifyFilterProfileTable [RFC 3413] SNMP-NOTIFICATION-MIB	New Row
Column Name (* = Part of Index)	Column Value
*snmpTargetParamsName	"@PSconfig_n", where n ranges from 0 to m-1 and m is the number of notification receiver TLV elements in the PS configuration file.
snmpNotifyFilterProfileName	"@PSconfig_n", where n ranges from 0 to m-1 and m is the number of notification receiver TLV elements in the PS configuration file.
snmpNotifyFilterProfileStorType	volatile
snmpNotifyFilterProfileRowStatus	active(1)

snmpNotifyFilterTable

Create one row for each TLV that has a non-zero <Filter Length>.

Table 6-12 snmpNotifyFilterTable

snmpNotifyFilterTable [RFC 3413] SNMP-NOTIFICATION-MIB	New Row
Column Name (* = Part of Index)	Column Value
* snmpNotifyFilterProfileName	"@PSconfig_n" , where n ranges from 0 to m-1 and m is the number of notification receiver TLV elements in the PS configuration file.
* snmpNotifyFilterSubtree	<Filter OID> from the TLV
snmpNotifyFilterMask	<Zero Length Octet String>
snmpNotifyFilterType	included(1)
snmpNotifyFilterStorageType	volatile
snmpNotifyFilterRowStatus	active(1)

snmpCommunityTable

Create one row with fixed values if 1 or more TLV's are present. This causes SNMPV1 and V2c Notifications to contain the community string in snmpCommunityName.

Table 6-13 snmpCommunityTable

snmpCommunityTable [RFC 2576] SNMP-COMMUNITY-MIB	First Row
Column Name (* = Part of Index)	Column Value
* snmpCommunityIndex	“@PSconfig”
snmpCommunityName	“public”
snmpCommunitySecurityName	“@PSconfig”
snmpCommunityContextEngineID”	<The PS engineID>
snmpCommunityContextName	<Zero length octet string>
snmpCommunityTransportTag	<Zero length octet string>
snmpCommunityStorageType	volatile
snmpCommunityStatus	active(1)

usmUserTable

Create one row with fixed values, if one or more TLVs are present. Other rows are created, one each time the engine ID of a trap receiver is discovered. This specifies the user name on the remote notification receivers to send notifications to.

One row in the usmUserTable is created. Then when the engine ID of each notification receiver is discovered, the agent copies this row into a new row and replaces the 0x00 in the usmUserEngineID column with the newly discovered value.

Table 6-14 usmUserTable

usmUserTable [RFC 3414] SNMP-USER-BASED-SM-MIB	First Row
Column Name (* = Part of Index)	Column Value
* usmUserEngineID	0
* usmUserName	“@PSconfig” When other rows are created, this is replaced with the <Security Name> field from the TLVelement.
usmUserSecurityName	“@PSconfig” When other rows are created, this is replaced with the <Security Name> field from the TLVelement.
usmUserCloneFrom	<don’t care> - cannot clone this row
usmUserAuthProtocol	None. When other rows are created, this is replaced with None or MD5, depending upon the security level of the v3 User.
usmUserAuthKeyChange	<don’t care> - write only
usmUserOwnAuthKeyChange	<don’t care> - write only
usmUserPrivProtocol	None. When other rows are created, this is replaced with None or DES, depending on the security level of the v3 User.
usmUserPrivKeyChange	<don’t care> - write only
usmUserOwnPrivKeyChange	<don’t care> - write only
usmUserPublic	<zero length string>
usmUserStorageType	volatile
usmUserStatus	active(1)

vacmSecurityToGroupTable

Create three rows with fixed values, if one or more TLVs are present.

These are the three rows with fixed values, which are used for the TLV entries with <Trap Type> set to 1, 2, or 3 or with a zero length <Security Name>.

Table 6-15 vacmSecurityToGroupTable

vacmSecurityToGroupTable [RFC 3415] SNMP-VIEW-BASED-ACM-MIB	First Row	Second Row	Third Row
Column Name (* = Part of Index)	Column Value	Column Value	Column Value
* vacmSecurityModel	SNMPv1(1)	SNMPv2c(2)	USM(3)
* vacmSecurityName	"@PSconfig"	"@PSconfig"	"@PSconfig"
vacmGroupName	"@PSconfigv1"	"@PSconfigv2"	"@PSconfigUSM"
vacmSecurityToGroupStorageType	volatile	volatile	volatile
vacmSecurityToGroupStatus	active(1)	active(1)	active(1)

6.3.3.1.4.7 IPCable2Home MIB Requirements

The PS MUST implement each MIB object listed in Annex A. If the Persistent column for a MIB object listed in Annex A contains the value Yes, the PS MUST retain the value of the object across a PS power cycle or re-boot, making the same value available for access by an SNMP manager immediately after provisioning complete (cabhPsDevProvState = pass(1)), following a re-boot that was available for access by that SNMP manager immediately before re-boot.

Required MIB objects are from the following MIB documents:

- Interfaces Group MIB [RFC 2863]
- DOCSIS Cable Device MIB [RFC 2669]
- CableLabs Definition MIB [Annex E.6]
- CableHome PSDev MIB [Annex E.4]
- CableHome CAP MIB [Annex E.1]
- CableHome CDP MIB [Annex E.2]
- CableHome CTP MIB [Annex E.3]
- CableHome Security MIB [Annex E.5]
- CableHome QoS MIB [Annex E.7]
- [draft-ietf-ipcdn-bpiplus-mib-05]
- IP MIB (SNMPv2) [RFC 2011]
- UDP MIB (SNMPv2) [RFC 2013]
- Diffie-Hellman USM Key [RFC 2786]
- INET Address MIB [RFC 3291]
- DOCS IF MIB [RFC 2670]
- IANA ifType MIB [IANAType]

In a IPCable2Home Residential Gateway or any other device with an embedded PS and embedded cable modem, the cable modem management entity and PS management entity (CMP) MUST respond to different and independent management IP addresses. J.112 and this Recommendation specify some of the same MIB objects but if a J.112-compliant cable modem and a IPCable2Home-compliant PS Element are embedded in the same device, each is required to maintain its own, separate instance of specified MIB objects, accessible through different management IP addresses, with the exception of the SNMP group of MIB 2 and SNMPv2 MIB, which MAY be common to and shared between the cable modem and the Portal Services Element, and MAY be accessible through either the cable modem management IP address or the PS management IP address.

In a PS with an embedded cable modem, software download of the single image of the combined cable modem software and Portal Services software, is controlled by the cable modem. The following docsDevSoftware group objects [RFC 2669] MUST NOT be implemented for a PS with an embedded cable modem, i.e., these objects MUST only be accessible through the cable modem management IP address in a PS with an embedded CM:

- docsDevSwServer
- docsDevSwFilename
- docsDevSwAdminStatus
- docsDevSwOperStatus

The docsDevSoftware Group of objects MUST be implemented in a Standalone PS. Modification of the docsDevSoftware objects (as specified in Section 11.8.4) by the cable operator for the purpose of downloading the standalone PS software image MUST result in proper secure software download operation.

In a PS with an embedded cable modem, cable modem MIB objects MUST only be visible and accessible when the manager accesses them through the cable modem management IP address, and MUST NOT be visible or accessible via any PS management IP address, with the exception of the SNMP group of MIB 2 and the SNMPv2 MIB which are allowed to be shared between the CM and PS management entities.

In a PS with an embedded cable modem, IPCable2Home-specified MIB objects MUST only be visible and

accessible when the manager accesses them through the PS management IP address (PS WAN-Man IP address) or through the PS LAN Server Router IP address, and MUST NOT be visible or accessible via the cable modem management IP address, with the exception of the SNMP group of MIB 2 and the SNMPv2 MIB which are allowed to be shared between the CM and PS management entities.

The general MIB hierarchy is illustrated in Figure 6-5. Specific OIDs required for individual MIBs are listed in Annex A.

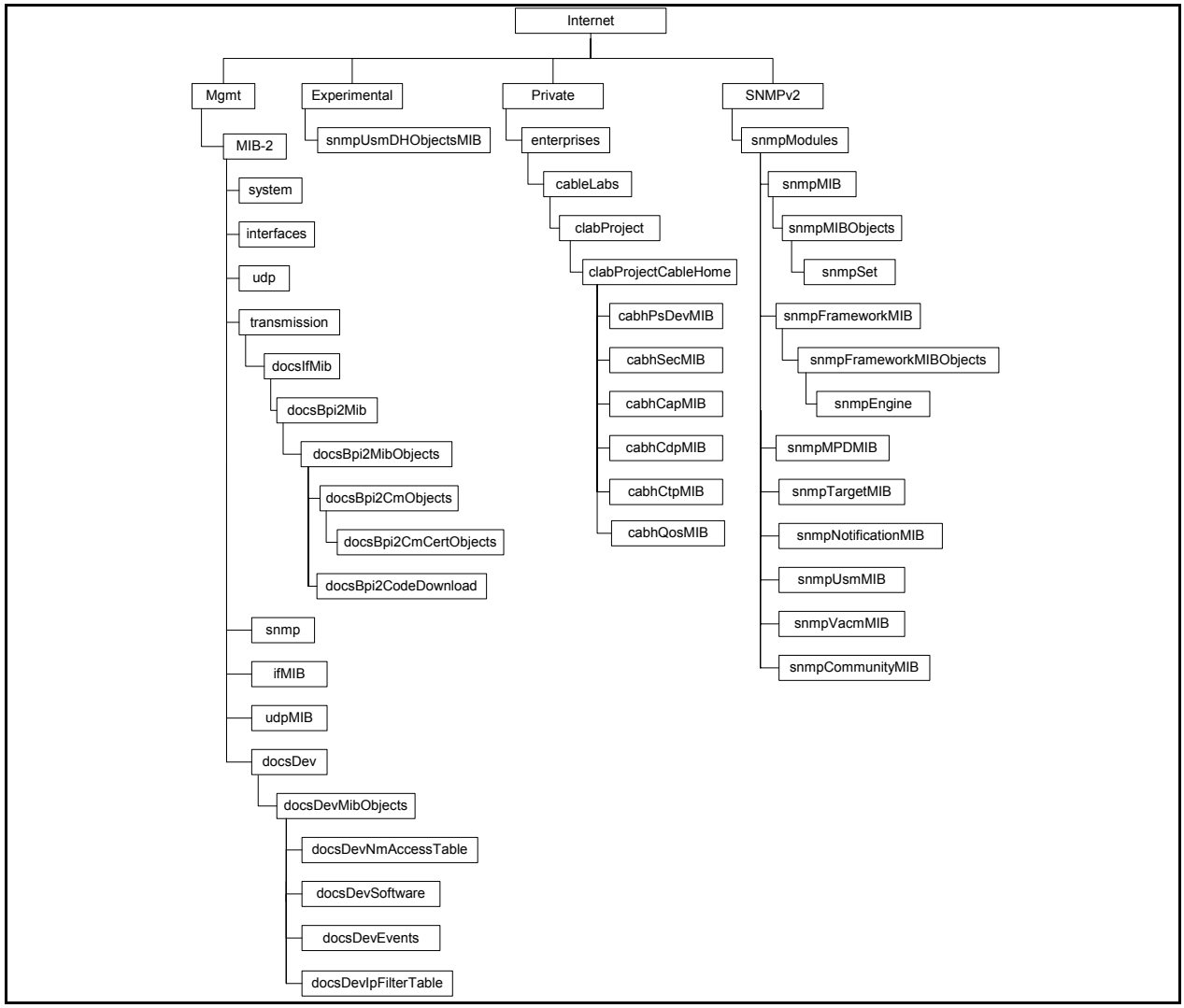

Figure 6-5 IPCable2Home MIB Hierarchy

6.3.3.1.4.8 Interfaces Group MIB

The Interfaces Group MIB [RFC 2863] provides a powerful tool to allow cable operators to understand the state of and see statistics for all of the physical interfaces on the Portal Service element. A *physical interface* is one for which a connector is exposed on the exterior of the device enclosure, and for which the object *ifConnectorPresent* is true. In order to enable the intelligent use of this MIB, an interface numbering scheme is essential. Therefore PS elements need to comply to the following requirements:

An instance of ifEntry MUST exist for the WAN-Data interface of the PS element, even if that interface is internal - as exists in the case of an Embedded PS utilizing an integrated chip design.

An instance of ifEntry MUST exist for each physical LAN interface of the PS element.

An instance of ifEntry MUST exist for an "Aggregated LAN Interfaces" interface, which is identified by the ifIndex value 255.

The PS ifTable interfaces MUST be numbered as shown in Table 6-16.

Table 6-16 Numbering Interfaces in the ifTable

Interface	Description
1	WAN-Man Interface
2	WAN-Data Interface
2+n	Each LAN Interface
255	Aggregated LAN Interface

If a given interface's ifAdminStatus = down, that interface MUST NOT accept or forward any traffic. The ifAdminStatus object corresponding to ifIndex value 255 MUST provide administrative control over all LAN interfaces and MUST be implemented as read-write.

The PS MUST assign the value other(1) to ifTable [RFC 2233] ifType entries corresponding to ifIndex 255. An embedded PS element MUST assign the value other(1) to ifTable ifType entries corresponding to ifIndex values 1 and 2. A standalone PS element MUST assign the appropriate IANAifType [IANAType] value to the ifTable ifType value corresponding ifIndex values 1 and 2.

The ifTable ifPhysAddress value corresponding to ifIndex 255 MUST be a zero length octet string.

The ifTable counters of WAN interfaces of ifIndex values 1 and 2 MUST be shared between the two interfaces. The ifTable counters for ifIndex value 255 MAY be implemented.

The Interface Stack (ifStack) group of [RFC 2233] MUST be implemented to identify relationships among the higher-layer "Aggregated LAN Interfaces" interface and the lower-layer LAN sub-interfaces. Figure 6-6 illustrates the use of the ifStack group for a PS with three LAN interfaces.

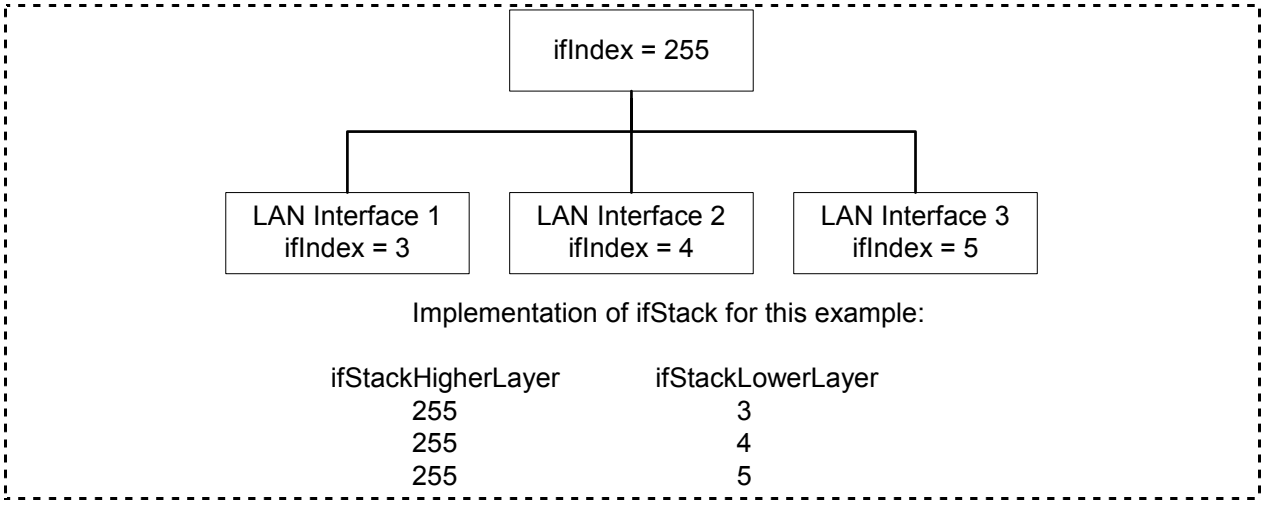

Figure 6-6 ifStack Implementation Example

6.3.3.1.4.9 ipNetToMediaTable Requirements

The ipNetToMediaTable [RFC 2011] maps IP addresses to physical addresses, and its use is straightforward if each IP address is associated to one physical interface and if each physical interface is associated to one physical address. The PS, however, implements different IP addresses that may apply to several physical interfaces, and associates the physical WAN interface to two hardware addresses. The PS MUST list in the ipNetToMediaTable each of the IP addresses that are part of its active configuration, creating one entry per distinct IP value and abiding by Table 6-17:

Table 6-17 PS ipNetToMediaTable

ipNetToMediaAddress	ipNetToMediaPhysAddress	ipNetToMediaIfIndex
WAN-Man IP Address	WAN-Man hardware address	1
WAN-Data IP Address	WAN-Data hardware address	2
DHCP server IP address	Zero length octet string	255
DNS server IP address	Zero length octet string	255
Server Router IP address	Zero length octet string	255

6.3.3.2 CMP Event Reporting Function

The CMP is required to support the handling and reporting of events generated by the PS, for the WAN Domain. Event messages defined by IPCable2Home for the PS element can be reported via SNMP Trap to the cable operator's notification receiver, via a System Log message sent to the cable operator's system log, or via a log local to the PS and accessible through specified MIB objects. Events defined for the PS are listed in Annex B Format and Content for Event, SYSLOG, and SNMP Trap. These are the same processes defined in DOCSIS specifications for event reporting in cable modems.

IPCable2Home Host devices are not required to support event messaging. Therefore, LAN Domain event messaging is not defined by this Recommendation.

Event Reporting for the WAN Domain

IPCable2Home uses the [RFC 2669] event reporting and control mechanisms for events generated in the PS (CMP). [RFC 2669] defines a standard format for reporting event information, regardless of the message type, including a local event log table in which certain entries will persist across reboot of the PS. Note that events may be generated by any part of a PS, but the CMP logs and/or reports the event either locally or to a Syslog or Trap server.

6.3.3.2.1 Event Reporting Function Goals

The goals of the CMP Event Reporting function are listed below:

- enable the transfer of unsolicited messages from the PS to the NMS across the WAN in the form of SNMP Traps and SYSLOG messages
- enable the logging of status and exception information in the PS Database (local log)
- enable access to local log status and exception information via MIB objects
- maintain compatibility with event reporting as defined in DOCSIS specifications

6.3.3.2.2 Event Reporting Function System Design Guidelines

The system design guidelines listed in Table 6-18 guided specification of the CMP Event Reporting Function.

Table 6-18 CMP Event Reporting Function System Design Guidelines

Reference	Event Reporting Function System Design Guidelines
EvRep 1	The PS will support the reporting of status and exception information as SNMP Notifications, SYSLOG messages, and volatile and non-volatile local log messages.
EvRep 2	The PS will support configurable event throttles and limits.
EvRep 3	The PS will support configurable event priorities.

6.3.3.2.3 Event Reporting Function System Description

Event reporting is a means for an element to report on status or an error condition in an unsolicited message. IPCable2Home supports four types of event reporting:

1. SNMP notify or trap
2. SYSLOG messaging
3. Non-volatile local log
4. Volatile local log

The use of the DOCSIS Device MIB [RFC 2669] to configure the PS for where to send SNMP traps (notifications) and SYSLOG messages and for event inhibiting and throttling values is required. Event notification by the PS is fully configurable. This Recommendation defines where the PS is to report events assigned a particular priority (ref.: Table 6-19) and the DOCSIS Device MIB allows the priority of each event to be configured. The DOCSIS Device MIB also maintains statistics for the occurrence of each event. The Event Table (docsDevEventTable) in the DOCSIS Device MIB includes an entry for each unique event reported by the PS, a count for the number of occurrences for each unique event entry, and the time at which the last entry was made for each event entry.

IPCable2Home defines the procedure for re-indexing the Event Table in the event that the PS is re-initialized such that volatile local log entries are lost. When volatile local log entries are lost the PS is required to re-index the Event Table such that the remaining (volatile) local log entries are sequentially indexed.

6.3.3.2.4 Event Reporting Function Requirements

PS requirements for CMP Event Reporting Function are specified in Sections 6.3.3.2.4.1 - 6.3.3.2.4.9.

6.3.3.2.4.1 Event Notification

The PS MUST generate asynchronous events that indicate important events and situations as specified (refer to Annex B). Events can be stored in an internal event LOG, stored in non-volatile memory, reported to other SNMP entities (as TRAP or INFORM SNMP messages), or sent as a SYSLOG event message to the SYSLOG server whose IP address is passed in DHCP Option 7 of the DHCP OFFER received from the Headend DHCP server through the PS WAN-Man Interface.

The PS MUST support the following event notification mechanisms:

- local event logging where certain entries in the local log can be identified to persist across a reboot of the PS
- SNMP TRAP and INFORM
- SYSLOG

The PS MUST implement the docsDevEvControlTable from [RFC 2669] to control reporting of events. The following BITs values for the [RFC 2669] object docsDevEvReporting MUST be supported by the PS:

- local-nonvolatile(0)
- traps(1)
- syslog(2)
- local-volatile(3)

SNMP SET request messages to the [RFC 2669] object docsDevEvReporting using the following values MUST result in a 'Wrong Value' error for SNMP PDUs:

- 0x20 = syslog only
- 0x40 = trap only
- 0x60 = (trap + syslog) only

An event reported by Trap, Syslog, or Inform MUST also generate a local log entry, whether volatile or non-volatile according to Table 6-19, and as described in Section 6.3.3.2.4.2.

6.3.3.2.4.2 Local Event Logging

The PS MUST maintain a single local-log event table that contains events stored as both local-volatile events and local-nonvolatile events. Events stored as local-nonvolatile events MUST persist across reboots of the PS. The local-log event-table MUST be organized as a cyclic buffer with a minimum of ten entries. The single local-log event-table MUST be accessible through the docsDevEventTable as defined in [RFC 2669].

Event descriptions MUST NOT be longer than 255 bytes, which is the maximum defined for SnmpAdminString.

The EventId is a 32 bit unsigned integer. EventIds ranging from 0 to ((2^31) - 1) are reserved . The EventId MUST be converted from the error codes defined in Annex B. The EventIds ranging from 2^31 to ((2^32)-1) MUST be used as vendor specific EventIds using the following format:

- Bit 31 set to indicate vendor specific event
- Bits 30-16 contain bottom 15 bits of vendor's SNMP enterprise number
- Bits 15-0 used by vendor to number their events

The [RFC 2669] object docsDevEvIndex provides for relative ordering of events in the log. The tagging of local log events as local-volatile and local-nonvolatile necessitates a method for synchronizing docsDevEvIndex values between the two types of events after a PS reboot. After a PS reboot, to synchronize the docsDevEvIndex values for volatile and non-volatile events, the following procedure MUST be used:

- The values of docsDevEvIndex for local log events tagged as local-nonvolatile MUST be renumbered beginning with 1.
- The local log MUST then be initialized with the events tagged as local-nonvolatile in the same order as they had been immediately prior to the reboot.
- Subsequent events recorded in the local log, whether tagged as local-volatile or local-nonvolatile, MUST use incrementing values of docsDevEvIndex.

A reset of the local log initiated through an SNMP SET of [RFC 2669] object docsDevEvControl MUST clear all events from the local log, including log events tagged as both local-volatile and local-nonvolatile.

6.3.3.2.4.3 SNMP TRAP and INFORM

The PS MUST support the SNMP Trap PDU as described in [RFC 3411]. The PS MUST support the SNMP INFORM PDU as described in [RFC 3411]. INFORM is a variation of trap and requires the receiving host to acknowledge the arrival of an InformRequest-PDU with an InformResponse-PDU.

When a standard SNMP trap is enabled in the PS, it MUST send notifications for any event in that category whose priority is either “error” or “notice”.

The PS MAY support vendor-specific events. If supported, vendor-specific PS events reportable via SNMP TRAP MUST be described in a private MIB that is distributed with the PS. When defining a vendor-specific SNMP trap, the OBJECTS statement of the private trap definition SHOULD contain at least the objects explained below:

- EvLevel
- EvIdText
- Event Threshold (if any for the trap)
- IfPhysAddress (the physical address associated with the WAN-Man IP address of the PS)

More objects can be contained in the OBJECTS statement as desired.

6.3.3.2.4.4 Syslog

SYSLOG messages issued by the PS MUST be in the following format:

<level>PortalServicesElement[vendor]: <eventId> text

Where:

Level - ASCII presentation of the event priority, enclosed in angle brackets, which is constructed as the bitwise of the default Facility (128) and event priority (0-7). The resulted level has the range between 128 and 135.

vendor - Vendor name for the vendor-specific SYSLOG messages or “CABLEHOME” for the standard

IPCable2Home messages.

EventId - ASCII presentation of the INTEGER number in decimal format, enclosed in angle brackets, that uniquely identifies the type of event. This EventID MUST be the same number that is stored in docsDevEvId object in docsDevEventTable. For the standard IPCable2Home events, this number is converted from the error code using the following rules:

- The number is an eight digit decimal number.
- The first two digits (left most) are the ASCII code (decimal) for the letter in the Error code.
- The next four digits are filled by 2 or 3 digits between the letter and the dot in the Error code with zero filling in the zap in the left side.
- The last two digits are filled by the number after the dot in the Error code with zero filling in the zap in the left.

For example, event D04.2 is converted into 68000402, and Event I114.1 is converted into 73011401.

Please note that this notion only uses a small portion of available number space reserved for IPCable2Home (0 to 2^31-1). The first letter of an error code is always in upper case.

text - for the standard messages, this string MUST have the textual description as defined in Annex B.

The example of the syslog event for the event D04.2: “Time of the day received in invalid format”:

<132>Portal ServicesElement[CABLEHOME]: <68000402> Time of the day received in invalid format.

The number 68000402 in the given example is the number assigned to this particular event.

6.3.3.2.4.5 Format of Events

The IPCable2Home Management Event messages MAY contain any of the following information:

- Event Counter - indicator of event sequence
- Event Time - time of occurrence
- Event Priority - severity of condition. [RFC 2669] defines eight levels of severity. The default event severity can be changed to a different value for each given event via the SNMP interface.
- Event Enterprise Number - This number identifies the event as either a standard event or a vendor-defined event.
- Event ID - identifies the exact event when combined with the Event Enterprise Number. Vendors define their own Event ID's. IPCable2Home standard management events are defined in Annex B. Each management event described in the annex is assigned a Event ID.
- Event Text - describes the event in human readable form
- PS WAN-Man-MAC address - describes the MAC address of the PS Element used for management of the box
- PS WAN-Data-MAC address - describes the MAC address of the PS Element optionally used for data

The exact format of this information for traps and informs is defined in Annex B. The format for SYSLOG messages is defined in the requirements portion of this subsection.

6.3.3.2.4.6 Event Priorities

[RFC 2669] document defines 8 different priority levels and the corresponding reporting mechanism for each level. The standard events specified in this document utilize these priority levels.

Emergency event (priority 1)

Reserved for vendor-specific 'fatal' hardware or software errors that prevent normal system operation and cause the reporting system to reboot. Each vendor may define its own set of emergency events. Examples of such events could be 'no memory buffers available', 'memory test failure' etc.

Alert event (priority 2)

A serious failure which causes the reporting system to reboot but the reboot is not caused by either hardware or software malfunctioning. After recovering from the event, the system MUST send the cold/ warm start notification.

Critical event (priority 3)

A serious failure that prevents the device from transmitting data but could be recovered without rebooting the system. After recovering from a Critical event, the PS MUST send the Link Up notification. Examples of such events could be PS Configuration File problems or the inability to get an IP address through DHCP.

Error event (priority 4)

A failure that could interrupt the normal data flow but does not cause device to reboot. Error events can be reported in real time by using either the TRAP or SYSLOG mechanism.

Warning event (priority 5)

A failure that could interrupt the normal data flow. Syslog and Trap reporting are enabled by default for this level.

Notice event (priority 6)

An event of importance that is not a failure and could be reported in real time by using either the TRAP or SYSLOG mechanism. Examples of the NOTICE events are 'Cold Start', 'Warm Start', 'Link Up' and 'SW upgrade successful'.

Informational event (priority 7)

An event of importance that is not a failure, but which could be helpful for tracing the normal operation of the device.

Debug event (priority 8)

Reserved for vendor-specific non-critical events

The priority associated with standard events MUST NOT be changed.

Table 6-19 shows the default notification types for the various event priorities. The PS MUST implement the default notification types as defined in Table 6-19 Default Notification Types for PS Event Priorities, for the eight event priorities. For example, the default notification type for Emergency and Alert events is to place them in the local-log as nonvolatile entries.

Table 6-19 Default Notification Types for PS Event Priorities

Event Priority	Local-non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-volatile (bit-3)	Note
1 Emergency	Yes	No	No	No	Vendor Specific
2 Alert	Yes	No	No	No	CableHome
3 Critical	Yes	No	No	No	CableHome
4 Error	Yes	Yes	Yes	No	CableHome
5 F	Yes	Yes	Yes	No	CableHome
6 Notice	No	Yes	Yes	Yes	CableHome
7 Informational	No	No	No	No	CableHome and Vendor Specific
8 Debug	No	No	No	No	Vendor Specific

The PS MUST support the ability to be configured to generate all notification types for each event priority level

listed in Table 6-19.

6.3.3.2.4.7 Standard Events

The PS MUST send the following generic SNMP traps, as defined in [RFC 3418] and [RFC 2863]:

- coldStart [RFC 3418]
- linkUp [RFC 2863]
- linkDown [RFC 2863]
- SNMP authentication-Failure [RFC 3418]

The PS MUST be capable of generating event notifications based on standard events listed in Annex B.

6.3.3.2.4.8 Event Throttling and Limiting

The PS MUST support SNMP TRAP/INFORM and SYSLOG throttling and limiting as described in [RFC 2669].

The PS MUST consider events identical if their EventIds are identical.

[RFC 2669] specifies four throttling states:

- unconstrained(1) causes traps and syslog messages to be transmitted without regard to the threshold settings.
- maintainBelowThreshold(2) causes trap transmission and syslog messages to be suppressed if the number of traps would otherwise exceed the threshold.
- stopAtThreshold(3) causes trap transmission to cease at the threshold, and not resume until directed to do so.
- inhibited(4) causes all trap transmission and syslog messages to be suppressed.

A single event MUST be treated as a single event for threshold counting, that is, an event causing both a trap and a syslog message is still treated as a single event.

6.3.3.2.4.9 Secure Software Download Event Reporting

Table B-1 in Annex B, Format and Content for Event, SYSLOG and SNMP Trap, describes events associated with Portal Services software upgrades, in three categories: Software Upgrade Initialization (SW UPGRADE INIT), Software Upgrade General Failure, and Software Upgrade Success. These events apply only to the standalone PS, since software upgrade (also referred to as secure software download) for a PS with an embedded cable modem is controlled and managed by the DOCSIS cable modem. Section 11.8, Software Download Into Embedded or Standalone PS Elements defines requirements for secure software download for the two classes of Portal Services elements. The embedded PS, as defined in Section 5.1.3.1, Embedded PS and Standalone PS, MUST NOT generate events categorized in Table B-1, Defined Events for IPCable2Home as "Software Upgrade Initialization" (SW UPGRADE INIT) events, "Software Upgrade General Failure" (SW UPGRADE GENERAL FAILURE) events, or "Software Upgrade Success" (SW UPGRADE SUCCESS) events.

6.3.3.3 CMP Discovery Function

6.3.3.3.1 Discovery Function Goals

The goals for the CMP Discovery function are listed below:

- Provide cable operators with visibility to IPCable2Home Host device and IPCable2Home Residential Gateway device attributes.
- Provide cable operators with visibility to applications implemented on IPCable2Home Host devices.
- Co-existence and interoperability between PS, IPCable2Home Hosts and LAN IP devices that are NOT compliant with this Recommendation.

Note: The goals for Discovery do NOT preclude the use of other discovery methods, protocols, etc. on the LAN but are only intended to specify the requirements for compliant devices. However, IPCable2Home Host devices MUST NOT interfere with correctly operating non- IPCable2Home LAN IP Devices.

Assumptions

The assumptions for the CMP discovery capability include the following:

- IPCable2Home Host devices, LAN IP devices, and IPCable2Home Residential Gateway devices implement the Internet Protocol (IPv4) suite of protocols
- IPCable2Home Hosts implement a Device Profile in XML format as described in Section 6.5.3.1.3 and a QoS Profile in XML format described Section 10.3.2.4.2.1

6.3.3.3.2 Discovery Function System Design Guidelines

The system design guidelines listed in Table 6-20 provided guidance in the development of the CMP Discovery function specification.

Table 6-20 PS Discovery System Design Guidelines

Reference	Discovery System Design Guidelines
Discovery 1	The PS and BP will support a protocol for discovering IPCable2Home Host devices connected to the home LAN.
Discovery 2	The PS will provide to the cable operator upon request information about devices added to the home LAN.
Discovery 3	The PS will provide to the cable operator upon request information about applications implemented on IPCable2Home Host devices.
Discovery 4	Discovery protocol message exchange within the home LAN will not appreciably degrade performance of the home LAN.
Discovery 5	Home LAN discovery protocol messaging will not propagate onto the WAN.

6.3.3.3.3 Discovery Function System Description

The purpose of the CMP Discovery Function is to provide the cable operator with information about the devices and applications available on a subscriber's LAN.

The Discovery specifies the PS to serve as a central repository of information about devices and applications available on the subscriber's LAN. Specified BP logical elements provide device-specific information about the device in which they reside, and a list of applications implemented in the device in which they reside.

The Discovery function consists of the following two steps:

1. The PS learns each IPCable2Home Host's IP address and MAC address. The PS learns this information directly for LAN-Trans devices when it receives and responds to their DHCP DISCOVER requests. Refer to Section 7.3.3.1.4 CDS Requirements. The PS is required to learn this information from LAN-Pass devices in order to support USFS functionality (refer to Section 8.3.3.4 for Upstream Selective Forwarding Switch Overview and Requirements), but this Recommendation does not prescribe how this is to be done.

2. The PS acquires device attributes and applications information from each BP. Each BP is required to send its Device Profile and QoS Profile to the PS. This is done through a "BP initiated" model, in which the BP sends the information to the CMP. The BP is permitted to initiate this information transfer at any time but is required to do so each time it acquires or renews its IP address lease. The PS receives this information and stores it, making it accessible to the cable operator through the PSDev MIB [Annex E.4].

The PS maintains information about the IPCable2Home Residential Gateway device, analogous to the BP's Device Profile, in the PS Database. This information, enabling the cable operator to discover attributes of the IPCable2Home Residential Gateway, is available via SNMP through the sysDescr, sysName, and sysLocation objects of MIB-2 [RFC 1213] and through the PS Device Profile Group of the PSDev MIB [Annex E.4].

6.3.3.3.4 Discovery Function Requirements

The PS MUST store the Device Profile information (ref.: Section 6.5.3.1 BP Device Profile) received in the BP_Init message from each BP, in the PS Database and make it accessible via the PS Device MIB IPCable2Home Host/BP

Device Profile Table (cabhPsDevBpProfileTable) [Annex E.4]. The PS is also required to store application information received from the QoS Profile resulting in discovery of this application information. See Section 10.3.2.4.2.

The PS MUST store its Device Profile attributes listed below in the PS Database and make them accessible to the SNMP entity via the PS Device Profile Group of the PS Device MIB [Annex E.4]:

- Device Type (cabhPsDevPsDeviceType)
- Manufacturer Universal Resource Locator (cabhPsDevPsManufacturerUrl)
- Device Model Universal Resource Locator (cabhPsDevPsModelUrl)
- Device Universal Product Code (cabhPsDevPsModelUpc)

6.3.3.4 CMP LAN Messaging Function

LAN Messaging refers to the exchange of messages between the PS and a BP. Although SNMP systems are prevalent in cable operators' data networks for the purpose of monitoring and configuring Cable Modem Termination Systems (CMTS) and cable modems (CM), SNMP is not prevalent among devices that cable data service subscribers have connected to their home LANs. Consequently, IPCable2Home defines an in-home messaging protocol to satisfy cable operators' needs to support their data service subscribers while maintaining compatibility with messaging protocols typically implemented in LAN-based data communications devices. This section describes the LAN messaging solution.

It is critical to note that a BP could reside in either the LAN-Trans or LAN-Pass domain. A BP in the LAN-Trans domain can easily address packets to the PS, since the PS Server Router address (cabhCdpServerRouter) is the LAN-Trans BP's default gateway, passed to the BP in DHCP Option Code 3. However, a LAN-Pass BP has no legitimate knowledge of the PS Server Router IP address. LAN messages sent to the PS from a LAN-Trans BP can use the PS Server Router IP address as the destination IP address. Another method has to be defined for the LAN-Pass BP.

One way to ensure LAN-Pass BP-to-PS messaging, and the method adopted for use, is to define a fixed, "well-known" IP address in the PS that the LAN-Pass BP will use as a destination. Since the PS is a layer-2 bridge for LAN-Pass devices, the USFS function will be relied upon to capture messages sent by a LAN-Pass BP to the well-known destination IP address. The packet(s) addressed to the well-known PS IP address that are captured by the USFS function are then processed by the PS. The address 192.168.0.1 is defined as the "well-known" PS IP address that LAN-Pass BPs are required to use as the destination IP address for BP-PS LAN messaging. This fixed, well-known PS IP address is not permitted to be assigned by the CDS to LAN-Trans devices. The well-known PS IP address defined above has the same value as the *default* value of cabhCdpServerRouter, but the well-known PS IP address defined for LAN messaging is fixed. It cannot be changed, while the value of cabhCdpServerRouter can be changed via PS Configuration File or SNMP Set command. The PS is required to respond to both addresses, if they are different.

Since a BP could reside in either address domain, it needs to support the addressing method defined for LAN-Trans as well as the addressing method defined for LAN-Pass BPs. In other words, BPs are required to support both LAN-Trans - to - PS addressing and LAN-Pass - to - PS addressing, and the PS is required to accept messages destined to either the fixed "well-known" PS IP address or the PS Server Router address (which could be the same or could be different). The BP will use the presence or absence of DHCP Option code 43 sub-option 101 value "CableHome1.1LAN-Trans" in the DHCP ACK received from its DHCP server to determine which addressing method it is required to use. If this value is present in the DHCP Option code 43 sub-option 101 the BP is required to send its BP_Init messages to its (the BP's) default gateway, i.e., the PS Server Router address. If the value is not present in the DHCP ACK the BP is required to send its BP_Init messages to 192.168.0.1.

The PS will reply to a BP using as the destination address the BP address the PS received as a source IP address, i.e., the PS replies by sending to the address from which it received the BP-initiated message. To a LAN-Pass BP, this message appears to originate from a device in the LAN-Trans domain.

Figure 6-7 summarizes the BP-to-PS addressing requirements for a compliant BP logical element.

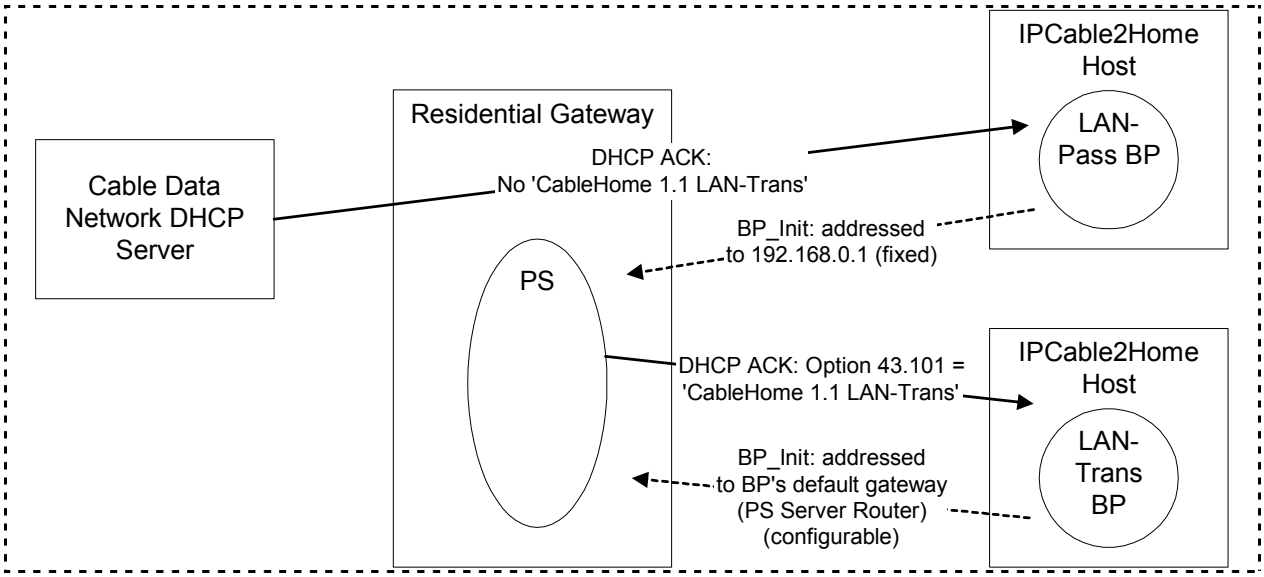

Figure 6-7 BP_Init Message Addressing

6.3.3.4.1 LAN Messaging Function Goals

Goals for LAN Messaging function are listed below:

- Support device and application discovery requirements by enabling the transfer of Device Profile information from BP logical elements in IPCable2Home Host devices to the PS element in compliant residential gateway devices.
- Specify an open, industry standard method for the exchange of Device Profile and prioritized Quality of Service Profile between the BP logical element in each IPCable2Home Host device and the PS logical element in a compliant residential gateway device.

6.3.3.4.2 LAN Messaging Function System Design Guidelines

The design guidelines listed in Table 6.3.3.4.3 guided specification of the LAN Messaging function.

6.3.3.4.3 LAN Messaging Function System Design Guidelines

Reference	LAN Messaging Function System Design Guidelines
LAN Msg 1	The PS and BP will support a protocol for exchanging XML-formatted information.
LAN Msg 2	The LAN messaging protocol will be an open standard.
LAN Msg 3	The LAN messaging protocol will be as compatible as possible with existing LAN IP Devices and Residential Gateway devices.

6.3.3.4.4 LAN Messaging Function System Description

Due to its flexibility, industry acceptance, and capabilities to pass configuration and status information, XML [XML1] was chosen as the information format for LAN (BP-PS) messaging. XML has gained acceptance as a communication protocol for the Internet, and is an open, non-proprietary format popular for its adaptation to disparate systems. XML benefits include its ability to enable the creation, modification, organization, and storage of information in any form tailored to the needs of management messages. XML document rules and character support provide additional benefit. The capabilities of XML make it a good fit for messages exchanged between PS and BP logical elements.

Simple Object Access Protocol (SOAP) [SOAP] is a member of the family of XML-associated protocols. It is a lightweight protocol for the exchange of information in a decentralized, distributed environment. SOAP is an XML-

based protocol that consists of three parts:

- an envelope the defines a framework for describing what is in a message and how to process it
- a set of encoding rules for expressing instances of application-defined datatypes, and
- a convention for representing remote procedure calls and responses

SOAP is specified for the exchange of Device Profiles and QoS Profiles between the PS and BP logical elements.

6.3.3.4.4.1 Simple Object Access Protocol (SOAP)

Encoding a Profile in XML is only the first step for the exchange of messages between IPCable2Home Residential Gateway and IPCable2Home Host devices. This Recommendation has to also provide conventions for the following:

- types of information to be exchanged
- how the information is to be expressed as XML
- how the information is sent from one logical element to another

Without these conventions, the PS and the BP cannot decode the information they're given, even if it is encoded in XML. These required conventions are provided by SOAP [SOAP]. Since this Recommendation specifies SOAP only for messaging within a subscriber's home LAN, not all of the SOAP messaging formats are required.

Transport Layer of SOAP

HTTP is the most commonly used transport mechanism for SOAP messaging. The PS and the BP are required to use HTTP as the transport mechanism for SOAP messaging to insure interoperability between various PS and BP implementations. In order to support this scheme, the PS implements an HTTP server listening on port 80 and the BP implements an HTTP client. The PS and the BP are each also required to have a SOAP Processing application running.

When the SOAP processing application running on a BP or on a PS receives a SOAP message, it processes that message by performing the following actions in the order listed below. The BP is prohibited from making modifications to the Device Profile or to the QoS Profile as a result any SOAP message other than the BP_Init_Response message received from the PS:

1. Identify all parts of the SOAP message intended for that application.

2. Verify that all mandatory parts identified in step 1 are supported by the application for this message and process them accordingly. If this is not the case then discard the message. The processor has the option to ignore optional parts identified in step 1 without affecting the outcome of the processing.

3. If applicable, send a response message as defined in later sections.

6.3.3.4.4.1.1 SOAP Message Formatting

This section introduces the format of SOAP messages necessary to support LAN messaging requirements.

The SOAP messaging that takes place between the PS and the BP (for the purpose of exchanging the device and QoS profiles) is initiated by the BP. This messaging is referred to as "BP_Init Operation".

This Recommendation defines two *confirmation code* tags used in SOAP messaging. The confirmation codes associated with these tags is described below:

Confirmation Codes

Confirmation code in a message indicates the success/failure details about the previous message in the transaction. A negative value indicates an error condition. Non-negative values indicate success. A positive nonzero value indicates an informative message. The CableHome-defined confirmation codes are listed in Table 6-21.

Table 6-21 CableHome LAN Messaging Confirmation Codes

Confirmation Code	Meaning
10	Unrecognized attribute present
0	Success
-10	A required attribute is missing
-20	Unacceptable value for an attribute
-30	Multiple errors encountered
-40	Error not otherwise classified or defined

CableHome defines two confirmation code tags: Device Confirmation Code and QoS Confirmation Code. Device Confirmation Code is the confirmation code specific to the Device Profile and QoS Confirmation Code is the confirmation code specific to the QoS Profile. The PS is permitted to send these in either order. The confirmation code values listed above apply as both Device Confirmation Codes and QoS Confirmation Codes.

6.3.3.4.4.2 BP-initiated SOAP Messaging (BP_Init Operation)

Figure 6-8 presents the message flow diagram for the messages exchanged between a BP and PS during BP-initiated SOAP messaging. The message sent by the BP to the PS is referred to as a *BP_Init* message. The response issued by a PS to the BP_Init message is a *BP_Init_Response*. The messaging shown in Figure 6-8 is a BP_Init message issued by the BP of its profile information to the PS (*BP_Init* message), and the PS response to the BP_Init message (*BP_Init_Response* message)

.

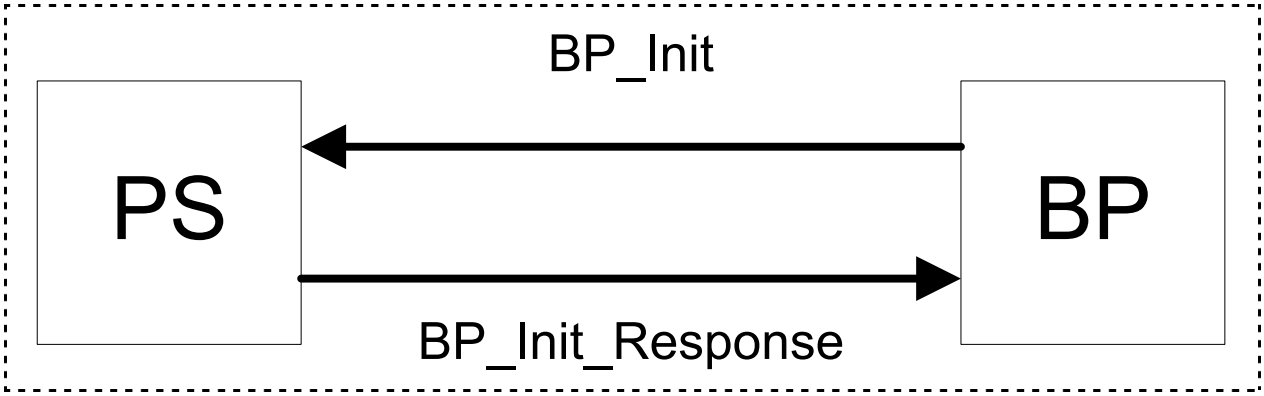

Figure 6-8 BP-initiated SOAP Messaging: BP_Init Operation

6.3.3.4.4.2.1 BP_Init Message Format

The format of the BP_Init message follows, using the transfer of the BP's Device Profile and QoS Profile to the PS as an example:

POST /DevQoSProfileService HTTP/1.1

HOST *IP Address of PS*

Content-Type: text/xml; charset="utf-8"

Content-Length: nnnn

SOAPAction: "/DevQoSProfileService"

```
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Body>
     <ch:BP_Init xmlns:m= IP Address of PS >
       <ch:BP_IP>

          IP Address of BP

       </ch:BP_IP>

       <ch:DeviceProfile>

          Device Profile from BP

       </ch:DeviceProfile>

       <ch:QoSProfile>

          QoS Profile from BP

        </ch:QoSProfile>

     </ch:BP_Init>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

6.3.3.4.4.2.2 BP_Init_Response Message Format

The format of the response message to the BP_Init message, the BP_Init_Response message, is shown below, using by way of example the response to the Device Profile and QoS Profile BP_Init message described above.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
   <SOAP-ENV:Body>
     <ch:BP_Init_Response xmlns:m= IP Address of PS >
       <ch:DeviceConfirmationCode>0</ch:DeviceConfirmationCode >

       <ch:QoSConfirmation Code>0</ch:QoSConfirmationCode>

       <ch:QoSProfile> QoS Profile from PS </ch:QoSProfile>
      </ch: BP_Init_Response>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

6.3.3.4.5 LAN Management Messaging Function Requirements

The PS MUST implement a HTTP server in accordance with Server requirements of [RFC 2616], listening on port 80.

The PS MUST implement an XML parser in accordance with [XML1].

The PS MUST implement a SOAP parser compliant with specifications described in [SOAP].

The PS MUST use HTTP as the transport mechanism for SOAP messaging to insure interoperability between

various PS and BP implementations.

The PS MUST run a SOAP-over-HTTP web service named DevQoSProfileService.

The PS MUST perform the following actions in the order listed when it receives a BP_Init SOAP message:

1. Identify all parts of the message intended for the PS.

2. Verify that the received message is formatted as specified in Section 6.3.3.4.4.2.1 and process the message. If the message does not contain all mandatory components, discard the message. The processor has the option to ignore optional parts identified in step 1 without affecting the outcome of the processing.

3. If the BP_Init message contained a Device Profile and/or a QoS Profile, send a BP_Init_Response message as defined in Section 6.3.3.4.4.1.1 SOAP Message Formatting.

4. If the message cannot be processed because it is incorrectly formatted, contains an invalid value, or does not conform with the CableHome specification or [SOAP] in some other way, return a status message to the sender with the appropriate confirmation code as described in Section 6.3.3.4.4.1.1 SOAP Message Formatting.

The PS MUST observe the following SOAP Syntax Rules:

- A SOAP message MUST be encoded using XML.
- A SOAP message MUST have a SOAP Envelope.
- A SOAP message MAY have a SOAP header.
- A SOAP message MUST have a SOAP Body.
- A SOAP message MUST use the SOAP Envelope namespaces.
- A SOAP message MUST use the SOAP Encoding namespace.
- A SOAP message MUST NOT contain a Document Type Declaration (DTD).
- A SOAP message MUST NOT contain XML Processing Instructions.
- The PS MUST use the following default namespaces:

  - for SOAP envelope syntax: http://schemas.xmlsoap.org/soap/envelope/

  - for SOAP encoding and data types: http://schemas.xmlsoap.org/soap/encoding/

  - for 'BP_Init_Response': IP Address of PS

The PS MUST accept and process each BP_Init message it receives with a destination IP address of 192.168.0.1 or with a destination IP address equal to the value of cabhCdpServerRouter.

The PS MUST ignore any BP_Init message received on any PS WAN Interface or with a destination IP address *not* equal to 192.168.0.1 or the value of cabhCdpServerRouter.

The PS MUST respond with a BP_Init_Response message to each BP_Init message received on its LAN interface and carrying a Device Profile, a QoS Profile, or both a Device Profile and a QoS Profile. The PS MUST send the BP_Init_Response message to the IP address which was the source IP address of the BP_Init message. The PS is not required to respond to BP_Init messages that carry neither a Device Profile nor a QoS Profile.

If the BP_Init message received by the PS contains a Device Profile, the BP_Init_Response message issued by the PS MUST contain a valid Device Confirmation Code.

If the BP_Init message received by the PS contains a QoS Profile, the BP_Init_Response message issued by the PS MUST contain a valid QoS Confirmation Code and MAY contain a QoS Profile.

The PS MUST NOT transmit a BP_Init_Response message out any PS WAN interface.

6.4 PS Logical Element CableHome Test Portal (CTP)

6.4.1 CTP Goals

The goals for the CableHome Test Portal include:

- Enable LAN IP Device and CableHome Host fault diagnostics
- Enable visibility to LAN IP Devices and CableHome Hosts, as well as access to the number and types of LAN IP Devices and CableHome Host
- Enable LAN IP Device and CableHome Host performance monitoring

6.4.2 CTP Design Guidelines

The Test Portal system design guidelines are listed in Table 6-22. A number of these guidelines are common with the CMP design guidelines. This list provided guidance for the specification of CTP functionality.

Table 6-22 CTP System Design Guidelines

Reference	CTP System Design Guidelines
CTP 1	The need exists for interfaces to support the management and diagnosis features and functions required to support cable-based services provisioned across the home network.
CTP 2	Local and remote monitoring capabilities are needed that can monitor home network operation and help the consumer and cable operator identify problem areas.
CTP 3	The cable network NMS requires a method to gather identification information about each IP device connected to the home network.
CTP 4	The cable network NMS requires a method to detect whether a connected device is in an operable state.

6.4.3 CTP System Description

The CTP (IPCable2Home Test Portal) contains the "remote tools" with which the NMS can collect further LAN device information. Tests must be run remotely, since getting past a network address translation (NAT) function in a router can be a challenge. For example, a WAN-to-LAN ping will not pass through a PS, unless the CAP has been preconfigured to pass this traffic. The CTP is a local proxy used to interpret and execute the remote fault/diagnostic class of SNMP messages it receives from the NMS operator. These LAN IP Device and IPCable2Home Host tests are defined based on problems likely to be encountered for CableHome 1.1 type of home networks: connectivity and throughput diagnostics.

These functions are termed the CTP Connection Speed Tool and CTP Remote Ping Tool. The Connection Speed and Remote Ping Tools enable the cable operator's customer support center and network operations center to learn more about the connection between the PS element and LAN IP Devices and IPCable2Home Hosts in the home.

6.4.3.1 CTP Connection Speed Tool Function

6.4.3.1.1 Connection Speed Tool Function Goals

The goal of the Connection Speed function is to enable the IPCable2Home system manager to remotely acquire metrics about the performance of the home LAN between the PS and a specific LAN IP Device or IPCable2Home Host.

6.4.3.1.2 Connection Speed Tool System Design Guidelines

Design guidelines listed in Table 6-22 *CTP System Design Guidelines* were used to guide specification of the Connection Speed Tool function.

6.4.3.1.3 Connection Speed Tool Function System Description

The Connection Speed Tool function is used to get a rough measure of the throughput performance across the link between the PS and a LAN IP Device or IPCable2Home Host. It sends a burst of packets between the PS and the LAN IP Device or IPCable2Home Host under test, and the round trip time is measured for the burst. Generally speaking, the NMS operator fills in a few parameters and triggers the function, and results are stored in the PS Database for later retrieval through the CTP MIB [Annex E.3].

The Connection Speed function relies on the LAN IP Devices and IPCable2Home Hosts to have a "loop-back function" or "echo-service" embedded. The Internet Assigned Numbers Authority (IANA) has assigned the echo service port 7 for both TCP and UDP [RFC 347]. The default value of the source IP address (cabhCtpConnSrcIp) is the same as the value of the PS LAN default gateway (cabhCdpServerRouter). The value of cabhCtpConnSrcIp can be set to any valid PS WAN-Data IP address or to any valid PS LAN Interface IP address. The PS WAN-Man IP address is not used as the source IP address for a CTP tool since when a PS WAN-Man IP address is present but a PS WAN-Data IP address is not, the PS is operating in Passthrough Primary Packet-handling mode and the cable operator can test LAN IP Devices and IPCable2Home Hosts directly from the NMS console if desired. This test feature works on LAN IP Devices and IPCable2Home Hosts in either the LAN-Trans or LAN-Pass address realms that implement the Echo Service function as described in [RFC 347].

The CTP Testable Requirements section below lists the parameters and responses for the Connection Speed Tool. Section 12.2.1.1 details the operation of the Connection Speed Tool.

6.4.3.1.4 Connection Speed Tool Function Requirements

The PS MUST implement the Connection Speed Tool, and MUST comply with the default values and value ranges defined for the Connection Speed Tool-specific objects of the CTP MIB [Annex E.3].

The PS SHOULD transmit the bytes of test data as fast as possible when running the Connection Speed Tool.

The PS MUST use Port 7 as the Destination Port when running the Connection Speed Tool.

The PS MUST NOT generate packets out any WAN Interface when running the Connection Speed Tool function.

When the NMS triggers the CTP to initiate the Connection Speed Tool by setting cabhConnControl = start(1), the PS MUST do the following:

- reset the timer
- set cabhCtpConnStatus = running(2)
- transmit the number of packets equal to the value of cabhCtpConnNumPkts, each of the size equal to the value of cabhCtpConnPktSize, to the IP address equal to the value of cabhCtpConnDestIp and port number 7, using the protocol specified by cabhCtpConnProto
- initiate the timer with the first bit transmitted
- terminate the timer when the last bit is received back from the target LAN IP Device or when the value of the timer is equal to the value of cabhCtpConnTimeOut, whichever occurs first
- when the timer is terminated, set cabhCtpConnStatus = complete(3) and report the appropriate event (refer to Annex B - CTP Events)
- store the value of the timer (in milliseconds) in cabhCtpConnRTT
- if the Connection Speed Tool test times out before the last bit is received from the target LAN IP Device or IPCable2Home Host, report the appropriate event (refer to Annex B - CTP Events)
- calculate the throughput as defined in the requirement below and store the value in cabhCtpConnThroughput

If the Connection Speed Tool is terminated by the NMS setting the object cabhCtpConnControl = abort(2) or for any other reason before the last bit is received from the target LAN IP Device and IPCable2Home Host or before the Connection Speed Tool test times out, the PS MUST set cabhCtpConnStatus = aborted(4) and report the appropriate event (refer to Annex B - CTP Events).

When the Connection Speed Tool function is executing, the PS MUST determine the average round-trip throughput between the PS and the LAN IP Device or IPCable2Home Host whose address is passed in cabhCtpConnDestIp (the target LAN IP Device) in kilobits per second, round the number to the nearest whole integer, and store the result in cabhCtpConnThroughput.

The PS MUST reset cabhCtpConnPktsSent, cabhCtpConnPktsRecv, cabhCtpConnRTT and cabhCtpConnThroughput each to a value of 0 when the Connection Speed Tool is initiated (i.e., when the value of cabhCtpConnControl is set to start(1)).

Connection Speed Tool RTT is measured at the PS as the time from the first bit of the first sent packet to the last bit of the last received packet. RTT is only valid if the number of received packets is equal to the number of transmitted packets.

The PS MUST allow the Connection Speed Tool destination IP address (cabhCtpConnDestIp) to be set to any valid IPv4 address of any LAN IP Device accessible through any LAN Interface of the PS running the CTP Connection Speed Tool.

Setting the Connection Speed Tool control object, cabhCtpConnControl, with the value start(1) MUST result in the execution of the Connection Speed Tool.

Setting the Connection Speed Tool control object, cabhCtpConnControl, with the value abort(2) MUST result in the termination of the Connection Speed Tool.

The default value of cabhCtpConnStatus is notRun(1), which indicates that the Connection Speed Tool has never been executed.

The PS MUST set the value of cabhCtpConnStatus to running(2) if the Tool has been instructed to start, has not been terminated, and if the Connection Speed Timer has not timed out.

The PS MUST set the value of cabhCtpConnStatus to complete(3) when the last packet sent by the Connection Speed Tool is received by the CTP.

The PS MUST set the value of cabhCtpConnStatus to aborted(4) if the Connection Speed Tool is terminated after it is initiated by an SNMP set of the value abort(2) to the object cabhCtpConnControl, or if the test is otherwise terminated before the last packet sent by the Connection Speed Tool is received and before the Connection Speed Tool timer (cabhCtpConnTimeOut) expires.

The PS MUST set the value of cabhCtpConnStatus to timedOut(5) if the Connection Speed Tool timer (cabhCtpConnTimeOut) expires before the last packet sent by the Connection Speed Tool is received by the CTP.

The PS MUST NOT use any IP address for the Connection Speed Tool source IP address (cabhCtpConnSrcIp) except a current, valid PS WAN-Data IP address (i.e., an active cabhCdpWanDataAddrIp object value) or a current, valid PS LAN Interface IP address. If an invalid value is configured for cabhCtpConnSrcIp, the PS MUST treat the execution of the test as an aborted case and set the Connection Speed Tool status object cabhCtpConnStatus to 'aborted' and report the appropriate event (see Table B-1).

6.4.3.2 CTP Ping Tool Function

6.4.3.2.1 Ping Tool Function Goals

The goal of the Ping Tool function is to enable the system manager to remotely test or verify connectivity between the PS and a specific LAN IP Device.

6.4.3.2.2 Ping Tool Function System Design Guidelines

Design guidelines listed in Table 6-22 "CTP System Design Guidelines" were used to guide specification of the Ping Tool function.

6.4.3.2.3 Ping Tool Function System Description

The Ping Tool function is called to test connectivity between the PS and individual LAN IP Devices or IPCable2Home Host devices. Results of multiple executions of the Ping Tool test can be assembled by the NMS to create a network scan of the LAN IP Devices or IPCable2Home Host devices. The DHCP table of the CDP has a

list of historical devices, but only the devices that employ DHCP. Ping may capture a current state including non-DHCP clients. To keep the PS simple, it is expected that the NMS increments the address and stores the results in the NMS tool to perform a scan of a LAN subnet.

The PING Tool is initiated by a series of SNMP set-request messages issued by the cable network NMS console to the PS management address.

Section 12.2.1.2 details the operation of the Ping Tool.

6.4.3.2.4 Ping Tool Function Requirements

The CTP Ping Tool MUST be implemented using the Internet Control Message Protocol (ICMP) "Echo" facility. The CTP will issue an ICMP Echo Request and the LAN IP Device is expected to return an ICMP Echo Reply.

The CTP MUST ignore, and exclude from the cabhCtpPingNumRecv count, any Echo Reply received after cabhCtpPingTimeOut expires.

The PS MUST implement the CTP Ping Tool, and MUST comply with the default values and value ranges defined for the Ping Tool-specific objects of the CTP MIB [Annex E.3].

When the NMS triggers the PS to initiate the Ping Tool by setting cabhPingControl = start(1), the PS MUST do the following:

- set cabhCtpPingStatus = running(2)
- issue as many Pings (ICMP requests) as specified by the value cabhCtpPingNumPkts, to the IP address defined by the value of cabhCtpPingDestIp, using the value of cabhCtpPingSrcIp as the source address of each request. The size of each test frame issued is the value of cabhCtpPingPktSize. A timeout for each ping (ICMP Echo Request/Response pair) is the value of cabhCtpPingTimeOut.
- if the value of cabhCtpPingNumPkts is greater than 1, wait the amount of time defined by the value of cabhCtpPingTimeBetween between each Ping request issued by the CTP.

If the CTP receives all Ping replies before their individual timeout timer expires, the PS MUST set cabhCtpPingStatus = complete(3) and report the appropriate event (refer to Annex B - CTP Events).

If the Ping Tool is terminated by the NMS setting the object cabhCtpPingControl = abort(2) or for any other reason before the last bit is received from the target LAN IP Device and before the timer is terminated, the PS MUST set cabhCtpPingStatus = aborted(4) and report the appropriate event (refer to Annex B - CTP Events).

If a timeout timer expires for at least one of the pings, before its reply is received from the target LAN IP Device, the PS MUST set cabhCtpPingStatus = timedOut(5) and report the appropriate event (refer to Annex B - CTP Events).

When the CTP Ping Tool function is initiated, the PS MUST determine the average round-trip time between the PS and the LAN IP Device or IPCable2Home Host device whose address is passed in cabhCtpPingDestIp (the target LAN IP Device), over the number of Ping requests defined by cabhCtpPingNumPkts, and store the result in cabhCtpPingAvgRTT. When the CTP Ping Tool function is initiated, the PS MUST determine the minimum and maximum round-trip times between the PS and the target LAN IP device, for the set of Ping requests defined by cabhCtpPingNumPkts, and store the values in cabhCtpPingMinRTT and cabhCtpPingMaxRTT, respectively.

If an ICMP error occurs during execution of the Ping Tool, the PS MUST increment the value of cabhCtpPingNumIcmpError and log the error in cabhCtpPingIcmpError. The last ICMP error that occurs will over-write the previous one written.

The PS MUST reset cabhCtpPingNumSent, cabhCtpPingNumRecv, cabhCtpPingAvgRTT, cabhCtpPingMaxRTT, cabhCtpPingMinRTT, cabhCtpPingNumIcmpError and cabhCtpPingIcmpError each to a value of 0 when the Ping Tool is initiated (i.e., when the value of cabhCtpPingControl is set to start(1)).

Ping Tool RTT is measured at the PS as the time from the last bit of each ICMP Echo Request packet transmitted by the CTP Ping Tool, to the time when the last bit of the corresponding ICMP Echo Reply packet is received.

The PS MUST allow the Ping Tool destination IP address (cabhCtpPingDestIp) to be set to any valid IPv4 address of any LAN IP Device or IPCable2Home Host device accessible through any LAN Interface of the PS running the CTP Ping Tool.

The PS MUST NOT generate packets out any WAN Interface when executing the Ping Tool function.

The PS MUST NOT use any IP address for the Ping Tool source IP address (cabhCtpPingSrcIp) except a current, valid PS WAN-Data IP address (i.e., an active cabhCdpWanDataAddrIp object value) or a current, valid PS LAN Interface IP address. If an invalid value is configured for cabhCtpPingSrcIp, the PS MUST treat the execution of the test as an aborted case and set the Ping Tool status object cabhCtpPingStatus to "aborted" and report the appropriate event (see Table B-1).

6.5 BP Logical Element - Management Boundary Point (MBP)

Section 5 defines the Boundary Point (BP), which is the IPCable2Home-defined logical element aggregating IPCable2Home-specified functionality of a IPCable2Home Host device. The Management Boundary Point (MBP) is the logical element of the BP responsible for IPCable2Home-defined discovery capabilities of the BP.

Discovery of IPCable2Home Host devices is the first step of the eventual management of IPCable2Home-specified functionality in these devices. This Recommendation enables discovery of CableHome Host devices through access to the Profile information via HTTP, from the CMP.

6.5.1 MBP Goals

The goal for the MBP is to fulfill IPCable2Home requirements for IPCable2Home Host device discovery and LAN messaging. The MBP is required to provide the cable operator with the Device Profile for each IPCable2Home Host device, through the PS acting as a proxy.

6.5.2 MBP System Design Guidelines

System design guidelines listed in Table 6-23 guided specification of the MBP.

Table 6-23 MBP System Design Guidelines

Reference	MBP System Design Guidelines
MBP 1	The MBP will maintain information about the attributes of the IPCable2Home Host device in which the BP resides.
MBP 2	The MBP will provide IPCable2Home Host device and application information to the IPCable2Home system manager during the BP initialization process.
MBP 3	The MBP will provide IPCable2Home Host device and application information to the IPCable2Home system manager periodically after BP initialization completes.

6.5.3 MBP System Description

The BP is required to maintain a Device Profile as described in Section 6.5.3.1.3 Device Profile Description and a QoS Profile described in Section 10.3.2.4.2.1 QoS Profile XML Schema.

The BP is further required to send the Device Profile to the PS, thereby providing the IPCable2Home system manager access to each IPCable2Home Host device's attribute information through the PS Device MIB [Annex E.4] via SNMP access over the cable data WAN. By providing access to the IPCable2Home Host device's attribute information in this fashion, the MBP satisfies the requirements for device discovery.

The BP is also required to support LAN messaging using SOAP over HTTP transport, as the means by which the Device Profile and QoS Profile are transferred from the BP to the PS.

6.5.3.1 BP Device Profile

The Device Profile and QoS Profile are XML-formatted structures containing information about the IPCable2Home Host device and the applications it implements. The Device Profile is used as a means for maintaining and communicating information about the IPCable2Home Host device. The BP is required to implement a Device Profile and provide its Device Profile information to the PS, which makes the information available through the PS Device MIB [Annex E.4]. The cable operator's data network NMS and other subscriber-

support organizations can obtain basic information about the IPCable2Home Host device by querying the PS Device MIB over the cable data network using SNMP Get-request messages.

6.5.3.1.1 Device Profile Goals

The goals of the BP Device Profile are listed below:

- aggregate information specific and unique to the IPCable2Home Host device implementing the BP
- provide the IPCable2Home system manager with information about the IPCable2Home Host device

6.5.3.1.2 Device Profile System Design Guidelines

System design guidelines listed in Table 6-24 guided the specification of the MBP Device Profile.

Table 6-24 MBP Device Profile System Design Guidelines

Reference	MBP Device Profile System Design Guidelines
MBP DevProf 1	The MBP will maintain a set of device-specific information about the IPCable2Home Host device in which the BP resides.
MBP DevProf 2	The format of the device-specific information will adhere to an open standard.
MBP DevProf 3	The format of the device-specific information maintained by an MBP will compatible with LAN IP Device operating systems, will be flexible to accommodate any kind or amount of device-specific information, and will be as compatible as possible with industry protocols and trends.

6.5.3.1.3 Device Profile Description

This Recommendation specifies implementation of a Device Profile and a QoS Profile in BP logical elements to support discovery of IPCable2Home Host Devices and to support the provisioning of QoS priorities in BPs. The Device Profile and QoS Profile are XML-formatted structures. The Device Profile contains a set of attributes that describe the IPCable2Home Host device. A Device Profile includes IPCable2Home-specified attributes and could include vendor-specified attributes as well. The QoS Profile contains a list of IANA-defined port numbers that reflect applications implemented by each device, the priority assigned to each application, and optional information about QoS priority for destination IP address and destination port number. The Device Profile is described in this section. The QoS Profile is described in the Section 10.3.2.4.2.1 QoS Profile XML Schema.

Table 6-25 presents a high-level description of the Device Profile required for BP elements.

Table 6-25 BP Device Profile Attributes

Attribute Name	Attribute Type	Use
Device Type	String	required
Manufacturer	String	required
Manufacturer's URL	String	optional
Hardware Revision	String	required
Hardware Options	String	optional
Serial Number	String	required
Model Name	String	optional
Model Number	String	optional
Model URL	String	optional
Model UPC	String	optional
Model Software OS	String	required
Model Software Version	String	required
LAN Interface Type (IANA ifType)	String	required
Number of Media Access Priorities	Integer	required
Physical Location	String	optional
Physical Address	String	required

Device Profile Attribute Details:

The *Device Type* attribute can have one of the following values: IPCable2Home Residential Gateway or IPCable2Home Host.

The *Manufacturer* attribute is the name of the device manufacturer.

The *Manufacturer's URL* attribute is the Uniform Resource Locator for the manufacturer's web site.

The *Hardware Revision* attribute is a string assigned by the manufacturer uniquely identifying a specific product hardware revision.

The *Hardware Options* attribute is a string assigned by the manufacturer identifying optional product hardware features implemented in the product.

The *Serial Number* attribute is the unique identifying serial number for the IPCable2Home Host device, assigned by the device manufacturer.

The *Model Name* attribute is the IPCable2Home Host device's model name or other identifying name assigned by the device manufacturer.

The *Model Number* attribute is the model number or other identifying value assigned by the device manufacturer.

The *Model URL* attribute is the Uniform Resource Locator for the model's web site.

The *Model UPC* attribute is the Universal Product Code value assigned to the device.

The *Model Software OS* attribute is the operating system implemented on the device.

The *Model Software Version* attribute is the version of software currently running on the device.

The *LAN Interface Type* attribute is a string containing the IANAifType value [IANAType] for ISO OSI Layer 2 networking technology implemented by the product.

The *Number of Media Access Priorities* attribute refers to the number of media access priorities the IPCable2Home Host device's LAN interface supports. This attribute and its uses are described in detail in Section 10 (QoS Section).

The *Physical Location* attribute is a value that can be assigned by the device owner indicating the physical location of the device, such as *Office* or *Living Room*.

The *Physical Address* attribute is the device's hardware address, such as the Media Access Control (MAC) address of an 802.3-based device.

6.5.3.1.4 Device Profile in XML Format

The Device Profile in XML format as required by IPCable2Home is shown below.

```
<xs: complexType name="ch:device">
    <xs:element name="ch:deviceType"            type="xs:string"/>
    <xs:element name="ch:manufacturer"          type="xs:string"/>
    <xs:element name="ch:manufacturerURL"       type="xs:string"/>
    <xs:element name="ch:hardwareRevision"      type="xs:string"/>
    <xs:element name="ch:hardwareOptions"       type="xs:string"/>
    <xs:element name="ch:serialNumber"          type="xs:string"/>
    <xs:element name="ch:modelName"             type="xs:string"/>
    <xs:element name="ch:modelNumber"           type="xs:string"/>
    <xs:element name="ch:modelURL"              type="xs:string"/>
    <xs:element name="ch:modelUPC"              type="xs:string"/>
    <xs:element name="ch:modelSoftwareOS"       type="xs:string"/>
```

```
        <xs:element name="ch:modelSoftwareVersion"          type="xs:string"/>
        <xs:element name="ch:lanInterfaceType"              type="xs:string"/>
        <xs:element name="ch:numberMediaAccessPriorities" type="xs:int"/>
        <xs:element name="ch:physicalLocation"              type="xs:string"/>
        <xs:element name="ch:physicalAddress"               type="xs:string"/>
    </xs:complexType>
```

6.5.3.1.5 Device Profile Requirements

The BP MUST implement a Device Profile as described in Section 6.5.3.1.4, consistent with XML formatting rules described in [XML1].

The BP MUST populate the Device Type attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with the string "CableHome Host" (without the quotes).

The BP MUST populate the Manufacturer attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value identifies the manufacturer of the IPCable2Home Host device in which the BP resides.

The BP MUST populate the Hardware Revision attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value accurately represents the manufacturer's hardware revision number for the IPCable2Home Host device in which the BP resides.

The BP MUST populate the Serial Number attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value is equal to the serial number uniquely identifying the IPCable2Home Host device in which the BP resides.

The BP MUST populate the Model Software OS attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value accurately represents the software operating system implemented on the IPCable2Home Host device in which the BP resides.

The BP MUST populate the Model Software Version attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value accurately represents the version of BP software implemented on the IPCable2Home Host device in which the BP resides.

The BP MUST populate the LAN Interface Type attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value is equal to the IANAifType [IANAType] representing the LAN technology supported by the IPCable2Home Host device in which the BP resides.

The BP MUST populate the Number of Media Access Priorities attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with an integer in the range 1 - 8 whose value is equal to the number of LAN interface priorities supported by the IPCable2Home Host device in which the BP resides.

The BP MAY populate the Manufacturer's URL attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value accurately and uniquely identifies a Uniform Resource Locator for the manufacturer of the IPCable2Home Host device in which the BP resides.

The BP MAY populate the Hardware Options attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value represents the hardware options of the IPCable2Home Host device in which the BP resides.

The BP MAY populate the Model Name attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value accurately and uniquely identifies the manufacturer's model name for the IPCable2Home Host device in which the BP resides.

The BP MAY populate the Model Number attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value accurately and uniquely identifies the manufacturer's model number for the IPCable2Home Host device in which the BP resides.

The BP MAY populate the Model URL attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value accurately and uniquely identifies a Uniform Resource Locator for the IPCable2Home Host device model in which the BP resides.

The BP MAY populate the Model UPC attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value accurately and uniquely identifies the Universal Product Code for the IPCable2Home Host device in which the BP resides.

The BP MAY populate the Physical Location attribute of the BP Device Profile (ref.: Section 6.5.3.1.4 Device Profile in XML Format) with a string whose value identifies the physical location of the IPCable2Home Host device in which the BP resides.

6.5.3.2 MBP LAN Messaging Function

6.5.3.2.1 MBP LAN Messaging Function Goals

The goals of the MBP LAN Messaging Function are listed in Section 6-7 LAN Messaging Function Goals.

6.5.3.2.2 MBP LAN Messaging Function System Design Guidelines

The MBP LAN Messaging Function System Design Guidelines are listed in Table 6.3.3.4.3 LAN Messaging Function System Design Guidelines.

6.5.3.2.3 MBP LAN Messaging Function System Description

The MPB LAN Messaging Function is as described in Section 6.3.3.4.4 LAN Messaging Function System Description.

6.5.3.2.4 MBP LAN Messaging Function Requirements

The BP MUST implement an Echo Service responder, such that the BP immediately echoes any IP packet received on Port 7 to the sender of the packet, bit for bit, changing only the source IP address and port for the destination IP address and port, and vice versa.

The BP MUST implement ICMP Echo and Echo Reply Message types (Type 8 and Type 0) and ICMP Timestamp and Timestamp Reply Message types (Type 13 and Type 14) as described in [RFC 792], and reply appropriately to Ping requests received on any interface.

The BP MUST implement an HTTP client in accordance with the Client requirements of [RFC 2616].

The BP MUST implement an XML parser in accordance with [XML1].

The BP MUST implement a SOAP parser in accordance with [SOAP].

The BP MUST use HTTP as the transport mechanism for SOAP messaging to insure interoperability between various PS and BP implementations.

If the BP received DHCP Option Code 43 sub-option 101 containing the string ‘CableHome 1.1 LAN-Trans’ in the DHCP ACK, the BP MUST address each BP_Init message to its default gateway (value of DHCP Option 3 received in the DHCP ACK).

If the BP did not receive Option Code 43 sub-option 101 containing the string ‘CableHome 1.1 LAN-Trans’ in the DHCP ACK, the BP MUST address each BP_Init message to IP address 192.168.0.1.

The BP MUST NOT transmit a BP_Init message more frequently than once per 20 seconds.

The BP MUST NOT transmit a BP_Init message any time other than the specific occasions listed in Section 10.4.1.4.1.1, “BP information to the PS using BP_Init Message,” on page 169.

The BP MUST NOT transmit a BP_Init message to any address other than the BP’s default gateway address or 192.168.0.1.

The BP MUST observe the following SOAP Syntax Rules:

- A SOAP message MUST be encoded using XML.
- A SOAP message MUST have a SOAP Envelope.

- A SOAP message MAY have a SOAP header.
- A SOAP message MUST have a SOAP Body.
- A SOAP message MUST use the SOAP Envelope namespaces.
- A SOAP message MUST use the SOAP Encoding namespace.
- A SOAP message MUST NOT contain a Document Type Declaration (DTD).
- A SOAP message MUST NOT contain XML Processing Instructions.
- The BP MUST use the following default namespaces:

  - for SOAP envelope syntax: http://schemas.xmlsoap.org/soap/envelope/

  - for SOAP encoding and data types: http://schemas.xmlsoap.org/soap/encoding/

  - for 'BP_Init': IP Address of PS

The BP MUST perform the following actions in the order listed when it receives a SOAP message:

1. Identify all parts of the SOAP message intended for the BP.

2. Verify that the received message is formatted as specified in Section 6.3.3.4.4.2.1 and process the message. If the message does not contain all mandatory components, discard the message. The processor has the option to ignore optional parts identified in step 1 without affecting the outcome of the processing.

3. If the message cannot be processed because it is incorrectly formatted, contains an invalid value, or does not conform with this Recommendation or [SOAP] in some other way, the BP MUST re-issue the BP_Init message, for a total of three attempts within a three minute period. If the BP does not receive a valid BP_Init_Response message after issuing three BP_Init messages within a three-minute period, the BP MUST stop retrying until it next renews or acquires its IP address lease.

6.5.3.3 MBP Discovery Function

6.5.3.3.1 MBP Discovery Function Goals

The goal for the IPCable2Home MBP Discovery functionality is to `provide the IPCable2Home system manager with information about the IPCable2Home Host device in which the BP resides.`

6.5.3.3.2 MBP Discovery Function System Design Guidelines

The design guidelines listed in Table 6-26 provided guidance for the specification of the MBP Discovery function.

Table 6-26 MBP Discovery Function System Design Guidelines

Reference	MBP Discovery System Design Guidelines
MBP Disc 1	The MBP will provide device-specific information about the IPCable2Home Host in which it resides to the cable operator through the PS acting as a proxy.
MBP Disc 2	The MBP will provide information about the applications implemented by a IPCable2Home Host device to the cable operator through the PS acting as a proxy.

6.5.3.3.3 MBP Discovery Function System Description

Each BP is required to implement a Device Profile in XML format as described in Section 6.5.3.1.4 Device Profile in XML Format. Each BP is also required to implement a QoS Profile described in Section 10.3.2.4.2.1 QoS Profile XML Schema. When the BP is operational and has completed initialization, it is required to send Device Profile and QoS Profile information to the PS using LAN Messaging described in Section 6.3.3.4 CMP LAN Messaging Function. By providing the PS with Device Profile and QoS Profile information, the BP enables the cable operator

to discover attributes of the IPCable2Home Host device in which the BP resides and the applications running on it, through the PS acting as a proxy for the cable operator's network management system.

6.5.3.3.4 Discovery Function Requirements

Upon receipt of any DHCPACK message [RFC 2131] addressed to itself, the BP MUST transmit a BP_Init message as described in Section 6.3.3.4.4.2 containing its Device Profile and its QoS Profile in the message body. The BP sends BP_Init at other times, including when its QoS Profile is refreshed as described in Section 10.4.1.4.1, "LAN Information Exchange," on page 169.

If the BP does not receive a valid BP_Init_Response message within one minute after the BP sends a BP_Init message, the BP MUST immediately retransmit the BP_Init message with the BP's Device Profile and QoS Profile in the message body, repeating the process for a total of three attempts or until the BP receives a valid BP_Init_Response message, whichever occurs first.

If the BP does not receive a valid BP_Init_Response message after sending a sequence of three BP_Init messages the BP MUST wait until it receives the next DHCPACK [RFC 2131] message and repeat the process.

7 PROVISIONING TOOLS

7.1 Introduction/Overview

The Portal Services element and LAN IP Devices must be properly initialized and configured in order to exchange meaningful information with one another and with elements connected to the cable network and the Internet. IPCable2Home provisioning tools provide the means for this initialization and configuration to occur seamlessly and with minimum user intervention. They also enable cable operators to add value to high-speed data service subscribers by defining processes through which the cable operator can facilitate and customize PS and LAN IP Device initialization and configuration. The three provisioning tools defined to accomplish this task are listed below:

- DHCP Portal (CDP) function in the Portal Services element
- Bulk Portal Services Configuration (BPSC) tool
- Time of Day Client in the Portal Services element

7.1.1 Goals

Goals of the Provisioning Tools are listed below:

- Enable the PS to acquire a network address on its WAN interface to be used for management of the PS
- Enable the PS to acquire one or more network addresses on its WAN interface to be used for the exchange of traffic between LAN IP Devices and the Internet or between IPCable2Home Host devices and the Internet
- Enable the PS to request and acquire configuration parameters in a configuration file
- Enable the PS to acquire current time of day from time of day services in the cable operator's data network
- Enable the PS to assign network address leases to LAN IP Devices and IPCable2Home Host devices
- Enable the PS to assign configuration parameters to LAN IP Devices and IPCable2Home Host devices

7.1.2 Assumptions

The Provisioning Tools operating assumptions are listed below:

- LAN IP Devices and IPCable2Home Host devices implement a DHCP client as defined by RFC 2131.
- The cable network provisioning system implements a DHCP server as defined by RFC 2131.
- If the cable network provisioning system's DHCP server supports DHCP Option 61 (client identifier option), the WAN-Man and all WAN-Data IP interfaces can share a common MAC address.
- LAN IP Devices and IPCable2Home Host devices may support various DHCP Options and BOOTP

Vendor Extensions, allowed by RFC 2132.

- Bulk PS configuration will be accomplished via the download of a PS Configuration File containing one or more parameters, using Trivial File Transfer Protocol (TFTP) [RFC 1350] or Hypertext Transfer Protocol (HTTP) [RFC 2616] with Transport Layer Security (TLS) [RFC 2246].
- The Headend DHCP server will provide a DHCP option, to the WAN-Management interface, which points to a Time of Day server, operating within the Headend network.

7.2 Provisioning Architecture

7.2.1 Provisioning Modes

Three provisioning modes are supported. They are referred to as DHCP Provisioning Mode (DHCP Mode), SNMP Provisioning Mode (SNMP Mode), and Dormant CableHome Mode. The three provisioning modes are compared in Table 7-1.

Table 7-1 Provisioning Modes

	DHCP Mode	SNMP Mode	Dormant CableHome Mode
DHCP Fields and Option Codes	Receives configuration file information in 'siaddr' and 'file' fields. Receives no Option 177.	Receives no configuration file information. Receives valid values for Option 177 sub-options 3, 6, and 51.	Receives no configuration file information and no Option 177, or receives an invalid combination of configuration file information and Option 177 sub-options.
PS Configuration File Trigger	Triggered by presence of TFTP server information in DHCP message	Triggered by NMS via SNMP message	PS receives no configuration file
PS Configuration File Requirement	PS Configuration File download is required	PS Configuration File download is not required	PS configuration file is not required

Specified behavior of the Provisioning Tools is dependent upon the Provisioning Mode in which the PS operates.

Section 13, Provisioning Processes, describes the sequence of events for DHCP and SNMP Provisioning Modes.

7.2.2 Provisioning Architecture Description

The provisioning architecture is illustrated in Figure 7-1. Portal Services elements will interact with server functions in the cable network over the HFC interface, or with IPCable2Home Host Devices to satisfy the system design guidelines listed in Section 7.3.1.

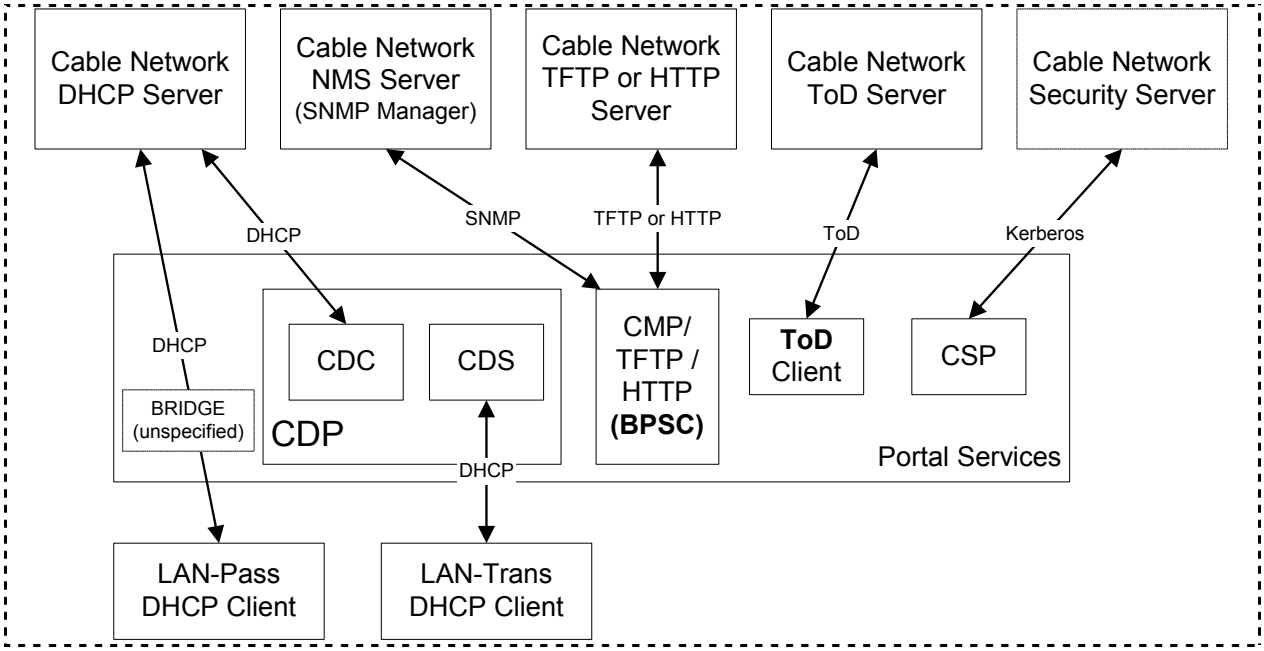

Figure 7-1 Provisioning Architecture

7.3 PS Logical Element - DHCP Portal (CDP)

The IPCable2Home DHCP Portal (CDP) is a logical sub-element of the PS logical element. The CDP has two primary roles: acquisition of network address leases for the PS and assignment of network address leases to LAN IP Devices and IPCable2Home Host devices in the LAN, and is one of the three provisioning tools introduced in Section 7.1. This section describes the Goals, System Design Guidelines, System Description, and Requirements pertaining to the CDP.

7.3.1 CDP Goals

The goals of the CDP include the following:

- Enable client functions in the PS to communicate with corresponding server functions in the cable data network
- Provide the PS with initial configuration parameters, giving it the ability to further configure itself

7.3.2 CDP System Design Guidelines

The following design guidelines drive the capabilities defined for the CDP:

Table 7-2 CDP System Design Guidelines

Number	CDP System Design Guidelines
CDP 1	Addressing mechanisms will be operator controlled, and will provide operator knowledge of and accessibility to IPCable2Home network elements and LAN IP Devices.
CDP 2	Address acquisition and management processes will not require human intervention (assuming that a user/household account has already been established).
CDP 3	Address acquisition and management will be scalable to support the expected increase in the number of LAN IP devices.
CDP 4	It is preferable for LAN IP Device addresses to remain the same after events such as a power cycle or Internet Service Provider switch.
CDP 5	Provide a mechanism by which the number of LAN IP Devices in the LAN-Trans realm can be monitored and controlled.
CDP 6	In-home communication will continue to work as provisioned during periods of Headend address server outage. Addressing support will be provided for newly added LAN IP Devices and address expirations during remote address server outages.
CDP 7	IP addresses will be conserved when possible (both globally routable addresses and private cable network management addresses).

7.3.3 IPCable2Home DHCP Portal System Description

The IPCable2Home DHCP Portal (CDP) is the logical entity that is responsible for addressing activities. The CDP address request and address allocation responsibilities within the IPCable2Home environment include:

- IP address assignment, IP address maintenance, and the delivery of configuration parameters (via DHCP) to LAN IP Devices in the LAN-Trans Address Realm.
- Acquisition of a WAN-Man and zero or more WAN-Data IP addresses and associated DHCP configuration parameters for the Portal Services (PS) element.
- Provide information to the IPCable2Home Name Portal (CNP) in support of LAN IP Device host name services.

The PS maintains two hardware addresses, one of which is to be used to acquire an IP address for management purpose, the other could be used for the acquisition of one or more IP address(es) for data. To prevent hardware address deception, the PS does not allow either of the two hardware addresses to be modified.

The Portal Services element requires an IP Address on the home LAN for its role on the LAN as a router (see Section 8, Packet Handling and Address Translation), DHCP Server (CDS), and DNS Server (see Section 9, Name Resolution). For each of these three Portal Service Element server and router functions, a LAN IP address is saved in the PS database. Each can be accessed via a different MIB object, which are listed below and in Table 7-2.

Router (default gateway) Address	cabhCdpServerRouter
Domain Name Server (DNS) Address	cabhCdpServerDnsAddress

Dynamic Host Configuration Server (DHCP) (CDS) Address cabhCdpServerDhcpAddress

The default value of cabhCdpServerRouter is 192.168.0.1. The default values of cabhCdpServerDnsAddress and cabhCdpServerDhcpAddress are also equal to 192.168.0.1. Any of these three CDP MIB objects can be changed without affecting the other two.

As shown in Figure 7-2, the CDP capabilities are embodied by two functional elements residing within the CDP:

- IPCable2Home DHCP Server (CDS)
- IPCable2Home DHCP Client (CDC)

Figure 7-2 also illustrates interaction between the CDP components and the address realms introduced in Section 5. The CDC exchanges DHCP messages with the DHCP server in the cable network (WAN Management address realm) to acquire an IP address and DHCP options for the PS, for management purposes. The CDC could also exchange DHCP messages with the DHCP server in the cable network (WAN Data address realm) to acquire zero (0), or more IP address(es) on behalf of LAN IP Devices in the LAN-Trans realm. The CDS exchanges DHCP messages with LAN IP Devices in the LAN-Trans realm, and assigns private IP addresses, grants leases to, and could provide DHCP options to DHCP clients within those LAN IP Devices.

LAN IP Devices in the LAN-Pass realm receive their IP addresses, leases, and DHCP options directly from the DHCP server in the cable network. The CDP bridges DHCP messages between the DHCP server in the cable network, and LAN IP Devices in the LAN-Pass realm.

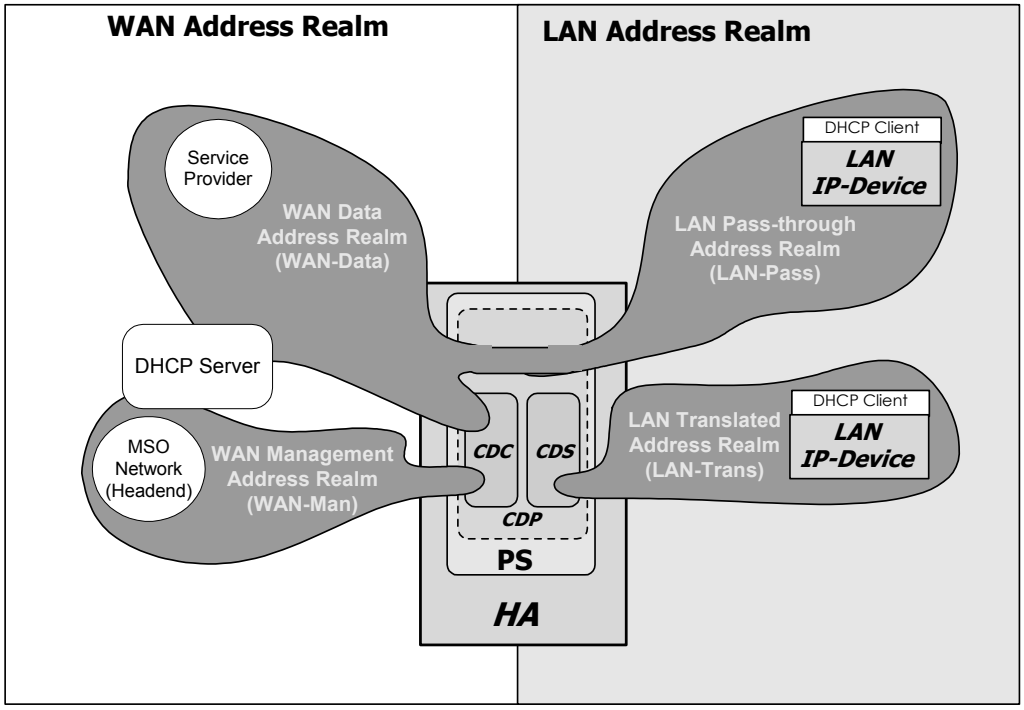

Figure 7-2 CDP Functions

7.3.3.1 DHCP Server (CDS) Sub-element

The CDS is a sub-element of the CDP logical element of the PS, and is the function responsible for allocating network address leases to LAN IP Devices in the LAN-Trans realm. It is also responsible for providing LAN IP Devices with configuration information via DHCP Option codes, as specified in RFC 2132. The CDS is required to perform this function whether or not the PS has an active WAN connection.

7.3.3.1.1 CDS Function Goals

Goals for the CDS Function include the following:

- allocate network address leases to LAN IP Devices in the LAN-Trans realm according to CDP MIB

settings and according to RFC 2131

- allocate configuration information according to RFC 2132
- satisfy goals for operation in the absence of a WAN connection by allocating LAN-Trans IP address leases and providing configuration information to LAN IP Devices upon request as long as the PS is operational, whether or not the PS has an active WAN connection
- do not allocate IP address leases and do not provide configuration information to LAN IP Devices for which the PS has been configured to treat as existing in the LAN-Pass realm

7.3.3.1.2 CDS Function System Design Guidelines

The design guidelines listed in Table 7-3 guided development of the CDS Function specifications.:

Table 7-3 IPCable2Home DHCP Server (CDS) Function System Design Guidelines

Number	CDS Function System Design Guidelines
CDS 1	Provide a means by which LAN IP Devices can acquire network address leases and configuration information for the LAN-Trans realm.
CDS 2	The mechanism for allocating LAN-Trans IP addresses and configuration information will operate whether the PS has a WAN connection to the cable operator's data network or not.
CDS 3	The mechanism for allocating LAN-Trans IP address leases and configuration information will not allocate IP address leases or provide configuration information for LAN IP Devices in the LAN-Pass realm.

7.3.3.1.3 CDS Function System Description

The CDS is a standard DHCP server as defined in RFC 2132, and responsibilities include:

- The CDS assigns addresses to and delivers DHCP configuration parameters to LAN IP Devices receiving an address in the LAN-Trans address realm. The CDS learns DHCP options from the NMS system and provides these DHCP options to LAN IP Devices. If DHCP options have not been provided by the NMS system (for example when the PS boots during a cable outage), the CDS relies on built-in default values (DefVals) for required options.
- The CDS is able to provide DHCP addressing services to LAN IP Devices, independent of the WAN connectivity state.
- The number of addresses supplied by the CDS to LAN IP Devices is controllable by the NMS system. The behavior of the CDS when a cable operator settable limit is exceeded is also configurable via the NMS. Possible CDS actions when the limit is exceeded include: (1) assign a LAN-Trans IP address and treat the WAN to LAN CAT interconnection as would normally occur if the limit had not been exceeded; and (2) do not assign an address to requesting LAN IP devices. An address threshold setting of 0 indicates the maximum threshold possible for the LAN-Trans IP address pool defined by the pool "start" (cabhCdpLanPoolStart) and "end" (cabhCdpLanPoolEnd) values.
- In the absence of time of day information from the Time of Day (ToD) server, the CDS uses the PS default starting time of 00:00.0 (midnight) GMT, January 1, 1970, updates the Expire Time for any active leases in the LAN-Trans realm to re-synchronize with DHCP clients in LAN IP Devices, and maintains leases based on that starting point until the PS synchronizes with the Time of Day server in the cable network.
- During the PS Boot process, the CDS remains inactive until activated by the PS.
- If the PS Primary Packet-handling mode (cabhCapPrimaryMode) has been set to Passthrough and the PS provisioning process has completed (as indicated by cabhPsDevProvState = pass(1)), then the CDS is disabled.

LAN IP Devices may receive addresses that reside in the LAN-Pass realm. As shown in Figure 7-2, LAN-Pass address requests are served by the WAN addressing infrastructure, not the PS. LAN-Pass addressing processes will occur when the PS is configured to operate in Passthrough Mode or Mixed Bridging/Routing Mode (see Section

8.3.4.3 Passthrough Requirements for more details). In these cases, DHCP interactions will take place directly between LAN IP Devices and cable data network servers, and this Recommendation does not specify the process.

Throughout this document, the terms **Dynamic Allocation and Manual Allocation** are used as defined in RFC 2132. The **CDS Provisioned DHCP Options**, cabhCdpServer objects in the CDP MIB, are DHCP Options that can be provisioned by the NMS, and are offered by the CDS to LAN IP devices assigned a LAN-Trans address. CDS Provisioned DHCP Options, cabhCdpServer objects, persist after a PS power cycle and the NMS system can establish, read, write and delete these objects. CDS Provisioned DHCP Options, cabhCdpServer objects, are retained during periods of cable outage and these objects are offered to LAN IP devices assigned a LAN-Trans address during periods of cable outage. The CDC persistent storage of DHCP options is consistent with RFC 2132, Section 2.1. The default values of CDS Provisioned DHCP Options, cabhCdpServer objects, are defined (Table 7-4) and the NMS can reset the CDS Provisioned DHCP Options, cabhCdpServer objects, and cabhCdpLanAddrTable to their default values, by writing to the cabhCdpSetToFactory MIB object.

The CDS Address Threshold (cabhCdpLanTrans) objects contain the event control parameters used by the CDS to signal the CMP to generate a notification to the Headend management system, when the number of LAN-Trans addresses assigned by the CDS exceeds the preset threshold.

The Address Count (cabhCdpLanTransCurCount) object is a value indicating the number of LAN-Trans addresses assigned by the CDS that have active DHCP leases.

The Address Threshold (cabhCdpLanTransThreshold) object is a value indicating when a notification is generated to the Headend management system. The notification is generated when the CDS assigns an address to the LAN IP Device that causes the Address Count (cabhCdpLanTransCurCount) to exceed the Address Threshold (cabhCdpLanTransThreshold).

The Threshold Exceeded Action (cabhCdpLanTransAction) is the action taken by the CDS while the Address Count (cabhCdpLanTransCurCount) exceeds the Address Threshold (cabhCdpLanTransThreshold). If the Threshold Exceeded Action (cabhCdpLanTransAction) allows address assignments after the count is exceeded, the notification is generated each time an address is assigned. The defined actions are a) assign a LAN-Trans address as normal, and b) do not assign an address to the next requesting LAN IP Device.

The Address Count (cabhCdpLanTransCurCount) continues to be updated during periods of cable outage.

The CDS MIB also contains the Address Pool Start (cabhCdpLanPoolStart) and Address Pool End (cabhCdpLanPoolEnd) parameters. These parameters indicate the range of addresses in the LAN-Trans realm that can be assigned by the CDS to LAN IP Devices.

The CDP LAN Address Table (cabhCdpLanAddrTable) contains the list of parameters associated with addresses allocated to LAN IP Devices with LAN-Trans addresses. These parameters include:

- The Client Identifiers , [RFC 2132], Section 9.14 (cabhCdpLanAddrClientID)
- The LAN IP address assigned to the client (cabhCdpLanAddrIp)
- An indication that the address was allocated either manually (via the CMP) or dynamically (via the CDP) (cabhCdpLanAddrMethod)

The CDS stores LAN IP Device identifying information in the cabhCdpLanAddrClientID MIB object. The CDS uses the value passed in the chaddr field of the DHCP REQUEST message sent by the LAN IP Device for this purpose.

The CDS creates a CDP Table (cabhCdpLanAddrTable) entry when it allocates an IP address to a LAN IP Device. The CDS can create CDP Table (cabhCdpLanAddrTable) entries during periods of cable outage.

The CDP Table (cabhCdpLanAddrTable) maintains a DHCP lease time for each LAN IP Device.

NMS-provisioned CDP Table (cabhCdpLanAddrTable) entries are retained during periods of cable outage and

persist across a PS power-cycle.

7.3.3.1.4 CDS Function Requirements

The PS MUST comply with the Server requirements of RFC 2131, section 4.3.

The PS MUST support Dynamic and Manual address allocation in accordance with RFC 2131, section 1.

PS Manual IP address allocation MUST be supported using CDP MIB's cabhCdpLanAddrTable entries created via the NMS system or PS Configuration file.

In support of Dynamic IP address allocation, the PS MUST be capable of creating, modifying and deleting cabhCdpLanAddrTable entries for devices allocated a LAN-Trans address.

The PS MUST retain Provisioned CDP LAN Address Management Table (cabhCdpLanAddrTable) entries during a cable outage and the entries MUST persist after a PS power cycle. The PS MUST be able to provide DHCP addressing services to LAN IP Devices when enabled by the PS, independent of the WAN connectivity state.

Upon PS reset or re-boot, the PS MUST NOT exchange DHCP messages with LAN IP Devices until the CDS is activated by the PS.

The PS MUST activate the CDS, i.e., the PS MUST begin responding to DHCP DISCOVER and DHCP REQUEST messages received through any PS LAN Interface, in any of the following conditions (see also Figure 13-2 IPCable2Home Provisioning Modes):

- When the PS is operating in DHCP provisioning mode, after the CDC has received a PS WAN-Man IP address lease and the PS has received and properly processed a PS configuration file
- When the PS is operating in SNMP provisioning mode, after the CDC has received a PS WAN-Man IP address lease, has authenticated with the Key Distribution Center (KDC) server, and has successfully enrolled with the NMS
- When the first CDC attempt to acquire a PS WAN-Man IP address lease fails
- When the PS is operating in DHCP provisioning mode and the first attempt to download or to process the PS configuration file fails
- When the PS is operating in SNMP provisioning mode and the attempt to authenticate with the KDC server fails
- When the PS is operating in SNMP provisioning mode and is triggered to download a PS configuration file before CDS operation is initiated, and the first attempt to download or to process the PS configuration file fails

The PS MUST assign a unique, available IP address from the range of addresses beginning with cabhCdpLanPoolStart and ending with cabhCdpLanPoolEnd, to each LAN-IP Device in the LAN-Trans realm that requests an IP address using DHCP, if the number of IP addresses already assigned by the CDS is less than the value of cabhCdpLanTransThreshold.

If the value of cabhCdpLanTransThreshold is 0, the PS MUST treat the threshold as if it has been assigned the largest value possible for the current LAN-Trans IP address pool size (as defined by the LAN-Trans IP address pool start (cabhCdpLanPoolStart) and end (cabhCdpLanPoolEnd) values).

The PS MUST maintain the Address Count parameter (cabhCdpLanTransCurCount) indicating the number of active LAN-Trans address leases granted to LAN IP devices.

The PS MUST increase the Address Count each time a lease for a LAN-Trans address is granted to a LAN IP Device and MUST decrease the Address Count each time a LAN-Trans address is released or a LAN-Trans address lease expires.

The PS MUST compare the Address Count parameter (cabhCdpLanTransCurCount) to the Address Threshold parameter (cabhCdpLanTransThreshold) after assigning a LAN-Trans address. If the Address Count parameter (cabhCdpLanTransCurCount) exceeds the Address Threshold parameter (cabhCdpLanTransThreshold), the PS MUST generate a notification in accordance with the event reporting mechanism defined in Section 6.3.3.2 CMP Event Reporting Function and Annex B. While the Address Count parameter (cabhCdpLanTransCurCount) exceeds the Address Threshold parameter (cabhCdpLanTransThreshold), the PS MUST be capable of the following threshold exceeded actions for the next DHCP DISCOVER from the LAN: assign a LAN-Trans addresses as normal or do not assign an address.

If cabhCdpLanTranCurCount equals or exceeds cabhCdpLanTransThreshold and a LAN IP Device requests and additional IP address lease, the PS MUST take specific action as indicated by the Threshold Exceeded Action (cabhCdpLanTransAction) provisioned parameter.

The PS MUST assign IP addresses and deliver DHCP configuration parameters listed in Table 7-4 for which the CDS has a valid value, only to LAN IP Devices receiving an address in the LAN-Trans address realm.

If the cable operator provisions values for a row in the cabhCdpLanAddrTable, the PS (CDS) MUST offer a lease for (i.e., attempt to assign) the provisioned cabhCdpLanAddrIp IP address, to the LAN IP Device whose hardware address corresponds to the provisioned cabhCdpLanAddrClientID, in response to a DHCP DISCOVER received from that LAN IP Device.

When the CDS assigns an active lease for an IP address to a LAN IP Device, the PS MUST remove that address from the pool of IP addresses available for assignment to LAN IP Devices.

If the CDS receives a lease request from a LAN IP device that it cannot satisfy due to the unavailability of addresses from the IP address pool (defined by cabhCdpLanPoolStart and CabhCdpLanPoolEnd), the PS MUST notify the event in accordance to Annex B and the event reporting mechanism defined in Section 6.3.3.2 CMP Event Reporting Function.

The PS MUST store the value passed in the chaddr field of the DHCP REQUEST message sent by the LAN IP Device when an active lease is created for the LAN IP Device.

The PS MUST support all CDP MIB objects, including all objects in the cabhCdpLanAddrTable, cabhCdpLanPool objects, cabhCdpServer objects, and cabhCdpLanTrans objects.

The CDS function of the PS MUST support the DHCP options indicated as mandatory in the CDS Protocol Support column of Table 7-4 CDS DHCP Options.

The CDS MUST include in DHCP OFFER and DHCP ACK messages it sends to its DHCP clients, the DHCP option code 43 sub-option 101 containing the string "CableHome1.1LAN-Trans" (without the quotation marks) as the sub-option information, *only* in response to DHCP DISCOVER and DHCP REQUEST messages that include DHCP option code 60 containing the string value *"CableHome1.1BP*" (without the quotation marks).

The CDS MUST NOT include DHCP option code 43 sub-option 101 in the DHCP OFFER and DHCP ACK messages it sends to any DHCP client that did not provide the string value "*CableHome1.1BP*" in DHCP option code 60, in its DHCP DISCOVER and DHCP REQUEST messages.

The CDS function of the PS MUST support offering the default values indicated in the CDS Factory Defaults column of Table 7-4 CDS DHCP Options, if the DHCP option has not been provisioned with other values.

If the PS Primary Packet-handling mode (cabhCapPrimaryMode) has been set to Passthrough and the PS provisioning process has completed (as indicated b cabhPsDevProvState = pass(1)), then the CDS function of the PS MUST be disabled.

The CDS function of the PS MUST NOT respond to DHCP messages that are received through any WAN Interface, nor originate DHCP messages from any WAN Interface.

The CDS function of the PS MUST NOT deliver any DHCP option with null value to any LAN IP Device.

The CDS MUST NOT offer a lease for IP address 192.168.0.1, i.e., the CDS MUST NOT transmit a DHCP offer or DHCP Ack message with the value 192.168.0.1 in the yiaddr field.

Table 7-4 CDS DHCP Options

Option Number	Option Function	CDS Protocol Support (M)andatory or (O)ptional	CDS Factory Defaults	MIB Object Name
0	Pad	M	N/A	N/A
255	End	M	N/A	N/A
1	Subnet Mask	M	255.255.255.0	cabhCdpServerSubnetMask
2	Time Offset	M	0	cabhCdpServerTimeOffset
3	Router Option	M	192.168.0.1	cabhCdpServerRouter
6	Domain Name Server	M	192.168.0.1	cabhCdpServerDnsAddress
7	Log Server	M	0.0.0.0	cabhCdpServerSyslogAddress
12	Host Name	M	N/A	N/A
15	Domain Name	M	Null String	cabhCdpServerDomainName
23	Default Time-to-live	M	64	cabhCdpServerTTL
26	Interface MTU	M	N/A	cabhCdpServerInterfaceMTU
43	Vendor Specific Information	M	Vendor Selected	cabhCdpServerVendorSpecific
43.101	Vendor Specific Information sub-option 101	M[1]	String: "CableHome 1.1 LAN-Trans"	N/A
50	Requested IP Address	M	N/A	N/A
51	IP Address Lease Time	M	3600 seconds	cabhCdpServerLeaseTime
54	Server Identifier	M	192.168.0.1	cabhCdpServerDhcpAddress
55	Parameter Request List	M	N/A	N/A
60	Vendor Class Identifier	M	N/A	N/A
61	Client-identifier	O	N/A	N/A

7.3.3.2 CDP DHCP Client (CDC) Function

7.3.3.2.1 CDC Function Goals

The goals of the CDP CDC Function include the following:

- acquire an IP address lease for the PS IP stack, used for management messaging and file transfer between the cable operator's network servers and the PS
- acquire configuration information from the cable operator's network DHCP server
- determine the Provisioning Mode in which the PS is to operate
- acquire one or more IP address lease(s) for mapping to LAN IP Devices in the LAN-Trans realm

7.3.3.2.2 CDC Function System Design Guidelines

The guidelines listed in Table 7-5 were used to guide specification of the CDC function:

[1]The CDS is required to include DHCP option code 43 sub-option 101 in the DHCP OFFER and DHCP ACK messages it sends to CableHome compliant LAN IP Devices *only*. CableHome compliance of LAN IP Devices is indicated by the presence of the string *CableHome1.1BP* in the DHCP DISCOVER and DHCP REQUEST messages.

Table 7-5 IPCable2Home DHCP Client (CDC) Function System Design Guidelines

Number	CDC Function System Design Guidelines
CDC 1	Provide a means by which the PS can acquire a network address lease and configuration information for its WAN-Man interface.
CDC 2	Provide a means by which the PS can acquire one or more network address leases and configuration information for its WAN-Data interface.
CDC 3	The mechanism for allocating LAN-Trans IP address leases and configuration information will not allocate IP address leases or provide configuration information for LAN IP Devices in the LAN-Pass realm.

7.3.3.2.3 CDC Function System Description

The CDC is a standard DHCP client as defined in RFC 2131, and responsibilities include:

- The CDC makes requests to Headend DHCP servers for the acquisition of addresses in the WAN- Man and may make requests to Headend DHCP servers for the acquisition of addresses in the WAN-Data address realms. The CDC also understands and acts upon a number of DHCP configuration parameters.
- The CDC makes a determination about which Provisioning Mode the PS is to operate in, based on information received in the DHCP ACKNOWLEDGE message from its DHCP server.
- The CDC supports acquisition of one WAN-Man IP address and zero or more WAN-Data IP addresses.
- The CDC supports the Vendor Class Identifier Option (DHCP option 60), the Vendor Specific Information option (DHCP Option 43), and the Client Identifier Option (DHCP option 61).
- In the default case, the CDC will acquire a single IP address for simultaneous use by the WAN-Man and WAN-Data IP interfaces. In order to minimize changes needed to existing Headend DHCP servers, the use of a Client Identifier (DHCP option 61) by the CDC is not required in this default case.

The CDP supports various DHCP Options and BOOTP Vendor Extensions, allowed by RFC 2132.

The CDC determines the provisioning mode in which the PS is to operate based upon information received from the DHCP server in the DHCP ACK message, as introduced in Section 5.5 IPCable2Home Operational Models.

DHCP Provisioning Mode of Operation:

The PS operates in DHCP provisioning mode if it receives a valid file name for the PS Configuration File in the *file* field and a valid IP address in the *siaddr* field of the DHCPACK message, and *does not* receive DHCP option 177 sub-options 3, 6, or 51.

Behavior of the PS when operating in DHCP Provisioning Mode is summarized below:

- requires a PS configuration file to be downloaded from a cable network file server
- defaults to using SNMPv1 and SNMPv2c for management messaging
- defaults to using the docsDevNmAccessTable of the DOCSIS Device MIB [RFC 2669] to control access to the PS Database via specified MIBs
- can be configured to use Transport Layer Security (TLS) [RFC 2246] to authenticate and encrypt the PS Configuration File (ref.: Section 11.9 PS Configuration File Security in DHCP Provisioning Mode)
- can be configured to operate in SNMPv3 Coexistence Mode, using Diffie-Hellman key management [RFC 2786], (ref.: Section 6.3.3.1.4.2.2)

SNMP Provisioning Mode of Operation:

The PS operates in SNMP provisioning mode if it receives DHCP option 177 with sub-option fields 3, 6, and 51, *does not* receive a valid file name in the *file* field and *does not* receive a valid IP address in the *siaddr* field of the DHCPACK message.

Behavior of the PS when operating in SNMP Provisioning Mode is summarized below:

- is not required to download a PS configuration file from the cable network file server. The PS can be triggered to download a PS configuration file at any time but will operate using factory default parameters without downloading a PS configuration file
- defaults to operating in SNMPv3 Coexistence Mode with SNMPv1 and SNMPv2 support *not* enabled (ref.: Section 11.4 Secure Management Messaging to the PS)
- defaults to using the User-based Security Model of SNMPv3 [RFC 3414] and View-based Access Control Model of SNMPv3 [RFC 3415] to control access to the PS Database via -specified MIBs (ref.: Section 11.4)
- uses Kerberos message exchanges with a Key Distribution Center server whose IP address is provided to the PS in DHCP Option 177 sub-option 51, and AP listener to authenticate SNMPv3 messages (ref.: Section 11.4.4.2 Security Algorithms for SNMPv3 in SNMP Provisioning Mode)
- can be configured to receive and process SNMPv1 and SNMPv2c messages as well as SNMPv3 messages

Dormant CableHome Mode:

The PS operates in Dormant CableHome Mode if it receives neither the combination of *file* field, *siaddr* field, or DHCP option code 177 sub-options to configure it for DHCP Provisioning Mode, nor the combination of these fields and sub-options to configure it for SNMP Provisioning Mode.

When the PS is operating in Dormant CableHome Mode, its behavior is required to be as described in Section 7.3.3.2.4, including the following. This mode of operation is designed to enable the PS to operate and perform residential gateway functions when connected to a cable data network that does not yet support CableHome provisioning and management systems:

- reject any SNMP messages received through any WAN interface
- disable the TFTP client function
- disable SYSLOG event reporting
- terminate the provisioning timer
- enable CNP, CAP, USFS, and CDS functionality

The PS is required to include certain DHCP option fields in DHCP DISCOVER and DHCP REQUEST messages it issues to cable network DHCP servers. The Vendor Class Identifier Option (DHCP option 60) defines a CableLabs device class. For this Recommendation, the Vendor Class Identifier Option will contain the string “CableHome1.1”, to identify a -compliant Portal Services (PS) logical element, whenever the CDC requests a WAN-Man or WAN-Data address.

The Vendor Specific Information option (DHCP Option 43) further identifies the type of device and its capabilities. It describes the type of component that is making the request (embedded or standalone, CM or PS), the components that are contained in the device (CM, MTA, PS, etc.), the device serial number, and also allows device specific parameters.

Details of the requirements for supporting DHCP options 60 and 43 are in Table 7-6 and Table 7-7. Details related to other optional and mandatory DHCP options are provided in Table 7-8.

The WAN-Data IP Address count parameter of the CDP MIB (cabhCdpWanDataIpAddrCount) is the number of IP address leases the CDC is required to attempt to acquire for the WAN side of NAT and NAPT mappings. The default value of cabhCdpWanDataIpAddrCount is zero, which means that, by default, the CDC will acquire only a WAN-Man IP address.

7.3.3.2.3.1 e DHCP Client Option 61

The PS element can have one or more WAN IP addresses associated with a one or more link layer (e.g. MAC) interfaces. Therefore, the CDC cannot rely solely on a MAC address as a unique client identifier value.

This Recommendation allows for the use of the Client Identifier Option (DHCP option 61), [RFC 2132] section 9.14, to uniquely identify the logical WAN interface associated with a particular IP address.

The PS is required to have two hardware addresses: one to be used to uniquely identify the logical WAN interface associated with the WAN-Man IP address (WAN-Man hardware address) and the other to be used to uniquely identify the logical WAN interface associated with WAN-Data IP addresses (WAN-Data hardware address).

7.3.3.2.3.2 WAN Address Modes

In order to enable compatibility with as many cable operator provisioning systems as possible, the CDC will support the following configurable WAN Address Modes:

WAN Address Mode 0:

The PS Element makes use of a single WAN IP Address, acquired via DHCP using the WAN-Man hardware address. The PS Element has one WAN-Man IP Interface and zero WAN-Data IP Interfaces. This Address Mode is only applicable when the PS Primary Packet-handling Mode (cabhCapPrimaryMode) is set to Passthrough (refer to Section 8.3.2). The cable operator's Headend DHCP server typically needs no software modifications to support this Address Mode. In WAN Address Mode 0, the value of cabhCdpWanDataIpAddrCount is zero.

WAN Address Mode 1:

The PS Element makes use of a single WAN IP Address, acquired via DHCP using the WAN-Man hardware address. The PS Element has one WAN-Man IP Interface and one WAN-Data IP Interface. These two Interfaces share a single, common IP address. This Address Mode is only applicable when the PS Primary Packet-handling Mode (cabhCapPrimaryMode) is set to NAPT. The cable operator's Headend DHCP server typically needs no software modifications to support this Address Mode. In WAN Address Mode 1, the value of cabhCdpWanDataIpAddrCount is zero.

WAN Address Mode 2:

The PS Element acquires a WAN-Man IP address using the unique WAN-Man hardware address, and is subsequently configured by the NMS to request one or more unique WAN-Data IP Address(es). The PS Element will have one WAN-Man and one or more WAN-Data IP Interface(s). All WAN-Data IP addresses will share a common hardware address that is unique from the WAN-Man hardware address. The two or more Interfaces (one WAN-Man and one or more WAN-Data) each has its own, unshared IP address. The CDP is configured by the cable operator to operate in WAN Address Mode 2 by writing a nonzero value to cabhCdpWanDataIpAddrCount, via the PS Configuration File or an SNMP set-request. This Address Mode is applicable when the PS Primary Packet-handling Mode (cabhCapPrimaryMode) is set to NAPT or NAT. The cable operator's Headend DHCP server might need software modification to include support for Client IDs (DHCP Option 61) so that it can assign multiple IP addresses to the single WAN-Data hardware address.

There are four potential scenarios for WAN-Data IP addresses:

1. The PS is configured to request zero WAN-Data IP addresses. No WAN-Data Client IDs are needed.

2. The PS is configured to request one or more WAN-Data IP addresses and there are no MSO-configured cabhCdpWanDataAddrClientId entries in the CDP MIB. The PS is required to auto-generate as many unique WAN-Data Client IDs as the value of cabhCdpWanDataIpAddrCount.

3. The PS is configured to request one or more WAN-Data IP addresses and there are at least as many MSO-configured cabhCdpWanDataAddrClientId entries as the value of cabhCdpWanDataIpAddrCount, i.e., the MSO has provisioned enough WAN-Data Client ID values. The PS does not auto-generate any Client IDs.

4. The PS is configured to request one or more WAN-Data IP addresses and there are fewer MSO-configured cabhCdpWanDataAddrClientId entries than the value of cabhCdpWanDataIpAddrCount, i.e., the MSO has provisioned some but not provisioned enough WAN-Data Client ID values. The PS is required to auto-generate enough additional unique WAN-Data Client IDs to bring the total number of unique WAN-Data Client IDs to the value of cabhCdpWanDataIpAddrCount.

If the cable operator desires for the PS to acquire one or more WAN-Data IP addresses, that are distinct from the WAN-Man IP address, the procedure is as follows:

For all WAN Address Modes, the PS first requests a WAN-Man IP address using the WAN-Man hardware address.

The procedure described below assumes the PS has already acquired a WAN-Man IP address:

1. The cable operator optionally provisions the PS with unique specific Client IDs, by writing values to the cabhCdpWanDataAddrClientId entries of the CDP MIB's cabhCdpWanDataAddrTable, via the PS

Configuration File or SNMP set-request message(s).

2. The cable operator configures the CDP to operate in WAN Address Mode 2 by writing cabhCdpWanDataIpAddrCount to a nonzero value through the PS Configuration File or SNMP set-request message.

3. After the CDP has been configured to operate in WAN Address Mode 2 as described in step 2), the PS checks to see if Client ID values have been provisioned by the NMS as described in step 1). If a number of Client ID values greater than or equal to the value of cabhCdpWanDataIpAddrCount have been provisioned, the PS uses these values in DHCP Option 61 when requesting the WAN-Data IP address(es). If Client ID values have not been provisioned, i.e., if the cabhCdpWanDataAddrClientId entries do not exist, or if the number of Client ID values provisioned is less than the value of cabhCdpWanDataIpAddrCount, the PS generates a number of unique Client ID values such that in combination with the provisioned Client IDs, the total number of unique Client IDs equals the value of cabhCdpWanDataIpAddrCount. The PS generates Client ID values by using the WAN-Data hardware address alone for the first requested WAN-Data IP address, and by concatenating the WAN-Data hardware address with a count that is 8 bits in length for the second and all subsequent WAN-Data IP addresses. If no Client IDs have been provisioned by the NMS, the first 8-bit count value is 0x02 (indicating the second requested WAN-Data IP address), the second count value is 0x03, and so on.

Example for the case when no Client IDs have been provisioned by the NMS:

Given WAN-Data hardware address 0xCDCDCDCDCDCD

PS-generated Client ID for the first requested WAN-Data IP address: 0xCDCDCDCDCDCD

PS-generated Client ID for the second requested WAN-Data IP address: 0xCDCDCDCDCDCD02

PS-generated Client ID for the third requested WAN-Data IP address: 0xCDCDCDCDCDCD03

PS-generated Client ID for the nth requested WAN-Data IP address: 0xCDCDCDCDCDCDn (n=<0xFF)

If some Client IDs have been provisioned by the NMS but the number is less than the value of cabhCdpWanDataIpAddrCount, the PS generates additional Client IDs as needed to bring the total number of Client IDs to the value of cabhCdpWanDataIpAddrCount. The PS will generate these additional Client IDs values by appending an 8-bit count value to the WAN-Data hardware address, starting with 0x02, unless that would duplicate a provisioned Client ID. If the Client IDs provisioned by the NMS follow the same format (hardware address with 8-bit count value), the PS is required to use a unique count value so as to not duplicate a provisioned Client ID.

Example for the case when Client IDs have been provisioned by the NMS (three provisioned Client ID values, cabhCdpWanDataIpAddrCount = 5):

Given WAN-Data hardware address 0xCDCDCDCDCDCD

First provisioned Client ID for the first WAN-Data IP address: 0x0A0A0A0A0A1A

Second provisioned Client ID for the second WAN-Data IP address: 0x0A0A0A0A0A2A

Third provisioned Client ID for the third WAN-Data IP address: 0x0A0A0A0A0A3A

First Client ID generated by the PS for the fourth requested WAN-Data IP address: 0xCDCDCDCDCDCD02

Second Client ID generated by the PS for the fifth requested WAN-Data IP address: 0xCDCDCDCDCDCD03

4. The PS adds the Client ID values it generates as cabhCdpWanDataAddrClientId entries to the end of the

cabhCdpWanDataAddrTable.

5. The PS (CDC) requests (repeating the DHCP DISCOVER process as needed) as many unique WAN-Data IP addresses as the value of cabhCdpWanDataIpAddrCount specifies, using the WAN-Data hardware address in the chaddr field of the DHCP message and the Client ID value(s) from step 3) in DHCP Option 61, beginning with the first cabhCdpWanDataAddrClientId entry of the cabhCdpWanDataAddrTable. The CDC is not permitted to request more WAN-Data IP addresses than the value of cabhCdpWanDataIpAddrCount, even if the number of provisioned Client IDs is greater than the value of cabhCdpWanDataAddrTable.

7.3.3.2.4 CDC Requirements

The PS MUST implement a DHCP client function in accordance with the Client requirements of RFC 2131.

The PS MUST implement a TFTP client function in accordance with the Client requirements of RFC 1350.

In both the Embedded and Standalone configurations, the PS MUST implement two unique WAN hardware addresses: the PS WAN-Man hardware address and the PS WAN-Data hardware address. The numerical value of the PS WAN-Data hardware address MUST follow sequentially the numerical value of the PS WAN-Man hardware address. The PS WAN-Man and PS WAN-Data hardware addresses MUST persist once they are set at the factory. The PS MUST NOT permit the modification of its factory-set PS WAN-Man and PS WAN-Data hardware addresses.

In both the Embedded PS and Standalone PS cases, the PS element MUST have WAN interface hardware addresses that are distinct from the cable modem's hardware address.

The PS MUST broadcast DHCP DISCOVER in accordance with client requirements of RFC 2131 and attempt to acquire a PS WAN-Man IP address lease during the PS boot process.

The PS MUST set cabhPsDevProvState to inProgress (2) when the PS broadcasts the DHCP DISCOVER message the first time following device reboot or PS reset. The PS is not required to set cabhPsDevProvState to inProgress(2) when renewing its IP address lease via DHCP.

The PS MUST use the PS WAN-Man hardware address in the *chaddr* field and in DHCP Option 61, in the DHCP DISCOVER and DHCP REQUEST messages, when requesting a WAN-Man IP address from the Headend DHCP server.

If the value of cabhCdpWanDataIpAddrCount is zero, the PS MUST use the WAN-Man IP Address for the WAN-Man and WAN-Data Interfaces.

If the value of cabhCdpWanDataIpAddrCount is greater than zero, the PS MUST request the same number of unique WAN-Data IP address(es) from the Headend DHCP server as the value of cabhCdpWanDataIpAddrCount.

The PS (CDC) MUST NOT attempt to acquire more WAN-Data IP addresses than the value of cabhCdpWanDataIpAddrCount.

The PS MUST use a unique cabhCdpWanDataAddrClientId in DHCP Option 61 for each WAN-Data IP address requested from the Headend DHCP server.

The PS MUST use the WAN-Data hardware address as the value in the DHCP message *chaddr* field for each WAN-Data IP address requested from the Headend DHCP server.

When the PS (CDC) requests WAN-Data IP addresses from the Headend DHCP server, the PS MUST use cabhCdpWanDataAddrClientId entries for DHCP Option 61 in the order the entries appear in the cabhCdpWanDataAddrTable, beginning with the first entry.

If a nonzero value is configured for cabhCdpWanDataIpAddrCount, and if the number of cabhCdpWanDataAddrClientId entries is less than the value of cabhCdpWanDataIpAddrCount, the PS MUST generate as many unique WAN-Data Client IDs as needed to bring the total number of cabhCdpWanDataAddrClientId entries to the value of cabhCdpWanDataIpAddrCount, and add each generated entry to the end of the cabhCdpWanDataAddrTable.

If the PS generates WAN-Data Client IDs, the first cabhCdpWanDataAddrClientId entry of the cabhCdpWanDataAddrTable MUST be the WAN-Data hardware address.

If the PS generates WAN-Data Client IDs, any cabhCdpWanDataAddrClientId entry generated by the PS other than

the first entry of the cabhCdpWanDataAddrTable MUST be the WAN-Data hardware address with an 8-bit count value appended to the end, beginning with 0x02, unless that value already exists as a cabhCdpWanDataAddrClientId entry, in which case the PS MUST generate the Client ID as the WAN-Data hardware address appended with the next available 8-bit count value.

The DHCP Option 43, sub-option 11 is a device specific parameter defined by this Recommendation. It indicates whether an address is being requested in the PS WAN Management or PS WAN Data realm. Table 7-6 indicates how the PS MUST set the values for DHCP Option 43, sub-option 11 for its WAN interfaces.

Table 7-6 DHCP Option 43, Sub-option 11 Values

Element Id	Description & Comments
PS WAN-Man = 0x01	Identifies the request for a WAN-Man realm address.
PS WAN-Data = 0x02	Identifies the request for a WAN-Data realm address

In the case of an Embedded PS with cable modem, the cable modem and PS element each send separate DHCP requests. Table 7-7 describes how the PS MUST set the contents of options 60 and 43 for the PS when the PS element is embedded with a cable modem, and separate PS WAN Management and PS WAN Data addresses are requested.

Table 7-7 DHCP Options for Embedded PS WAN-Man and WAN-Data Address Requests

DHCP Request Options	Value	Description
Embedded Portal Services DHCP Request for WAN Management Address		
CPE Option 60	"CableHome1.1"	
CPE Option 43 sub-option 1	request sub-option vector	List of sub-options (within option 43) to be returned by server. None defined.
CPE Option 43 sub-option 2	"EPS"	Embedded PS
CPE Option 43 sub-option 3	"ECM:EPS"	List of embedded devices (Embedded CM and embedded PS)
CPE Option 43 sub-option 4	e.g.,"123456"	CM/PS Device serial number
CPE Option 43 sub-option 5	e.g., "v3.2.1"	CM/PS Hardware Version Number
CPE Option 43 sub-option 6	e.g., "1.0.2"	CM/PS Software Version Number
CPE Option 43 sub-option 11	PS WAN-Man (0x01)	Defines that an address is being requested in the PS WAN Management realm
CPE Option 43 sub-option 12	e.g., "ABC Inc. CM-PS123..."	CM/PS System Description from sysDescr
CPE Option 43 sub-option 13	e.g., "CM-PS123-1.0.2...."	CM/PS Fireware Rev from docsDevSwCurrentVers
CPE Option 43 sub-option 14	e.g., "1.2.3..."	Firewall Policy File Version from cabhSecFwPolicyFileCurrentVersion
Embedded Portal Services DHCP Request for WAN-Data Address		
CPE Option 60	"CableHome1.1"	
CPE Option 43 sub-option 1	request sub-option vector	List of sub-options (within option 43) to be returned by server. None defined.
CPE Option 43 sub-option 2	"EPS"	Embedded PS
CPE Option 43 sub-option 3	"ECM:EPS"	List of embedded devices (Embedded CM and embedded PS)
CPE Option 43 sub-option 4	e.g.,"123456"	CM/PS Device serial number

CPE Option 43 sub-option 11	PS WAN-Data (0x02)	Defines that an address is being requested in the PS WAN-Data realm

Table 7-8 describes to what the PS MUST set the contents of options 60 and 43, when the PS is a standalone device.

Table 7-8 DHCP Options for Stand-alone PS WAN-Man and WAN-Data Address Requests

DHCP Request Options	Value	Description
Stand-alone Portal Services DHCP Request for WAN Management Address		
CPE Option 60	"CableHome1.1"	
CPE Option 43 sub-option 1	request sub-option vector	List of sub-options (within option 43) to be returned by server. None defined.
CPE Option 43 sub-option 2	"SPS"	Stand-alone PS
CPE Option 43 sub-option 3	"SPS"	List of Embedded devices (Standalone PS only)
CPE Option 43 sub-option 4	e.g., "123456"	Device serial number
CPE Option 43 sub-option 5	e.g., "v3.2.1"	CM/PS Hardware Version Number
CPE Option 43 sub-option 6	e.g., "1.0.2"	CM/PS Software Version Number
CPE Option 43 sub-option 11	PS WAN-Man (0x01)	Defines that an address is being requested in the PS WAN Management realm
CPE Option 43 sub-option 12	e.g., "ABC Inc. CM-PS123..."	CM/PS System Description from sysDescr
CPE Option 43 sub-option 13	e.g., "CM-PS123-1.0.2..."	CM/PS firmware revision from docsDevSwCurrentVers
CPE Option 43 sub-option 14	e.g., "1.2.3..."	Firewall Policy File version from cabhSecFwPolicyFileCurrentVersion
Standalone Portal Services DHCP Request for WAN-Data Address		
CPE Option 60	"CableHome1.1"	
CPE Option 43 sub-option 1	request sub-option vector	List of sub-options (within option 43) to be returned by server. None defined.
CPE Option 43 sub-option 2	"SPS"	Stand-alone PS
CPE Option 43 sub-option 3	"SPS"	List of Embedded devices (Stand-alone PS only)
CPE Option 43 sub-option 4	e.g., "123456"	Device serial number
CPE Option 43 sub-option 11	PS WAN-Data (0x02)	Defines that an address is being requested in the PS WAN-Data realm

For a detailed description of the contents of the PS sysDescr object, see Section 6.3.3.1.4 SNMP Agent Function Requirements.

The PS MUST support the DHCP Options indicated as mandatory in the CDC Protocol Support column in Table 7-9. Table 7-9 lists the DHCP Options that are mandatory and optional for the CDC to support.

Table 7-9 CDC DHCP Options

Option Number	Option Function	CDC Protocol Support (M)andatory
0	Pad	M
255	End	M
1	Subnet Mask	M
2	Time Offset Option	M
3	Router Option	M
4	Time Server Option	M
6	Domain Name Server	M
7	Log Server (syslog)	M
12	Host Name	M
15	Domain Name	M
23	Default Time-to-live	M
26	Interface MTU	M
43	Vendor Specific Information	M
50	Requested IP Address	M
51	IP Address Lease Time	M
54	Server Identifier	M
55	Parameter Request List	M
60	Vendor Class identifier	M
61	Client-identifier	M
177	Suboption 3 - Service Provider's SNMP Entity Address	M
177	Suboption 6 - Kerberos Realm Name of the Provisioning Realm	M
177	Suboption 51 - Kerberos Server IP address	M

The PS MUST include DHCP options listed as mandatory in Table 7-10 in DHCP DISCOVER and DHCP REQUEST messages sent to the cable network DHCP server

Table 7-10 CDC DHCP Options in DISCOVER and REQUEST Messages

Option Number	Option Function	CDC Protocol Inclusion (M)andatory
255	End	M
43	Vendor Specific Information	M
50	Requested IP Address	M
55	Parameter Request List	M
60	Vendor Class Identifier	M
61	Client-identifier	M

The PS MUST request DHCP options listed as mandatory in Table 7-11, within the DHCP Option 55 (Parameter Request List) [RFC 2132] sent in the DHCP DISCOVER and DHCP REQUEST messages.

Table 7-11 CDC DHCP Options Requested within Option 55

Option Number	Option Function	CDC Protocol Inclusion (M)andatory
1	Subnet Mask	M
2	Time Offset Option	M
3	Router Option	M
4	Time Server Option	M
6	Domain Name Server	M
7	Log Server (syslog)	M
15	Domain Name	M
23	Default Time-to-live	M
26	Interface MTU	M
51	IP address Lease Time	M
54	Server Identifier	M
177	PacketCable Compatible Client Configuration Option	M

The PS MUST support a Service Provider's SNMP Entity Address (DHCP Option 177 sub-option 3) configured as an IPv4 address. The format of DHCP Option 177 sub-option 3 is described below:

The sub-option length MUST be 5 octets. The length octet MUST be followed by a single octet that indicates the specific address type that follows. This type octet MUST be set to 1 to indicate an IPv4 address. The type octet MUST be followed by 4 octets of IPv4 address.

Code	Length	Type	Address			
3	5	1	a1	a2	a3	a4

The PS MUST support a Kerberos Realm Name (DHCP Option 177 sub-option 6). A Kerberos realm name is required by the PS to permit a DNS lookup for the address of the service provider's Key Distribution Center (KDC) entity. The format of DHCP Option 177 sub-option 6 is described below:

The realm name MUST be encoded per the domain style realm name described in RFC 1510. The realm name MUST be all capital letters and conform to the syntax described in RFC 1035 section 3.1. The sub-option is encoded as follows:

Code	Length	Kerberos Realm Name			
6	n	k1	k2	. . .	k_n

The PS MUST support a Kerberos server IP address (DHCP Option 177 sub-option 51). The Kerberos server IP address sub-option informs the PS of the network address of one or more Key Distribution Center servers.

The encoding of the KDC Server Address sub-option will adhere to the format of an IPv4 address using the default port. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4. If multiple KDC servers are listed they MUST be listed in decreasing order of priority. The KDC Server Address sub-option is encoded as follows:

Code	Length	Address 1				Address 2		
51	n	a1	a2	a3	a4	a1	a2	. . .

Whenever the first PS WAN-Data interface does not have a current DHCP lease, that first PS WAN-Data interface MUST default to the following IP parameters:

"Fallback" WAN-Data IP address: 192.168.100.5

Netmask: 255.255.255.0

Default Gateway: 192.168.100.1

The purpose for the "Fallback" WAN-Data IP address is to enable access to the cable modem's diagnostic IP address (192.168.100.1) from a LAN IP Device. The "Fallback" WAN-Data IP address MUST only be used as the WAN IP address portion of the Dynamic NAT or NAPT tuple of a C-NAT and C-NAPT address mapping, respectively. If the PS is operating in WAN Address Mode 2 and is required to attempt to acquire multiple WAN-Data IP address leases and the PS is unable to acquire the leases after issuing three DHCP DISCOVER messages (in accordance with DHCP retry procedures specified in Section 7.3.3.2.4, CDC Requirements), the PS MUST use the "Fallback" WAN-Data IP address as the WAN portion of each Dynamic NAT tuple, until the PS acquires the necessary WAN-Data IP address lease(s) from a DHCP server through a PS WAN interface.

The PS MUST NOT use the "Fallback" WAN-Data IP address when the PS is configured to operate in Passthrough Primary Packet-handling mode.

The PS MUST NOT use the "Fallback" WAN-Data IP address for any C-NAT or C-NAPT mappings when the PS has a current PS WAN-Man and PS WAN-Data IP address lease. If a DHCP server on the PS WAN interface offers a lease to the PS (CDC) for the IP address 192.168.100.5, i.e., the same address as the "Fallback" WAN-Data IP address, the PS (CDC) MAY accept the lease and use the address as the WAN-Data IP address for a C-NAT or C-NAPT mapping.

Even when using the 192.168.100.5 default WAN-Data IP address, the PS MUST continue to perform a DHCP DISCOVER every 10 seconds until a valid DHCP lease is granted to that PS WAN-Data interface (or the WAN-Man interface, if the WAN-Man and WAN-data are sharing one IP address).

When a PS is acquiring a WAN-Management IP address for its WAN-Man interface, the PS MUST always insert its WAN hardware address into the Client ID (DHCP option 61) field in the DHCP Discover message.

If during its attempt to acquire a lease for the PS WAN-Man IP address the CDC receives no DHCP OFFER, the PS MUST log Event ID 68000100 in the local log and re-broadcast a DHCP DISCOVER message (i.e., restart the provisioning sequence in the event of this failure condition) - repeating the DHCP lease acquisition attempt up to 5 times. If on its fifth attempt to acquire a PS WAN-Man IP address lease the CDC receives no DHCP OFFER, the PS MUST use the "Fallback" WAN IP address, netmask, and default gateway as described above and continue to attempt to acquire a valid WAN-Man IP address by broadcasting DHCP DISCOVER out its WAN interface every 10 seconds until a valid DHCP lease is granted for the WAN-Man IP address.

If during the process of acquiring a lease for the PS WAN-Man IP address the CDC receives, in the DHCP ACK [RFC 2131] from the DHCP server in the cable network, a valid IP address in the 'siaddr' field and a valid file name in the 'file' field and does not receive DHCP Option 177 sub-option 3, sub-option 6, or sub-option 51 (valid combination 1), the PS MUST set cabhPsDevProvMode to dhcpmode(1) and attempt to synchronize time of day with the ToD server as described in Section 7.5.4 Time of Day Client Function Requirements.

If during the process of acquiring a lease for the PS WAN-Man IP address the CDC receives a DHCP ACK from the DHCP server in the cable network containing DHCP Option 177 with a valid IP address (SNMP Entity's address) in sub-option 3, a valid Kerberos realm name in sub-option 6, and a valid IP address (Kerberos server IP address) in sub-option 51, and does not receive a valid IP address in the 'siaddr' field and does not receive a valid file name in the 'file' field (valid combination 2), the PS MUST set cabhPsDevProvMode to snmpmode(2) and the PS MUST initiate operation of the CDS and attempt to synchronize time of day with the ToD server and to authenticate with the KDC server as described in Section 11.3.4 Authentication Infrastructure Requirements.

If during the process of acquiring a lease for the PS WAN-Man IP address the CDC receives, in the DHCP ACK from the DHCP server in the cable network, any combination of DHCP Option 177 sub-options 3, 6, and 51, 'siaddr' field, and 'file' field other than the two valid combinations described above, the PS has received an invalid DHCP configuration, and the PS MUST log the appropriate event and re-broadcast a DHCP DISCOVER message (i.e., restart the provisioning sequence in the event of this invalid condition) - repeating the entire DHCP lease acquisition process up to 5 times.

If on its fifth attempt to acquire a lease for the PS WAN-Man IP address the CDC receives, in the DHCP ACK from the DHCP server in the cable network, any combination of DHCP Option 177 sub-options 3, 6, and 51, 'siaddr'

field, and 'file' field other than the two valid combinations described above, the PS MUST do the following on the assumption that it is connected via cable modem to a cable data network that does not support IPCable2Home provisioning (Dormant CableHome mode):

- Disable the SNMP agent (CMP) for WAN interface access. Leave the SNMP agent enabled for message received through the LAN interface (i.e., for SNMP messages addressed to the PS Server Router address).
- Disable the TFTP client
- Disable SYSLOG event reporting
- Accept the offered (CPE) IP address lease and use it as the PS WAN-Data address in the CAP Mapping Table, including assigning the address to cabhCdpWanDataAddrIp and populating the other entries of the CDP WAN-Data Address Table (cabhCdpWanDataAddrTable). The PS will be operating without a WAN-Man IP address, which is different from any of the WAN Address Modes described in Section 7.3.3.2.3.2.
- Terminate the provisioning timer
- Set the value of cabhPsDevProvMode to dormantCHmode(3)
- Set the value of cabhPsDevProvState to fail(3)
- Enable the CDS
- Enable the CAP and USFS functionality
- Enable the CNP
- Enable the firewall
- Operate with parameters that have been provisioned in the past, including those values of persistent MIB objects. The PS operating in Dormant CableHome Mode MUST NOT reset its MIB objects to factory default settings.

When a PS operating in WAN Address Mode 2 (as described in Section 7.3.3.2) is acquiring a WAN-Data IP address for a WAN-Data interface that will use an IP address distinct from the WAN-Man interface, the PS MUST include the Client Identifier option (cabhCdpWanDataAddrClientId) in the DHCP Discover message. To enable these unique WAN-Data Client IDs, the CDC MUST enable the NMS system to create cabhCdpWanDataAddrClientId entries in the cabhCdpWanDataAddrTable.

If a PS is operating in WAN Address Mode 2 (as described in Section 7.3.3.2) the PS MUST attempt to obtain an IP address, via DHCP, for each unique client ID (cabhCdpWanDataAddrClientId) in the cabhCdpWanDataAddrTable, up to the limit defined by cabhCdpWanDataIpAddrCount.

The PS MUST continue to retransmit the broadcast DHCP DISCOVER message implementing a randomized exponential backoff algorithm, consistent with that described in RFC 2131, until it acquires a valid PS WAN-Man IP and/or PS WAN-Data IP address lease, as needed.

If the PS (CDC) is successful in acquiring the WAN-Man IP address (i.e., receives a DHCP ACK from a DHCP server via the PS WAN-Man Interface) on its first attempt, and if the PS is operating in DHCP Provisioning Mode, the PS MUST attempt Time of Day time synchronization with the ToD server by issuing a ToD request as described in Section 7.5.4, before attempting to download the PS Configuration File.

If the PS (CDC) is unsuccessful in acquiring the WAN-Man IP address (i.e., the DHCP request times out in accordance with RFC 2131) on its first attempt, the PS MUST trigger the CDS (i.e., initiate CDS operation), so that the CDS can serve DHCP requests from LAN IP Devices in the LAN-Trans realm.

The PS CDC Function MUST only respond to DHCP messages that are received through, or send DHCP messages through, a WAN Interface.

When the WAN-Man DHCP lease expires, the PS MUST clear all row entries from the cabhCdpWanDnsServerTable.

7.4 PS Function - Bulk Portal Services Configuration (BPSC)

7.4.1 Bulk Portal Services Configuration Function Goals

The primary goals of the BPSC function are to request, receive, and process PS and firewall configuration parameters.

7.4.2 Bulk Portal Services Configuration Function System Design Guidelines

The guideline identified in Table 7-12 guided specification of capabilities for the Bulk PS Configuration function:

Table 7-12 Bulk Portal Services System Design Guidelines

Number	Bulk PS Configuration System Design Guidelines
BPSC 1	Provide a mechanism by which the PS can download and process PS and Firewall Configuration Files.

7.4.3 Bulk Portal Services Configuration Function System Description

Bulk Portal Services configuration is typically carried out during the provisioning of the PS element, via the processing of configuration settings contained within a configuration file. However, this process may be initiated at any time. Within this section the term "configuration file" is used to mean either the PS Configuration File or the Firewall Configuration File. Specific requirements for either type of configuration file will be labeled with the appropriate file label, i.e., PS Configuration File or Firewall Configuration File. The Bulk PS Configuration tool consists of the following components:

- The format of the Configuration File
- Modes of triggering the download process
- Means of authenticating the file
- Means of reporting back the status of the configuration file download and other considerations

Bulk PS Configuration (BPSC) is a tool that MSOs can use to change PS and Firewall configuration settings in bulk, via a Configuration File. Typically, the Configuration File will contain many settings, since the primary usefulness afforded by Configuration Files use is the ability to change a number of configuration settings with minimal cable operator intervention. However, it is expected that the Firewall Configuration File will only be used for firewall-specific settings.

The Bulk PS Configuration process can behave the same as successive SNMP sets executed by an operator manually. The Configuration File is a tool meant to make operators more productive and to make large configuration changes less error prone.

It is significant to note that a PS operating in SNMP Provisioning Mode does not need a PS Configuration File loaded before it can operate. It is expected that a PS operating in SNMP Provisioning Mode will initialize itself to a known state and a PS could run for a lifetime without having a PS Configuration File loaded. However, a PS will accept and process a PS Configuration File when one is provided.

7.4.4 Bulk Portal Services Configuration Function Requirements

A PS operating in DHCP Provisioning Mode MUST download and process a PS Configuration File.

A PS operating in SNMP Provisioning Mode MUST be capable of operating without a PS Configuration File, but MUST be capable of downloading and processing a PS Configuration File if triggered as described in Section 7.3.3.2. The PS is not required to download a Firewall Configuration File in either DHCP or SNMP Provisioning Mode.

MIB object settings passed in the PS Configuration File take precedence over and MUST over-write existing MIB object settings.

7.4.4.1 Configuration File Format Requirements

PS or firewall configuration data MUST be contained in a file, which is downloaded via TFTP or HTTPS. The

Configuration File MUST consist of a number of configuration settings (1 per parameter), each of the form "Type Length Value (TLV)". Definitions of these terms are provided in Table 7-13.

Table 7-13 TLV Definitions

Type	A single-octet identifier which defines the parameter
Length	A two-octet field specifying the length of the Value field (not including Type and Length fields)
Value	A set of octets Length long containing the specific value for the parameter

The configuration settings MUST follow each other directly in the file, which is a stream of octets (no record markers). The PS MUST be capable or properly receiving and processing a configuration file that is padded to an integral number of 32-bit words, and be able to properly receive and process a configuration file that is not padded to an integral number of 32-bit words. See Section 7.3.3.1.1 for a definition of the pad. Configuration settings are divided into three types:

- Configuration settings which are required to be present
- Additional or optional IPCable2Home-specified configuration settings which MAY be present
- Vendor-specific configuration settings.

A PS or Firewall Configuration File MAY contain many different parameters, but the only parameters that MUST be included in any configuration file are the PS Message Integrity Check (MIC) (Type 53) and the End of Data Marker (Type 255).

To allow uniform management of the PS, the PS MUST support a Configuration File that is up to 64K-bytes long.

Each Portal Services element MUST support configuration parameter Types 0, 9, 10, 21, 28, 32, 33, 34, 38, 43, 53 and 255, which are described in this section. Each TLV parameter in the Firewall Configuration File describes a firewall attribute. Since the IPCable2Home firewall is configured via access to the IPCable2Home Security MIB (ref: Section 11.6.4 Firewall Requirements), a Firewall Configuration File typically includes TLV type 28 configuration settings, which contain SNMP MIB objects. Vendor-specific firewall configuration information is permitted to be passed to the PS in the Firewall Configuration File using the vendor-specific configuration setting type 43 (TLV-43). If the configuration file does not contain the required attributes, the PS MUST reject the file.

The size of the value in the Length field for any configuration parameter included in a IPCable2Home configuration file MUST be 2 octets.

The Length value for each Type described in the TLV descriptions in this section is the actual length in octets of the Value field.

7.4.4.1.1 Pad Configuration Setting

This has no Length or Value fields and is only used following the end of data marker to pad the file to an integral number of 32-bit words.

Type	Length	Value
0	---	---

7.4.4.1.2 Software Upgrade Filename

The filename of the software upgrade file for the IPCable2Home device. The filename is a fully qualified directory-path name. The file is expected to reside on a TFTP server identified in a configuration setting option.

Type	Length	Value
9	Variable	filename

7.4.4.1.3 SNMP Write-Access Control

This object makes it possible to disable SNMP "Set" access to individual MIB objects. Each instance of this object controls access to all of the writeable MIB objects whose Object ID (OID) prefix matches. This object may be repeated to disable access to any number of MIB objects.

Type	Length	Value
10	n	OID prefix plus control flag

Where n is the size of the ASN.1 Basic Encoding Rules [ISO8025] encoding of the OID prefix plus one byte for the control flag.

The control flag may take values:

0 - allow write-access

1 - disallow write-access

Any OID prefix may be used. The Null OID 0.0 may be used to control access to all MIB objects. (The OID 1.3.6.1 will have the same effect.)

When multiple instances of this object are present and overlap, the longest (most specific) prefix has precedence.

Thus, one example might be

someTable disallow write-access

someTable.1.3 allow write-access

This example disallows access to all objects in someTable except for someTable.1.3.

7.4.4.1.4 Software Upgrade TFTP Server

The IP address of the TFTP server, on which the software upgrade file for the IPCable2Home device resides.

Type	Length	Value
21	4	ip1, ip2, ip3, ip4

7.4.4.1.5 SNMP MIB Object with extended Length

This object allows arbitrary SNMP MIB objects to be Set via the TFTP-Registration process, where the value is an SNMP variable binding (VarBind) as defined in RFC 3416. The VarBind is encoded in ASN.1 Basic Encoding Rules, just as it would be if part of an SNMP Set request.

Type	Length	Value
28	Variable	variable binding

The PS MUST treat the variable binding, in a Type 28 TLV, as if it were part of an SNMP Set Request with the following caveats:

- It MUST treat the request as fully authorized (it cannot refuse the request for lack of privilege).
- SNMP Write-Control provisions (see previous section) do not apply.
- No SNMP response is generated by the PS.
- This object MAY be repeated with different VarBinds to "Set" a number of MIB objects. All SNMP Sets in a Configuration File MUST be treated as if simultaneous. Each VarBind MUST be limited to 65535 bytes.

7.4.4.1.6 Manufacturer Code Verification Certificate

The Manufacturer's Code Verification Certificate (M-CVC) for Secure Software Downloading. Refer to Section 11.8.4.4.2 Network Initialization.

Type	Length	Value
32	Variable	Manufacturer CVC (DER-encoded ASN.1)

7.4.4.1.7 Co-signer Code Verification Certificate

The Co-signer's Code Verification Certificate (C-CVC) for Secure Software Downloading. Refer to Section 11.8.4.4.2 Network Initialization.

Type	Length	Value
33	Variable	Co-signer CVC (DER-Encoded ASN.1)

7.4.4.1.8 SNMPv3 Kickstart Value

(ref.: Section C.1.2.8 DOCSIS 1.1 RFI Specification SP-RFIv1.1-I09-020830)

Compliant Portal Services elements MUST understand the following TLV and its sub-elements and be able to kickstart SNMPv3 access to the PS regardless of whether the PS is operating in NmAccess Mode or Coexistence Mode (see Section 6.3.3 CMP System Description and Section 6.3.3.1.4.2 Network Management Mode Requirements).

Type	Length	Value
34	n	Composite

Up to 5 of these objects may be included in the configuration file. Each results in an additional row being added to the usmDHKickstartTable and the usmUserTable and results in an agent public number being generated for those rows.

7.4.4.1.8.1 SNMPv3 Kickstart Security Name

Type	Length	Value
34.1	2-16	UTF8 Encoded security name

For the ASCII character set, the UTF8 and the ASCI I encodings are identical. Normally, this will be specified as one of the IPCable2Home built-in USM users, e.g., "CHAdministrator".

The security name is NOT zero terminated. This is reported in the usmDHKickStartTable as usmDHKickStartSecurityName and in the usmUserTable as usmUserName and usmUserSecurityName.

7.4.4.1.8.2 SNMPv3 Kickstart Manager Public Number

Type	Length	Value
34.2	n	Manager's Diffie-Hellman public number expressed as an octet string.

This number is the Diffie-Hellman public number derived from a privately (by the manager or operator) generated random number and transformed according to RFC 2786. This is reported in the usmDHKickStartTable as usmKickstartMgrPublic. When combined with the object reported in the same row as usmKickstartMyPublic, it can be used to derive the keys in the related row in the usmUserTable.

7.4.4.1.9 SNMP Notification Receiver

(ref: [DOCSIS9])

Type	Length	Value
38	n	Composite

This PS Configuration File element specifies a Network Management Station that will receive notifications from the PS when it is in Coexistence network management mode. This TLV (38) consists of several sub-TLVs inside the TLV configuration file element. Up to 10 of these elements may be included in the PS Configuration File. Section 6.3.3.1.4.6 Mapping TLV Fields Into Created SNMPv3 Table Rows provides detail about how this configuration file element is mapped into SNMPv3 functional tables.

All multi-byte fields of this sub-TLV MUST be placed in the network byte order.

7.4.4.1.9.1 Sub-TLV 38.1 - IP Address of trap receiver

IPv4 address of the trap receiver, in binary.

Type	Length	Value
38.1	4	IP address

7.4.4.1.9.2 Sub-TLV 38.2 - UDP Port number of the trap receiver

UDP Port number of the trap receiver, in binary.

Type	Length	Value
38.2	2	UDP Port

If this sub-TLV is not present in a configuration file, the default value 162 is used.

7.4.4.1.9.3 Sub-TLV 38.3 - Type of trap sent by the PS (Note 2)

Trap type.

Type	Length	Value
38.3	2	Trap type

The PS MUST support the following trap type values:

1 = SNMP v1 trap in an SNMP v1 packet

2 = SNMP v2c trap in an SNMP v2c packet

3 = SNMP inform in an SNMP v2c packet

4 = SNMP v2c trap in an SNMP v3 packet

5 = SNMP inform in an SNMP v3 packet

7.4.4.1.9.4 Sub-TLV 38.4 - Timeout

Timeout, in milliseconds, used for sending SNMP inform messages.

Type	Length	Value
38.4	2	0 - 65535

7.4.4.1.9.5 Sub-TLV 38.5 - Retries

Number of retries when sending an inform, after sending the inform the first time.

Type	Length	Value
38.5	2	0 - 65535

7.4.4.1.9.6 Sub-TLV 38.6 - Notification Filtering Parameters

Type	Length	Value
38.6	n	Filter OID

Where n is the size of the ASN.1-encoded Filter Object Identifier.

Filter OID is an ASN.1-formatted Object Identifier of the snmpTrapOID value that identifies the notifications to be sent to the notification receiver. This notification and all below it will be sent.

If this Sub-TLV is not present, the notification receiver will receive all notifications generated by the SNMP agent.

7.4.4.1.9.7 Sub-TLV 38.7 - Security Name to use when sending SNMP V3 Notification

Type	Length	Value
38.7	2 - 16	UTF8-encoded security name

This sub-TLV is not required for Trap type = 1, 2, or 3. The PS MUST ignore sub-TLV 38.7 if the trap type in sub-TLV 38.3 is 1, 2, or 3. If sub-TLV 38.7 is not supplied for a Trap type of 4 or 5, the PS MUST send the SNMPv3 Notification in the noAuthNoPriv security level using the security name "@PSconfig". (Note 2)

SecurityName

The SNMPv3 Security Name to use when sending an SNMPv3 Notification. Only used if Trap Type is set to 4 or 5. This name MUST be a name specified in a Config File TLV Type 34 as part of the DH Kickstart procedure. The notifications MUST be sent using the Authentication and Privacy Keys calculated by the PS during the DH Kickstart procedure.

Notes:

1. Upon receiving one of these TLV elements, the PS MUST make entries to the following tables in order to cause the desired trap transmission: snmpNotifyTable, snmpTargetAddrTable, snmpTargetParamsTable, snmpNotifyFilterProfileTable, snmpNotifyFilterTable, snmpCommunityTable, usmUserTable, vacmSecurityToGroupTable, vacmAccessTable, and vacmViewTreeFamilyTable

2. Trap Type: The community String for traps in SNMP V1 and V2 packets MUST be "public". The Security Name in traps and informs in SNMP V3 packets where no security name has been specified MUST be "@PSconfig" and in that case the security level MUST be NoAuthNoPriv.

3. Filter OID: SNMP V3 allows the specification of which Trap OID's are to be sent to a trap receiver. The filter OID in the config element specifies the OID of the root of a trap filter sub-tree. All Traps with a Trap OID contained in this trap filter sub-tree MUST be sent to the trap receiver.

4. The PS Configuration File is permitted to also contain TLV MIB elements (TLV-28) that make entries to any of the 10 tables listed in Note 1. The PS MUST ignore TLV MIB elements that use index columns that start with the characters "@PSconfig".

5. The PS MUST process TLV-38 only if the PS has entered SNMP V3 Coexistence Mode during processing of the PS Configuration File.

7.4.4.1.10 Vendor-specific Information

If vendor-specific information is provided to the PS, it MUST be encoded in the vendor-specific information field (VSIF) (code 43) using the Vendor ID field to specify which TLV tuples apply to which vendors' products. The vendor ID MUST be the first sub-TLV embedded inside VSIF. If the first TLV inside the VSIF is not a Vendor ID, the PS Configuration File MUST be ignored.

This configuration setting is permitted to appear in a configuration file multiple times and the same Vendor ID is permitted to appear multiple times. The PS MUST reject the configuration file if more than one Vendor ID Sub-TLV is present inside a single VSIF.

Vendor-specific sub-types are allowed to be added after Type 43.1.

Type	**Length**	**Value**
43	N	vendor-specific settings

Sub-TLV 43.1 - Vendor ID type

Vendor identification specified by the three-byte Organization Unique Identifier of the PS vendor.

Type	**Length**	**Value**
43.1	3	v1, v2, v3

7.4.4.1.11 PS Message Integrity Check (PS MIC)

Type	**Length**	**Value**
53	20	A 160-bit (20 octet) SHA hash

This parameter contains a hash (PS MIC) calculated by a Secure Hash Algorithm (SHA-1), defined in NIST, FIPS PUB 180-1: Secure Hash Standard, April 1995 [FIPS 180-1]. This TLV is only used in the configuration file immediately before the end of data marker.

7.4.4.1.12 End-of-Data Marker

This is a special marker for end of data. It has no Length or Value fields.

Type	Length	Value
255	---	---

7.4.4.2 BPSC Triggering Requirements

Transfer of the configuration file, from the TFTP server or HTTPS server in the cable data network to the PS, is initiated by an event referred to as a trigger. Requirements for triggering the transfer of a PS Configuration File or Firewall Configuration File from the TFTP server or HTTPS server to the PS follow.

The mode of triggering the PS Configuration File download is dependent upon the Provisioning Mode in which the PS is operating. The CMP MUST read the value of cabhPsDevProvMode (see Section 7.3.3.2.4) prior to initiating any PS Configuration File download. The method of triggering for the Firewall Configuration File download is not dependent upon the Provisioning Mode.

7.4.4.2.1 PS Configuration File Download Trigger for DHCP Provisioning Mode

If the PS receives the TFTP or HTTPS server address in the 'siaddr' field and the PS Configuration File name in the 'file' field of the DHCP ACK, the PS MUST combine the server address and PS Configuration File name to form a URL-encoded value and write that value into PSDev MIB object cabhPsDevProvConfigFile. The PS MUST use the following format for the URL-encoded value for the TFTP server IP address and PS Configuration File name:

tftp://IPv4_address_of_the_TFTP_server/full_path_ to_the_PS_Configuration_File/PS_Configuration_File_name

The PS MUST use the following format for the URL-encoded value for the HTTPS server IP address and PS Configuration File name:

https://IPv4_address_of_the_HTTPS_server/full_path_ to_the_PS_Configuration_File/PS_Configuration_File_name

Download of the PS Configuration File, by a PS operating in DHCP Provisioning Mode, is triggered by the presence of the PS Configuration File location (TFTP or HTTPS server IP address) and name in the DHCP message issued to the PS (CDC) by the DHCP server in the cable network. Refer to Section 7.3.3.2.4 CDC Requirements.

If the PS is operating in DHCP Provisioning Mode (as indicated by the value of cabhPsDevProvMode), after the PS (CDC) receives a DHCP ACK from the DHCP server in the cable network, and the IP address in the 'siaddr' field does not match the first IP address in DHCP option 72, then the PS MUST issue a TFTP Get request to the server identified in the DHCP message 'siaddr' field to download the configuration file.

If the PS is operating in DHCP Provisioning Mode (as indicated by the value of cabhPsDevProvMode), after the PS (CDC) receives a DHCP ACK from the DHCP server in the cable network, and the IP address in the ‘siaddr' field matches the first IP address in DHCP option 72, and the cabhPsDevTodSyncStatus MIB object has a value of '1' (ToD access succeeded), then the PS MUST establish a TLS session as defined in Section 11, and issue a HTTP Get request to the server identified in the DHCP message 'siaddr' field, to download the configuration file.

If the PS is operating in DHCP Provisioning Mode (as indicated by the value of cabhPsDevProvMode), after the PS (CDC) receives a DHCP ACK from the DHCP server in the cable network, and the IP address in the ‘siaddr' field matches the first IP address in DHCP option 72, and the cabhPsDevTodSyncStatus MIB object has a value of '2' (ToD access failed), the PS MUST wait until the cabhPsDevTodSyncStatus MIB object has a value of '1' (ToD access succeeded), before establishing a TLS session as defined in Section 11, and issuing an HTTP Get request to the server identified in the DHCP message 'siaddr' field, to download the configuration file.

Modification of cabhPsDevProvConfigFile MUST NOT trigger a PS operating in DHCP Provisioning Mode to download a configuration file. A PS operating in DHCP Provisioning Mode MUST treat cabhPsDevProvConfigFile as a read-only object.

7.4.4.2.2 PS Configuration File Download Trigger for SNMP Provisioning Mode

If the PS is operating in SNMP Provisioning Mode (as indicated by the value of cabhPsDevProvMode), PS Configuration File download MUST NOT occur before completion of the SNMP v3 setup process (refer to Section 11.4 Secure Management Messaging to the PS, for details about the SNMP setup process).

If the PS is operating in SNMP Provisioning Mode (as indicated by the value of cabhPsdevProvMode), the PS element MUST NOT initiate a PS Configuration File download if the cabhPsDevTodSyncStatus MIB object has a

value of '2' (ToD access failed).

Once the PS, operating in SNMP Provisioning Mode (as indicated by the value of cabhPsDevProvMode), issues a TFTP request to download a PS Configuration file (subject to conditions described in other requirements, below), the PS MUST complete the download phase. When the PS (CMP) has successfully downloaded the requested PS Configuration File, it MUST process the file before issuing a TFTP request for another PS Configuration File.

The PS MUST attempt to download and process the configuration file whose name and address are specified in cabhPsDevProvConfigFile when it receives an SNMP Set command for the cabhPsDevProvConfigFile object, if the following conditions are true:

- the PS is operating in SNMP Provisioning Mode
- the cabhPsDevTodSyncStatus MIB object has a value of '1' (ToD access succeeded), and
- cabhPsDevProvConfigFileStatus = idle(1)

The format of cabhPsDevProvConfigFile MUST be a URL- encoded TFTP server IP address and configuration file name.

If the PS (CMP) operating in SNMP Provisioning Mode receives an SNMP set request from the NMS to update the value of cabhPsDevProvConfigFile and cabhPsDevProvConfigFileStatus = busy(2), or if the cabhPsDevProvConfigHash object does not have a valid value, then the PS MUST reject the set request.

7.4.4.2.3 Firewall Configuration File Trigger

The Firewall Configuration File download is triggered when the value used to SET the cabhSecFwPolicyFileURL MIB object, by either the PS Configuration File or by a SNMP SET command, is different than the value of the cabhSecFwPolicySuccessfulFileURL MIB. If the value used to SET the cabhSecFwPolicyFileURL MIB object, by either the PS Configuration File or by a SNMP SET command, is the same as the value of the cabhSecFwPolicySuccessfulFileURL MIB, the Firewall Configuration File download MUST NOT be triggered.

7.4.4.2.4 Post-trigger Operation

Once triggered, the PS MUST use an RFC 1350 compliant TFTP or RFC 2616 HTTP client to download the configuration files.

A signaling mechanism is necessary to inform the management entity that the PS is currently processing a configuration file. The PS Dev MIB object cabhPsDevProvConfigFileStatus is defined to serve as this signaling mechanism.

If a PS is not currently requesting, downloading, or processing a configuration file, it MUST set cabhPsDevProvConfigFileStatus = idle(1). When the PS has issued a TFTP request for a configuration file specified in cabhPsDevProvConfigFile, it MUST set cabhPsDevProvConfigFileStatus = busy(2).When the PS completes the processing of the PS Configuration File, the PS MUST set cabhPsDevProvConfigFileStatus = idle(1).

Once triggered to download a configuration file, the PS element MUST continue to attempt to download the specified configuration file from the specified location until the configuration file is successfully downloaded and the hash successfully computed as described in Section 7.4.4.3 Configuration File Check and SNMP Provisioning Mode Authentication Requirements. The PS MUST use an adaptive timeout for TFTP and HTTPS based on binary exponential backoff as described below, if the first attempt is not successful, until the PS successfully receives the requested file from the server in the cable data network:

- each retry is 2^n second(s) following the previous attempt, where the PS Configuration File Retry Counter or the Firewall Configuration File Retry Counter, n = [0, 1, 2, 3, 4, or 5]
- n = 0 for the first retry, then is incremented by one for each subsequent attempt until n = 5
- if the PS does not successfully acquire the requested PS Configuration File following the attempt with n = 5, n is to be reset to 0 and the PS is to restart the WAN-Man IP address acquisition process via DHCP.
- if the PS does not successfully acquire the requested Firewall Configuration File following the attempt with n = 5, n is to be reset to 0 and the PS is to continue normal operation, i.e., the PS is not to restart the WAN-Man IP address acquisition process.

The PS MUST exchange TFTP and HTTPS messages only through the PS WAN-Man Interface. The PS MUST

reject any configuration file not received through the PS WAN-Man Interface.

When the download of the configuration file is complete and the configuration file is properly authenticated as described in Section 7.4.4.3 PS Configuration File Check and SNMP Provisioning Mode Authentication Requirements, the PS MUST process the TLVs contained within the file as defined below. See Section 7.4.4.4 Configuration File Processing and Status Reporting Requirements, for specifics of error handling and event generation while processing the configuration file.

The PS MUST use parameters extracted from the configuration file to set the managed objects in the PS database. This process is functionally equivalent to an SNMP SET operation, but it does not rely on the user or view-based access permissions. The PS MUST unconditionally update managed objects in the PS database corresponding to recognized OIDs.

The PS MUST translate configuration file TLV-28 elements into a single SNMP PDU containing (n) MIB OID/instance and value components (SNMP varbinds). In accordance with RFC 3416, the single configuration file-generated SNMP PDU will be treated "as if simultaneous" and the PS MUST behave consistently, regardless of the order in which TLV-28 elements appear in the configuration file or SNMP PDU. The single configuration file-generated SNMP PDU requirement is consistent with SNMP PDU packet behaviors received from an SNMP manager: SNMP PDU varbind order does not matter, and there is no defined MAX SNMP PDU limit. Once a single SNMP PDU is constructed, the PS processes the SNMP PDU and determines the PS configuration acceptance/rejection based on the rules for configuration file processing, described in Section 7.4.4.4 PS Configuration File Processing and Status Reporting Requirements. In processing the SNMP PDU, the PS MUST support CreateAndGo for row creation.

The PS MUST update the size of the PS Configuration file in the MIB object cabhPsDevProvConfigFileSize.

The PS MUST update the number of TLVs processed (i.e., the TLVs that are intended to change the PS configuration per their own Value field) and the number of TLVs ignored (i.e., the TLVs intended to change the PS configuration per their own Value fields that are not successful) from a PS Configuration File, in the MIB objects cabhPsDevProvConfigTLVProcessed and cabhPsDevConfigTLV Rejected, respectively[2]. Configuration parameter Types 255 (End-of-Data Marker), 53 (PS MIC), 0 (Pad Configuration Setting), and Type and Length field pairs that encompass sub-TLVs do not specify values in Value fields intended to change PS configuration and thus MUST NOT be counted in the values of cabhPsDevProvConfigTLVProcessed and cabhPsDevConfigTLVRejected.

7.4.4.3 Configuration File Check and SNMP Provisioning Mode Authentication Requirements

The algorithm used to authenticate the configuration file depends upon the provisioning mode in which the PS is operating (see Section 5.5 IPCable2Home Operational Models). The PS supports two provisioning modes: DHCP Provisioning Mode and SNMP Provisioning mode. Two methods of configuration file authentication are supported for DHCP Provisioning Mode, depending upon the information received in the 'siaddr' field of the DHCP ACK message.

The following sections describe the security algorithms and requirements needed to check the configuration file Hash based on the provisioning mode of the PS element. The PS element MUST support both security algorithms specified in Sections 7.4.4.3.1 PS Configuration File Check for DHCP Provisioning Mode and 7.4.4.3.2 PS Configuration File Authentication Algorithm for SNMP Provisioning Mode.

7.4.4.3.1 PS Configuration File Check for DHCP Provisioning Mode

When operating the DHCP Provisioning Mode, the PS will use a hash-based check of the configuration file, or it will authenticate the message in which the file is transferred, depending upon the configuration of the cable operator's provisioning system.

The PS MUST conduct the hash-based configuration file check described below:

1. When the configuration file Generator of the Provisioning System creates a new PS Configuration File or modifies an existing file, the Config File Generator will create a SHA-1 hash of the contents of the PS

[2]Per these definitions a TLV that does not successfully configure the PS is counted twice, once by each of cabhPsDevProvConfigTLVProcessed and cabhPsDevProvConfigTLVRejected. A TLV that successfully configures the PS is counted only by cabhPsDevProvConfigTLVProcessed.

Configuration File, taken as a byte string. The end of data marker and any padding that follow it are not included in the hash calculation.

2. The Config File Generator adds the hash value, calculated in Step 1, to the PS Configuration File as the last TLV setting (immediately before the end of data marker) using a type 53 TLV. The PS Configuration File is then made available to the appropriate TFTP server.

3. The PS element downloads the PS Configuration File.

4. The PS MUST update the cabhPsDevProvConfigHash MIB object with the hash value from the hash TLV created in steps 1 and 2.

5. The PS element MUST compute a SHA-1 hash over the contents of the PS Configuration File excluding the hash TLV (used to configure the cabhPsDevProvConfigHash MIB object), the end of data marker, and any padding that follows. If the computed hash and the value of the cabhPsDevProvConfigHash MIB object are the same, the PS Configuration File integrity is verified and the configuration file MUST be processed; otherwise, the file MUST be rejected.

7.4.4.3.2 PS Configuration File Authentication Algorithm for SNMP Provisioning Mode

The procedure for checking the PS Configuration File Hash by the PS element in SNMP Provisioning Mode follows:

1. When the Config File Generator of the Provisioning System creates a new PS Configuration File or modifies an existing file, the Config File Generator will create a SHA-1 hash of the entire content of the PS Configuration File, taken as a byte string. The end of data marker and any padding that follow it are not included in the hash calculation.

2. The NMS sends the hash value calculated in step 1 to the PS element via SNMP SET. The PS updates its cabhPsDevProvConfigHash MIB object with the new value.

3. The NMS sends the Name and location of the PS Configuration File via SNMP SET. The PS updates its cabhPsDevProvConfigFile MIB object with the new value.

4. The PS element downloads the named file from the configured TFTP server. If the PS Configuration File contains TLV type 53 the PS MUST ignore it.

5. The PS element MUST compute a SHA-1 hash over the contents of the PS Configuration File excluding the TLV 53 if it exists, the end of data marker and any padding that follows. If the computed hash and the value of the cabhPsDevProvConfigHash MIB object are the same, the PS Configuration File integrity is verified and the configuration file MUST be processed; otherwise, the file MUST be rejected.

7.4.4.3.3 Firewall Configuration File Check

The PS is required to use the Firewall Configuration File check on the Firewall Configuration File as described in this section if the file is provided in SNMP Provisioning Mode or DHCP Provisioning Mode without the use of HTTPS/TLS as defined in Section 11.9 PS Configuration File Security in DHCP Provisioning Mode.

If the Firewall Configuration File was downloaded without the use of HTTP/TLS, the PS MUST follow the procedure defined in steps 1) through 5) below to check the integrity of the Firewall Configuration File:

1. The Firewall Configuration File generator will create a SHA-1 hash of the entire contents of the Firewall Configuration File, taken as a byte string.

2. The provisioning system sends the hash value calculated in step 1 to the PS element in one of two ways:
   a) modifies the cabhSec2FwPolicyFileHash MIB object via a type 28 TLV in the PS Configuration File
   b) sends an SNMP Set command to update the cabhSec2FwPolicyHash MIB object

3. The provisioning system sends the name and location of the Firewall Configuration File to trigger the download of the Firewall Configuration File in one of two ways:
   a) modifies the cabhSec2FwPolicyFileURL MIB object via a type 28 TLV in the PS Configuration File
   b) sends an SNMP Set command to update the cabhSec2FwPolicyURL MIB object

4. If the cabhSecFwPolicyFileOperStatus is not inProgress(1) and the value used to SET the cabhSec2FwPolicyFileURL MIB object is different than the value of the cabhSec2FwPolicySuccessfulFileURL MIB, then the PS element MUST immediately download the named file from the configured server.

5. The PS MUST compute a SHA-1 hash over the entire contents of the Firewall Configuration File and compare the computed hash to the hash represented by the value of the cabhSec2FwPolicyFileHash MIB object. If the computed hash and the value of the cabhSec2FwPolicyFileHash MIB object are the same, the integrity of the Firewall Configuration File is verified and the PS MUST use Firewall Configuration File to configure the firewall, otherwise the PS MUST reject the file.

7.4.4.4 Configuration File Processing and Status Reporting Requirements

The PS MUST report configuration file download status and error conditions using the Event Reporting process described in Section 6.3.3.2 CMP Event Reporting Function.

Table 7-14 identifies success and failure modes that might be encountered with PS Configuration File download and processing, and the action that the PS MUST take when it detects these modes.

Table 7-14 Configuration File Processing Success & Failure Modes

Failure Mode	Action
TFTP failed - Get Request sent, no response received	Report an event (Event ID 68000500) and retry TFTP.
TFTP failed - configuration file not found	Report an event (Event ID 68000600) and retry TFTP.
TFTP failed - out of order packets	Report an event (Event ID 68000700) and retry TFTP.
TFTP download failed - exceeded maximum number of retries	Report an event (Event ID 68000900) and reset.
TFTP download successful	Report an event (Event ID 68001000 if TLS was not used or Event ID 68003200 if TLS was used) and begin configuration file check or authentication.
Configuration file fails authentication check	Report an event (Event ID 68000800) and reset. Do not attempt to process the file.
Configuration File is too large	Report an event (Event ID 73040102) and reset. Do not attempt to process the file.
No End Of Data marker	Report an event (Event ID 7340102) and reset. Do not attempt to process the file.
Duplicate TLV-28 OID	Report an event (Event ID 73040102), reject the configuration file, and reset. Preserve all object values that existed before the attempt to process this bad configuration file.
Recognized Type but bad Value or valid TLV-28 OID but bad MIB value	Report an event (Event ID 73040102), reject the configuration file, and reset. Preserve all object values that existed before the attempt to process this bad configuration file.
An unrecognized SNMP OID is encountered	Disregard the subject TLV and report an event (Event ID 73040100). Continue to process the file.
Type field is not valid for PS	Disregard the subject TLV and report an event (Event ID 73040101). Continue to process the file.

Refer to Annex B for a list of events including those listed in Table 7-14 and information about how events are

reported.

7.4.4.4.1 Unsuccessful Configuration File Download Attempt - TFTP or HTTPS Retries Permitted

If the PS Configuration File Retry Counter is less than 5 and the TFTP or HTTPS Get Request times out, the PS Configuration File is not found on the server, or the TFTP or HTTPS Get failed due to out of order packets, the PS MUST initiate operation of the CDS and CNP functions, report the appropriate event, and retry the attempt to download the PS Configuration File, in accordance with the retry algorithm described in Section 7.4.4.2.4 Post-trigger Operation.

If the Firewall Configuration File Retry Counter is less than 5 and the TFTP or HTTP Get Request times out, the Firewall Configuration File is not found on the server, or the TFTP or HTTP Get failed due to out of order packets, the PS MUST continue normal operations, report the appropriate event, and retry the attempt to download the Firewall Configuration File, in accordance with the retry algorithm described in Section 7.4.4.2.4 Post-trigger Operation.

7.4.4.4.2 Unsuccessful Configuration File Download Attempt - TFTP or HTTPS Retries Exhausted

If the PS Configuration File Retry Counter is equal to 5 and the PS has not successfully downloaded the PS Configuration File, the PS MUST report the event identified in Table 7-14, “Configuration File Processing Success & Failure Modes,” on page 130 for indicating failure of the PS Configuration File download process and release its PS WAN-Man IP address in accordance with RFC 2131, and restart the WAN-Man IP address acquisition process via DHCP.

If the Firewall Configuration File Retry Counter is equal to 5 and the PS has not successfully downloaded the PS Configuration File, the PS MUST report the event identified in Table 7-14 Configuration File Processing Modes for indicating failure of the Firewall Configuration File download process and continue normal operations. If the Firewall Configuration File is not successfully downloaded the PS MUST function as it did prior to the failed Firewall Configuration File download attempt.

7.4.4.4.3 Successful PS Configuration File Download

Successful download of the PS Configuration File is defined as complete and correct reception by the PS element the contents of the PS Configuration File within the TFTP timeout period and computation by the PS the hash values for the PS Configuration File with no errors resulting from the computation.

If the PS successfully downloads the PS Configuration File, the PS MUST reset the PS Configuration File Retry Counter to zero and report the event identified for ‘Failure Mode’ TFTP Download Successful in Table 7-14 Configuration File Processing Modes.

7.4.4.4.4 Unsuccessful PS Configuration File Download

If the PS Configuration File fails the Configuration File Check as specified in Section 7.4.4.3 Configuration File Check and SNMP Provisioning Mode Authentication Requirements or in Section 11.9 PS Configuration File Security in DHCP Provisioning Mode, the PS MUST stop the provisioning process, reject the PS Configuration File, report the appropriate event, and restart the WAN-Man IP acquisition process via DHCP.

If the PS Configuration File contains no End-of-Data TLV (TLV-255), no PS MIC TLV (TLV-53), or is too large to process, the PS MUST stop the provisioning process, reject the PS Configuration File, report the appropriate event, and restart the WAN-Man IP address acquisition process via DHCP.

If the PS Configuration File contains duplicate TLV-28 elements (duplicate means two or more SNMP MIB objects have an identical object identifier (OID)), the PS MUST stop the provisioning process, reject the PS Configuration File, report the appropriate event, and restart the WAN-Man IP address acquisition process via DHCP.

If the PS Configuration File contains a recognized Type field but bad Value field or a valid TLV 28 OID but bad MIB value, the PS MUST stop the provisioning process, reject the PS Configuration File, report the appropriate event, and restart the WAN-Man IP address acquisition process via DHCP.

If the PS Configuration File contains an unrecognized Type field or a TLV 28 element with an unrecognized OID,

the PS MUST ignore that TLV, report the appropriate event, and continue processing the PS Configuration File.

7.4.4.4.5 Successful Firewall Configuration File Download

Successful download of the Firewall Configuration File is defined as complete and correct reception of the file by the PS element within the TFTP or HTTPS timeout period and error-free file validation as defined by the integrity check procedure described in Section 7.4.4.3 Configuration File Check and SNMP Provisioning Mode Authentication Requirements . After the PS successfully downloads the Firewall Configuration File, the PS MUST update the cabhSec2FwPolicySuccessfulFileURL MIB with the same value as the cabhSec2FwPolicyFileURL MIB.

If the PS successfully downloads the Firewall Configuration File, the PS MUST reset the Firewall Configuration File Retry Counter to zero and report Event ID 68003200 (ref.: Table B-1 Defined Events for IPCable2Home). After the PS successfully downloads and processes the Firewall Configuration File, the firewall MUST function as configured by the downloaded file.

7.4.4.4.6 Unsuccessful Firewall Configuration File Download

If the Firewall Configuration File fails the Configuration File Check as specified in Section 7.4.4.3 Configuration File Check and SNMP Provisioning Mode Authentication Requirements, the PS MUST continue normal operations, reject the Firewall Configuration File and report the appropriate event identified in Table B-1 Defined Events for IPCable2Home.

If the Firewall Configuration File contains duplicate TLV-28 elements (duplicate means two or more SNMP MIB objects have an identical object identifier (OID)), the PS MUST continue normal operations, reject the Firewall Configuration File and report the appropriate event identified in Table B-1 Defined Events for IPCable2Home.

If the Firewall Configuration File contains a recognized Type field but bad Value field or value TLV-28 OID but bad MIB value, the PS MUST continue normal operations, reject the Firewall Configuration File and report the appropriate event identified in Table B-1 Defined Events for IPCable2Home.

If the Firewall Configuration File contains an unrecognized Type field or a TLV-28 element with an unrecognized OID, the PS MUST ignore that TLV, report the appropriate event identified in Table B-1 Defined Events for IPCable2Home, and continue processing the Firewall Configuration File.

If the download of the Firewall Configuration File fails for any reason, the firewall MUST function as configured prior to the failed download attempt.

7.5 PS Function - Time of Day Client

7.5.1 Time of Day Client Function Goals

The goal of the Time of Day client function of the PS is to acquire the current time of day from the Time of Day server in the cable operator's network.

7.5.2 Time of Day Client Function System Design Guidelines

The guideline identified in Table 7-15 guided specification of the capabilities defined for the PS Time of Day Client function:

Table 7-15 Time of Day Client System Design Guidelines

Number	Time of Day Client System Design Guidelines
TOD 1	Provide a mechanism by which the PS can achieve time synchronization with the Headend network

7.5.3 Time of Day Client Function System Description

The Portal Services element makes use of an [RFC 868 compliant Time of Day client, in order to achieve time synchronization with a time server on the Headend network. Time synchronization is essential for PS security functions as well as event messaging.

When the CDC DHCP client requests an IP Address - from the Headend DHCP server - for the WAN-Man interface, the DHCP client will receive the IP address of the Headend ToD server within DCHP Option 4. The DHCP client will also receive the Time Offset (from UTC), within DHCP Option 2.

Once the WAN-Man IP stack begins use of the IP address it received from DHCP, it should send an [RFC 868 time query to the ToD Server. If the ToD server responds with a valid response, the PS will begin using this time of day for event message time stamps and security functions.

7.5.4 Time of Day Client Function Requirements

The Portal Services element MUST implement a Time of Day Client.

The Portal Services Time of Day Client MUST comply with the Time of Day Protocol [RFC 868] and make use of the UDP Protocol only.

Upon reset, the Portal Services Element MUST initialize its time to 00:00.0 (midnight) GMT, January 1, 1970.

The Portal Services Element MUST attempt Time of Day time synchronization with the Time Servers provided in DHCP Option 4 of the DHCP ACK, received by the WAN-Man interface during WAN-Man DHCP lease acquisition.

If the PS receives DHCP Option 4 (Time Server Option) in the DHCP ACK, the PS MUST save the IP address of the Time Server from which the PS accepted a response as the value of cabhPsDevTimeServerAddr.

The PS MUST combine the time retrieved from the Time Server with the time offset provided by DHCP Option 2, to create the current local time.

The Portal Services Element MUST make use of the current local time calculated from the time retrieved from the ToD server and time offset received by DHCP Option 2 for any functions requiring time of day, and which need only be accurate to the nearest second.

The priority for the system time of day clock for an Embedded PS is as follows:

- First priority: time of day acquired from the ToD server
- Second priority: time of day acquired from the cable modem
- Third priority: time initialized to January 1, 1970

An Embedded PS MUST use the most recent valid time of day acquired from the ToD server for the system time of day clock, even if this means overwriting the system time acquired by the CM.

If an Embedded PS is unable to acquire time of day from the ToD server, it MUST use time of day acquired by the cable modem for the system time of day clock.

If an Embedded PS is unable to acquire time of day from the ToD server, and is unable to acquire valid time of day from the cable modem, it MUST use time of day initialized in the boot process to January 1, 1970 for the system time of day clock.

The priority for the system time of day clock for a Standalone PS is as follows:

- First priority: time of day acquired from the ToD server
- Second priority: time initialized to January 1, 1970

A Standalone PS MUST use the most recent valid time of day acquired from the ToD server for the system time of day clock.

If a Standalone PS is unable to acquire time of day from the ToD server, it MUST use time of day initialized in the boot process to January 1, 1970 for the system time of day clock.

The PS element MUST continue to attempt to communicate with the Time of Day server, until local time is established. The specific timeout for Time of Day Requests is implementation dependent. However, the PS Time of Day client MUST NOT exceed more than 3 ToD requests in any 5 minute period. At minimum, the PS Time of Day client MUST issue at least 1 ToD request per 5 minute period, until local time is established.

If the ToD server does not respond with a valid response the PS MUST do the following, not necessarily in the order listed:

- set the value of cabhPsDevTodSyncStatus to '2' (ToD access failed),
- if there are active leases in the LAN-Trans realm as indicated by a nonzero value for cabhCdpLanTransCurCount, set cabhCdpLanAddrCreateTime to the current time and set cabhCdpLanAddrExpireTime to the value of cabhCdpLanAddrCreateTime plus the value of cabhCdpServerLeaseTime for each active lease (Expire Time = CreateTime + LeaseTime),
- log the failure and generate a standard event defined in Annex B, and
- continue to retry communication with the ToD server until local time is established, and
- if triggered to do so, attempt to download the PS Configuration File as described in Section 7.4.4.2.4 Post-trigger Operation.

If the ToD server does respond with a valid response the PS MUST do the following, not necessarily in the order listed:

- set the value of cabhPsDevTodSyncStatus to '1' (ToD access succeeded),
- if there are active leases in the LAN-Trans realm as indicated by a nonzero value for cabhCdpLanTransCurCount, set cabhCdpLanAddrCreateTime to the current time and set cabhCdpLanAddrExpireTime to the value of cabhCdpLanAddrCreateTime plus the value of cabhCdpServerLeaseTime for each active lease (Expire Time = CreateTime + LeaseTime),
- if triggered to do so, attempt to download the PS Configuration File as described in Section 7.4.4.2.4 Post-trigger Operation.

If the value of cabhPsDevTodSyncStatus is '1', i.e., if local time has already been established, it is not necessary for the Time of Day client to issue a ToD request.

The PS MUST send and receive ToD messages only through a WAN-Man Interface.

7.6 BP Function - DHCP Client

7.6.1 BP DHCP Client Function Goals

The goal of the BP DHCP client function is to acquire an IP address lease and configuration parameters for the BP from the system DHCP server.

7.6.2 BP DHCP Client Function System Design Guidelines

The guideline listed in Table 7-16 guided specification of the BP DHCP Client function.:

Table 7-16 BP DHCP Client Function System Design Guidelines

Number	BP DHCP Client Function System Design Guidelines
BP DHC 1	Provide a means by which the BP can acquire a network address lease and configuration information.

7.6.3 BP DHCP Client Function System Description

The DHCP Client function of the BP is responsible for acquiring an IP address lease from a system DHCP server. The server could be the CDS Function of the CDP sub-element of the PS or it could be a DHCP server in the cable operator's data network, depending upon how the PS packet handling mode is configured. The BP DHCP Client function also acquires configuration information passed in DHCP Option fields from the system DHCP server.

7.6.4 BP DHCP Client Function Requirements

The BP MUST implement a DHCP client function in accordance with the Client requirements of RFC 2131.

Upon reset the BP MUST issue a DHCP DISCOVER broadcast message to acquire an IP address lease.

The BP MUST support the DHCP Options and sub-options indicated as mandatory (M) in Table 7-17.

The BP MUST include the following DHCP option codes, in each DHCP DISCOVER and DHCP REQUEST message it sends:

- DHCP Option code 55 Parameter Request List
- DHCP Option code 60 Vendor Class Identifier, with the string “CableHome1.1BP”
- DHCP Option code 255 End

Table 7-17 BP DHCP Client Required DHCP Options

Option Number	Option Function	Support (M)andatoryor (O)ptional	Factory Default Value
0	Pad	-	N/A
255	End	M	N/A
1	Subnet Mask	M	N/A
2	Time Offset	O	0
3	Router Option	M	N/A
6	Domain Name Server	M	N/A
7	Log Server	M	N/A
12	Host Name	O	N/A
15	Domain Name	M	Null String
23	Default Time-to-live	M	N/A
26	Interface MTU	M	N/A
43	Vendor Specific Information	M	Vendor Selected
50	Requested IP Address	M	null value or vendor selected
51	IP Address Lease Time	M	N/A
54	Server Identifier	M	N/A
55	Parameter Request List	M	N/A
60	Vendor Class Identifier	M	“CableHome1.1BP”
61	Client-identifier	O	N/A

8 PACKET HANDLING & ADDRESS TRANSLATION

8.1 Introduction/Overview

8.1.1 Goals

The key goals which drive the packet handling capabilities include:

- Provide cable friendly address translation functionality, enabling cable operator visibility and manageability of home devices while preserving cable based sourced based routing architectures.
- Prevent unnecessary traffic on the cable and home network.
- Conservation of globally routable public IP addresses as well as cable network private management addresses.
- Facilitate in-home IP traffic routing by assigning network addresses to LAN IP Devices such that they reside on the same logical subnetwork.

8.1.2 Assumptions

- It is assumed that when cable operator provisioning servers provide multiple globally routable IP

addresses to customer devices in a home, these addresses will not necessarily reside on the same subnet.

- Changing Internet service providers is assumed to occur relatively infrequently, occurring at a rate similar to a household changing its primary long distance carrier.

8.2 Architecture

This section describes the key concepts behind the IPCable2Home packet handling and address translation functionality.

8.3 PS Logical Element - IPCable2Home Address Portal (CAP)

The IPCable2Home Address Portal (CAP) is a logical sub-element of the Portal Services logical element. Its functions are to route traffic between the LAN and the WAN, route LAN-to-LAN traffic, and to perform address and port translation functions.

8.3.1 CAP Goals

The goals of the CAP are listed below and in Section 8.1.1:

- Route IP packets between LAN IP Devices, and between LAN IP Devices and the Portal Services' default gateway on the WAN
- Provide Network and Port Address Translation (NAPT) capability for mapping between a single global IP address on the PS WAN Interface and one or more private IP addresses in the LAN
- Provide Network Address Translation (NAT) capability for 1-to-1 mapping between global IP addresses on the PS WAN Interface and private IP addresses on the LAN
- Keep traffic between LAN IP Devices on the LAN and do not permit it to traverse the WAN

8.3.2 CAP System Design Guidelines

The system design guidelines listed in Table 8-1 guided specification of the IPCable2Home Address Portal functionality.

Table 8-1 CAP System Design Guidelines

Number	CAP System Design Guideline
CAP 1	Addressing mechanisms will be operator controlled, and will provide operator knowledge of and accessibility to IPCable2Home devices.
CAP 2	Addressing will do nothing that will compromise current cable network routing architectures (for example source based routing, MPLS).
CAP 3	Traffic management mechanisms will insulate the cable network from traffic generated by in house peer-to-peer communications.
CAP 4	IP Addresses will be conserved when possible (both globally routable addresses and private cable network management addresses).

8.3.3 CAP System Description

Address translation and packet handling functionality is provided by the functional entity known as the IPCable2Home Addressing Portal (CAP). The CAP encompasses the following address translation and packet forwarding elements:

- IPCable2Home Address Translation (CAT)
- IPCable2Home Passthrough Function
- Upstream Selective Forwarding Switch (USFS)

As shown in Figure 8-1, the CAT function provides a mechanism to interconnect the WAN-Data address realm and LAN-Trans address realm (via address translation), while Passthrough provides a mechanism to interconnect the WAN-Data address realm and the LAN-Pass address realm (via bridging). The CAT function is compliant with

Traditional Network Address Translation (NAT) [RFC 3022] section 2. As with Traditional NAT, there are two variations of CAT, referred to as IPCable2Home Network Address Translation (C-NAT) Transparent Routing and IPCable2Home Network Address and Port Translation (C-NAPT) Transparent Routing. C-NAT Transparent Routing is the IPCable2Home compliant version of Basic NAT [RFC 3022] section 2.1 and C-NAPT Transparent Routing is the IPCable2Home compliant version of NAPT [RFC 3022] section 2.2.

Per [RFC 3022], C-NAT transparent routing is "a method by which IP addresses are mapped from one group to another, transparent to end users," and C-NAPT transparent routing "is a method by which many network addresses and their TCP/UDP (Transmission Control Protocol/User Datagram Protocol) ports are translated into a single network address and its TCP/UDP ports." Also, per [RFC 3022], the purpose of C-NAT and C-NAPT functionality is to "provide a mechanism to connect a realm with private addresses to an external realm with globally unique registered addresses."

The IPCable2Home Passthrough function is a IPCable2Home specified bridging process that interconnects the WAN-Data Address Realm and the LAN-Pass Address Realm without address translation.

The Upstream Selective Forwarding Switch (USFS) defines a function within the CAP with the capability of confining home networking traffic to the home network, even when home networking devices generating this traffic reside on different logical IP subnets. Specifically, this function forwards traffic sourced from an IP address in one of the LAN Address realms, destined to IP addresses in one of the LAN Address realms, directly to its destination. This direct forwarding functionality prevents the traffic from traversing the HFC network, and interconnects the LAN-Trans and LAN-Pass Address Realms.

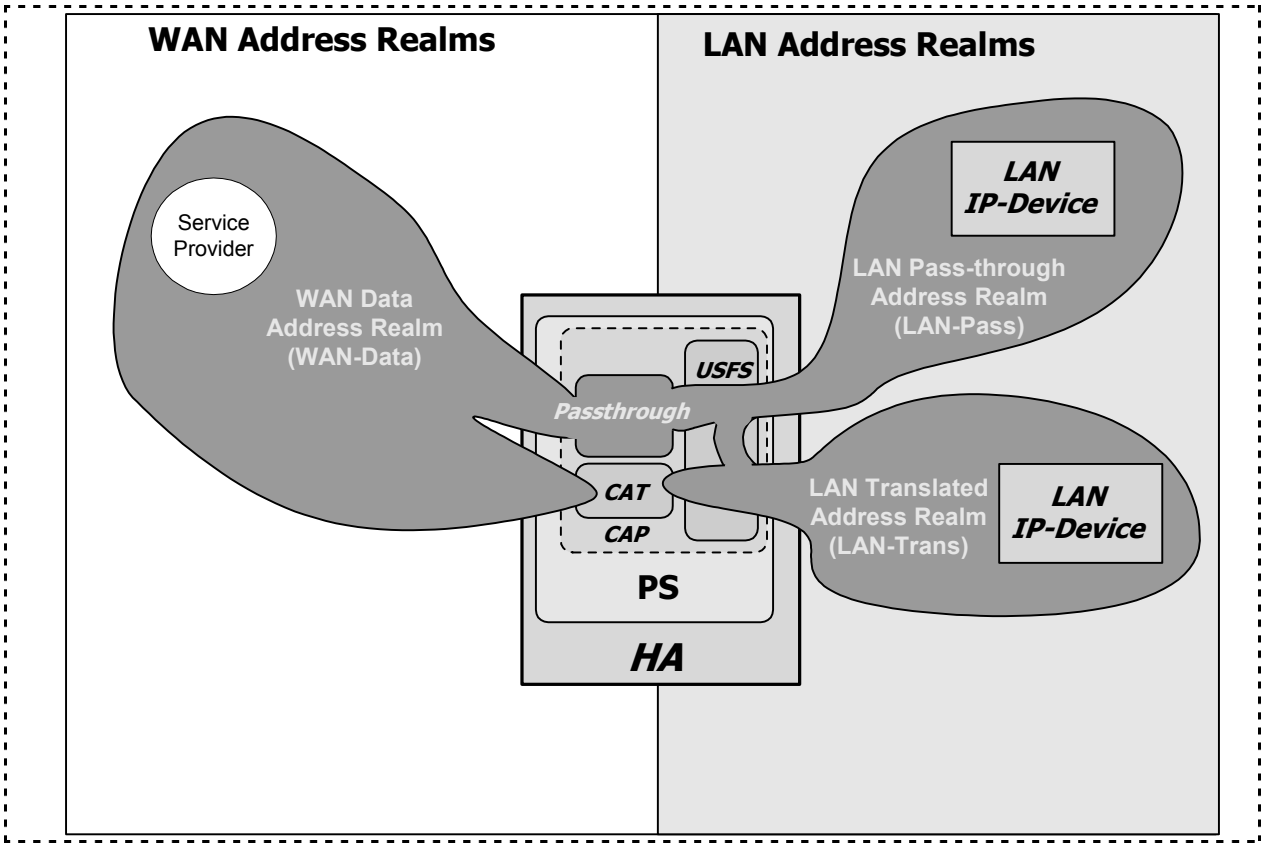

Figure 8-1 IPCable2Home Address Portal (CAP) Functions

Throughout this document, the terms Address Binding, Address Unbinding, Address Translation, and Session are used as defined in [RFC 2663]. In addition, IPCable2Home defines the term Mapping as the information required to perform C-NAT Transparent Routing and C-NAPT Transparent Routing.

In particular, a C-NAT Mapping is defined as a tuple of the form (WAN-Data IP address, LAN-Trans IP address) providing a one-to-one mapping between WAN-Data addresses and LAN-Trans addresses. Similarly, a C-NAPT Mapping is defined as a tuple of the form (WAN-Data IP address and TCP/UDP port, LAN-Trans IP address and TCP/UDP port) providing a one-to-many mapping between a single WAN-Data address and multiple LAN-Trans

addresses. For ICMP traffic (such as ping), an ICMP sequence number is used in place of the TCP/UDP port number.

LAN-to-WAN traffic is defined as packets sourced by LAN IP Devices destined to devices on the WAN side of the PS. WAN-to-LAN traffic is defined packets sourced by WAN hosts destined to LAN IP devices. LAN-to-LAN traffic is defined as packets sourced by LAN IP Devices destined to LAN IP Devices on the same or different subnet.

8.3.3.1 Packet Handling Modes

The Portal Services element is configurable, via the cabhCapPrimaryMode MIB object, to operate in one of three Primary Packet-handling Modes when handling LAN-to-WAN and WAN-to-LAN traffic: Passthrough Mode, C-NAT Transparent Routing Mode, and C-NAPT Transparent Routing Mode. Further, the C-NAT or C-NAPT primary modes may also operate in a Mixed Mode described below.

In Passthrough mode, the CAP acts as a transparent bridge [ISO DIS 10038 MAC Bridges] between the WAN-Data realm and LAN-Pass realm. In Passthrough mode, forwarding decisions are made primarily at OSI Layer 2 (data link layer). In this mode, the CAP does not perform any C-NAT or C-NAPT Transparent Routing functions.

The CAP supports OSI Layer 3 (network layer) forwarding in both the C-NAT Transparent Routing Mode and the C-NAPT Transparent Routing Mode, described below.

In C-NAT Mode, the PS element (CDC) acquires one or more IP addresses used for WAN-Data traffic during the PS boot process. After acquisition, via DHCP, these IP addresses are used as the WAN-Data IP address portion of Dynamically created C-NAT Mapping tuples. These WAN IP addresses make up a pool of addresses available for Dynamically created C- NAT Mappings. If an available IP address exists in the WAN-Data IP address pool, the CAP creates a Dynamic C-NAT Mapping when it first sees LAN-to-WAN IP traffic that does not have an existing Mapping. If no available IP address exists in the WAN-Data IP address pool, the Dynamic C-NAT Mapping can not be created, and this traffic is dropped, and an event is generated (see Annex B).

The LAN-Trans IP address portion of the Dynamically created C-NAT Mapping tuples is provided by the pool of IP addresses defined by the cable operator in the IPCable2Home CDP MIB. The CAP enters the tuple of the unique WAN-Data IP address and a unique LAN-Trans IP address in the CAP Mapping Table, along with other parameters including WAN and LAN Port numbers, the Mapping Method, and the transport protocol used for the Mapping. The port number will not be translated by the CAP for C-NAT Mappings: the source and destination port numbers in the UDP or TCP header will be unchanged. When the PS is operating in NAT primary packet handing mode (cabhCapPrimaryMode = nat(2)), the CAP will enter the value 0 into the WAN and LAN port number entries of the CAP Mapping Table. The CAP will also enter the value 0 into the WAN and LAN port number entries of the CAP Mapping Table for provisioned static port forwarding entries of the CAP Mapping Table when the PS is operating in NAPT primary packet handling mode (cabhCapPrimaryMode = napt(1)). For the case of a static port forwarding entry provisioned in the CAP Mapping Table for a PS operating in NAPT primary packet handling mode, the 0-value port number entry will serve two purposes: (1) indicate to the CAP that the port numbers are not to be translated, i.e., that the ports are “wild carded”, and (2) indicate to anyone reading the CAP Mapping Table that this static port mapping is effectively a C-NAT mapping, thereby providing a distinction between static port forwarding entries (C-NAT mappings) (port number 0) and C-NAPT Mappings (nonzero port number). Refer to Section 8.3.3.2 Static Port Forwarding Wild Cards for more information about static port forwarding operation of the CAP.

Dynamic C-NAT Mappings for UDP traffic are destroyed when an inactivity timeout period, cabhCapUdpTimeWait, expires. Dynamic C-NAT Mappings for TCP traffic are destroyed when an inactivity timeout period, cabhCapTcpTimeWait, expires or a TCP session terminates. Dynamic C-NAT Mappings for ICMP traffic are destroyed when an inactivity timeout period, cabhCapIcmpTimeWait, expires. In addition, Static C-NAT Mappings may be created or destroyed when the NMS system writes to or deletes from the cabhCapMappingTable MIB table.

In C-NAPT Mode (the factory default mode for the system) the PS element (CDC) acquires one IP address, used for WAN-Data traffic. After acquisition, via DHCP, this IP address is used as the WAN-Data IP address portion of Dynamically created C-NAPT Mapping tuples. If the WAN-Data IP address has been acquired, Dynamic C-NAPT Mappings are created when the CAP first sees LAN-to-WAN IP traffic that does not have an existing Mapping. If the WAN-Data IP address has not been acquired (i.e., does not have an active DHCP lease), the Dynamic C-NAPT Mapping can not be created, and this traffic is dropped, and a standard event is generated (see Annex B).

Dynamic C-NAPT Mappings for UDP traffic are destroyed when an inactivity timeout period,

cabhCapUdpTimeWait, expires. Dynamic C-NAPT Mappings for TCP traffic are destroyed when an inactivity timeout period, cabhCapTcpTimeWait, expires or a TCP session terminates. Dynamic C-NAPT Mappings for ICMP traffic are destroyed when an inactivity timeout period, cabhCapIcmpTimeWait, expires. In addition, Static C-NAPT Mappings may be created or destroyed when the NMS system writes to or deletes from the cabhCapMappingTable MIB table.

Figure 8-2 shows a typical Dynamic C-NAPT Mapping process with a TCP packet. In this example, the PS is configured to operate in NAPT mode and already has obtained a WAN IP address, and the LAN IP Device has already obtained an IP in the LAN-Trans realm.

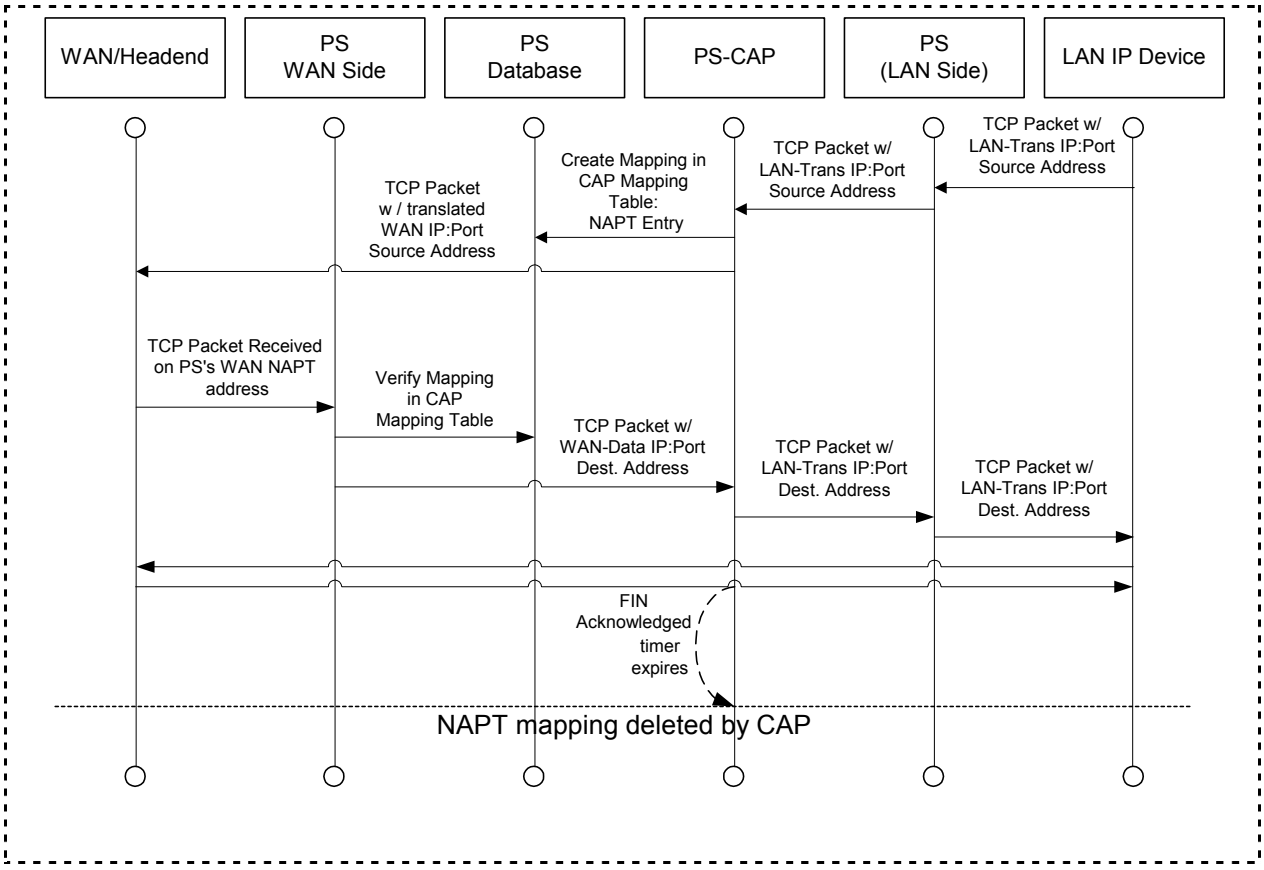

Figure 8-2 PS Configuration (CAP Mapping Table - NAPT) Sequence Diagram

It is also possible for the PS to operate in a Mixed Bridging/Routing Mode. In this case, the NMS sets the primary mode to C-NAT or C-NAPT Transparent Routing, and the NMS writes one or more MAC addresses belonging to LAN IP Devices, whose traffic is to be bridged, into the Passthrough Table (cabhCapPassthroughTable). In this Mixed Mode, the PS examines MAC addresses of received frames to determine whether to transparently bridge the frame or to perform any C-NAT or C-NAPT Transparent Routing functions at the IP layer. In the case of LAN- to-WAN traffic, the PS examines the source MAC address, and if that MAC address exists in the cabhCapPassthroughTable, the frame is transparently bridged to the WAN-Data interface. In the case of WAN- to-LAN traffic, the PS examines the destination MAC address, and if that MAC address exists in the cabhCapPassthroughTable, the frame is transparently bridged to the appropriate LAN interface. If the MAC address does not exist in the cabhCapPassthroughTable, the packet is processed by higher layer functions, including the C-NAT/C-NAPT Transparent Routing function.

It is assumed that when the PS is in Routing mode (C-NAT/C-NAPT), that it will process broadcast traffic in accordance with [RFC 919], [RFC 922], [RFC 1812], and [RFC 2644]. It is also assumed that when the PS is in Passthrough Mode, that broadcast traffic will be bridged to all interfaces.

When the PS is in Mixed Bridging/Routing Mode, and receives broadcast traffic sourced from a device in Passthrough Table, the PS is expected to bridge the broadcast to all interfaces. When the PS is in Mixed Bridging/Routing Mode, and receives broadcast traffic on any WAN interface, the PS is expected to bridge the

broadcast to all LAN interfaces.

It should be noted that the USFS functionality (Section 8.3.3.4) is applied in each of the three primary packet-handling modes, and regardless of whether or not Mixed mode is in use. USFS forwarding decisions will take precedence over other forwarding decisions that could potentially forward traffic from the LAN to the WAN.

8.3.3.2 Static Port Forwarding Wild Cards

When the PS is provisioned to operate in C-NAPT primary packet handling mode and a C-NAPT binding is statically created with the port number set to zero then the CAP will handle inbound traffic in a special way. The CAP will forward all inbound traffic not associated with an existing C-NAPT session or an existing C-NAPT static binding to the LAN IP address specified in this special type of C-NAPT binding.

The CAP will process packets as follows:

1. Check all incoming packets to see if they are associated with an existing session specified by a C-NAPT dynamic binding. If this is the case then the packet it translated as specified and then forwarded.

2. If not then the CAP checks to see if there is a static C-NAPT binding associated with the packet. If this is the case then the packet it translated as specified and then forwarded.

3. If not then the CAP checks to see if there is a static C-NAPT binding for this WAN IP address with the port number set to 0. If this is the case then the CAP translates the IP address to the LAN IP Addr specified in this special C-NAPT static binding. Note that C-NAPT does not translate the port in this case. After the address translation the packet is forwarded.

Note: If none of the above is true then the packet is dropped.

8.3.3.3 Virtual Private Network (VPN) Support in the CAP

The PS is required to implement a *VPN Passthrough* feature that allows IPSec [RFC 2401]-based VPN clients to exchange keys using Internet Key Exchange protocol [RFC 2409]. A single VPN client in the home at a time is supported, and that client is assumed to satisfy the following conditions:

- the LAN IP Device is in the LAN-Trans realm, i.e., it has a LAN-Trans IP address
- the LAN IP Device uses IPSec as the VPN protocol
- the LAN IP Device uses Internet Key Exchange to dynamically exchange encryption keys with the VPN server

This Recommendation does not limit the number of VPN clients in the LAN-Pass realm (i.e., LAN IP Devices whose MAC address is in the PS Passthrough Table) that can simultaneously access VPN servers outside the home.

For the VPN client to operate properly a firewall policy file must be active in the PS that opens the proper ports for incoming (WAN-to-LAN) traffic, most notably port 500, for IKE traffic.

When keys are dynamically exchanged using IKE [RFC 2406] prior to initiation of an IPSec session the CAP will translate network addresses as usual and will additionally associate port 500 as an inbound port for the private (LAN-Trans) IP address of the device that initiated the VPN connection. This will ensure that incoming IKE messages will be properly forwarded to the VPN client. IPsec sessions are defined in the CAP by the port used for inbound and outbound traffic, the port used for key exchange, the VPN server address and the VPN client address.

Even though the firewall has opened port 500, incoming traffic on port 500 will only be forwarded by the CAP after an IPSec session has been initiated by a client in the LAN-Trans address realm.

If a second VPN client in the home attempts to initiate an IPsec session with a different VPN server the CAP will shift the ports used on the WAN-Data IP address for traffic and key exchange and translate these ports to the standard ports on the VPN Client IP address in the LAN-Trans realm. Additional VPN clients can be supported as well. However, the CAP does not support more than one VPN client in the home connecting to the same VPN server.

IPsec has three modes that can be used for VPNs. The PS is required to support Encapsulating Security Payload Tunneling mode [RFC 2406]. Support for Encapsulating Security Payload Transport mode [RFC 2406] and IP Authentication Header mode [RFC 2402] are not required.

8.3.3.4 Upstream Selective Forwarding Switch Overview

In some cases, a LAN IP Device in the LAN-Pass address realm will reside on a different logical IP subnet than other LAN IP Devices connected to the same PS element. It is important to prevent the traffic between these LAN IP Devices from traversing the HFC network. Preventing this unwanted HFC traffic is the function that is provided by the Upstream Selective Forwarding Switch (USFS).

Specifically, the USFS routes traffic - that is sourced from within the home network and is destined to the home network - directly to its destination. LAN IP Device sourced traffic whose destination IP address is outside the LAN address realm is passed unaltered to the CAP bridging/routing functionality.

The USFS functionality makes use of the IP Address Translation Table (as defined in [RFC 2011]) within the PS element. This table, the [RFC 2011] ipNetToMediaTable, contains a list of MAC Addresses, their corresponding IP Addresses, and PS Interface Index numbers of the physical interfaces that these addresses are associated with. The USFS will refer to this table in order to make decisions about directing the flow of LAN-to-WAN traffic. In order to populate the ipNetToMediaTable the PS learns IP and MAC addresses and their associations. For every associated physical interface, the PS learns all of the LAN-Trans and LAN-Pass IP addresses along with their associated MAC bindings, and this learning can occur via a variety of methods. Vendor specific IP/MAC address learning methods may include: ARP snooping, traffic monitoring, and consulting CDP entries. Entries are purged from the ipNetToMediaTable after a reasonable inactivity timeout period has expired.

The USFS inspects all IP traffic received on PS LAN interfaces. If the destination IP address is found (via the ipNetToMediaTable) to reside on a PS LAN interface, the original frame's data-link destination address is changed from that of the default gateway address to that of the destination LAN IP Device, and the traffic is forwarded to the QoS Forwarding and Media Access (QFM) functionality (ref.: Section 10.2 PS Logical Element CQP) in the PS to be forwarded out on the proper PS LAN interface according to the packet priority. If a match to the destination IP address is not found in the ipNetToMediaTable, the packet is passed, in its original form, to the C-NAT/C-NAPT transparent routing function or the Passthrough bridging function (depending on the active packet handling mode).

8.3.3.5 Multicast

The CAP supports WAN-to-LAN Multicast traffic by transparently bridging downstream IGMP messaging [RFC 2236] and downstream IP Multicast [ID-IGMP] packets. In addition, when in C-NAT/C-NAPT Transparent Routing Mode, the CAP performs address translation on upstream IGMP messages sourced by LAN IP Devices residing in the LAN-Trans domain. The CAP forwards WAN-originated IGMP traffic to the LAN to allow the advertisements to reach LAN IP Devices. A LAN IP Device will determine which multicast it wishes to join and will send a multicast “join” message. The multicast source will then be able to pass data to the LAN IP Device. When the multicast service is no longer desired, the LAN IP Device can either ignore the service and the stream will time out, or the LAN IP Device can send an IGMP “leave” message to the chain to tear down the streaming traffic. Figure 8-3 provides a detailed example of IGMP and Multicast processes passing through a PS.

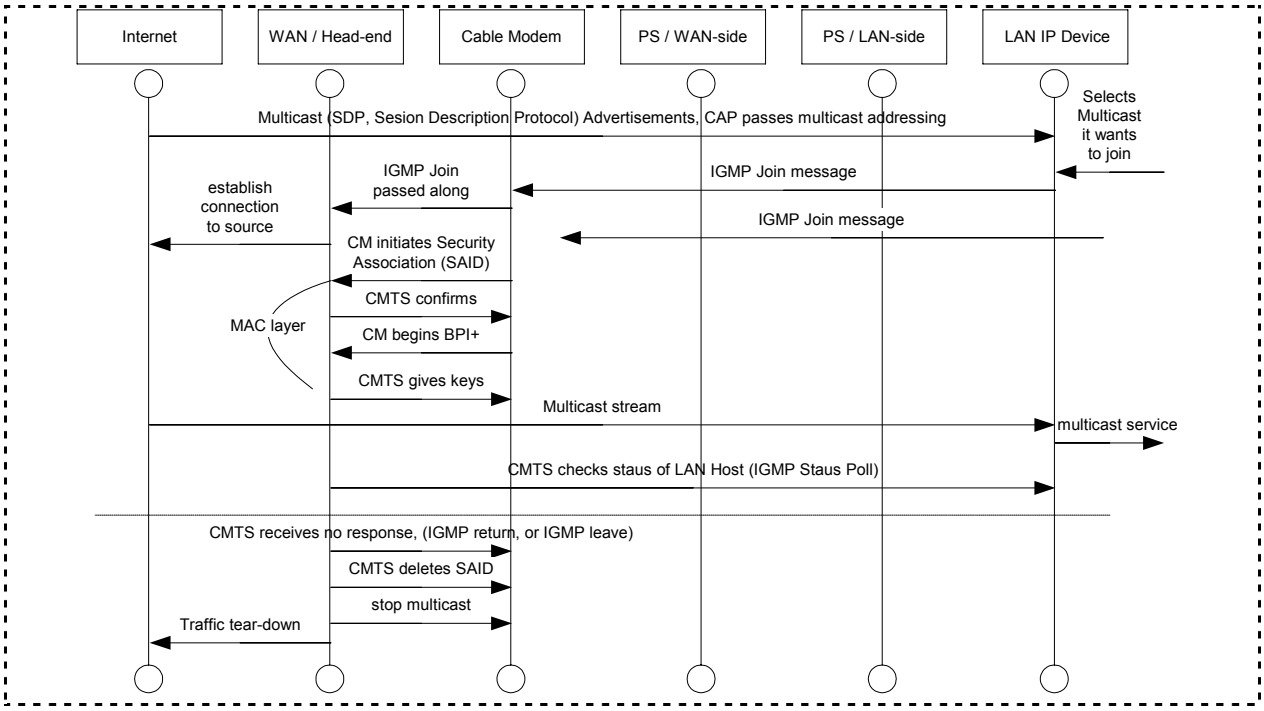

Figure 8-3 Multicast via IGMP Sequence

8.3.3.6 IPCable2Home Packet Handling Examples

This section provides an informative look at processing involved for packet handling. Figure 8-4 shows an example of possible packet processing steps for LAN-to-WAN uni-cast traffic, and Figure 8-5 shows an example of possible packet processing steps for WAN-to-LAN uni-cast traffic.

Note: These examples are informative only and do not imply any requirements on implementation.

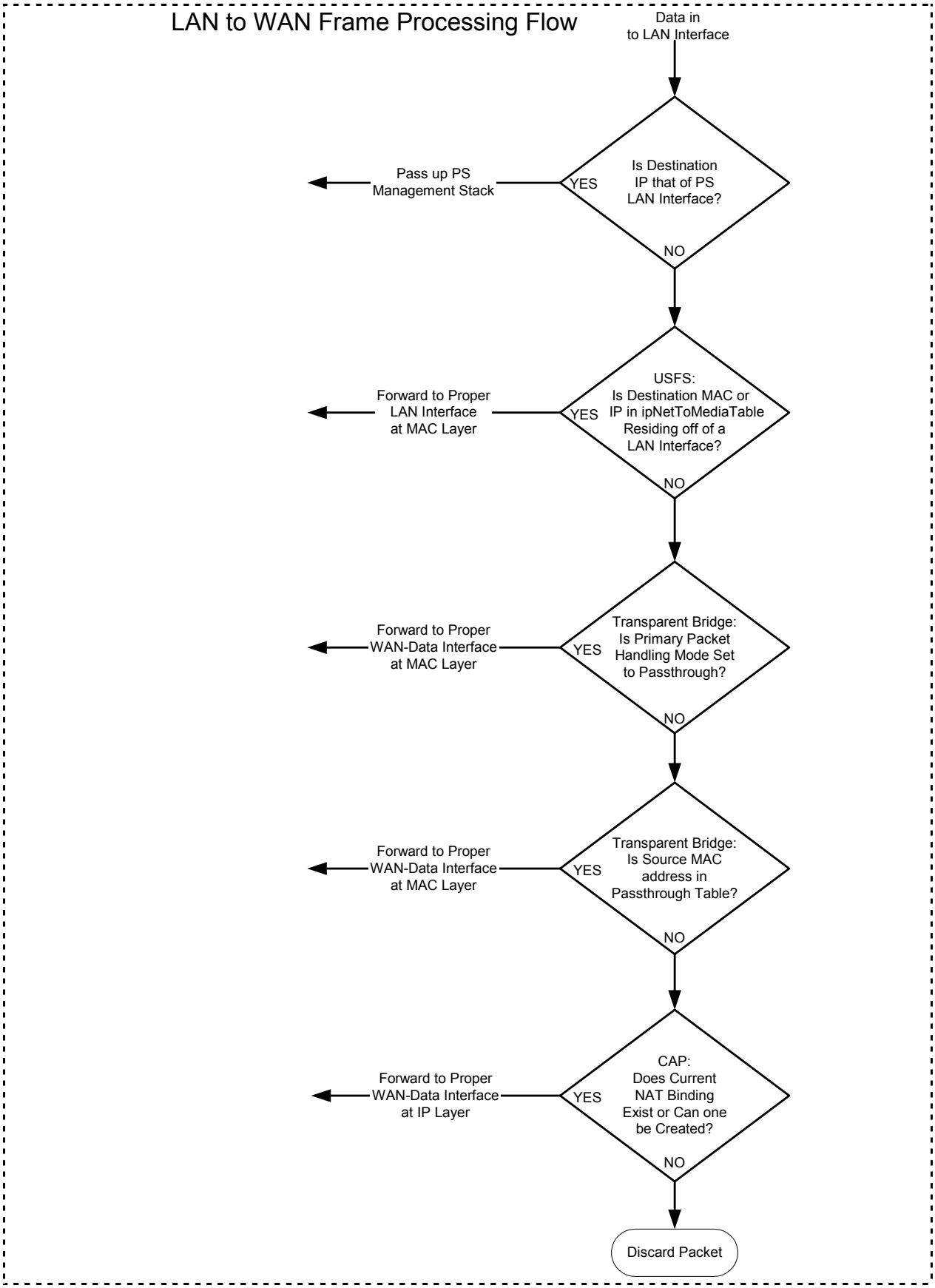

Figure 8-4 LAN-to-WAN Packet Processing Example

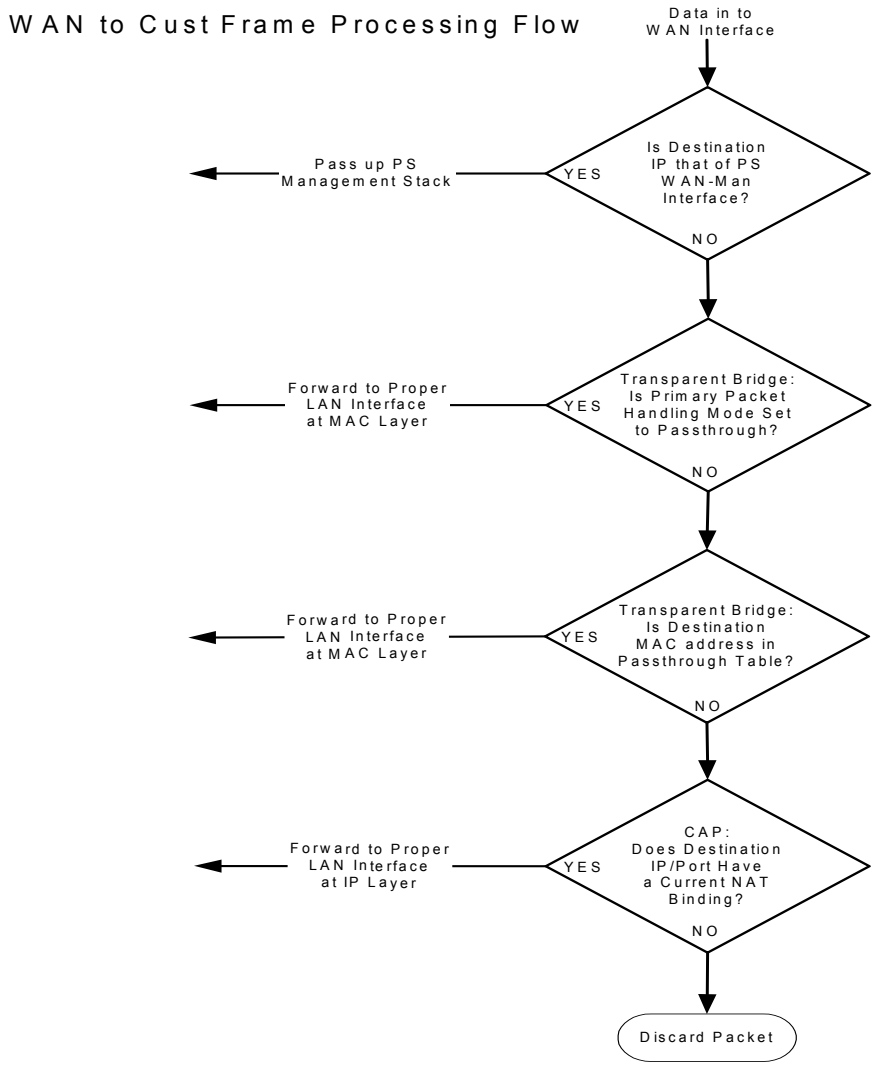

Figure 8-5 WAN-to-LAN Packet Processing Example

8.3.4 CAP Requirements

8.3.4.1 General Requirements

All logical IP interfaces on the Portal Services element MUST be compliant with [RFC 1122] and [RFC 1123], Sections 3 and 4, to enable standard communication with Internet Hosts.

The PS MUST support WAN-to-LAN Multicast traffic by transparently bridging WAN-to-LAN IGMP messaging and WAN-to-LAN IP Multicast packets as defined in [RFC 2236].

If the Primary Packet-handling Mode, cabhCapPrimaryMode, is set to Passthrough, all LAN-to-WAN IGMP

messaging MUST be transparently bridged.

If the Primary Packet-handling Mode, cabhCapPrimaryMode, is set to C-NAPT, the source IP address for all LAN-to-WAN IGMP messages, sourced from LAN IP Devices residing in the LAN-Trans Domain, MUST be translated to the WAN-Data IP address being used for C-NAPT mappings, and then forwarded out to the WAN.

If the Primary Packet-handling Mode, cabhCapPrimaryMode, is set to C-NAT, the source IP address for all LAN-to-WAN IGMP messages - sourced from LAN IP Devices residing in the LAN-Trans Domain that have an IP address that is part of an existing C-NAT mapping - MUST be translated to the WAN-Data IP address being used in that C-NAT mapping, and then forwarded out to the WAN.

8.3.4.2 Packet Handling Requirements

The PS MUST support Passthrough Mode, C-NAT Transparent Routing Mode, and C-NAPT Transparent Routing Mode, and the PS MUST support the selection of this Primary Packet-handling Mode, via the cabhCapPrimaryMode MIB object.

If the Primary Packet-handling Mode, cabhCapPrimaryMode, is set to C-NAT, the PS MUST make certain there exists an available Headend supplied IP address in the WAN-Data IP Address Pool (with a current DHCP lease) before attempting to use this IP address as part of a C-NAT Mapping. If the CAP is unable to create a C-NAT Mapping, due to WAN-Data IP Address Pool depletion, it MUST generate a standard event (as defined in Annex B).

The PS MUST set the WAN and LAN port numbers (cabhCapMappingWanPort and cabhCapMappingLanPort, respectively) of the CAP Mapping Table equal to zero for each Dynamic C-NAT Mapping it creates.

If the cable operator creates or changes a row in the CAP Mapping Table, i.e., if a row is created via the static mapping method (cabhCapMappingMethod = static(1)), and the port number objects of the row (cabhCapMappingLanPort and cabhCapMappingWanPort) are not specified, the PS MUST enter zero for cabhCapMappingLanPort and cabhCapMappingWanPort for that row.

The PS MUST NOT translate the port number for any packet whose IP address appears in the CAP Mapping Table with a port number of zero.

If the Primary Packet-handling Mode, cabhCapPrimaryMode, is set to C-NAPT, the PS MUST make certain there exists a current WAN IP address (with a current DHCP lease from Headend provisioning) before attempting to use this IP address as part of a C-NAPT Mapping. If the CAP is unable to create a C-NAPT Mapping, due to not having a current WAN IP Address or due to port number depletion, it MUST generate a standard event (as defined in Annex B).

LAN-to-LAN uni-cast traffic MUST never be routed or bridged out a WAN interface.

When the DHCP lease of a WAN-Data IP address - that is part of C-NAT or C-NAPT mapping - expires, all mappings associated with that IP address MUST be deleted from cabhCapMappingTable.

8.3.4.3 Passthrough Requirements

When the CAP's Primary Packet-handling Mode, cabhCapPrimaryMode, is set to Passthrough mode, the PS MUST act as a transparent bridge, as defined in [ISO DIS 10038 MAC Bridges], between the WAN-Data realm and LAN-Pass realm, and MUST NOT perform any C-NAT or C-NAPT Transparent Routing functions. Even when the Primary Packet-handling Mode is set to Passthrough, USFS processing MUST take precedence over LAN-to-WAN bridging decisions.

8.3.4.4 C-NAT and C-NAPT Transparent Routing Requirements

When the Primary Packet-handling Mode (cabhCapPrimaryMode) is set to C-NAT the PS MUST support C-NAT address translation processes in accordance with the basic NAT requirements defined in [RFC 3022].

When the Primary Packet-handling Mode (cabhCapPrimaryMode) is set to C-NAPT the PS MUST support C-NAPT address translation processes in accordance with the basic NAPT requirements defined in [RFC 3022].

Regardless of the Primary Packet-handling Mode, the PS MUST support the creation and deletion of Static C-NAT and C-NAPT Mappings, by enabling the NMS system to read, create, and delete (via the CMP) Static CAP Mapping (cabhCapMappingTable) entries.

NMS created Static C-NAT and C-NAPT Mappings MUST persist across PS reboots.

The PS MUST support the creation of Dynamic C-NAT and C-NAPT Mappings, initiated by LAN-to-WAN TCP, UDP, or ICMP traffic. The PS MUST enable the NMS system to read (via the CMP) Dynamic CAP Mapping (cabhCapMappingTable) entries.

The PS MUST support the deletion of Dynamic C-NAT and C-NAPT Mappings if a given Mapping is associated with a TCP session and that TCP session terminates or the TCP inactivity timeout, cabhCapTcpTimeWait, for that Mapping elapses.

The PS MUST support the deletion of Dynamic C-NAT and C-NAPT Mappings if a given Mapping is associated with a UDP session and the UDP inactivity timeout, cabhCapUdpTimeWait, for that Mapping elapses.

The PS MUST support the deletion of Dynamic C-NAT and C-NAPT Mappings if a given Mapping is associated with an ICMP session and the ICMP inactivity timeout, cabhCapIcmpTimeWait, for that Mapping elapses.

Dynamic C-NAT and C-NAPT Mappings MUST NOT persist across PS reboots.

8.3.4.5 Virtual Private Network Support Requirements

When the CAP is operating in C-NAT or C-NAPT Primary Packet-handling mode (as indicated by the value of cabhCapPrimaryMode), the PS MUST recognize IPSec sessions initiated by VPN clients in the LAN-Trans realm, create appropriate mappings in the CAP Mapping Table, and map port 500 for inbound (WAN-to-LAN) traffic to the LAN-Trans IP address bound to the LAN IP Device that initiated the session.

When the CAP is operating in C-NAT or C-NAPT Primary Packet-handling mode (as indicated by the value of cabhCapPrimaryMode) and it recognizes an IPSec session when another one has already been mapped in the CAP Mapping Table to a different VPN server, the PS MAY create mappings for the new session, e.g., by port shifting.

If inbound traffic on port 500 is received by the CAP and there is no active IPSec VPN session then the packets received through port 500 MUST be discarded.

The PS MUST support IPsec sessions using Encapsulating Security Payload Tunneling mode, [RFC 2406].

8.3.4.6 Static Port Forwarding Support Requirements

When the Primary Packet-handling Mode (cabhCapPrimaryMode) is set to C-NAPT and there is a C-NAPT static binding with the WAN port number set to 0 then the PS MUST translate the IP addresses specified in the binding for packets that are not associated with a existing dynamic or static C-NAPT binding.

8.3.4.7 Mixed Bridging/Routing Mode Requirements

The PS MUST support Mixed Bridging/Routing Mode as described in Section 8.3, where the CAP Primary Packet-handling Mode, cabhCapPrimaryMode, is set to C-NAT or C-NAPT Transparent Routing and where the CAP will also transparently bridge traffic for particular MAC addresses. If the CAP Primary Packet-handling Mode, cabhCapPrimaryMode, is set to C-NAT or C-NAPT Transparent Routing and the NMS has written a MAC address, belonging to a LAN IP Device, into the cabhCapPassthroughTable, the PS MUST transparently bridge LAN-to-WAN traffic sourced by this MAC address and WAN-to-LAN traffic destined for this MAC address.

When in Mixed Bridging/Routing Mode, as described in Section 8.3, the USFS function MUST be applied to all LAN originated traffic received.

8.3.4.8 USFS Requirements

Upstream Selective Forwarding Switch (USFS) functionality MUST be applied to packet processing, regardless of the CAP's packet-handling mode (Passthrough, C-NAT, C-NAPT, or mixed Bridging/Routing).

The PS element MUST learn all LAN-Trans IP, LAN-Pass IP, and MAC addresses of LAN IP Devices, associated with each of its active physical network interfaces. IP addresses and MAC addresses learned by the PS element, and PS physical interface index numbers MUST be accessible to the NMS system (through the CMP) via the [RFC 2011] ipNetToMediaTable. The PS element MUST delete entries from the ipNetToMediaTable, when an inactivity timeout expires.

The USFS function MUST inspect all IP traffic originating on PS LAN interfaces, to determine if the destination IP address of a packet is that of a device residing on a PS LAN interface. If the destination IP address in a packet inspected by the USFS is that of a LAN IP Device residing off of a PS LAN interface, the USFS function MUST

replace the MAC Layer Destination address, within the packet's Layer 2 header, with the MAC address of that destination LAN IP Device and forward the frame to the QoS Forwarding and Media Access (QMA) entity (See Section 10.3.1) in the PS to be forwarded out on the proper physical LAN interface according to the packet priority.

The USFS MUST NOT forward any packet destined for a LAN IP Device out any WAN Interface.

9 NAME RESOLUTION

9.1 Introduction/Overview

9.1.1 Goals

The goals of name resolution include:

- Provide Domain Name Service (DNS) from a server in the PS to DNS clients within LAN IP Devices, even during cable connection outages.
- Enable subscribers to refer to local devices via intuitive device names rather than by IP address.
- Via recursive queries to remote DNS servers, provide answers to LAN DNS clients when queried for resolution of non-local hostnames.
- Provide easy DNS service recovery upon re-establishment of cable connectivity after an outage.

9.1.2 Assumptions

The operating assumptions for the naming services include:

- The DNS server in the PS element is the only DNS server authoritative for LAN IP Devices in the LAN-Trans realm.
- The PS element will not provide DNS service to LAN IP Devices in the LAN-Pass realm.
- If the PS element makes use of multiple WAN-Data addresses, the WAN DNS Server information obtained during the most recent WAN-Data address acquisition process (DHCP) will be used.

9.2 Architecture

9.2.1 System Design Guidelines

Table 9-1 Name Resolution System Design Guidelines

Reference	System Design Guideline
Name Rsln 1	Provide Domain Name Service (DNS) from a server in the PS to DNS clients within LAN IP Devices, for name resolution of LAN IP Devices (independent of the state of the WAN connection).
Name Rsln 2	Provide DNS answers, via recursive queries beginning with a cable network DNS server, for DNS clients within LAN IP Devices, for resolution of non-local hostnames.

9.2.2 System Description

This section provides an overview of the IPCable2Home name resolution services within the PS element.

9.2.2.1 Name Resolution Functional Overview

The IPCable2Home Naming Portal (CNP) is a service running in the PS that provides a simple DNS server for LAN IP Devices in the LAN-Trans address realm. The CNP is not used by LAN IP Devices in the LAN-Pass address realm, because they will be directly served by DNS servers external to the home.

Typically, LAN IP Devices in the LAN-Trans realm are configured by the CDP to use the CNP as their Domain

Name Server. The CNP service in the LAN-Trans realm does not depend on the state of the WAN connection. The CNP performs the following tasks:

- Resolves hostnames for LAN IP Devices, returning their corresponding IP addresses.
- Provide DNS answers, via recursive queries beginning with a DNS server in the cable network, for queries that cannot be resolved via local PS information. This action occurs only when WAN DNS server information is available in the PS. Otherwise, the CNP returns an error indicating that the name cannot be resolved.

Making the CNP the primary DNS server on the LAN avoids the need to reconfigure LAN IP Devices when the state of the WAN connection changes. It also permits changing external DNS server assignment without LAN IP Device reconfiguration.

9.2.2.2 Name Resolution Operation

When queried to resolve a hostname, the CNP function of the PS performs the lookup process shown in Figure 9-1. The CNP responds to initial standard DNS queries [RFC 1035], directed to cabhCdpServerDnsAddress, for all name lookups. It is the responsibility of the CNP to make recursive queries to external DNS servers - beginning with the first cabhCdpWanDnsServerIp entry in the CDP's cabhCdpWanDnsServerTable - when queried by a LAN IP Device and to respond to that LAN IP Device with either an answer or an error message.

The CNP relies on the CDP's cabhCdpLanAddrTable, to learn the hostnames associated with the current IP addresses of active LAN IP Devices. As long as a LAN IP Device maintains an active DHCP lease with the CDP and has provided a hostname to the CDP (as part of its IP address acquisition process) its name can be resolved by the CNP. If the hostname requested for resolution cannot be found in the cabhCdpLanAddrTable, the CNP performs recursive queries to external DNS servers (of which the initial one is learned by the CDC via DHCP options).

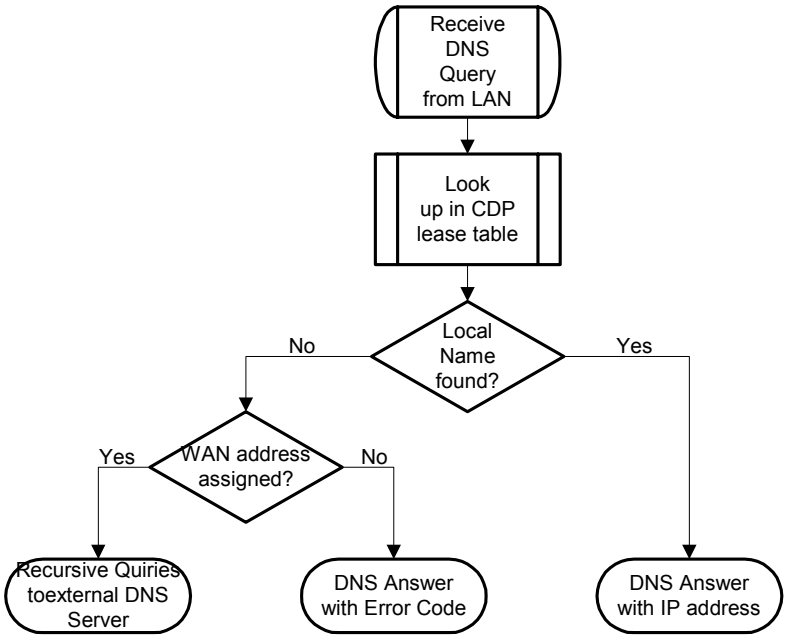

Figure 9-1 CNP Packet Processing

A standard DNS query specifies a target domain name (QNAME), query type (QTYPE), and query class (QCLASS), and asks for Resource Records that match. The CNP will respond to the DNS queries with QCLASS = IN, and QTYPE = A, NS, SOA or PTR as defined in [RFC 1035]. Support for zone transfers and DNS over TCP is not required.

Since the CNP is an authoritative DNS server inside the LAN-Trans realm, it will provide Start of Authority (SOA) and Authoritative Nameserver (NS) records on request. An example of the SOA record fields (see Section 3.3.13 of [RFC 1035]) follows:

Table 9-2 SOA Record Fields

[RFC 1035] RDATA field	IPCable2Home CDP MIB Object
MNAME	cabhCdpServerDomainName
RNAME	Not specified
SERIAL	Not specified
REFRESH	Not specified
RETRY	Not specified
EXPIRE	Not specified
MINIMUM	Not specified

The MNAME field is the domain name of the LAN-trans address realm. The CNP uses the value stored in cabhCdpServerDomainName as the LAN-trans address realm domain name.

The RNAME field is the mailbox of the responsible person for the domain. If the PS maintains an E-mail address for an administrator, this information could be specified in this field.

The SERIAL field is an unsigned 32-bit number, used to identify the version of the zone information. But since this Recommendation does not specify zone transfers, value of this field is not specified.

9.3 Name Resolution Requirements

The CNP MUST comply with the standard DNS message format and support standard DNS queries, as described in [RFC 1034], [RFC 1035].

The CNP is a stateless server that MUST be able to receive queries and send replies in UDP packets [RFC 768].

The CNP MUST support recursive mode, as defined in [RFC 1034].

The CNP answers name queries, beginning with local information within the PS, and its response messages MUST contain an answer or an error.

The CNP MUST only respond to DNS queries addressed to cabhCdpServerDnsAddress.

The CNP MUST NOT respond to any DNS queries addressed to the PS WAN-Man or WAN-Data IP addresses.

Upon receiving an initial hostname resolution query from a LAN IP Device, the CNP MUST access the CDP's cabhCdpLanAddrTable to look up hostnames associated with IP addresses that are leased to LAN IP Devices.

Regardless of the existence of any cabhCdpWanDnsServerIp entries in the CDP MIB cabhCdpWanDnsServerTable, if the hostname can be resolved by the CNP from local data, the CNP MUST respond to the hostname resolution query with the IP address of the named LAN IP Device.

If the queried host name can not be resolved by the CNP from local data, and the CDP's cabhCdpWanDnsServerTable is populated with at least one cabhCdpWanDnsServerIp entry, the CNP function of the PS MUST attempt to resolve the hostname query via recursive queries to external DNS servers, starting with queries to the DNS server, represented by the first cabhCdpWanDnsServerIp entry in the cabhCdpWanDnsServerTable.

If the host name can not be resolved by the CNP from local data and no cabhCdpWanDnsServerIp entries exist in the cabhCdpWanDnsServerTable, the CNP function of the PS MUST respond to the host name resolution query with the appropriate error specified by [RFC 1035].

The CNP MUST respond to DNS queries of type QCLASS = IN, and QTYPE = A, NS, SOA or PTR.

The CNP responses to DNS queries MUST comply with Section 3.3 of [RFC 1035], with Authoritative Answer bit set to '1' in the Header Section (see Section 4.1.1 of [RFC 1035]).

Since the CNP is an authoritative DNS server inside the LAN-Trans realm, it MUST provide Start of Authority (SOA) and Authoritative Nameserver (NS) records on request. The SOA record fields (see Section 3.3.13 of [RFC 1035]) MUST contain an entry for the MNAME field that is equal to the value of the CDP's cabhCdpServerDomainName MIB object.

If cabhCdpServerDomainName is not set, the CNP MUST still provide DNS referral service to LAN IP Devices.

10 QUALITY OF SERVICE

10.1 Introduction

This section describes the IPCable2Home environment for enabling home networking applications to utilize QoS resources. These resources provide a management mechanism that prioritizes data flows to support real-time application traffic, such as VoIP, A/V streaming, and video gaming, by using prioritized media access and queuing. IPCable2Home QoS is complementary to the IPCablecom & J.112 QoS mechanisms, which allow QoS traffic management over the HFC network.

This Recommendation defines the necessary PS and BP element and sub-element QoS requirements that enable applications to establish different levels of QoS within the home network and for operators to communicate the desired priority treatment to the IPCable2Home-enabled applications on the home network.

10.1.1 Goals

The goals for IPCable2HomeQoS include:

- Enable home networking applications to establish prioritized data transmission among Hosts as well as between the Hosts and the Residential Gateway using compliant messaging.
- Enable home networking applications to establish prioritized data sessions between the CMTS and Residential Gateway device using IPCablecom compliant messaging.

10.1.2 Assumptions

The following assumptions were made for IPCable2Home QoS:

- To avoid problems with NAT functions in the CAP element, IPCablecom 1.0 compliant applications will use IPCable2Home LAN-Pass addressing as defined in Section 7 & Section 9.
- Applications that could benefit from QoS could be embedded in IPCable2Home Host devices connected via a home networking technology.
- IPCable2Home Host applications could include IPCablecom services.

Note: Any device that would like to receive QoS for operator services will have to comply with this Recommendation and the device's operating system and network stack will need to have appropriate QoS capabilities.

10.2 QoS Architecture

The IPCable2Home quality-of-service (CqoS) architecture is composed of IPCable2Home functional elements (PS and BP) and sub-elements in the PS and BPs. Developers of IPCable2Home networking equipment (e.g., hardware and software) implement one or more of these elements depending on the desired feature set of these products. Specified minimum sets of capabilities are required to participate in the Q-Domain. The basic CqoS elements are presented in Section 10.2.2.

10.2.1 System Design Guidelines

The overall IPCable2Home QoS system design guidelines are listed in Table 10-1 below.

Table 10-1 IPCable2Home QoS System Design Guidelines

Number	QoS System Design Guidelines
QoS 1	QoS Media Access: IPCable2Home will define a mechanism that controls transmission access using priorities on shared media for the PS and BP logical elements. It will provide prioritized media access to various devices and applications on the home network.
QoS 2	QoS Forwarding: The PS will support a queuing mechanism that prioritizes packets that are received from multiple interfaces and are to be retransmitted /forwarded through LAN interfaces.
QoS 3	QoS Characteristics Management: IPCable2Home will specify a signaling and management mechanism for communication of QoS characteristics between the PS and BPs desiring QoS within a home network. This mechanism will be aggregated and managed in the PS.

10.2.2 IPCable2Home QoS System Description

The CqoS Architecture is composed of the following entities:

- Q-Domain
- Portal Services element (PS)
- Boundary Point element (BP)
- IPCable2Home Quality-of-Service Portal sub-element (CQP)
- IPCable2Home Quality-of-Service Boundary Point sub-element (QBP)

The cable data network equipment manages the IPCable2Home QoS functions but is not within the Q-Domain.

10.2.2.1 CQP Sub-Element

The PS element includes a IPCable2Home Quality of Service Portal (CQP) sub-element. The CQP acts as a CqoS portal for IPCable2Home compliant applications. Its primary function is to enable priorities based QoS for the devices within the home network. It performs priorities based queuing/forwarding and media access for the traffic originating from the PS as well as for the traffic transiting through the PS. It is also responsible for communication of QoS characteristics to various devices within the home.

The CQP also supports the delivery of QoS messaging across the HFC network for IPCablecom applications. IPCablecom compliant messaging includes QoS messaging and other messages related to the aspects of a specific service such as policy decisions and application of two phase reservation models. (From CH 1.0.)

10.2.2.2 QBP Sub-Element

The BP element includes a IPCable2Home Quality of Service Boundary Point (QBP) sub-element. It performs priorities based media access for the traffic originating from the BP. It is also responsible for the reception of QoS characteristics from the PS.

10.2.2.3 QoS Functionality in CQP and QBP

CQP and QBP sub-elements consists of one or more of the following functionalities:

- **QoS prioritized Forwarding and Media Access (QFM):** Specifies prioritized queuing and packet forwarding and prioritized shared media access in the PS. This functionality is part of the PS only.
- **QoS Characteristics Server (QCS):** This functionality is responsible for maintaining a repository of QoS characteristics for various devices and applications within the home network and also for communication of these characteristics to these devices and applications. This functionality is a part of the PS only.
- **QoS Characteristics Client (QCC):** This functionality, with the aid of QCS, determines QoS characteristics that a particular application/device needs to use. It resides within the BP only.

10.2.2.4 Q-Domain

The Q-Domain defines the sphere of direct influence of CQoS functionality. The Q-Domain exists on a per-home basis and is independent of the Addressing Realms. Individual homes are separate and have independent Q-Domains. The CQP and QBP elements bound the Q-Domain within a given home.

10.2.2.5 Physical Device Classes & CqoS Functional Elements

An example of the relationship between the IPCable2Home Devices and the CqoS functional elements is presented in Figure 10-1.

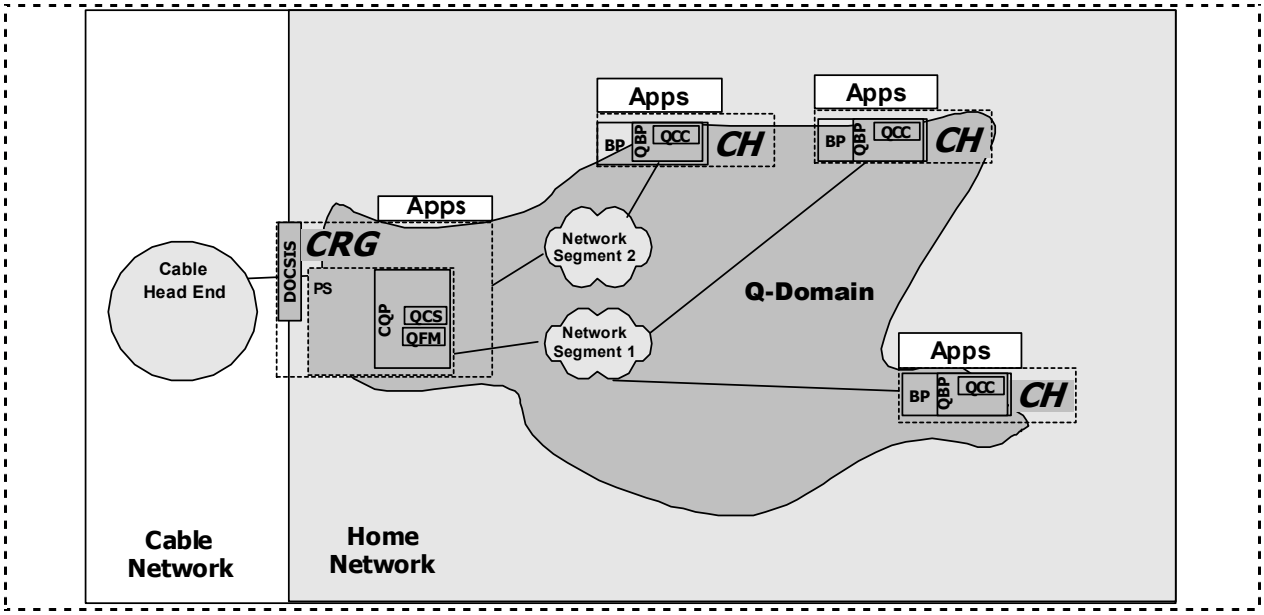

Figure 10-1 Example of CqoS Functional Elements

10.2.2.6 IPCable2Home Priorities and their Mappings

10.2.2.6.1 IPCable2Home Priorities

This Recommendation defines three different QoS priorities. They are:

- IPCable2Home Generic Priorities
- IPCable2Home Queuing Priorities
- IPCable2Home Media Access Priorities

10.2.2.6.1.1 IPCable2Home Generic Priorities

This Recommendation defines eight IPCable2Home Generic Priority levels, 0 through 7, 7 being the highest and 0 being the lowest. Cable operators can assign one of these eight priorities to an application. Out of the three types of priorities defined, only the IPCable2Home Generic Priority value for an application can be set by a cable operator. The other two priorities, IPCable2Home Queuing Priorities and IPCable2Home Media Access Priorities, are derived from this IPCable2Home Generic Priority, depending on the capabilities of the hardware and software in the device. The higher the IPCable2Home Generic Priority assigned to an application, the higher preference is given to that application's packets for packet forwarding and media access functionalities.

10.2.2.6.1.2 IPCable2Home Queuing Priorities

In the PS, packets may arrive from multiple interfaces and be destined for a single interface. Hence, each interface needs to implement a queueing function. In order to provide prioritized QoS for traffic within the home passing through the PS, this Recommendation specifies prioritized queueing functionality per interface in the PS. For this purpose, an individual queue within an interface is designated with a certain queuing priority. This is defined as IPCable2Home Queuing Priority. This IPCable2Home queuing priority needs to be identified for each packet to be

transmitted on each PS interface so that the packet can be placed in an appropriate queue. This queuing priority is derived from the IPCable2Home Generic Priority assigned to the application that sent the packet, using the number of queues supported by an interface on the PS. This mapping is specified in Section 10.2.2.6.2.

10.2.2.6.1.3 IPCable2Home Media Access Priorities

This Recommendation defines a prioritized QoS media access system in which traffic over a shared media is prioritized based on the assigned packet priority. Thus, a shared media technology needs to support prioritized QoS such that a packet with higher priority is given preferential access to the shared media, versus a packet with lower priority. Various shared media technologies support varying number of media access priorities. (e.g. HomePNA support eight media access priorities, HomePlug support four). IPCable2Home Media Access Priority for the packet is derived from its IPCable2Home Generic Priority based on the number of media access priorities supported by the interface's layer-2 shared media technology. This mapping is defined in Section 10.2.2.6.3. IPCable2Home Media Access Priority values are logical levels that represent a level of preference that an application-packet should get for media access. IPCable2Home Media Access Priority mapping is separate and distinct from native media access priority mappings defined by layer-2 shared media technologies, to keep IPCable2Home Media Access Priority mapping independent of layer-2 technologies.

10.2.2.6.2 Mapping of IPCable2Home Generic Priorities to IPCable2Home Queuing Priorities

As explained in Section 10.2.2.6.1.2, the PS performs prioritized queuing for each of its interfaces. There are 8 IPCable2Home Generic Priorities defined, hence an ideal scenario would be that an interface has 8 queues and each is assigned with a queuing priority from 0 to 7. However, the number of queues implemented for an interface in the PS varies on the implementation. The number of queues supported by an interface will be stored in the PS database and readable via a MIB object cabhPriorityQosPsIfAttribIfNumQueues. If an interface implements N (1<=N<=8) queues, the various queues in an interface will be designated with IPCable2Home Queuing Priorities from 0 (lowest) to N-1 (highest). When a packet enters the PS, the packet's IPCable2Home Queuing Priority needs to be determined based on its Generic Priority so a packet can be placed in an appropriate queue. This mapping between the two priorities is specified in Table 10-2.

In Table 10-2, eight Generic Priorities are expressed in the fist column. In the adjacent columns of the table, the number of queues supported for the interface is presented as a range from 8 to 1. Table entries represent IPCable2Home Queuing Priorities for packets ranging from 0 to N-1.

Once a packet's IPCable2Home Queuing Priority is determined from Generic Priority using Table 10-2, a packet is placed in a queue that is designated for that specific IPCable2Home Queuing Priority.

Table 10-2 IPCable2Home Queuing Priority Mappings

Generic IPCable2Home Priority	Number of Queues Supported by the Interface (N)							
	8	7	6	5	4	3	2	1
7	7	6	5	4	3	2	1	0
6	6	5	4	3	3	2	1	0
5	5	4	3	2	2	1	1	0
4	4	3	2	2	2	1	1	0
3	3	2	1	1	1	1	0	0
2	2	1	1	1	1	0	0	0
1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Note: The following paragraph illustrates how IPCable2Home Queuing Priorities Mapping should be used:

If an incoming data packet has a Generic Priority of 7, and is destined for an outgoing interface that supports only three queues (N=3); then the IPCable2Home Queuing Priority for that packet would be 2. Three queues for that particular interface would be designated with priorities of '0' (lowest), '1' and '2' (highest). This particular packet would be placed in the queue with the priority designation of two for that interface.

10.2.2.6.3 Mapping of IPCable2Home Generic Priorities to IPCable2Home Media Access Priorities

As discussed in Section 10.2.2.6.1.3, various layer-2 technologies support varying number of media access priorities. Hence, eight IPCable2Home Generic Priorities defined for applications need to be mapped to the appropriate number of IPCable2Home Media Access priorities, based on number of media access priorities (1<=M<=8) supported by a layer-2 technology interface. The number of native media access priorities (M) supported by the particular layer-2 shared media technology of each interface on the PS and BP is stored in the PS and BP respectively. The number of media access priorities supported by PS interfaces is available through the MIB object cabhPriorityQosPsIfAttribIfNumPriorities in the PS. The number of media access priorities supported by the BP interface is available in the PS though the MIB object cabhPsDevBpNumberInterfacesPriorties. The mapping between these two priorities is defined in Table 10-3.

Table 10-3 is very similar to Table 10-2, except the mapping of IPCable2Home Generic Priority values is performed using the number of media access priorities (M) supported by a particular layer-2 shared media technology. The entries in the table represent IPCable2Home Media Access Priorities. Thus, if a layer-2 technology supports M media access priorities, then the IPCable2Home Media Access Priorities for that technology would range from 0 (lowest) to M-1 (highest). These IPCable2Home Media Access Priority values represent relative logical levels. The higher the IPCable2Home Media Access priority value for the packet, the higher preference it should be granted for accessing the shared media. Implementers of this Recommendation should make sure that packets are given required relative preferential access to the shared media, as described by the IPCable2Home Media Access Priority mapping.

Note: The following paragraph illustrates how IPCable2Home Media Access Priorities Mapping should be used:
If a IPCable2Home Generic Priority value for an application packet is 7 (highest), and the layer-2 technology on which the packet is being transmitted supports 4 media access priorities, then referring to Table 10-3, the packet's IPCable2Home Media Access Priority value would be 3 (highest). However, if a Generic Priority value for a packet is 2, the IPCable2Home Media Access Priority value for the aforementioned technology would be 1 (second lowest). Previously, the required IPCable2Home mapping may be different from native mappings used by the shared media technologies.

See Appendix I for examples of differences between the IPCable2Home Media Access Priority mapping and native layer-2 technology mappings**.**

Table 10-3 IPCable2Home Media Access Priority Mappings

IPCable2Home Generic Priority	# Media Access Priorities Supported (N) in the LAN							
	8	7	6	5	4	3	2	1
7	7	6	5	4	3	2	1	0
6	6	5	4	3	3	2	1	0
5	5	4	3	2	2	1	1	0
4	4	3	2	2	2	1	1	0
3	3	2	1	1	1	1	0	0
2	2	1	1	1	1	0	0	0
1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

10.3 PS Logical Sub-Element CQP

The CQP contains the QFM and QCS functionalities as shown in Figure 10-1. The QFM functionality is described in Section 10.3.1. The QCS functionality is described in Section 10.3.2.

10.3.1 QoS Forwarding and Media Access (QFM)

The Quality of Service Forwarding and Media access functionality (QFM) in the PS is responsible for prioritized forwarding and media access for the packets going through the PS onto the home LAN. This section provides description of the QFM functionality in the PS and specifies associated PS requirements.

10.3.1.1 QoS Forwarding and Media Access Goals

The goals for the QoS Forwarding and Media Access functionality include:

- To order the packets arriving from multiple LAN interfaces to the PS and forward them to a destination LAN interface according to their priorities and LAN interfaces' queuing capabilities.
- Provide prioritized access to the shared media during the packet transmission based on the packet priority and capabilities of shared media for prioritized access.

10.3.1.2 QoS Forwarding and Media Access Design Guidelines

Table 10-4 QFM System Design Guidelines

Number	QFM System Design Guidelines
QFM.1	The QFM should operate on packets to and from the LAN-Trans and LAN-Pass address realms.
QFM.2	The QFM will determine the packet priority using the information available in the PS MIB maintained by QCS.
QFM.3	The QFM will order incoming packets to exit through LAN interfaces according to their priorities.
QFM.4	The QFM should be able to work with different number of queues per interface.
QFM.5	The QFM will provide prioritized access to the shared media on each interface according to the packet priority.
QFM.6	The QFM should map IPCable2Home Generic Priority of the packet to IPCable2Home Media Access priority according to the defined mapping.
QFM.7	The QFM should be able to operate with interfaces that support different numbers of priorities for media access.

10.3.1.3 QoS Forwarding and Media Access Design Assumptions

- Each PS LAN interface may support less than eight queues.
- Maximum number of queues supported a PS LAN interface is eight.
- Each PS LAN networking technology may support less than eight media access priorities.
- Maximum number of media access priorities supported by a PS LAN networking technology is eight.

10.3.1.4 QoS Forwarding and Media Access System Description

The QFM provides the PS a mechanism to order and transmit packets out of the PS to a LAN host according to assigned priorities. It is through the assignment of priorities to packets and the action of the QFM that packets passing through the PS over the home LAN are provided prioritized access to the host transmission interfaces and to the shared LAN media. Any packet going out of the PS on a LAN interface should be processed by the QFM regardless of its source.

Once the QFM receives a packet destined for a particular LAN interface, it performs the following three actions before the packet is transmitted onto the destination LAN interface:

1. Classification process to identify the Generic Priority of the packet
2. Prioritized queuing
3. Prioritized media access

10.3.1.4.1 Classification of the Packet to identify IPCable2Home Generic Priority

When the PS needs to transmit a packet over the LAN interface, it examines the packet to identify a IPCable2Home Generic Priority for the packet. The PS reads the destination IP and destination port of the packet. The PS database stores a classifier table (cabhPriorityQosDestPriorityListTable) that uses destination IP and destination port values to determine the Generic Priority of the packet. Wild carding (0) is allowed for destination port field but not for destination IP. Hence the PS first tries to find a specific entry that matches packet's destination IP and destination port to determine the priority. If a specific entry is not found, the PS tries to determine priority using destination IP only. If no entry is found in the classifier table for packet's destination IP and destination port, then the PS assigns a Generic Priority value of 0 to the packet. The PS uses the assigned IPCable2Home Generic Priority value to determine the packet's IPCable2Home Queuing Priority and IPCable2Home Media Access Priority.

10.3.1.4.2 Prioritized Queuing

The number of queues supported by an interface on the PS, to which the packet is destined, may not be the same as

the eight IPCable2Home Generic Priority values defined by this Recommendation. Hence the PS maps IPCable2Home Generic Priority value of the packet to a IPCable2Home Queuing Priority value as defined in Section 10.2.2.6.1.2. Then the PS places the packet in an appropriate queue of the destination interface that corresponds to this mapped IPCable2Home Queuing Priority value.

For each outgoing interface, the QFM polls all of the queues on that interface according to their priorities to extract packets out to be transmitted on the shared media. Every time the QFM is to extract a packet from the queues for a particular PS interface, it always starts its polling with the highest priority queue first. If the highest priority queue has no packets to be sent, the QFM polls the next highest priority queue of the remaining queues in the hierarchy until it finds a packet to be sent in one of the queues. Packets are extracted from each queue in the order they arrive. Thus, the queuing scheme used by the QFM may be described as First in, First Out with Priorities, and Highest Priority Queue First.

10.3.1.4.3 Prioritized Media Access

Once the QFM extracts a packet from the set of queues of an interface, the packet needs to be transmitted on the shared LAN media with an appropriate priority. Hence, the QFM maps the IPCable2Home Generic Priority value of the packet to the IPCable2Home Media Access Priority value as explained in Section 10.2.2.6.3, using Table 10-3. This value determines the level of preference the packet should use for accessing the shared media. Therefore, vendors need to insure that relative media access preferences, as required by IPCable2Home Media Access Priority values, are maintained when transmitting packets over the shared LAN media.

10.3.1.4.4 IPCablecom Applications Support

Since the goal of QoS is to provide QoS over home network only, this Recommendation does not give special consideration for access network QoS. However, a LAN IP Device may host an IPCablecom application [j.161], [J.163], in which case the PS can be configured for Passthrough packet handling so as to bridge QoS messaging between the IPCablecom application on the home network and the CMTS.

Since the PS will simply forward IPCablecom QoS messaging while in Passthrough mode, there is no dependency on the NMS to support this function. Therefore, this CQP function remains the same for both DHCP Provisioning Mode and SNMP Provisioning Mode (see Section 5.5).

10.3.1.5 QoS Forwarding and Media Access Requirements

10.3.1.5.1 Packet Classification Requirements

When PS needs to transmit a packet over a LAN interface, the PS MUST determine IPCable2Home Generic Priority for the packet from its destination IP and destination port values using PS classifier table, (cabhPriorityQosBpDestTable) stored in the PS database [Annex E.7]. The PS MUST always try to find a specific entry in the PS database that matches both the destination IP and destination port of the packet to determine the priority. If a specific entry is not found then the PS MUST try to find an entry that matches only the destination IP of the packet. If there is no entry in the PS database that matches the destination IP of the packet, then the PS MUST assign IPCable2Home Generic Priority value 0 to the packet.

10.3.1.5.2 Prioritized Queuing Requirements

The PS MUST store the number of queues implemented by each of its interface in the PS database that can be accessed via a cabhPriorityQosPsIfAttribIfNumQueues MIB [Annex E.7].

The PS MUST map the IPCable2Home Generic Priority value of the packet identified during the classification process to IPCable2Home Queuing Priority value as defined in Section 10.2.2.6.1.2 using the number of queues (cabhPriorityQosPsIfAttribIfNumQueues) implemented by an interface on which the packet is to be transmitted. The PS MUST queue the packet appropriately on the destination interface according to this mapped IPCable2Home Queuing Priority value.

For each LAN interface, the PS MUST poll various queues on that interface according to their priorities to extract packets out to be transmitted on the shared media. Every time the PS is to extract a packet from the various queues for a particular interface, the PS MUST always start its polling with the highest priority queue first. If the highest priority queue has no packets to be sent, the PS MUST poll the next highest priority queue of the remaining queues in the hierarchy, until it finds the next available highest priority packet to be sent. PS MUST always extract packets from each queue in the order they arrive.

10.3.1.5.3 Prioritized Media Access Requirements

The PS MUST store the number of native layer-2 media access priorities supported by each of its interface in the PS database that can be accessible via a MIB cabhPriorityQosPsIfAttribIfNumPriorities [Annex E.7].

After the packet is extracted from the queues of a particular interface the PS MUST map Generic Priority of the packet to IPCable2Home Media Access Priority, as defined in Section 10.2.2.6.1.3, using the number of media access priorities supported (cabhPriorityQosPsIfAttribIfNumPriorities) by that interface. The PS MUST transmit the packet through the shared media technology such that its relative preferential access to the media, as required by IPCable2Home Media Access Priority value, is maintained.

10.3.1.5.4 IPCablecom Applications Support Requirements

The PS MUST act as a transparent bridge and forward IPCablecom [J.161], [J.163] QoS messaging between the CMTS and IPCablecom applications. Application data is associated to a CM service flow according to a classifier that is created in the CM interface, based on the information included in the IPCablecom messages (such as RSVP PATH).

Since the PS requirement for IPCable2Home is to just forward IPCablecom QoS messaging, there is no dependency on the NMS to support this function. Therefore, this CQP function remains the same for both DHCP Provisioning Mode and SNMP Provisioning Mode (see Section 5.5).

10.3.2 PS QoS Characteristics Server (QCS)

The QoS Characteristics Server (QCS) functionality in the PS is responsible for management of application priorities in the home network on behalf of a cable operator. This section provides the description of the QCS functionality and associated PS requirements.

10.3.2.1 QoS Characteristics Server Goals

- To establish a set of criteria by which applications and network stacks can assign and use QoS characteristics for traffic within the home network.
- To provide a mechanism for the head-end to communicate the desired QoS Characteristics to the PS and then to IPCable2Home Hosts (BPs). Specifically, the assignment of QoS characteristics is related to the priority information per application type.

10.3.2.2 QoS Characteristics Server Design Guidelines

Table 10-5 QCS Design Guidelines

Number	System Design Guidelines
QCS.1	QCS will be provided priorities information for each application from Network Management Server (NMS) in the Headend
QCS.2	Priorities information supplied to the QCS will be controlled by cable operators (individual PS or mass PS update control)
QCS.3	Priorities information supplied to the QCS may be updated by the headend and BPs (QCCs) will acquire this updated information from the QCS
QCS.4	QCS will use a defined message content protocol (XML) and message transport protocol (SOAP) for distribution of priority information to BPs
QCS.5	QCS will use a defined messaging content interface (MIB) for providing priorities information of various applications in the home LAN to Network Management Server (NMS) in the headend
QCS.6	QCS aids QoS Forwarding and Media Access (QFM) functionality to determine a priority of the application packet

10.3.2.3 QoS Characteristics Server Assumptions

- IPCable2Home defines a format for exchanging messages between PS and BP.

- IPCable2Home defines a protocol for exchanging information between PS and BP.
- IPCable2Home Hosts can have more than one service/application.

10.3.2.4 QoS Characteristics Server System Description

The QCS maintains a "database" of information in the PS Database as described in Section 5.4. The QCS receives application priority information from the headend, via initial configuration of the PS, or through a MIB interface in the CMP. The QCS also gathers application information from various BPs in the home LAN and assigns priorities to them. The QCS communicates this application priority information to the BPs (QCCs) to be used for prioritized media access by BPs. The information maintained by the QCS is used by the QFM functionality in the PS for prioritized forwarding and prioritized media access of packets going through it.

The rest of Section 10.3.2.4 is devoted to describing the exchange of information that occurs between the headend and the PS over the WAN and between the PS and BPs over the LAN.

10.3.2.4.1 WAN Information exchange

From the WAN side, the cable operator headend provides to the PS mapping of different applications and the priorities that they should use in a configuration file, or using SNMP SETs. NMS, at the headend, can read and update (change/modify/delete) these application priorities in the PS database using SNMP via a MIB interface.

10.3.2.4.1.1 Application ID to IPCable2Home Generic Priority Mappings from headend to the PS

The headend provides the PS with a list of Application Ids and their IPCable2Home Generic Priorities that a cable operator wishes these applications to use. This information is supplied to the PS through a configuration file at the time of PS initialization, or via SNMP SET commands from the headend. The PS stores this information in the PS database that is accessible via a MIB table, cabhPriorityQosMasterTable [Annex E.7]. The PS makes use of this table as a priority master table to identify the priorities for various applications on the BPs over the home LAN.

PS can also receive requests from the NMS to update (add/modify/delete) these IPCable2Home Generic Priorities for applications in its master table using SNMP. In response to these requests, the PS updates (add/modify/delete) priority master table (cabhPriorityQosMasterTable). Such updates to the application priorities gets communicated to the BPs during subsequent LAN Information exchanges, which is described in Section 10.3.2.4.2.

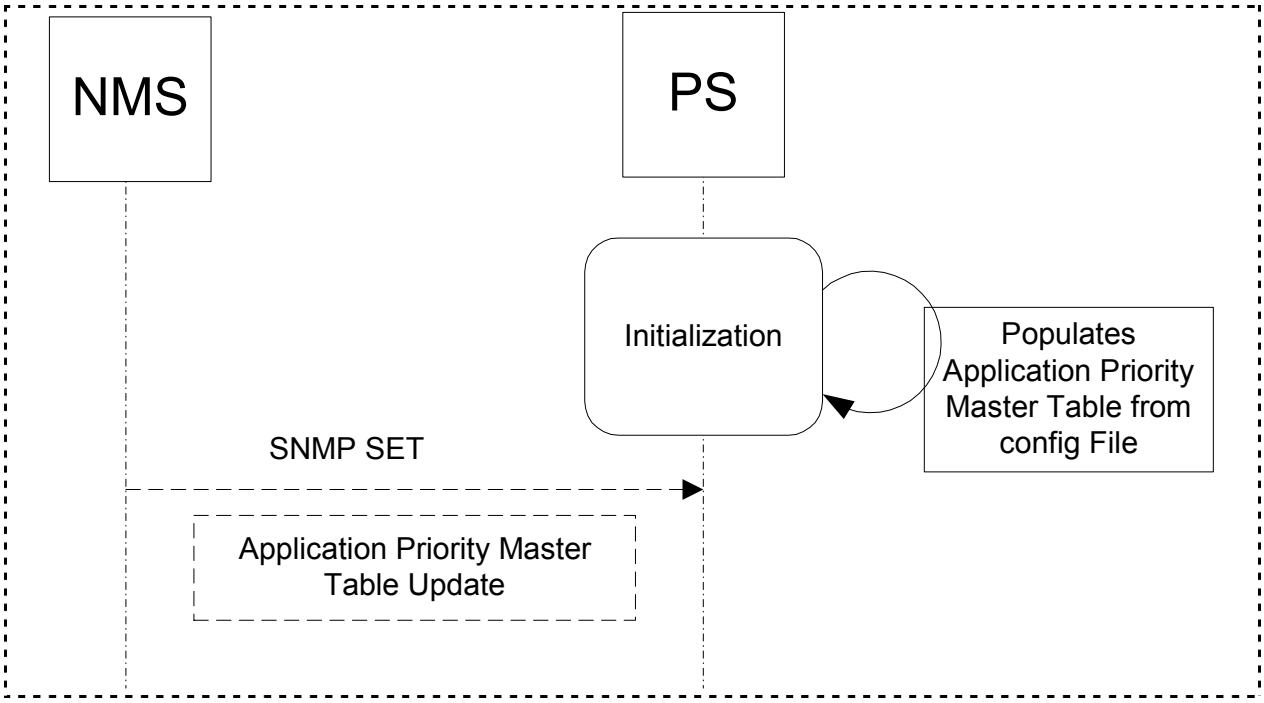

Figure 10-2 WAN Information Exchange and Processing at the PS

10.3.2.4.2 LAN Information Exchange

On the LAN side, a BP communicates its applications and sessions (destination IP and port) information to the PS in order to obtain their priorities. Once the PS receives this information, it determines appropriate priorities by looking them up in the priority master table and conveys them back to the BP. This information is exchanged between the PS and BP using QoSProfile XML schema (described below in Section 10.3.2.4.2.1) and BP Initiated SOAP Messaging (BP_Init Operation) as described in Section 6.3.3.4.4.2.

10.3.2.4.2.1 QoSProfile XML Schema

The QoSProfile XML schema contains two XML complex sequences: QoSApplicationList and the DesPriorityList. The QoSApplicationList contains four elements: BpIpAddress, ApplicationId, DefaultCHpriority, and a sequence of DestPriorityList. The DestPriorityList complex type, which is considered a secondary sequence in QoSApplicationList, contains three elements: DestIp, DestPort, and IpPortPriority. Each element has a defined type as mentioned in Table 10-6. The defined types are references from the W3C XML schema definitions [XML1].

The ApplicationId element is the application server port number for each BP application assigned by IANA [IANA1]. Although applications are identified by the IANA port number, communication may also occur on other port numbers. BP communicates a list of ApplicationIds for all the applications installed on it to the PS in the BP_Init Message (described in Section 10.4.1.4.1.1).

The DefaultCHpriority element is the default IPCable2Home Priority for an application. The BP may provide a value for this element in the QoSProfile. That value will be overwritten by the value supplied by the PS, via the BP_Init_Response Message (described later in Section 10.3.2.4.2.3), after consulting the application priority master table in the PS database (cabhPriorityQosMasterTable).

The BP includes DestPriorityListEntry sequence(s) in the QoSProfile for an application-session with another device. The DestPriorityListEntry sequence(s) are associated to the ApplicationId element in the QoSProfile XML schema. DestIP and DestPort elements respectively, correspond to the destination IP and destination port number of the application-session (socket connection) that is established by the BP. These entries are used to determine the priority (IpPortPriority) of the traffic, passing through the PS, based on specific destination IP address and port number as specified in the entry. Wild carding (0) is allowed only for DestPort, but not for DestIP. The BP may provide a value for IpPortPriority element in the QoSProfile. The PS overwrites that value with the DefaultCHpriority, supplied in the BP_Init_Response Message, after consulting the application priority master table in the PS database (cabhPriorityQosMasterTable).

A BP is always required to transmit the entire QoSProfile XML schema to the PS whenever it sends the BP_Init message.

Table 10-6 QoS Profile XML Schema

```
<xs:complexType name= “ch:QoSProfile”/>
      <xs:element name =”ch:QoSApplicationList”   type=”ch:QoSApplicationListEntry minOccurs=”1”
maxOccurs=”4”/>”
</xs:complexType>

<xs:complexType name= “ch:QoSApplicationListEntry>
   <xs:sequence>
       <xs:element name=”ch:BpIpAddress”              type=”xs:string”/>
       <xs:element name=”ch:ApplicationId”             type=”xs:int”/>
       <xs:element name=”ch:DefaultCHPriority”       type=”xs:int”/>
       <xs:element name=”ch:DestPriorityList”   type=”ch:DestPriorityListEntry minOccurs=”0”
maxOccurs=”4”/>”
    </xs:sequence>
</xs:complexType>

<xs:complexType name= “ch:DestPriorityListEntry>
   <xs:sequence>
        <xs:element name=”ch:DestIp”            type=”xs:string”/>
        <xs:element name=”ch:DestPort”          type=”xs:int”/>
        <xs:element name=”ch:IpPortPriority”   type=”xs:int”/>
    </xs:sequence>
</xs:complexType>
```

10.3.2.4.2.2 BP information to the PS using BP_Init Message

A BP is required to send its applications and sessions information to the PS in the QoSProfile XML schema format using BP_Init Message, as described in Section 6.3.3.4.4.2.1, on the following three different occasions:

- DHCP lease acquisition or renewal
- Application update (addition or deletion) in a BP
- Establishment and termination of application-session with another device by a BP

Refer to Section 10.4.1.4.1.1.1 for detailed description of BP information exchange under each of the above three occasions.

10.3.2.4.2.3 Priority information from the PS to BP using BP_Init_Response

The processing of the QoSProfile XML schema by the PS is exactly the same in all the three different occasions (mentioned above in Section 10.3.2.4.2.2), when it receives the BP_Init Message. The processing of QoSProfile XML schema is described below:

Upon the receipt of the QoSProfile XML schema from the BP in the BP_Init message, the PS determines values for DefaultCHPriority (part of QoSApplicationListEntry) and IpPortPriority (part of DestPriorityListEntry) elements for all the applications in the QoSProfile by looking up the priority master table in the PS database (cabhPriorityQosMasterTable). The PS updates the BP QoSProfile with these priorities by overwriting the values that BP may have provided in its original QoSProfile.

The PS then stores this BP application priority information, represented by the updated QoSProfile, in the PS database that is accessible via the MIB tables, cabhPriorityQosBpTable and cabhPriorityQosBpDestTable [Annex E.7]. The PS completely replaces the old BP application priority information that may have been stored in its database with the new information represented by the updated QoSProfile. Such a complete replacement of the old BP application priority information addresses the processing of both the addition as well as the deletion of a new application or a session in the BP and keeps the processing complexity in the PS to a minimum level.

The cabhPriorityQosBpTable represents information about various applications and their priorities on a particular BP in the home LAN. The cabhPriorityQosBpDestTable represents destination IP and port

specific priorities for different BP application-sessions. The QFM functionality in the PS utilizes the information represented by the cabhPriorityQosBpDestTable for its prioritized queuing and prioritized media access in the PS.

After updating the database with BP application priority information, the PS sends BP QoSProfile, updated with priority information, to the BP using BP_Init_Response Message as described in Section 6.3.3.4.4.2.2. This updated QoSProfile conveys to the BP appropriate priority information that it is required to use for its applications.

10.3.2.5 QoS Characteristics Server Requirements

10.3.2.5.1 WAN Information Exchange Requirements

The PS MUST store a list of Application Ids and their IPCable2Home Generic Priorities, provided by a cable operator, in the PS database, that is accessible via a Application Priority Master MIB table, cabhPriorityQosMasterTable [Annex E.7]. The PS MUST support updates (add/modify/delete) to this priority master table (cabhPriorityQosMasterTable) through a configuration file at the time of PS initialization, or via SNMP SET commands from the headend.

10.3.2.5.2 LAN Information Exchange Requirements:

The processing of the QoSProfile XML schema by the PS is identical in all three different occasions (mentioned above in Section 10.3.2.4.2.2), when it receives the BP_Init Message.

The PS MUST be able to process BP QoSProfile XML schema (as described in Section 10.3.2.4.2.1) containing its applications and sessions (destination IP and port) information received in the BP_Init Message (as described in Section 6.3.3.4.4.2.). When PS receives QoSProfile XML schema from the BP (on any of the three occasions as explained in Section 10.3.2.4.2.2) in the BP_Init message, the PS MUST determine values for DefaultCHPriority (part of QoSApplicationListEntry) and IpPortPriority (part of DestPriorityListEntry) elements for all the applications in the QoSProfile by looking up the priority master table in the PS database (cabhPriorityQosMasterTable). The PS MUST update the BP QoSProfile with these priority values by overwriting the values that BP may have provided in its original QoSProfile.

The PS then MUST store this BP application priority information, represented by the updated QoSProfile, in the PS database that is accessible via MIB tables, cabhPriorityQosBpTable and cabhPriorityQosBpDestTable [Annex E.7]. The PS MUST completely replace the old BP application priority information that may have been stored in its database with the new information represented by the updated QoSProfile.

After updating the PS database with BP application priority information, the PS MUST send the entire BP QoSProfile, updated with priority information, to the BP using BP_Init_Response Message, as described in Section 6.3.3.4.4.2.2.

10.4 BP Logical Sub-Element QBP

10.4.1 QoS Characteristics Client (QCC)

10.4.1.1 QoS Characteristics Client Goals

- To provide a mechanism for a IPCable2Home Host to receive desired QoS Characteristics from the PS. These QoS characteristics are communicated to the PS from the headend.
- To establish a set of criteria in a IPCable2Home Host by which its applications and network stacks can assign and use QoS characteristics for its application traffic.

10.4.1.2 QoS Characteristics Client System Assumption

IPCable2Home Compliant Host (BP) can have more than one service/application on it.

10.4.1.3 QoS Characteristics Client System Guidelines

Table 10-7 QCC Design Guidelines

Number	System Design Guidelines
QCC.1	QCC will be provided application priorities information from QCS.
QCC.2	Priorities controlled by QCS will be updated dynamically and QCC will request updated priority information from QCS.
QCC.3	QCC will use a defined message content protocol (XML) and message transport protocol (SOAP) for communicating priority information to the PS.
QCC.4	The QCC will provide prioritized access to the shared media of its LAN interface according to the packet priority.

10.4.1.4 QoS Characteristics Client System Description

This section provides an overview of the key concepts of the QoS Characteristics Client (QCC) in the BP.

The messaging of the QCC is closely related to the messaging of the QCS described in Section 10.3.2.4.2. The QCC in the BP is a counterpart to the QCS in the PS. The QCC performs all of the QoSProfile message exchanges with the PS (as described in Section 10.3.2.4.2) on behalf of the BP, using BP Initiated SOAP Messaging (Section 6.3.3.4.4.2). Thus the QCC obtains priority information for various applications and application-sessions on the BP. The QCC maintains an internal database to store the application priority information that it receives from the QCS, and uses this information to prioritize its application streams.

The QCC is also responsible for mapping the IPCable2Home Generic Priority of the application packet to the IPCable2Home Media Access Priority, using the number of media access priorities supported by the BP interface, as specified in Section 10.2.2.6.3.

The QCC is responsible for the following two main functions in the BP:

- LAN Information Exchange
- Prioritized Media Access for BP applications

Note: The rest of Section 10.4.1.4 is devoted to describing these two key functions of the QCC.

10.4.1.4.1 LAN Information Exchange

As described in Section 10.3.2.4.2, a BP is required to communicate its applications and sessions (destination IP and port) information to the PS in order to obtain their priorities. After the PS sends priority information to the BP, it stores this information in its database and uses it for prioritized media access. This BP is required to send its information to the PS, using QoSProfile XML schema (described below in Section 10.3.2.4.2.1) and BP Initiated SOAP Messaging (BP_Init Operation), as described in Section 6.3.3.4.4.2.

10.4.1.4.1.1 BP information to the PS using BP_Init Message

A BP is always required to convey its information to the PS in the QoSProfile XML schema format (Table 10-6), using BP_Init Message, as described in Section 6.3.3.4.4.2.1. A BP always sends its entire QoSProfile schema to the PS. As described in Section 10.3.2.4.2.2, a BP is required to send BP_Init Message with its entire QoSProfile schema to the PS on the following three different occasions:

- DHCP lease acquisition or renewal
- Application update (addition or deletion) in a BP
- Establishment or termination of application-session with another device by a BP

10.4.1.4.1.1.1 BP device and application information to the PS upon BP DHCP lease acquisition or renewal

After a BP receives DHCPACK message [RFC 2131] addressed to itself, either at the time of DHCP lease acquisition or DHCP lease renewal, it is required to send its device and application priority information to the PS, using BP_Init Message. The BP device information is sent using Device Profile XML schema (defined in Section 6.5.3.1.4), and application priority information is sent using QoSProfile XML schema.

The BP Device Profile sent to the PS contains a number of media access priorities (XML element: numberMedia

AccessPriorities) supported by an interface on a BP. This information exchange and processing is described in Section 6.5.3.3, “MBP Discovery Function,” on page 91. Using this information, the PS populates cabhPsDevBpNumberInterfacePriorities [Annex E.4] MIB, which is a part of the cabhPsDevBpProfileTable [Annex E.4] MIB.

The BP QoSProfile sent to the PS after BP DHCP lease acquisition or lease renewal, contains a list of applications on the BP (QoSApplicationListEntry). It may also optionally contain destination IP address and port specific entries (DestPriorityListEntry) associated to an application. This information is formatted according to the QoSProfile XML schema, as described in Table 10-6. The BP may optionally provide values for DefaultCHPriority and IpPortPriority XML elements.

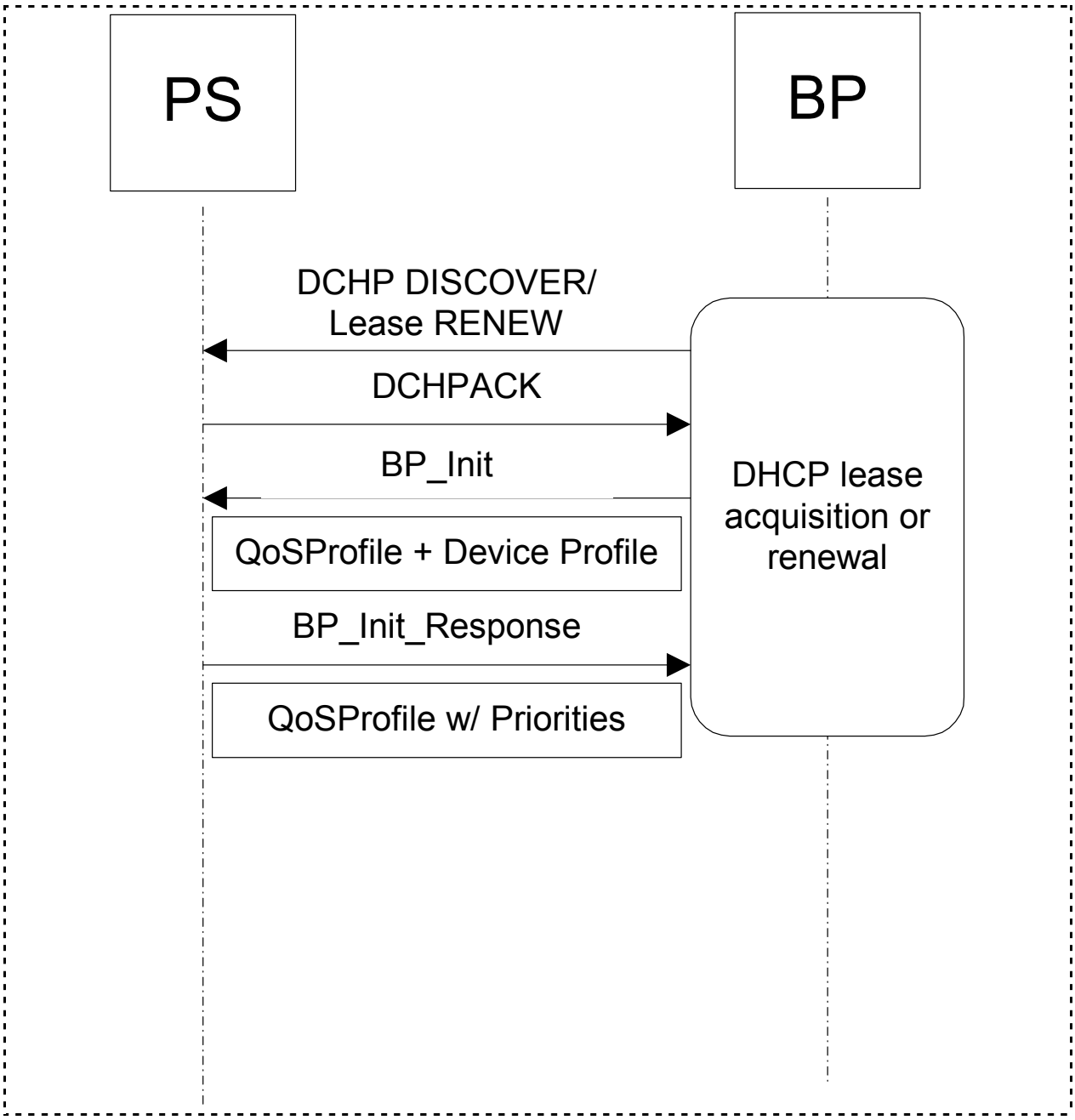

Figure 10-3 Information Exchange upon BP Lease Acquisition or Renewal

10.4.1.4.1.1.2 BP application information to the PS upon application update in the BP

When a new application is added on the BP, the BP adds an entry for this application (QoSApplicationListEntry) in

its existing QoSProfile XML schema. The BP may also optionally populate the DefaultCHPriority element associated with this ApplicationId in the QoSProfile. It may also include DestPriorityListEntry sequence for this ApplicationId. The BP is then required to send this new QoSProfile XML schema to the PS using BP_Init Message.

When an application is removed from the BP, the BP is required to delete all the entries (QoSApplicationListEntry as well as DestPriorityListEntry) related to that particular application from its QoSProfile. The BP is then required to send this modified QoSProfile to the PS using BP_Init Message.

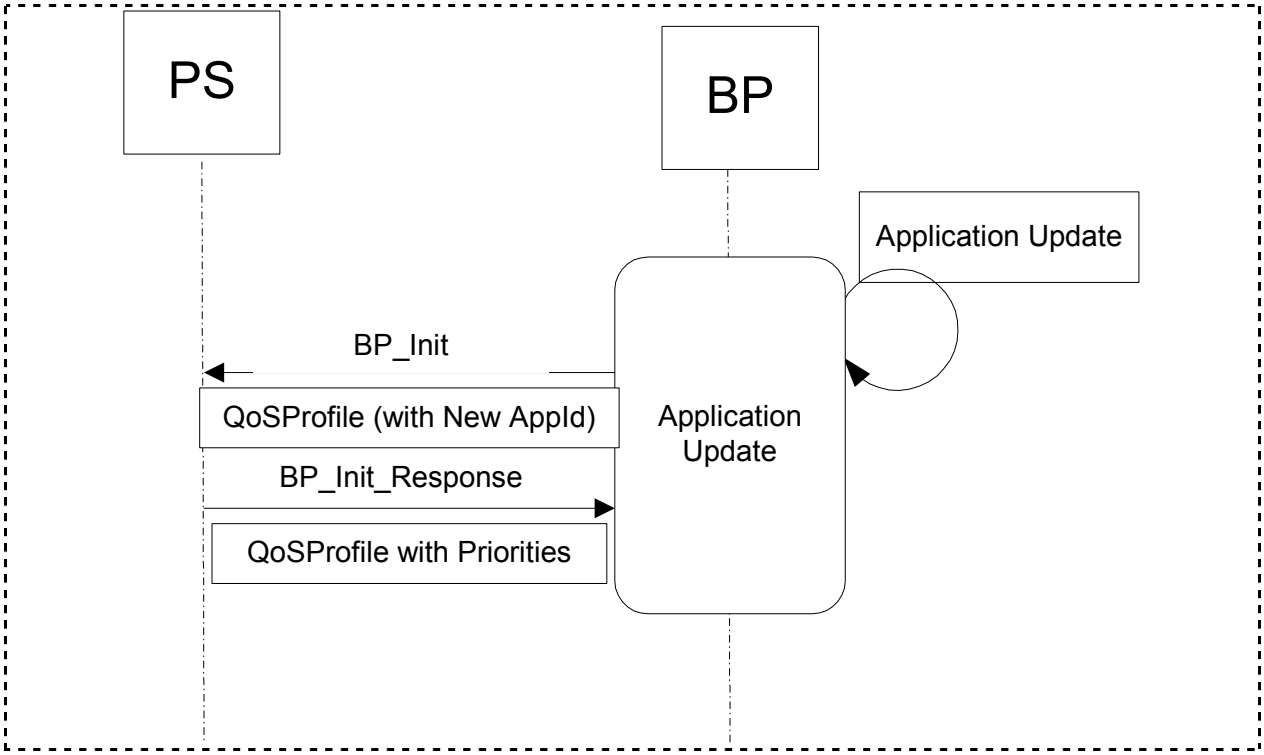

Figure 10-4 Information Exchange upon BP Application Update

10.4.1.4.1.1.3 BP application information to the PS upon application-session establishment or termination

After an application on a BP establishes a session with another device, the BP adds session's destination IP and destination port information, (DestPriorityListEntry) associated to that application (Application ID) in its QoSProfile XML schema (Table 10-6). The BP may optionally populate the IpPortPriority element in the DestPriorityListEntry. The BP then sends this QoSProfile XML schema to the PS using BP_Init Message so that the PS can create entries in its classifier table (cabhPriorityQosBpDestTable), after identifying a priority (IpPortPriority) for the entry by using the priority master table. These classifier entries are utilized by the QFM functionality in the PS to determine the priorities of the packets by examining their destination IP and port; (if they happen to pass through the PS). Using these entries in the classifier table, the QFM performs prioritized queuing and prioritized media access as described in Section 10.3.1.4.

Once the BP terminates a session, the BP deletes the corresponding destination IP and port specific entry, DestPriorityListEntry, from its QoSProfile XML schema and sends this updated QoSProfile to the PS using BP_Init Message so that the PS can delete the entries in its classifier table.

These destination IP and port specific entries in the PS classifier table (cabhPriorityQosBpDestTable) can be used for providing prioritized packet forwarding and prioritized media access for the traffic going from the PS to a non-compliant sink-only device.

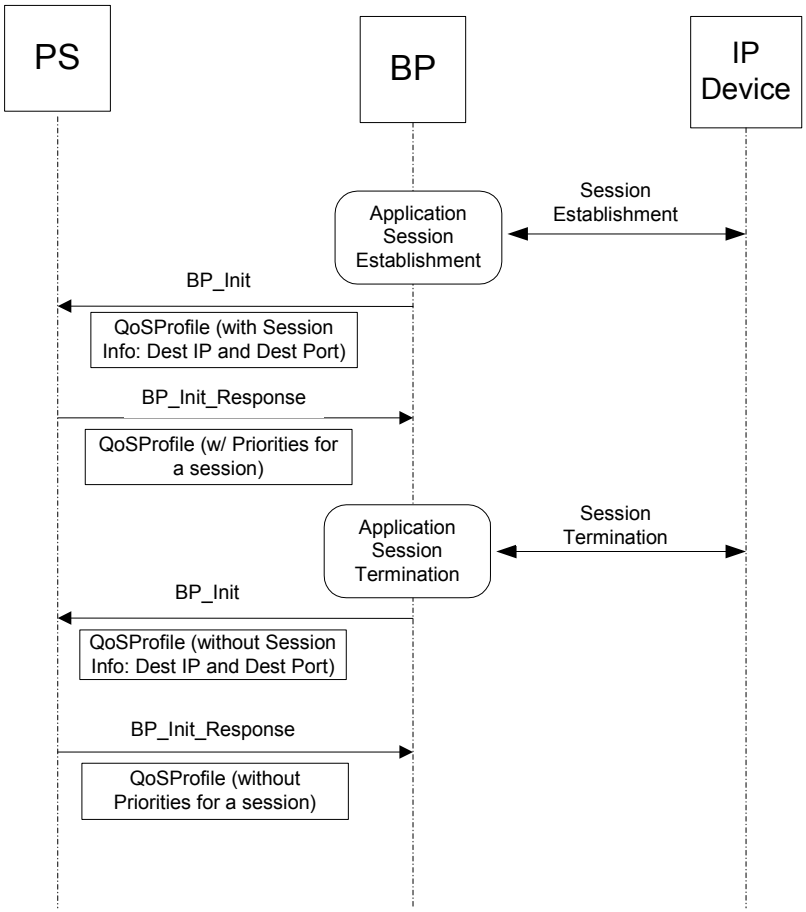

Figure 10-5 Information Exchange upon BP Session Establishment & Termination

10.4.1.4.1.2 Reception of priorities information from the PS in the BP_Init_Response

A BP receives priorities information for its applications (DefaultCHPriority) and application-sessions (IpPortPriority) in the BP_Init_ Response Message from the PS in the QoSProfile XML schema format. From the perspective of a BP, the process of receiving and processing of the QoSProfile XML schema, after it receives BP_Init_Response Message from the PS, is exactly the same for all the three occasions (as mentioned above in Section 10.4.1.4.1.1), when it sends BP_Init Message.

Upon receipt of this information, the BP completely replaces its previously stored QoSProfile XML schema with the newly received QoSProfile in its database. The BP uses the priority information supplied in this QoSProfile XML schema to determine priorities for its applications (Application Id) and application-sessions (identified by destination IP and destination Port).

10.4.1.4.2 Prioritized Media Access

The BP uses application priority information that it receives from the PS in QoSProfile XML schema (Table 10-6) to identify a IPCable2Home Generic Priority for all packets to be transmitted on its LAN interface. If destination IP address and port number for an application-packet matches with the DestIP and DestPort of any of the DestPriorityListEntry sequences in the QoSProfile XML schema, the BP uses a priority value specified by IpPortPriority of that DestPriorityListEntry sequence as IPCable2Home Generic Priority for that packet. Otherwise, the BP uses DefaultCHpriority corresponding to CHApplicationId as a IPCable2Home Generic Priority for the packet. The BP maps this IPCable2Home Generic Priority of the packet to a IPCable2Home Media Access Priority as specified in Section 10.2.2.6.3, using the numberMediaAccessPriorities element of the BP Device Profile XML schema (Section 6.5.3.1). The BP then transmits the packet through its shared media technology in such way that packet's relative preferential access to the shared media as required by IPCable2Home Media Access Priority value, is maintained.

10.4.1.5 QoS Characteristics Client Requirements

10.4.1.5.1 LAN Information Exchange Requirements

This section specifies BP requirements for the information exchange that it needs to perform in order to obtain priorities information from the PS for its applications and sessions.

10.4.1.5.1.1 BP Information to the PS using BP_Init Message

In order to receive their priorities information, a BP MUST communicate its applications and sessions (destination IP and port) information to the PS in the QoSProfile XML schema format (Table 10-6) using BP_Init Message, as described in Section 6.3.3.4.4.2.1. A BP MUST send BP_Init Message with its entire QoSProfile schema to the PS on the following three different occasions:

- DHCP lease acquisition or renewal
- Application update (addition or deletion) in a BP
- Establishment or termination of application-session with another device by a BP

10.4.1.5.1.1.1 BP device & application information to the PS upon BP DHCP lease acquisition/renewal

After a BP receives DHCPACK message [RFC 2131] addressed to itself, either at the time of DHCP lease acquisition or DHCP lease renewal, the BP is required to send its device and application priority information to the PS using BP_Init Message as specified in Section 6.5.3.3.4

The BP MUST include its list of applications (QoSApplicationListEntry) in the QoSProfile sent to the PS after its DHCP lease acquisition or renewal. The BP MAY include destination IP address and port specific entries (DestPriorityListEntry) associated to an application in this QoSProfile. The BP MAY also provide values for DefaultCHPriority and IpPortPriority XML elements in this QoSProfile.

10.4.1.5.1.1.2 BP application information to the PS upon application update in the BP

When a new application is added on the BP, the BP MUST add an entry for this application (QoSApplicationListEntry) in its existing QoSProfile XML schema. The BP MAY optionally populate the DefaultCHPriority element associated with this ApplicationId in the QoSProfile. The BP MAY also include DestPriorityListEntry sequence for this ApplicationId.

When an application is removed from the BP, the BP MUST delete all the entries (QoSApplicationListEntry as well as DestPriorityListEntry) related to that particular application from its QoSProfile.

After such update to the QoSProfile XML schema, the BP is then required to send this new QoSProfile XML schema to the PS using BP_Init Message.

10.4.1.5.1.1.3 BP application information to the PS upon application-session establishment or termination

When an application on a BP establishes a session with another device, the BP MUST add the session's destination IP and destination port information (DestPriorityListEntry) associated to that application (Application ID) in its QoSProfile XML schema (Table 10-6). The BP MAY "wild card" (0) the DestPort element. The BP MUST NOT "wild card" the DestIP element. The BP MAY optionally populate the IpPortPriority element in the DestPriorityListEntry.

When an application on a BP terminates a session, the BP MUST delete the corresponding destination IP and port specific entry, DestPriorityListEntry, from its QoSProfile XML schema.

After such update to the QoSProfile XML schema, the BP is then required to send this new QoSProfile XML schema to the PS using BP_Init Message so that the PS can update (add/delete) the entries in its classifier table (cabhPriorityQosBpDestTable).

These destination IP and port specific entries in the PS classifier table (cabhPriorityQosBpDestTable) MAY be used for providing prioritized packet forwarding and prioritized media access for the traffic going from the PS to a non-compliant sink-only device.

10.4.1.5.1.2 Priorities information from the PS to a BP in the BP_Init_Response

A BP MUST be able to process priorities information for its applications (DefaultCHPriority) and application-sessions (IpPortPriority) that it receives from the PS in the QoSProfile XML schema format (Table 10-6) using BP_Init_ Response Message. Upon receipt of this information, the BP MUST completely replace its previously stored QoSProfile XML schema with the newly received QoSProfile XML schema.

10.4.1.5.2 Prioritized Media Access Requirements

The BP MUST use application priority (DefaultCHPriority or IpPortPriority) information that it receives from the PS in QoSProfile XML schema (Table 10-6) to identify a IPCable2Home Generic Priority for all packets to be transmitted on its LAN interface. If destination IP address and port number for an application-packet matches with the DestIP and DestPort of any of the DestPriorityListEntry sequences in the QoSProfile XML schema, then the BP MUST use a priority value specified by IpPortPriority of that DestPriorityListEntry sequence as IPCable2Home Generic Priority for that packet. Otherwise, the BP MUST use DefaultCHpriority corresponding to CHApplicationId as a IPCable2Home Generic Priority for the packet. The BP MUST map this IPCable2Home Generic Priority of the packet to a IPCable2Home Media Access Priority as specified in Section 10.2.2.6.3, using the numberMediaAccessPriorities element of the BP Device Profile XML schema (Section 6.5.3.1). The BP then MUST transmit the packet through its shared media technology in such way that the packet's relative preferential access to the shared media, as required by IPCable2Home Media Access Priority value, is maintained.

11 SECURITY

11.1 Introduction/Overview

This section defines the security interfaces, protocols and functional requirements needed to secure the PS and its operations.

The delivery of reliable multi-media IP services to client devices on a home network requires a secure residential gateway along with the security mechanisms to protect these services from illegal access, monitoring, and disruption. The purpose of any security technology is to protect value, including revenue based services. Threats to a revenue stream exist when a user of the network perceives the value, expends effort and money, and invents a technique to get around making the necessary payments (See Annex C). Some network users will go to extreme lengths to steal when value is perceived. The addition of security technology to protect value has an associated cost; the more money expended, the greater the security (security effectiveness is thus basic economics).

The security architecture focuses on securing the LAN from network attacks as well as securing communications between the PS and the Headend servers. The PS functionality can provide the foundation for other applications and services served by the cable operator to the home LAN. Security can exist for these applications independent of the IPCable2Home security architecture. IPCablecom specifies interfaces for a multi-media applications and has its own security architecture. For all references to IPCablecom security, please refer to [J.170].

11.1.1 Goals

The goals for the security model include:

- Employ a cost effective security technology to force any user with the intent to steal or disrupt network services to spend an unreasonable amount of money or time.
- Secure the IPCable2Home network used to offer high value cable-based services so that it is at least as secure as the CableModem and IPCablecom technologies on the hybrid fiber-coax (HFC) network.
- Where possible, align security mechanisms with the CableModem and IPCablecom security Recommendations.
- From the LAN, it is the intent of security architecture to assist the operator with a secure identity to make it hard for the average subscriber to gain unauthorized access to the HFC network and cable-based services.

11.1.2 Assumptions

The assumptions for the security environment include:

- It is assumed the Embedded PS, has a J.112 or J.122 cable modem.
- The home network includes less security for low value services.
- Back office configurations are not specified and IPCable2Home assumes minimal configurations by the cable operator to operate in the specified modes.

11.2 Security Architecture

The e security architecture is based on the architecture as defined in the Reference Architecture Section 5 of this Recommendation. The architecture defines a Portal Services (PS) element, which includes Management, Provisioning, Security, and QoS functions.

The architecture also includes the following set of Headend elements: Cable Modem Termination System (CMTS), Dynamic Host Configuration Protocol (DHCP) [RFC 2131] server, Network Management System, Trivial File Transfer Protocol (TFTP) server in the cable network, TFTP client in the PS, Hypertext Transfer Protocol (HTTP) server in the cable network, HTTP client in the PS, Transport Layer Security (TLS) [RFC 2246] server in the cable network, TLS client in the PS, and a Key Distribution Center (KDC) server in the cable network.

The security architecture focuses on securing the LAN from network attack, as well as securing communications between the PS and the Headend servers.

11.2.1 System Design Guidelines

The security design requirements are listed below in Table 11-1. This list provided guidance for the development of the security architecture.

Table 11-1 Security System Design Guidelines

Reference	Security System Design Guidelines
SEC1	Include the design necessary to communicate the authentication credentials for elements.
SEC2	Authentication credentials for PS and critical back office servers will be provided. The credentials will define specific usage and ensure a source of trust.
SEC3	Network management messages between the cable Headend and PS can be authenticated and optionally encrypted to protect against unauthorized monitoring and control.
SEC4	The firewall will accept configuration files in a standard language and format.[3]
SEC5	The cable operator will have the ability to remotely manage compliant firewall products through configuration file or SNMP commands
SEC6	The firewall will include a default set of rules for an expected minimum set of functionality.
SEC7	Provide the necessary support for IPCablecom through the firewall.
SEC8	A minimum set of requirements will be placed on the firewall filtering capabilities for packet, port, IP addresses, and time of day
SEC9	A detailed firewall event logging interface will allow the cable operator to monitor and review firewall activity as configured.
SEC10	The firewall will support commonly used applications in specific scenarios.
SEC11	The firewall will protect the LAN and WAN from common network attacks.

[3]1. The Firewall Configuration File Requirements are defined in Section 7.4 PS Function - Bulk Portal Services Configuration (BPSC)

SEC12	The management of the events and rulesets for the firewall will be defined in detail via the Security MIB.
SEC13	The cable operator will have the ability to securely download software images to the PS element.
Sec14	The cable operator will have the ability to authenticate and optionally encrypt the transport of configuration files for the PS or firewall.

This section limits the scope of the specified security architecture to meet these primary system security requirements. However, it is acknowledged that in some cases additional security is desired and can be added by the cable operator as needed. The concerns of individual cable operators or manufacturers can result in additional security protections. This specification does not restrict the use of further protections, as long as they do not conflict with the intent and guidelines of this specification.

11.2.2 System Description

The security architecture includes the following security elements:

- Security-Domain
- Portal Services function (PS)
- **Cable** Security Portal function (CSP)
- Firewall (FW)
- Key Distribution Center (KDC)
- HTTPS Server with TLS

The architecture defines the PS Element within the residential gateway. Security exists only in a few of the specified interfaces, as the System Design Guidelines require. Figure 11-1 illustrates the relationship between the various elements which contain security.

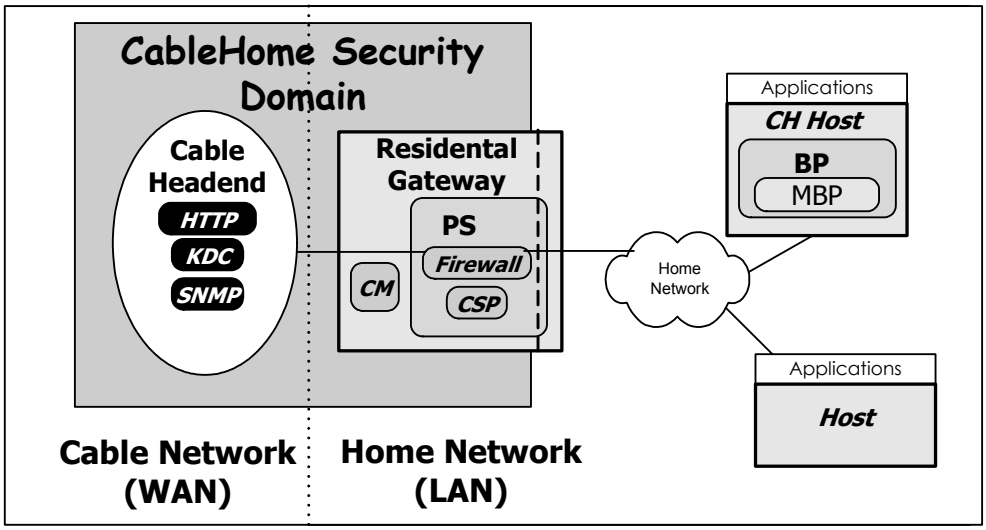

Figure 11-1 IPCable2Home Security Elements

11.2.2.1 Security Domain

The Security Domain is defined in Figure 11-1, and encompasses the PS element in the residential gateway and the illustrated Headend servers, with specified security. The Security Domain defines the boundary of the sphere of direct influence where security functionality is extended to the residential gateway from the cable network's Headend. The PS element is wholly within the Security Domain, with the exception of the LAN side USFS functionality. The CSP and Firewall act as the boundary elements between the Security-Domain and the non-secure domain.

11.2.2.2 PS Related Security Sub-Elements

The PS includes the following security elements:

- Cable Security Portal (CSP)
- Firewall (FW)

The CSP acts as a security portal for other PS sub-elements such as negotiating the SNMPv3 keys either through Diffie-Helman or Kerbero, as required. The CSP ensures there is security for SNMPv3 between the NMS and the PS, when turned on by the cable operator. The CSP provides the ability to validate and verify digital certificates for the purposes of authentication and encryption. The CSP initiates, manages, and closes a TLS session for secure downloading of the PS configuration file and firewall configuration file, if instructed by the cable operator during the DHCP exchange.

The PS firewall functionality provides protection to the user, as well as the HFC network, from unwanted traffic coming from the WAN, LAN, or PS address realms. Such traffic can include deliberate attacks on the in-home network, as well as traffic limiting for parental control applications. The security requirements include specific rules for remote management by the cable operator.

11.2.2.3 Key Distribution Center (KDC) Server

The Key Distribution Center (KDC) server is required if the cable operator deploys IPCable2Home with SNMP provisioning mode. If a KDC server is available in the Headend, it will be used to provide mutual authentication and key distribution services with the use of the Kerberos protocol. If available, the KDC will communicate with the CSP function to establish these services.

11.3 PS Device Authentication Infrastructure

This section describes authentication for the PS device and its communication to the KDC and the HTTPS server.

11.3.1 Device Authentication Infrastructure Goals

It is important to establish the secure identity of the PS element to assist with the following goals:

- Reduce the possibility of device and software cloning, as well as theft of service. The gateways are in a widely distributed environment where the consumer has in-home physical access to the gateway. Providing a secure identity reduces risk of tampering with the gateway hardware device.
- Establish the source of trust. The PKI provides an established source of trust which is rooted within the Manufacturer base.

11.3.2 Authentication Infrastructure System Design Guidelines

Table 11-2 Authentication Infrastructure System Design Guidelines

Reference	Security System Design Guidelines
SEC1	Include the design necessary to communicate the authentication credentials for IPCable2Home elements.
SEC2	Authentication credentials for CPE and critical back office servers will be provided. The credentials will define specific usage and ensure a source of trust.

11.3.3 Authentication Infrastructure System Description

For security purposes, it is important to know with whom you are communicating prior to exchanging any meaningful information. Authentication provides a secure identity. There are three parts to authentication: the identity credential, the checking of the identity credential for validity, and the common means to securely communicate the identity information. An industry standard identification credential, X.509 certificates, in conjunction with [RFC 3280] for certificate use, and Kerberos, which is a common communications protocol for mutual authentication is specified. X.509 certificates are exchanged between the PS Element and the KDC during the Kerberos PKINIT exchange, which is wrapped in the AS Request and AS Reply messages. The PS Element

Certificate provides the identity of the associated PS Element by cryptographically binding the PS Element WAN-Man MAC address to a public key certificate. Each side validates the information in the certificate and verifies the certificate chain back to the root for each chain. Once the trust has been established, the information for the SNMPv3 keys is sent from the KDC to the PS Element. This authentication section describes the use of Kerberos and X.509 certificates.

11.3.4 Authentication Infrastructure Requirements

11.3.4.1 Element Authentication via Kerberos

Authentication is specified when a KDC that supports IPCable2Home, is available in the Headend. If a KDC is available, it is recommended that the cable operator provision the PS Element in SNMP Provisioning Mode (as described in Section 5.5), to take advantage of the specified mutual authentication protocol with the use of Kerberos, using the PKINIT extension. Kerberos provides a protocol to secure mutual authentication in order to provide keying material and communication establishment only between authenticated parties on the IPCable2Home network. Because this authentication model has been specified by another ITU project, i.e., IPCablecom, IPCable2Home references the IPCablecom model when appropriate.

Various Kerberos MIB objects are required by IPCablecom Some MIB objects to cover the Kerberos functionality needed by IPCable2Home have been defined. These MIB objects are defined in the Security MIB and described in the MIB Object sections of this chapter.

Communication between the KDC and PS is initiated by the PS immediately after the DHCP options are processed during provisioning, if the DHCP options require the PS to initiate communication to the KDC. The DHCP options specified in Section 7.3.3.2.4 require option 177, sub-option 51, which contains the value for the KDC's IP Address to be included with the other DHCP options, and MUST be used by the PS to establish communication between the PS and KDC. Even though IPCablecom requires a DNS name as part of the DHCP options, DNS is not required for IPCable2Home and, therefore, the IP address of the KDC is required for the PS to be able to find the appropriate KDC.

11.3.4.1.1 Kerberos/PKINIT

When the PS Element is provisioned in SNMP Provisioning Mode, the use of Kerberos with the PKINIT public key extension for authenticating IPCable2Home elements and supporting key management requirements is specified. IPCable2Home elements (clients) authenticate themselves to the KDC with the PKINIT protocol. Once authenticated to the KDC, clients will receive a Kerberos ticket for authenticating themselves to a particular server.

In SNMP provisioning mode, the PS Element, the NMS (i.e., SNMP Manager) and KDC MUST follow the specification for Kerberos/PKINIT, as defined in [J.170] sections 6.4 and 6.5, unless otherwise noted in this specification. The IPCable2Home KDC is equivalent to or the same as the IPCablecom MSO KDC (IPCablecom specifies the use of several KDCs). The IPCable2Home specification uses the term Network Management Systems (NMS) to provide SNMP functionality. In referencing the IPCablecom suite of specifications, it is noted that IPCablecom uses the term provisioning server to denote SNMP functionality. This SNMP functionality in general is be compatible within both specifications. However, they are not identical as IPCablecom and IPCable2Home specific information is specified. The PS element MUST act as the client to the KDC. In the IPCablecom Security Specification, the MTA is the client and is expected that IPCable2Home implementations will use the client functionality specified for the MTA, for the PS element. The PS element makes use of Kerberos for SNMP key management, as well as for device authentication. The certificates used in PKINIT for IPCable2Home are specified in the PKI Section of this document. Where IPCablecom specifies an MTA device certificate, IPCable2Home provides a certificate for the PS Element (PS Element Certificate), and implementations of PS Elements MUST include the PS Element Certificate.

The following sections for Kerberos functionality from [J.170] do not apply to IPCable2Home:

- section 6.4.2.1.3 Pre-Authenticator for Provisioning Sever Location
- section 6.4.6 MTA Principal Names
- section 6.4.7 Mapping of MTA MAC Address to MTA FQDN
- section 6.4.9 Service Key Versioning
- section 6.4.10 Kerberos Cross-Realm Operation

- section 6.5.2.1 Rekey Messages
- section 6.5.3 Kerberized IPSec
- section 6.4.5 Kerberos Server Locations and Naming Conventions

11.3.4.1.2 IPCable2Home Specific Authentication Variables

The model IPCablecom specifies includes some specific variable names for Kerberos in the IPCablecom Network Architecture. In order for IPCable2Home to use the IPCablecom model, the following variable names MUST to be changed:

- Replace pktcKdcToMtaMaxClockSkew as defined in the IPCablecom Security Spec, with KdcToClientMaxClockSkew.
- Replace pktcSrvrToMtaMaxClockSkew as defined in the IPCablecom Security Spec, with SrvrToClientMaxClockSkew.
- Replace mtaprovsrvr as defined in the IPCablecom Security Specification, with provsrvr.

IPCable2Home Kerberos implementations MUST ignore the Object Identifier (OID) field portion, which reads clabProjIPCablecom (2), within the AppSpecificTypedData, within the KRB-ERROR messages.

11.3.4.1.3 profile for Kerberos Server Locations and Naming Conventions

Kerberos Realm names MAY use the same syntax as a domain name. However, Kerberos Realms MUST be in all capitals. Kerberos Realm details MUST be followed according to [J.170], Appendix B.

The KDC conventions listed in [J.170], Section 6.4.5.2 are considered informative with the expectation that the KDC will perform the necessary functions in the back office to exchange the appropriate information with the NMS (provisioning server or SNMP manager). The PS element has provided the KDC with the provisioning server IP address in the AS Request, as the necessary information to make appropriate contact between the KDC and provisioning server.

A PS Element principal name MUST be of type NT-SRV-INST with exactly two components, where the first component MUST be the string "PSElement" (not including the quotes), and the second component MUST be the WAN-Man-MAC address:

PSElement/<WAN-Man-MAC>

where <WAN-Man-MAC> is the WAN Management MAC address of the PS Element. The format the <WAN-Man-MAC> MUST be "XX:XX:XX:XX:XX:XX" (not including the quotes), where X is a hexadecimal character of the MAC address. Hexadecimal characters a-f MUST be in lower case.

A NMS Element principal name MUST be of type NT-SRV-HST with exactly two components, where the first component MUST be the string "provsrvr" (not including the quotes), and the second component MUST be the service provider's SNMP entity address:

provsrvr/<SNMP entity address>

The <SNMP entity address> MUST be the service provider's SNMP entity IP address (CDC DHCP Option 177, sub-option 3) in dotted notation enclosed in square brackets (e.g. [12.34.56.78]).

11.3.4.2 Public Key Infrastructure (PKI)

Public key certificates, which comply with the [ITU-T X.509] specification and the IETF [RFC 3280] are used.

11.3.4.2.1 Generic Certificate Requirements

This section describes what is commonly referred to as the generic structure, since all certificates share these requirements. All certificates specified in this section MUST include the following information:

- **Certificate Version**- The Version of the certificates MUST be [ITU-T X.509], v3, and noted as v2 in the actual certificate. All certificates MUST comply with [RFC 3280], except where the non-compliance with the RFC is explicitly stated in this chapter of this document. Any non-compliance request by this document for content does not imply non-compliance for format. Any specific non-compliance request for format will be explicitly described.

- **Public Key Type** – RSA Public Keys are used throughout the certificate hierarchies described in Section 11.3.4.2.2. The subjectPublicKeyInfo.algorithm OID used MUST be 1.2.840.113549.1.1.1 (rsaEncryption). The public exponent for all RSA keys MUST be F_4 – 65537.
- **Extensions**- The extensions (subjectKeyIdentifier, authorityKeyIdentifier, KeyUsage, and BasicContraints) MUST follow [RFC 3280]. All other certificate extensions, if included, MUST be marked as non-critical. The encoding tags are [c:critical, n:non-critical; m:mandatory, o:optional] and are identified in the table for each certificate.
- **subjectKeyIdentifier** – The subjectKeyIdentifier extension included in all certificates as required by [RFC 3280] (e.g., all certificates except the device and ancillary certificates) MUST include the keyIdentifier value composed of the 160-bit SHA1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length and number of unused bits from the ASN1 encoding) (See [RFC 3280]).
- **authorityKeyIdentifier** – The authorityKeyIdentifier extension included in all certificates as required by [RFC 3280], MUST include the subjectKeyIdentifier from the issuer's certificate (see [RFC 3280]), with the exception of root certificates.
- **KeyUsage** – The keyUsage extension MUST be used for all Certification Authority (CA) certificates and Code Verification Certificates (CVCs). For CA certificates, the keyUsage extension MUST be marked as critical with a value of keyCertSign and cRLSign. For CVC certificates, the keyUsage extension MUST be marked as critical with a value of digitalSignature and keyEncipherment. The end-entity certificates MAY use the keyUsage extension as listed in [RFC 3280].
- **BasicConstraints** – The basicConstraints extension MUST be used for all CA and CVC certificates and MUST be marked as critical. The values for each certificate for basicConstraints MUST be marked as specified in the certificate description Table 11-2 through Table 11-13.
- **Signature Algorithm** – The signature mechanism used MUST be SHA-1 [FIPS 186] with RSA Encryption. The specific OID is 1.2.840.113549.1.1.5.
- **SubjectName and IssuerName** – If a string cannot be encoded as a PrintableString, it MUST be encoded as a UTF8String (tag [UNIVERSAL 12]).

  When encoding an X.500 Name:

- Each RelativeDistinguishedName (RDN) MUST contain only a single element in the set of X.500 attributes.
- The order of the RDNs in an X.500 name MUST be the same as the order in which they are presented in this specification.
- **serialNumber** – The serial number MUST be a unique, positive integer assigned by the CA to each certificate (i.e., the issuer name and serial number identify a unique certificate). Cas MUST force the serialNumber to be a non-negative integer. The Manufacturer SHOULD NOT impose or assume a relationship between the serial number of the certificate and the serial number of the modem to which the certificate is issued.

  Given the uniqueness requirements above, serial numbers can be expected to contain long integers. Certificate users MUST be able to handle serialNumber values up to 20 octets. Conformant Cas MUST NOT use serialNumber values longer than 20 octets.

11.3.4.2.2 Certificate Hierarchies

There are three distinct certificate hierarchies used:

1. Manufacturer Chain is used to identify authorized manufacturers;
2. Code Verification Chain is used to identify compliant software images;
3. Service Provider Chain is used to identify devices on the Service Provider's network for mutual authentication to the subscriber's devices.

The certificate hierarchies described in this document can apply to all related projects needing certificates. Each project can adopt this hierarchy as there is an opportunity to move to a more generic, shared certificate structure.

Also, each project can make specific adjustments in the requirements for that particular project. It is a goal to create a PKI which can be re-used for every project. There can be differences in the end-entity certificates required for each project. However, in the cases where end-entity certificates overlap, one end-entity certificate could be used for several services within the cable infrastructure. For example, IPCablecom requires a KDC for the service provider and IPCable2Home also requires a KDC for the service provider. If the service provider is running both network architectures on their systems, they can use the same KDC and the same KDC certificate for communication on both systems, i.e., IPCablecom and IPCable2Home. In this case, the IPCable2Home KDC is equivalent to the IPCablecom MSO KDC (IPCablecom specifies the use of several KDCs).

In Figure 11-2, the term Certificate Authority is abbreviated as CA and Code Verification Certificate is abbreviated as CVC.

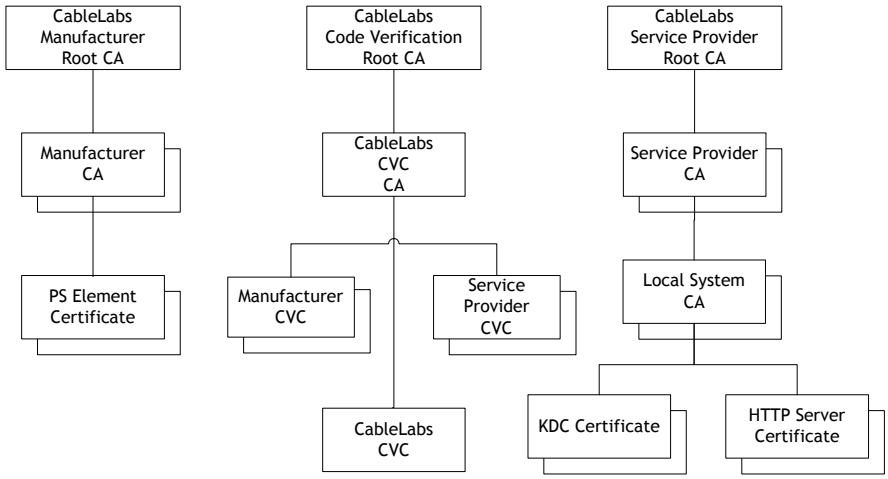

Figure 11-2 IPCable2Home Certificate Hierarchy

11.3.4.2.2.1 Manufacturer Certificate Hierarchy

The Manufacturer certificate hierarchy, or Manufacturer chain, is rooted at a Manufacturer Root CA, which is used to issue Manufacturer Certification Authority (CA) certificates for a set of authorized manufacturers. Manufacturers use their CA to issue individual PS Element Certificates. This chain is used for authentication of devices in the home.

The information contained in the following tables are the specific values for the required fields according to [RFC 3280]. These specific values for the Manufacturer Certificate hierarchy MUST be followed according to Table 11-3, Table 11-4, and Table 11-5. If a required field is not specifically listed in the tables, then the guidelines in [RFC 3280] MUST be followed. The generic extensions MUST also be included as specified in PKI Section 11.3.4.2.

Manufacturer Root CA Certificate

The Manufacturer Root CA Certificate (see Table 11-3) MUST be verified as part of the certificate chain containing the Manufacturer Root CA Certificate, Manufacturer CA Certificate and the PS Element Certificate.

Table 11-3 Manufacturer Root CA Certificate

Manufacturer Root CA Certificate	
Subject Name Form	C=<country> O=<Company Name> CN=[Company Name] Manufacturer Root CA
Intended Usage	This certificate is used to issue Manufacturer CA Certificates.
Signed By	Self-Signed
Validity Period	20+ years
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign), subjectKeyIdentifier [n,m], basicConstraints [c,m](cA=true)

Manufacturer CA Certificate

The Manufacturer CA Certificate MUST be verified as part of a certificate chain containing the Manufacturer Root CA Certificate, Manufacturer CA Certificate, and the PS Element Certificate.

The state/province, city, and manufacturer's facility are optional attributes. A manufacturer MAY have more than one manufacturer's CA certificate. If a manufacturer is using more than one manufacturer CA certificate, the PS element MUST have access to the appropriate certificate as verified by matching the issuer name in the PS Element Certificate with the subject name in the Manufacturer CA Certificate. The authorityKeyIdentifier of the PS Element Certificate MUST be matched to the subjectKeyIdentifier of the manufacturer certificate as described in [RFC 3280].

Table 11-4 Manufacturer CA Certificate

Manufacturer CA Certificate	
Subject Name Form	C=<country> O=<CompanyName> [ST=<state/province>] [L=<city>] OU= <organization unit> [OU=<Manufacturer's Facility>] CN=<CompanyName> Mfg CA
Intended Usage	This certificate is issued to each Manufacturer by a certificate authority Manufacturer Root CA and can be provided to each PS Element either at manufacture time, or during a field code update. This certificate appears as a read-only parameter in the PS element. This certificate issues PS Element Certificates. This certificate, along with the Manufacturer Root CA Certificate and the PS Element Certificate, is used to authenticate the PS element identity. The optional listing for manufacturer's facility can be the facility name and/or facility location.
Signed by	Hierarchy's Manufacturer Root CA
Validity Period	20 Years
Modulus Length	2048

Extensions	keyUsage[c,m](keyCertSign, cRLSign), subjectKeyIdentifier [n,m], authorityKeyIdentifier [n,m], basicConstraints[c,m](cA=true, pathLenConstraint=0)

The Company Name in the Organization (O) field MAY be different than the Company Name in the Common Name (CN) field.

PS Element Certificate

The PS Element Certificate MUST be verified as part of a certificate chain containing the Manufacturer Root CA Certificate, Manufacturer CA Certificate and the PS Element Certificate.

The state/province, city, product name and manufacturer's facility are optional attributes.

The PS Element WAN-Man MAC address MUST be expressed as six pairs of hexadecimal digits separated by colons, e.g., "00:60:21:A5:0A:23". The Alpha HEX characters (A-F) MUST be expressed as uppercase letters.

A PS Element Certificate is permanently installed and not renewable or replaceable. Therefore, the PS Element Certificate has a validity period greater than the expected operational lifetime of the specific device.

Table 11-5 PS Element Certificate

PS Element Certificate	
Subject Name Form	C=<country> O=<Company Name> [ST=<state/province>] [L=<city>] OU=<organization unit> [OU=<Product Name>] [OU=<Manufacturer's Facility>] CN=<WAN-Man MAC Address>
Intended Usage	This certificate is issued by the Manufacturer CA and installed in the factory. The NMS server cannot update this certificate. This certificate appears as a read-only parameter in the PS Element. This certificate is used to authenticate the PS element identity.
Signed By	Manufacturer CA
Validity Period	20+ years
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,m](digitalSignature, keyEncipherment, dataEncipherment), anyExtendedKeyUsage[n,m] (id-kp-clientAuth), authorityKeyIdentifier [n,m]

11.3.4.2.2.2 Code Verification Certificate Hierarchy

The Code Verification Certificate (CVC) hierarchy, or code verification chain, is rooted at a Code Verification Root CA, which issues the Code Verification CA certificate. The Code Verification CA is used to issue CVCs to a set of authorized manufacturers and service providers. Code Verification CA also issues the CVC. This chain is specifically used to authenticate software downloads. The IPCable2Home PKI allows for Manufacturer CVCs, a CVC and Service Provider CVCs.

The information contained in the following tables are the specific values for the required fields according to [RFC 3280]. These specific values for the Code Verification Certificate hierarchy MUST be followed according to Table 11-6, Table 11-7, Table 11-8, Table 11-9, and Table 11-10 below. If a required field is not specifically listed in the tables, the guidelines in [RFC 3280] MUST be followed. The generic extensions MUST also be included as specified in PKI Section 11.3.4.2.

Code Verification Root CA Certificate

This certificate MUST be verified as part of the certificate chain containing the Code Verification Root CA Certificate, the Code Verification CA, and the Code Verification Certificates.

Table 11-6 Code Verification Root CA Certificate

Code Verification Root CA Certificate	
Subject Name Form	C=<country> O=<Company Name> CN= [Company Name] CVC Root CA
Intended Usage	This certificate is used to sign Code Verification CA Certificates
Signed By	Self-signed
Validity Period	20+ years
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign), subjectKeyIdentifier [n,m], basicConstraints [c,m](cA=true)

Code Verification CA Certificate

The Code Verification CA Certificate MUST be verified as part of a certificate chain containing the Code Verification Root CA Certificate, Code Verification CA Certificate, and the Code Verification Certificate. A Stand-Alone PS MUST only support one CVC CA at a time.

Table 11-7 Code Verification CA Certificate

Code Verification CA Certificate	
Subject Name Form	C=<country> O=<Company Name> CN= [Company Name] CVC CA
Intended Usage	This certificate is issued to the certificate authority by the Code Verification Root CA. This certificate issues Code Verification Certificates.
Signed By	Hierarchy's Code Verification Root CA
Validity Period	20 years
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign), subjectKeyIdentifier [n,m], authorityKeyIdentifier [n,m], basicConstraints [c,m](cA=true, pathLenConstraint=0)

Manufacturer Code Verification Certificate

This certificate MUST be verified as part of the certificate chain containing the Code Verification Root CA Certificate, Code Verification CA Certificate, and Code Verification Certificates.

Table 11-8 Manufacturer Code Verification Certificate

Manufacturer Code Verification Certificate	
Subject Name Form	C=<country> O=<CompanyName> [ST=<state/province>] [L=<city>] CN=<CompanyName> Mfg CVC
Intended Usage	The Code Verification CA issues this certificate to each authorized Manufacturer. It is used in the policy set by the cable operator for secure software download. The CompanyName in the O and CN fields may be different.
Signed By	Code Verification CA
Validity Period	up to 10 years
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,m](digitalSignature, keyEncipherment), extendedKeyUsage [c,m] (id-kp-codeSigning), authorityKeyIdentifier [n,m]

Code Verification Certificate

The Code Verification Certificate MUST be verified as part of a certificate chain containing the Code Verification Root CA Certificate, Code Verification CA Certificate, and Code Verification Certificate.

Table 11-9 Code Verification Certificate

Code Verification Certificate	
Subject Name Form	C=<country> O=<Company Name> CN= <Company Name>CVC
Intended Usage	The Code Verification CA issues this certificate. It is used to authenticate certified code. It is used in the policy set by the cable operator for secure software download.
Signed By	Code Verification CA
Validity Period	up to 10 years
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,m](digitalSignature, keyEncipherment), extendedKeyUsage [c,m] (id-kp-codeSigning), authorityKeyIdentifier [n,m]

Service Provider Code Verification Certificate

The Service Provider Code Verification Certificate MUST be verified as part of a certificate chain containing the Code Verification Root CA Certificate, Code Verification CA Certificate, and Service Provider Code Verification Certificate.

Table 11-10 Service Provider Code Verification Certificate

Service Provider Code Verification Certificate	
Subject Name Form	C=<country> O=<CompanyName> [ST=<state/province>] [L=<city>] CN=<CompanyName> Service Provider CVC
Intended Usage	The Code Verification CA issues this certificate to each authorized Service Provider. It is used in the policy set by the cable operator for secure software download. The CompanyName in the O and CN fields may be different.
Signed By	Code Verification CA
Validity Period	up to 10 years
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,m](digitalSignature, keyEncipherment), extendedKeyUsage [c,m] (id-kp-codeSigning), authorityKeyIdentifier [n,m]

11.3.4.2.2.3 Service Provider Certificate Hierarchy

The Service Provider certificate hierarchy, or Service Provider chain, is rooted at a Service Provider Root CA, which is used to issue certificates for a set of authorized Service Providers. The Service Provider CA can be used to issue optional Local System CA Certificates or ancillary certificates. If the Service Provider CA does not issue the ancillary certificates, the Local System CA will. The ancillary certificates are the end entity certificates on the cable operator's network.

The information contained in the following tables are the specific values for the required fields according to [RFC 3280]. These specific values for the Service Provider Certificate hierarchy MUST be followed according to Table 11-11 through Table 11-14 below. If a required field is not specifically listed in the tables, the guidelines in [RFC 3280] MUST be followed. The generic extensions for IPCable2Home MUST also be included as specified in PKI Section 11.3.4.2.

Service Provider Root CA Certificate

This certificate MUST be verified as part of the certificate chain containing the Service Provider Root CA Certificate, Service Provider CA Certificate, optional Local System CA Certificate, and Ancillary Certificates.

Table 11-11 Service Provider Root CA Certificate

Service Provider Root CA Certificate	
Subject Name Form	C=<country> O=<Company Name> CN=<Company Name> Service Provider Root CA
Intended Usage	This certificate is used to issue Service Provider CA Certificates.
Signed By	Self-signed
Validity Period	20+ years
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign), subjectKeyIdentifier [n,m], basicConstraints [c,m](cA=true)

Service Provider CA Certificate

The Service Provider CA certificate MUST be verified as part of the certificate chain containing the Service

Provider Root CA Certificate, Service Provider CA Certificate, optional Local System CA Certificate, and Ancillary Certificates.

Table 11-12 Service Provider CA Certificate

Service Provider CA Certificate	
Subject Name Form	C=<country> O=<CompanyName> CN=<CompanyName> Service Provider CA
Intended Usage	The Service Provider Root CA issues this certificate to each Service Provider. In order to make it easy to update this certificate, each network element is configured with the OrganizationName attribute of the Service Provider CA Certificate SubjectName. This is the only attribute in the certificate that must remain constant. This certificate appears as a read-write parameter in the MIB object that identifies the OrganizationName attribute for the IPCable2Home Kerberos realm. The IPCable2Home element does not accept Service Provider certificates that do not match this value of the OrganizationName attribute in the SubjectName. If the Headend contains a KDC that supports IPCable2Home, then the PS element needs to perform the first PKINIT exchange with the KDC right after a reboot, at which time its MIB tables are not yet configured. At that time, the IPCable2Home Kerberos client MUST accept any Service Provider OrganizationalName attribute, but it MUST later check that the value added into the MIB object for this realm is the same as the one in the initial PKINIT reply. This CA issues Local System CA certificates or ancillary certificates.
Signed By	Service Provider Root CA
Validity Period	20 years
Modulus Length	2048
Extensions	keyUsage[c,m](keyCertSign, cRLSign), subjectKeyIdentifier [n,m], authorityKeyIdentifier [n,m], basicConstraints[c,m](cA=true, pathLenConstraint=1)

The Company Name in the Organization (O) field MAY be different than the Company Name in the Common Name (CN) field.

Local System CA Certificate

This certificate is optional for the service provider. If this certificate exists, it MUST be verified as part of the certificate chain containing the Service Provider Root CA Certificate, Service Provider CA Certificate, optional Local System CA Certificate, and Ancillary Certificates.

Table 11-13 Local System CA Certificate

Local System CA Certificate	
Subject Name Form	C=<country> O=<CompanyName> OU=<Local System Name> CN=<CompanyName> Local System CA
Intended Usage	This certificate is optional, and if it exists, is issued by the Service Provider CA. This CA issues ancillary certificates. Network servers are allowed to move freely between regional Cas of the same service provider.
Signed By	Service Provider CA
Validity Period	20 years
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,m](keyCertSign, cRLSign), subjectKeyIdentifier [n,m], authorityKeyIdentifier [n,m], basicConstraints[c,m](cA=true, pathLenConstraint=0)

The Company Name in the Organization (O) field MAY be different than the Company Name in the Common Name (CN) field.

KDC Certificate

This certificate MUST be verified as part of the certificate chain containing the Service Provider Root CA Certificate, Service Provider CA Certificate, optional Local System CA Certificate, and Ancillary Certificates (e.g., the KDC Certificates).

The KDC Certificate MUST include the Kerberos PKINIT subjectAltName as specified in [J.170] subsection "Key Distribution Center Certificate."

Table 11-14 KDC Certificate

KDC Certificate	
Subject Name Form	C=<country> O=<Company Name> [OU=<Local System Name>] OU= <Company Name> Key Distribution Center CN=<DNS Name>
Intended Usage	This certificate is issued either by the Service Provider CA or the Local System CA. It is used to authenticate the identity of the KDC to the Kerberos clients during PKINIT exchanges. This certificate is passed to the PS element inside the PKINIT reply.
Signed By	Service Provider CA or the Local System CA
Validity Period	20 years
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,o](digitalSignature) authorityKeyIdentifier[n,m](keyIdentifier=<subjectKeyIdentifier value from CA certificate>) subjectAltName[n,m] (see [J.170], Appendix C)

HTTPS server Server Certificate

This certificate MUST be verified as part of the certificate chain containing the Service Provider Root CA Certificate, Service Provider CA Certificate, optional Local System CA Certificate, and Ancillary Certificates (e.g., the KDC Certificates).

Table 11-15 HTTPS Server Certificate

HTTPS Server Certificate	
Subject Name Form	C=<country> O=<Company Name> [OU=<Local System Name>] OU= <Company Name> HTTPS Server CN=<DNS Name>
Intended Usage	This certificate is issued either by the Service Provider CA or the Local System CA. It is used to authenticate the identity of the HTTPS server to the HTTP clients for the TLS session during provisioning. This certificate is passed to the PS element inside the TLS Server Certificate message.
Signed By	Service Provider CA or the Local System CA
Validity Period	20 years
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,m](digitalSignature, keyEncipherment, dataEncipherment), anyExtendedKeyUsage[n,m] (id-kp-serverAuth), authorityKeyIdentifier [n,m]

11.3.4.2.3 Certificate Validation

IPCable2Home certificate validation involves validation of a linked chain of certificates from the end entity certificates up to the valid Root. For example, the signature on the PS Element Certificate is verified with the Manufacturer CA Certificate, and then the signature on the Manufacturer CA Certificate is verified with the Manufacturer Root CA Certificate. The Manufacturer Root CA Certificate is self- signed, and is received from a trusted source in a secure way. The public key present in the Manufacturer Root CA Certificate is used to validate the signature on the same certificate.

The exact rules for certificate chain validation MUST fully comply with [RFC 3280], where they are referred to as "Certificate Path Validation." In general, [ITU-T X.509] certificates support a liberal set of rules for determining if the issuer name of a certificate matches the subject name of another. The rules are such that two name fields MAY be declared to match, even though a binary comparison of the two name fields does not indicate a match. [RFC 3280] recommends that certificate authorities restrict the encoding of name fields, so that an implementation can declare a match or mismatch, using simple binary comparison. IPCable2Home security follows this recommendation. Accordingly, the DER-encoded tbsCertificate.issuer field of a IPCable2Home certificate MUST be an exact match to the DER-encoded tbsCertificate.subject field of its issuer certificate. An implementation MAY compare an issuer name to a subject name by performing a binary comparison of the DER-encoded tbsCertificate.issuer and tbsCertificate.subject fields.

The validation of validity periods for nesting is not checked and intentionally not enforced, which is compliant with current standards. At the time of issuance, the validity start date for any end-entity certificate MUST be the same as or later than the start date of the issuing CA certificate validity period. After a CA certificate is renewed, the start dates of end-entity certificates MAY be earlier than the start date of the issuing CA certificate. The validity end date for end entity certificates MAY be before, the same as, or after the validity end date for the issuing CA, as specified in the IPCable2Home Certificate tables.

11.3.4.2.3.1 Validation for the Manufacturer Chain and Root Verification

The KDC MUST validate the linked chain of manufacturer certificates. Usually the first certificate in the chain is not explicitly included in the certificate chain that is sent over the wire. In the cases where the Manufacturer Root CA Certificate is explicitly included over the wire, it MUST already be known to the verifying party ahead of time

to verify this certificate. The Manufacturer Root CA Certificate sent over the wire MUST NOT contain any changes to the certificate, with the possible exception of the certificate serial number, validity period, and the value of the signature. If changes other than the certificate serial number, validity period, and the value of the signature, exist in the Manufacturer Root CA certificate that was passed over the wire in comparison to the known Manufacturer Root CA Certificate, the KDC making the comparison MUST fail the certificate verification.

11.3.4.2.3.2 Validation for the Code Verification Chain and Root Verification

A back office server can check the validity of the Code Verification Chain prior to beginning the software download process. For details, see the secure software download Section 11.8 of this document.

11.3.4.2.3.3 Validation for the Service Provider Chain and Root Verification

The PS Element MUST validate the linked chain of Service Provider certificates. Usually the first certificate in the chain is not explicitly included in the certificate chain that is sent over the wire. In the cases where the Service Provider Root CA Certificate is explicitly included over the wire, it MUST already be known to the verifying party ahead of time to verify this certificate. The Service Provider Root CA Certificate MUST NOT contain any changes to the certificate, with the possible exception of the certificate serial number, validity period, and the value of the signature. If changes other than the certificate serial number, validity period, and the value of the signature, exist in the Service Provider Root CA Certificate that was passed over the wire in comparison to the known Service Provider Root CA Certificate, the PS element making the comparison MUST fail the certificate verification.

11.3.4.2.4 Certificate Revocation

Certificate revocation is out of scope of this Recommendation.

11.4 Secure Management Messaging to the PS

The security algorithm used to initialize SNMP management messaging depends upon the provisioning mode of the PS element (see Section 5.5). In IPCable2Home, there are three provisioning modes: DHCP Provisioning Mode, SNMP Provisioning mode, and Dormant mode. DHCP Provisioning Mode has additional sub-modes that identify whether it is configured for NmAccess Mode or Coexistence Mode. SNMP Provisioning Mode requires SNMPv3 for management messaging.

The following sections describe the security algorithms and requirements needed to initialize SNMP management messaging, based on the provisioning mode of the PS element. The PS element MUST support the SNMPv3 security algorithms specified in Section 11.4.4.1.2 and Section 11.4.4.2.

11.4.1 Goals of Secure Management Messaging

Securing management messages include the following goals:

- Provide options to encrypt network management messages to the PS
- Provide options to authenticate network management messages to the PS
- If possible, provide security on management messaging that will not require additional protocols to be implemented
- Provide guidelines and minimum requirements for the encryption and authentication algorithms

11.4.2 Secure Management Messaging System Design Guidelines

Reference	Security System Design Guidelines
SEC3	Network management messages between the cable Headend and PS can be authenticated and optionally encrypted to protect against unauthorized monitoring and control.

11.4.3 Secure Management Messaging System Description

The use of SNMP of management to the PS from the cable operators network. SNMP has been adopted into cable

industry products for several years. The cable operator back office can support SNMPv1, v2 or v3. The PS is required to support management messaging for all three versions of SNMP. There is no security, per se, built into SNMPv1 or v2. SNMPv3 provides basic authentication and encryption algorithms defined in [RFC 3410] – [RFC 2576] and IPCable2Home specifies the use of the RFC defined security. SNMPv3 does not specify how the keys are set up to start the encryption and authentication process, and therefore, some details to generate and establish key exchange are specified. The details are listed within the next section.

11.4.4 Secure Management Messaging Requirements

11.4.4.1 Security Algorithms for SNMP in DHCP Provisioning Mode

In DHCP Provisioning Mode, the PS element can be configured for NmAccess Mode or Coexistence Mode. In Coexistence Mode the PS element can be configured for SNMPv1, SNMPv2, and/or SNMPv3 management messaging.

11.4.4.1.1 NmAccess Mode

If the PS Element is provisioned in DHCP Provisioning Mode with NmAccess Mode, the SNMP-based network management within the PS Element does not use SNMPv3 and therefore does not need to initialize SNMPv3 security functions. Initialization of the SNMPv1/v2 management link is defined in Section 6.3.3.1.

11.4.4.1.2 CoexistenceMode

If the PS Element is provisioned in DHCP Provisioning Mode with Coexistence Mode and the management messaging protocol is determined to be SNMPv3 (see Section 6.3.3.1), then the PS Element MUST use SNMPv3 security specified by [RFC 3414]. The PS MUST support SNMPv3 authentication and SNMPv3 privacy. The cable operator is strongly encouraged to turn on SNMPv3 authentication at all times. The use of SNMPv3 privacy is recommended if the cable operator can handle the additional load for encryption.

In order to establish SNMPv3 keys in DHCP provisioning mode, all IPCable2Home SNMP interfaces MUST utilize the SNMPv3 initialization and key changes procedure as defined in section 2.2 of the DOCSIS 1.1 Operations Support Systems Interface specification, [ANSI/SCTE 23-3 2003] (replace “CM” wording with “PS element” and replace “DOCSIS 1.1 compliant” wording with “IPCable2Home compliant”).

To support SNMPv3 initialization and key changes in DHCP provisioning mode, the PS element MUST also be capable of receiving TLVs of type 34, 34.1, and 34.2, as defined in section C.1.2.8 of the DOCSIS 1.1 Radio Frequency Interface specification, [J.112 Annex B] and implement the key-change mechanism specified in [RFC 2786] which includes the usmDHKickstartTable MIB object.

11.4.4.1.3 SNMPv3 Key Initialization

For each of up to 5 different security names, the Ultimate Authorization (CHAdministrator) generates a pair of numbers. First, the CHAdministrator generates a random number Rm.

Then, the CH Administrator uses the DH equation to translate Rm to a public number z. The equation is as follows:

z = g ^ Rm MOD p

where g is from the set of Diffie-Hellman parameters, and p is the prime from those parameters.

The PS configuration file is created to include the (security name, public number) pair. The PS MUST support a minimum of 5 pairs. For example:

TLV type 34.1 (SNMPv3 Kickstart Security Name) = CHAdministrator

TLV type 34.2 (SNMPv3 Kickstart Public Number) = z

The PS MUST support the VACM entries defined in Section 6.3.3.1.4.5. Only VACM entries specified by the corresponding security name in the PS configuration file MUST be active.

During the PS boot process, the above values (security name, public number) MUST be populated in the usmDHKickstartTable.

At this point:

usmDHKickstartMgrpublic.1 = “z” (octet string)

usmDHKickstartSecurityName.1 = “CHAdministrator”

When usmDHKickstartMgrpublic.n is set with a valid value during the registration, a corresponding row is created in the usmUserTable with the following values:

usmUserEngineID: localEngineID

usmUserName: usmDHKickstartSecurityName.n value

usmuserSecurityName: usmDHKickstartSecurityName.n value

usmUserCloneFrom: ZeroDotZero

usmUserAuthProtocol: usmHMACMD5AuthProtocol [RFC 2104]

usmuserAuthKeyChange: (derived from set value)

usmUserOwnAuthKeyChange: (derived from set value)

usmUserPrivProtocol: usmDESPrivProtocol

usmUserPrivKeyChange: (derived from set value)

usmUserOwnPrivKeyChange: (derived from set value

usmUserPublic

usmUserStorageType: permanent

usmUserStatus: active

Note: For (PS) dhKickstart entries in usmUserTable, Permanent means it MUST be written to but not deleted and is not saved across reboots.

After the PS has completed initialization (indicated by a value of ‘1’ (pass) for cabhPsDevProvState):

1. The PS generates a random number xa for each row populated in the usmDHKickstartTable which has a non-zero length usmDHKickstartSecurityName and usmDHKickstartMgrPublic.

2. The PS uses DH equation to translate xa to a public number c (for each row identified above).

   C = g ^ xa MOD p

where g is from the set of Diffie-Hellman parameters, and p is the prime from those parameters.

At this point:

usmDHKickstartMyPublic.1 = “c” (octet string)

usmDHKickstartMgrPublic.1 = “z” (octet string)

usmDHKickstartSecurityName.1 = “CHAdministrator”

3. The PS calculates shared secret sk where sk = z ^xa mod p.

4. The PS uses sk to derive the privacy key and authentication key for each row in usmDHKickstartTable and sets the values into the usmUserTable.

As specified in [RFC 2786], the privacy key and the authentication key for the associated username, "CHAdministrator" in this case, is derived from sk by applying the key derivation function PBKDF2 defined in PKCS#5 v2.0.

privacy key ←- PBKDF2(salt = 0xd1310ba6,

iterationCount = 500,

keyLength = 16,

prf = id-hmacWithSHA1) [RFC 2104]

authentication key ←-- PBKDF2(salt = 0x98dfb5ac,

iterationCount = 500,

keyLength = 16 (usmHMACMD5AuthProtocol) [RFC 2104],

prf = id-hmacWithSHA1) [RFC 2104]

At this point, the PS (CMP) has completed its SNMPv3 initialization process and MUST allow appropriate access level to a valid securityName with the correct authentication key and/or privacy key.

The PS MUST properly populate keys to appropriate tables as specified by the SNMPv3-related RFCs and [RFC 2786].

5. The following describes the process that the manager uses to derive the PS's unique authentication key and privacy key.

   The SNMP manager accesses the contents of the usmDHKickstartTable using the security name of 'dhKickstart' with no authentication.

   The PS MUST provide pre-installed entries in the USM table and VACM tables to correctly create user 'dhKickstart' of security level noAuthNoPriv that has read-only access to system group and usmDHkickstartTable.

   If the PS is in Coexistence Mode and is configured to use SNMPv3 the Group specification for the dhKickstart View MUST be implemented as follows:

   dhKickstart Group

   vacmGroupName 'dhKickstart'

   vacmAccessContextPrefix ''

   vacmAccessSecurityModel 3 (USM)

   vacmAccessSecurityLevel NoAuthNoPriv

   vacmAccessContextMatch exact

   vacmAccessReadViewName 'dhKickstartView'

   vacmAccessWriteViewName ''

   vacmAccessNotifyViewName ''

   vacmAccessStorageType permanent

vacmAccessStatus active

The VACM View for the dhKickstart view MUST be implemented as follows:

dhKickstartView subtree 1.3.6.1.2.1.1 (System Group) and 1.3.6.1.3.101.1.2.1 (usmDHkickstartTable)

The SNMP manager gets the value of the PS's usmDHKickstartMypublic number associated with the securityName for which the manager wants to derive authentication and privacy keys. Using the private random number, the manager can calculate the DH shared secret. From that shared secret, the manager can derive operational authentication and confidentiality keys for the securityName that the manager is going to use to communicate with the PS.

11.4.4.1.4 Diffie-Hellman Key Changes

The PS MUST support the key-change mechanism specified in the above section as well as [RFC 2786].

11.4.4.2 Security Algorithms for SNMPv3 in SNMP Provisioning Mode

If the PS Element is provisioned in SNMP Provisioning Mode, the SNMP-based network management within the PS Element MUST run over SNMPv3 with security specified by [RFC 3414]. The PS MUST support SNMPv3 authentication and SNMPv3 privacy. The cable operator is strongly encouraged to turn on SNMPv3 authentication at all times. The use of SNMPv3 privacy is recommended if the cable operator can handle the additional load for encryption. In order to establish SNMPv3 keys in SNMP provisioning mode, the PS MUST utilize Kerberized SNMPv3 key management as specified in Section 11.4.4.2.1.

11.4.4.2.1 Kerberized SNMPv3

The Kerberized key management profile specific for SNMPv3 MUST be followed as defined in section 6.5.4 in [J.170].

11.4.4.2.2 SNMPv3 Encryption Algorithms

The encryption Transform Identifiers for Kerberized SNMPv3 key management MUST be followed as defined in section 6.3.1 in [J.170].

11.4.4.2.3 SNMPv3 Authentication Algorithms

The authentication algorithms for Kerberized SNMPv3 key management MUST be followed as defined in section 6.3.2 in [J.170].

11.4.4.2.4 SNMPv3 Engine Ids

Because the SNMP Manager and Client MUST verify that the SNMPv3 Engine ID in the AP Request and AP Reply messages are based on the appropriate Kerberos principal name in the ticket [J.170], the following defines the rule to be used in generating SNMPv3 Engine:

- The SNMPv3 Engine ID follows the format defined in [RFC 3411], i.e., the first bit is set to 1 (one) and the appropriate value is used for the first four bytes [RFC 3411];
- The fifth byte carries the value 4 (four) to indicate that the following bytes, up to 27, are to be considered as text. These up to 27 bytes are defined as follows:
    - Up to the first 25 characters of the Kerberos principal name are used for the engine ID bytes starting on the 6th byte.
    - The above sequence of bytes, indicating the Kerberos principal name, is followed by a byte to be considered as an 8bit Hex value. Each different value identifies a particular SNMP engine in the device (element or NMS server). The value 0 (zero) MUST not be used.
    - The text string that starts on the 6th byte terminates with a Null character.

Note that other formats are possible by following the approach in [RFC 3411]. The above selection, though, is intended to reduce implementation complexity that would be required if all of the approaches in [RFC 3411] were allowed.

11.4.4.2.5 Populating the usmUserTable

SNMPv3 security settings for the cable operator "CHAdministrator" user are defined in Section 6.3.6.3 View-based Access Control Model (VACM) Requirements. The CHAdministrator is the ultimate authority for management of the Portal Services element. Other users can also be defined. In this section, a USM user is defined specifically for the provisioning process. In particular, it is defined to enable a notification receiver to be specified for the cabhPsDevProvEnrollTrap and cabhPsDevInitTrap, which the PS is required to send during the provisioning process (ref.: Table 13-1 Flow Descriptions for PS WAN-Man Provisioning Process for DHCP Provisioning Mode, step CHPSWMD-11; Table 13-2 Flow Descriptions for PS WAN-Man Provisioning Process for SNMP Provisioning Mode, step CHPSWMS-11 and step CHPSWMS-13; and Section 13.3.3 Provisioning Enrollment/Provisioning Complete Informs).

The msgSecurityParameters in SNMPv3 messages carry a msgUserName field that specifies the user on whose behalf the message is being exchanged and with whose security information the fields msgAuthenticationParameters and msgPrivacyParameters are produced. For the SNMP engine of a IPCable2Home element to process these messages, the necessary information is required to be entered in the usmUserTable [RFC 3414] for the element engine.

The usmUserTable MUST be populated with the following information in the PS Element right after the AP Reply message is received:

- usmUserEngineID: the local SNMP engine ID as defined in Section 11.3.3.2.4, SNMPv3 Engine Ids
- usmUserName: CHAdministratorxx:xx:xx:xx:xx:xx, where xx:xx:xx:xx:xx:xx is the device's WAN-Man hardware address
- usmUserSecurityName: CHAdministratorxx:xx:xx:xx:xx:xx, where xx:xx:xx:xx:xx:xx is the device's WAN-Man hardware address
- usmUserCloneFrom: 0.0
- usmUserAuthProtocol: indicates the authentication protocol selected for the user, from the AP Reply message
- usmUserAuthKeyChange: default value ""
- usmUserOwnAuthKeyChange: default value ""
- usmUserPrivProtocol: indicates the encryption protocol selected for the user, from the AP Reply message
- usmUserPrivKeyChange: default value ""
- usmUserOwnPrivKeyChange: default value ""
- usmUserPublic: default value ""
- usmUserStorageType: permanent
- usmUserStatus: active

New SNMPv3 users MAY be created with standard SNMPv3 cloning, as defined in [RFC 3414].

The VACM Security To Group Table [RFC 3415] MUST be populated with the following information in the PS right after the AP Reply message is received:

- vacmSecurityModel: 3(usm)
- vacmSecurityName: CHAdministratorxx:xx:xx:xx:xx:xx
- vacmGroupName: CHAdministratorSNMP
- vacmSecurityToGroupStatus: active

The VACM Access Table [RFC 3415] MUST be populated with the following information, linked to the vacmSecurityToGroupTable defined above, in the PS right after the AP Reply message is received:

- vacmAccessContentPrefix: ""
- vacmAccessSecurityModel: 3(usm)
- vacmAccessSecurityLevel: AuthNoPriv
- vacmAccessContextMatch: exact(1)

- vacmAccessReadViewName: CHAdministratorView
- vacmAccessWriteViewName: CHAdministratorView
- vacmAccessNotifyViewName: CHAdministratorNotifyView
- vacmAccessStorageType: permanent
- vacmAccessStatus: active

Seven rows of the VACM View Tree [RFC 3415] MUST be populated with the following information in the PS right after the AP Reply message is received:

- vacmViewTreeFamilyName: CHAdministratorNotifyView
- vacmViewTreeFamilySubtree: cabhPsDevProvEnrollTrap
- vacmViewTreeFamilyType: included
- vacmViewTreeFamilyMask: “”

- vacmViewTreeFamilyName: CHAdministratorNotifyView
- vacmViewTreeFamilySubtree: cabhPsDevBase
- vacmViewTreeFamilyType: included
- vacmViewTreeFamilyMask: “”

- vacmViewTreeFamilyName: CHAdministratorNotifyView
- vacmViewTreeFamilySubtree: docsDevSoftware
- vacmViewTreeFamilyType: included
- vacmViewTreeFamilyMask: “”

- vacmViewTreeFamilyName: CHAdministratorNotifyView
- vacmViewTreeFamilySubtree: cabhPsDevInitTrap
- vacmViewTreeFamilyType: included
- vacmViewTreeFamily Mask: “”

- vacmViewTreeFamilyName: CHAdministratorNotifyView
- vacmViewTreeFamilySubtree: cabhPsDevBase
- vacmViewTreeFamilyType: included
- vacmViewTreeFamily Mask: “”

- vacmViewTreeFamilyName: CHAdministratorNotifyView
- vacmViewTreeFamilySubtree: docsDevEventTable
- vacmViewTreeFamilyType: included
- vacmViewTreeFamily Mask: “”

- vacmViewTreeFamilyName: CHAdministratorNotifyView
- vacmViewTreeFamilySubtree: cabhPsDevProv
- vacmViewTreeFamilyType: included
- vacmViewTreeFamily Mask: “”

11.5 CqoS in the PS

CqoS is a transparent bridge for IPCablecom and LAN-to-LAN QoS. The IPCablecom DqoS messages between the MTA and the CMTS, CMS, or CM are secured by the IPCablecom Security Specification. It is not within the scope of IPCable2Home to add security for IPCablecom messages. IPCable2Home in-home, LAN-to-LAN QoS messaging is not secured since the threat for attacks within the home are seen as extremely low. It is not within the scope of IPCable2Home to add security for IPCablecom messages. Since there is no security requirement for the PS element to secure CqoS messages originated on the WAN, there is no dependency on the back office servers to support this function.

11.6 Firewall in the PS

Security issues have been a major concern for companies relying on networks for decades, and now there is increasing awareness of security and privacy issues for home users with the always on CM. Because the average subscriber might lack some technical knowledge or even if not, lacks the time to keep their home computers in top-notch secure operation, a firewall has become a necessary first line of defense in protecting the unsecured computers and other LAN IP devices in the home.

11.6.1 Goals and Assumptions of IPCable2Home Firewall

Goals:

- Provide the cable operator with a standard and interoperable configuration for the firewall
- Provide the cable operator with a minimum set of required functionality for the firewall
- Enable monitoring of the firewall events using the event messaging mechanism
- Protect the home network and LAN IP devices on that network from unwanted WAN-to-LAN traffic
- Protect the HFC from unwanted LAN-to-WAN traffic.
- Protect the PS from attacks and traffic deemed as unwanted by the cable operator.
- Ensure the firewall will process packets at sufficient rates so packet filtering does not introduce a performance bottleneck, regardless of the complexity or size of the ruleset.
- Ensure support of identified applications through the firewall for specified scenarios.
- Provide the cable operator the ability to monitor and change rules used by the firewall
- Ensure that the proper security configurations (e.g. filtering rules and policies) exist on the firewall system.
- Identify the types of attacks the firewall will log and specify the log so the operator can view the log as needed.
- Support IPCablecom through the firewall
- Notify the administrator of defined important events in real time
- Provide a factory default ruleset to ensure consistent baseline resets of the firewall

Assumptions:

- The firewall treats all packets destined to or coming from the LAN according to the current policy regardless of address mode, CAT or Pass-through. (e.g. address mode has no affect on the firewall operations).
- The firewall begins operation immediately after the provisioning complete message, regardless of provisioning mode.
- SNMP, in particular SNMP messaging directed at the IPCable2Home Management Portal (CMP), can be used to configure IPCable2Home firewall rulesets. Thus, the ruleset is represented, externally as a collection of MIB objects.
- Policy MIB objects control the logging actions taken by the firewall packet filter
- The firewall will apply filtering rules and policies conjunction with checking the translated addresses known to the CAT in the PS.

11.6.2 Firewall System Design Guidelines

Firewall system design guidelines listed in Table 11-16 guided IPCable2Home firewall specifications.

Table 11-16 IPCable2Home Security System Design Guidelines

Reference	Security System Design Guidelines
SEC4	The firewall will accept configuration files in a standard language and format.[4]
SEC5	The cable operator will have the ability to remotely manage compliant firewall products through configuration file or SNMP commands
SEC6	The compliant firewall will include a default set of rules for an expected minimum set of functionality.
SEC7	Provide the necessary support for IPCablecom through the firewall.
SEC8	A minimum set of requirements will be placed on the firewall filtering capabilities for packet, port, IP addresses, TOD, etc.
SEC9	A detailed firewall event logging interface will allow the cable operator to monitor and review firewall activity as configured.
SEC10	The firewall will support commonly used applications in specific scenarios.
SEC11	The firewall will protect the LAN and WAN from common network attacks.
SEC12	The management of the events and rulesets for the firewall will be defined in detail via the Security MIB.

11.6.3 Firewall System Description

Typically, firewalls are built using a combination of the following components: Packet Filtering (PF), Stateful Packet Filtering (SPF), Application Level Gateway (ALG), and Application Server Proxy (ASP). A packet-filtering module is probably the most common firewall component because it determines which packet streams are blocked and which are allowed to cross the firewall. Each individual packet decision is based on static configuration information (the ruleset) configured into the firewall's filtering mechanisms (policy) so that the packet will be allowed or denied, based on the inspection of packet header fields: source and destination IP addresses, source and destination protocol port numbers, protocol type, etc. Depending on the desired level of security, a great number of filters might need to be configured on a firewall. The cable operator will need to balance the ruleset complexity with customer needs. This Recommendation attempts to specify a rich set of configuration filters, managed via MIB objects, so the various types of services (protocols and applications) can be individually configured, if needed.

A stateful packet filtering (SPF) module uses accumulated state information from packets that belong to the same connection when making packet-dropping decisions. The SPF differentiates between different protocols and handles each protocol's connection correctly. The SPF module stores and utilizes information found in the packet's network layer and transport layer headers.

An application level gateway (ALG) is a component that knows how to extract information required for connection tracking from the packet's application layer. As some protocols incorporate connection control information at the application layer, the SPF will incorporate ALGs to perform the connection tracking. The specific ALG (e.g. FTP-ALG, IPSEC-ALG) is required for handling each such protocol needed to support IPCable2Home. For example, the FTP protocol in active mode incorporates the TCP port number that will be used later on for the data transfer. Therefore, it is required to use an FTP ALG to track the state of all FTP connections. See Annex D for more information on ALG requirements.

An application specific proxy (ASP), another typical firewall, can filter, based on the application layer protocol unique features, or messages specifically for the client-server protocols. There are security benefits in the use of ASPs. For one, it is possible to add access control lists to protocols, requiring users or systems to provide some level of authentication before access is granted. In addition to being protocol specific, an ASP understands the protocol and can be configured to block only subsections of the protocol. The ASP allows the operation of NAT-unfriendly applications when the Portal Service is operating in either of its two transparent routing modes: C-NAT

[4] 1. The Firewall Configuration File Requirements are defined in Section 7.4 PS Function - Bulk Portal Services Configuration (BPSC).

or C-NAPT. For example, an FTP ASP can be configured to block the traffic from unauthenticated users, while granting authenticated users selective access to the "put" and "get" commands, depending on which directions these commands are issued.

The particular combination of packet filer, SPF AGLs and ASPs on a given firewall product, constitutes a trade off between performance and the security level. Typically, being a network layer mechanism, packet filters tend to yield better performance than ALGs/ASPs that are application layer mechanisms. A compromise solution becoming increasingly popular consists of the use of stateful packet filtering (SPF), where state information accumulated from packets that belong to the same connection is kept and used in making packet-dropping decision.

SPFs and ASPs both include filtering against the security policy to achieve the desired level of security for a site. However, while the security policy determines the allowed services and the way in which they are used across the firewall, the security policy does not spell out the specific configuration for the firewall. The ruleset is expressed in human readable form, then interpreted by the firewall, and implemented into the filtering policy in the internal language of the firewall. The filters inspect each packet and determines which packets the firewall forwards and which it rejects.

The following is a high-level diagram of the firewall and the roles of the various firewall components referenced by this specification.

Note: This diagram does not indicate any specific technical architecture or implementation. It is only for logical reference.

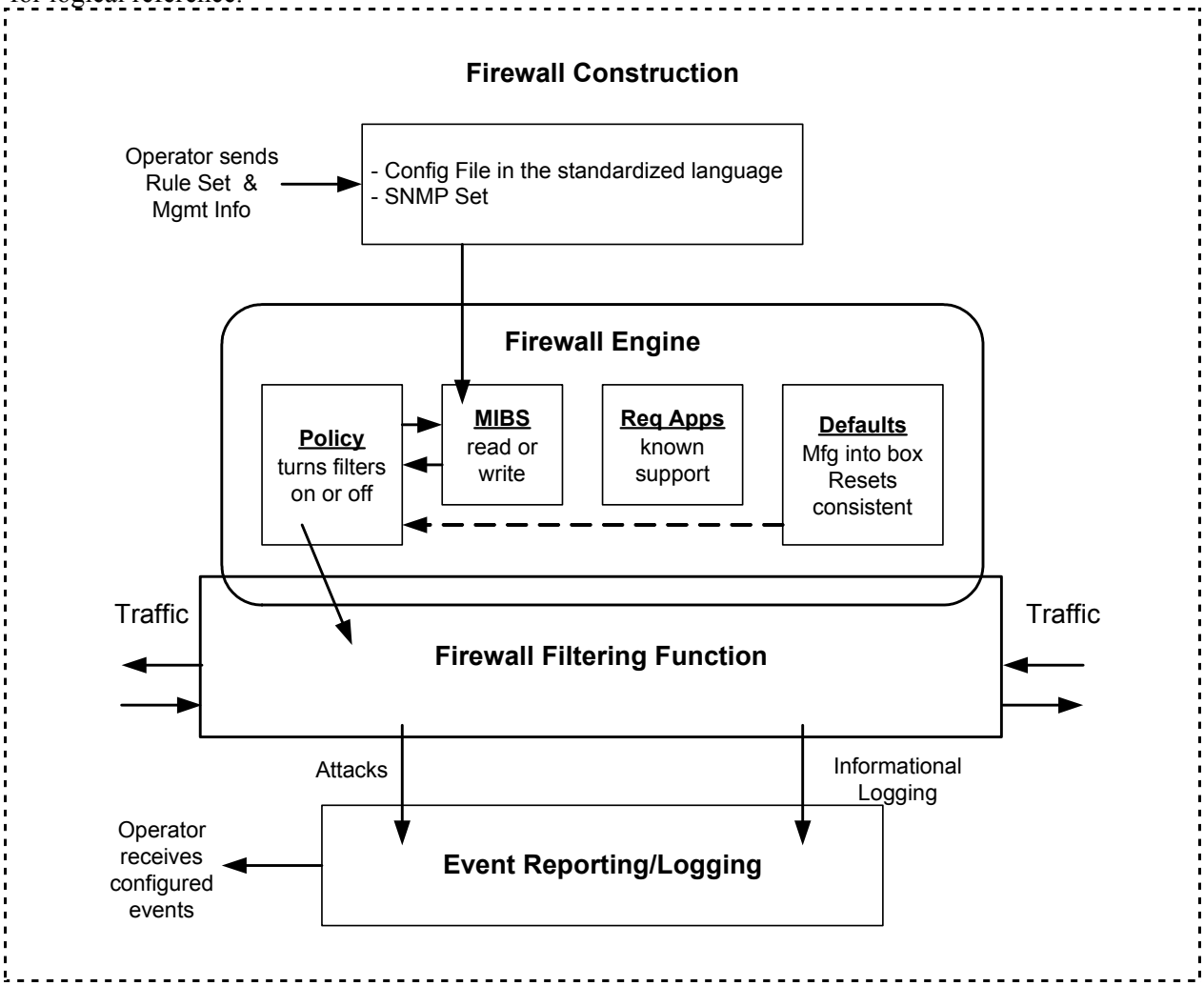

Figure 11-3 Firewall Logical Reference

11.6.4 Firewall Requirements

11.6.4.1 Configuration File Language for the Firewall

A cable operator chosen ruleset can be configured into the firewall via a PS configuration file or firewall configuration file download. Within this section the term configuration file is used to mean either the PS configuration file or firewall configuration file. The language and format for the configuration file containing the ruleset applicable to a particular firewall product is not only defined, but how that file is used in the firewall to configure the SPF and ASP components, will be implementation specific.

The PS MUST be able to receive and interpret a firewall configuration file constructed, using TLVs in ASN.1 format with BER [ISO8025] encoding. Inside the firewall, the compiler translates the policy language into a vender specific internal format. TLV type 28 MUST be used for all the firewall MIB objects. The language of the PS configuration file and the firewall configuration file is the same. The requirements for firewall configuration file processing are defined in Section 7.

11.6.4.2 Firewall Configuration

The PS allows, but does not require remote management of firewall functions by the cable operator. The firewall MUST accept rulesets configured in bulk, via the specified PS or Firewall configuration files. When a configuration file contains firewall filtering rules, the rules can be treated as either an incremental or a complete configured ruleset, as set by the cabhSec2FwClearPreviousRuleset object. Upon configuration file download and process completion, the firewall rules from the configuration file MUST be immediately applied and available for use once the cabhPsDevProvState MIB object has a value of pass(1), without rebooting the PS. If the cabhSec2FwPolicySelection object is set to configuredRuleset, the new rules MUST be immediately applied in processing. When the PS processes a configuration file with firewall rulesets, the PS MUST NOT lose the incremental rules sent via SNMP Set, or rules sent via previous configuration files, unless the cabhSec2FwClearPreviousRuleset object is set to complete(2) or incrementDefault(3). For rules set via SNMP, the rule or MIB object value MUST be activated (or appropriately available if activation is not currently allowed) immediately after processing the SNMP Set message without rebooting the PS. For example, if one of the firewall filters is updated via SNMP set, but the cabhSec2FwPolicySelection object is currently set to factoryDefault, the PS will then update the configured ruleset with the modified rule, but currently operate the PS using the factory default policy until the cable operator changes the object to configuredRuleset, which now includes the newly configured rule.

The firewall MUST check and apply the configured rules in the docsDevFilterIpTable as described in [RFC 2669], unless otherwise stated in this Recommendation.

If the PS cannot process the configuration file for any reason, the PS MUST send the appropriate event for processing failure, and the firewall MUST use the ruleset selected by the cabhSec2FwPolicySelection object and enabled by the cabhSec2FwEnable object.

11.6.4.3 Firewall Policy

The firewall policy instructs the firewall to filter traffic based on particular rules. The policy accepts the rulesets to be applied by the filtering function since the filtering function alone has no meaning, as it is only a set of capabilities. The firewall filtering capabilities, combined with the firewall policy, provide firewall protection for the LAN. The firewall filters actively inspect each packet or connection with the policy to apply the two allowed actions: allow or deny. If there is a conflict of rules within the policy, the firewall MUST resolve the conflict as described for the docsDevFilterIpTable in [RFC 2669], unless otherwise stated in this Recommendation.

The firewall is designed to protect the home computer system from attacks and unwanted traffic. Traffic is generalized into categories of WAN, LAN and PS sourced traffic. By default, if there is no rule for traffic initiated by non-LAN IP addresses (LAN IP addresses are defined as LAN-Trans and LAN-Pass addresses), the firewall MUST deny this traffic. By default, if there is no rule configured for traffic initiated from a LAN IP address destined to a WAN IP address, the firewall MUST allow this traffic. All packets not explicitly allowed by a configured rule MUST be checked to see if they MUST be allowed due to state.

A standard way for policies to be communicated to the firewall, a default policy, and support for IPCablecom are specified. The default policy serves as the standard factory default settings; the operator can choose to reset the box to these settings at any time. The factory default policy allows management of the PS and enables the PS for most traffic initiated from the LAN to the WAN. Cable operators can create any configuration needed to support any application through the firewall for each customer. The policy can be set by PS configuration file, firewall

configuration file, or SNMP Set messages.

The PS may receive traffic for the IPCablecom MTA. Therefore, it is appropriate to take a brief look at the support needed for the MTA. Support for IPCablecom, described in Section 11.6.4.3.3, consists of the IPCable2Home factory default policy plus the needed protocols to enable IPCablecom messaging to traverse the firewall. Annex D also notes which ports must be opened for the MTA. Support for IPCablecom enables provisioning, management, and services through the firewall.

The Factory Default Policy, as defined in Table 11-17, is required to be installed at the time of manufacture and always available in the box to reset to a baseline level of filtering. All configured rulesets are provisioned by cable operator. The Factory Default Policy is not labeled as a "Ruleset" since it is not specified in the required configuration file language and format, instead the requirements are listed for the default policy and implementation is vendor specific, since it is implemented at the time of manufacture.

IPCable2Home currently specifies a Firewall Factory Default Policy built into the PS at manufacture time and a method for the cable operator to configure rulesets into the PS as needed. This section describes the general firewall policy concept as it relates to address realms, the factory default policy, the IPCablecom ruleset information, and the cable operator configured ruleset.

11.6.4.3.1 Firewall Policy and Address Realms

The firewall filters based policy with a specific ruleset configuration. If there is no configuration of the firewall by a cable operator, the firewall is set to the factory default policy. The policy includes filtering rules for source and destination IP addresses and the concept of direction is derived from the word's source and destination and therefore, is not specified.

The concept of IP addressing realms are defined in this specification for WAN and LAN IP addresses. The PS is in the LAN, but packets originating from or destined to the PS are not referred to as LAN traffic for the purpose of firewall filtering. Instead, the specific PS IP address is called out. Packets originating from or destined to the PS are indicated by the use of the WAN-Man IP address, PS server router IP address or to the fixed 192.168.0.1 IP address (which can be, but may not necessary be, the same as the PS server router IP address). Accordingly, the firewall will distinguish traffic to and from the PS in the Factory Default Policy. The LAN IP addresses are not distinguished between addressing modes as the firewall does not filter based on IPCable2Home IP address modes. The PS WAN-Data IP address MUST be considered part of the LAN IP address realm since the WAN-Data IP address only proxies packets with the CAT translated IP addresses (e.g. LAN-Trans IP addresses).

11.6.4.3.2 e Factory Default Policy

The Factory Default Policy provides for normal PS functionality as well as most traffic initiated from Hosts. The Factory Default Policy MUST be hard-coded into the PS at the time of manufacture. The PS MUST always use the Factory Default Policy when the cabhSec2FwPolicySelection object is set to factoryDefault(1).

The Factory Default Policy MUST support the protocols required by IPCable2Home with the exception of the TOD protocol, which is not specified beyond the provisioning process and therefore not included in the Factory Default Policy, as the firewall does not become active until after the provisioning state is passed. If the cabhSec2FwPolicySelection object is set to factoryDefault during the provisioning process (e.g. reboot), the PS MUST activate the factory default policy immediately after the cabhPsDevProvState MIB object has a value of pass(1) without rebooting the PS. If the cabhSec2FwPolicySelection object is set to factoryDefault via SNMP, the PS MUST activate the factory default policy immediately without rebooting the PS. The Factory Default Policy MUST NOT include any time of day restrictions or limits on the number of sessions or connections to be supported simultaneously, unless otherwise specified in Annex D, Applications through the CAT and Firewall.

Table 11-17 specifies the Factory Default Policy. Both LAN address realms, LAN-Trans and the LAN-Pass, are treated the same for the Factory Default Policy and are labeled LAN IP Address. The firewall MUST be able to look up addresses in the CAT mapping table to apply policy based on the real Host device IP Address. PS Addresses MUST NOT include any PS WAN-Data IP addresses. PS WAN-Data addresses belong to LAN IP traffic and as such are treated as LAN IP Addresses. The table bases information on session initiation, not on allowed traffic. Therefore, the Firewall Factory Default Policy MUST be implemented for session initiation and not for allowed traffic. Traffic returning at the request of the initiator is understood as state information for a session and the firewall will check the session state after checking the policies to ensure a packet is not denied that is part of a current session. Table 11-17 MUST be implemented as the Firewall Factory Default Policy.

Table 11-17 IPCable2Home Firewall Factory Default Policy

Column Headings Identify Session Initiation	Source: WAN IP Address Dest: PS WAN-Man IP Address	Source: WAN IP Address Dest: LAN IP Address	Source: PS WAN-Man IP Address Dest: WAN IP Address	Source:PS WAN-Man IP Address Dest: LAN IP Address	Source: LAN IP Address Dest: PS Server Router IP Address	Source: LAN IP Address Dest: PS 192.168.0.1	Source: LAN IP Address Dest: PS WAN-Man IP Address	Source: LAN IP Address Dest: WAN IP Address	Relation-ship Scenarios Required
AOL IM	Deny	Allow	Deny	Deny	Deny	Deny	Deny	Allow	All
DHCP	Deny	Deny	Allow	Deny	Allow	Allow	Deny	Allow	All
DNS	Deny	Deny	Allow	Deny	Allow	Allow	Deny	Allow	All
FTP	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Allow	All
HTTP	Deny	Deny	Deny	Deny	Allow	Allow	Deny	Allow	All
HTTPS (This is HTTP over TLS)	Deny	Deny	Allow	Deny	Allow	Allow	Deny	Allow	All
ICMP Echo Requests & Timestamp (Ping & Traceroute)	Allow	Allow	Deny	Allow	Allow	Allow	Deny	Allow	All
IPSec	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Allow	One-to-one, Single
Kerberos	Deny	Deny	Allow	Deny	Deny	Deny	Deny	Deny	All
Microsoft Messenger	Deny	Allow	Deny	Deny	Deny	Deny	Deny	Allow	All
MSN Messenger	Deny	Allow	Deny	Deny	Deny	Deny	Deny	Allow	All
POP3	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Allow	All
SMTP	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Allow	All
SNMP	Allow	Deny	Allow	Deny	Allow	Allow	Deny	Deny	All
SNMP Trap	Deny	Deny	Allow	Deny	Allow	Allow	Deny	Deny	All
Syslog	Deny	Deny	Allow	Deny	Deny	Deny	Deny	Deny	All
Telnet	Deny	Deny	Deny	Deny	Allow	Allow	Deny	Allow	All
TFTP	Deny	Deny	Allow	Deny	Deny	Deny	Deny	Allow	All
Yahoo Messenger	Deny	Allow	Deny	Deny	Deny	Deny	Deny	Allow	All
Windows Messenger	Deny	Allow	Deny	Deny	Deny	Deny	Deny	Allow	One-to-one; One-to-multi

11.6.4.3.3 IPCablecom Ruleset

If the cable operator deploys IPCablecom, the firewall may need to pass traffic to and from the MTA, depending upon network and device configuration. If operating a IPCablecom network, the protocols defined by the IPCablecom set of Recommendations MUST NOT be broken by the firewall. The cable operator may need to configure the firewall for any additional rules to ensure IPCablecom will be enabled through the firewall. The following Table 11-18, is a list of specifications which have unique port requirements for communication with the MTA. However, it is not a comprehensive list of all the IPCablecom specifications.

Table 11-18 Relevant IPCablecom 1.x Specifications for IPCable2Home Firewall

Description	Specification
Audio/Video Codecs Specification	[J.161]
Dynamic Quality of Service Specification	[J.163]
Network-Based Call Signaling Protocol Specification	[J.162]
MTA Device Provisioning Specification	[J.167]
Security Specification	[J.170]
Management Event Mechanism Specification	[J.164]
Audio Server Protocol Specification	[J.175]
Call Management Server Signaling Specification	[J.178]

The list of the required IPCablecom protocols to the MTA have been taken from the indicated specifications. The IANA assigned port numbers to open the ports needed by the IPCablecom specified protocols through the firewall are listed in Annex D, Applications Through CAT and the Firewall. The IPCablecom defined protocols include the following:

- Provisioning SNMPv3, DHCP, DNS, TFTP, SYSLOG
- Media Stream RTP, RTCP
- QoS RSVP
- Security Kerberos, IPSec
- Network Call Signaling MGCP, SDP (Note: SDP does not require any specific port.)

11.6.4.3.4 Configured Ruleset and Current Version

The cable operator can send any firewall ruleset needed to the PS via configuration file or SNMP Set. When a cable operator sends rules to the PS, this is known as the configured ruleset or current version. The configured ruleset MUST be stored in non-volatile memory (e.g. persist across reboots). This requirement ensures the PS can re-enable this ruleset if the firewall is enabled and the policy selection is set to configuredRuleset. The defined firewall filters are set with the configured ruleset. The bulk of the firewall filtering MIB objects are specified in [RFC 2669], with an additional schedule table added in the Security MIB. The mib objects are grouped together in a filter table. This filter table is the configured ruleset.

PS processing and application of a configured ruleset sent by the cable operator depends upon the contents of the configuration file and the value of the cabhSec2FwClearPreviousRuleset object. The configured ruleset can be an increment to the existing configured ruleset or a complete replacement. An increment to the factory default policy is also possible. If the cable operator increments the firewall rulesets against the factory default policy, the PS MUST populate the filter table with the factory default rules prior to populating the table with the cable operator configured rules. This provides visibility for the cable operator to be able to see all the filtering rules. Functional details and requirements for this feature are stated in the cabhSec2FwClearPreviousRuleset MIB object description in Section 11.6.4.7.1.

11.6.4.4 Firewall Filtering

This section specifies requirements for the firewall's packet filtering component. The specified packet filter examines individual packets and determines whether to allow or deny their passage across the firewall. More specifically, the packet filter inspects packet header fields and makes per-packet decisions based upon the contents of those fields and configured ruleset.

11.6.4.4.1 Minimum Set of Filtering Capability

For the purpose of IPCable2Home, a simple NAT or packet filter is not sufficient. In order to provide a flexible and secure solution the firewall MUST implement an Application Specific Proxy (ASP) or a Stateful Packet Filtering (SPF) firewall. Additionally, specific requirements for these filtering techniques are needed in order to provide a sufficient level of testable, reliable, and interoperable products for the cable industry. The firewall's ASP/SPF component controls traffic flows associated with application-layer protocols that cannot be effectively and transparently controlled through static filtering. The filtering mechanisms will examine applications that are dynamically established over IP, TCP, UDP, or ICMP sessions. Port, IP address, and scheduling activity is managed as related to a "session" within the firewall. Also, the application specific proxy allows the operation of NAT-unfriendly applications when the Portal Service is operating in either of its two transparent routing modes: C-NAT or C-NAPT.

Regardless of the type of firewall that is implemented, the PS firewall MUST be session aware and able to track information on an IP address pair (source and destination) in conjunction with the current policy for the specified IP address. A session consists of a pairing of IP addresses on a per request basis. This request includes matching the request with the allowed policy for that session which consists of IP address, application port and curfew.

The firewall's packet filter architecture specifies separate inbound (WAN-to-LAN) and outbound (LAN-to-WAN) packet filters and PS. The inbound packet filter examines packets coming into the PS WAN interface. The outbound packet filter examines packets coming into the PS LAN interface. Separate rules can be applied to inbound and outbound packet filters. Packets destined to the PS from the WAN or LAN are filtered at the firewall prior to forwarding to any of the PS non-firewall components (CAP, CDP, CNP, CSP, CQP, and CPM).

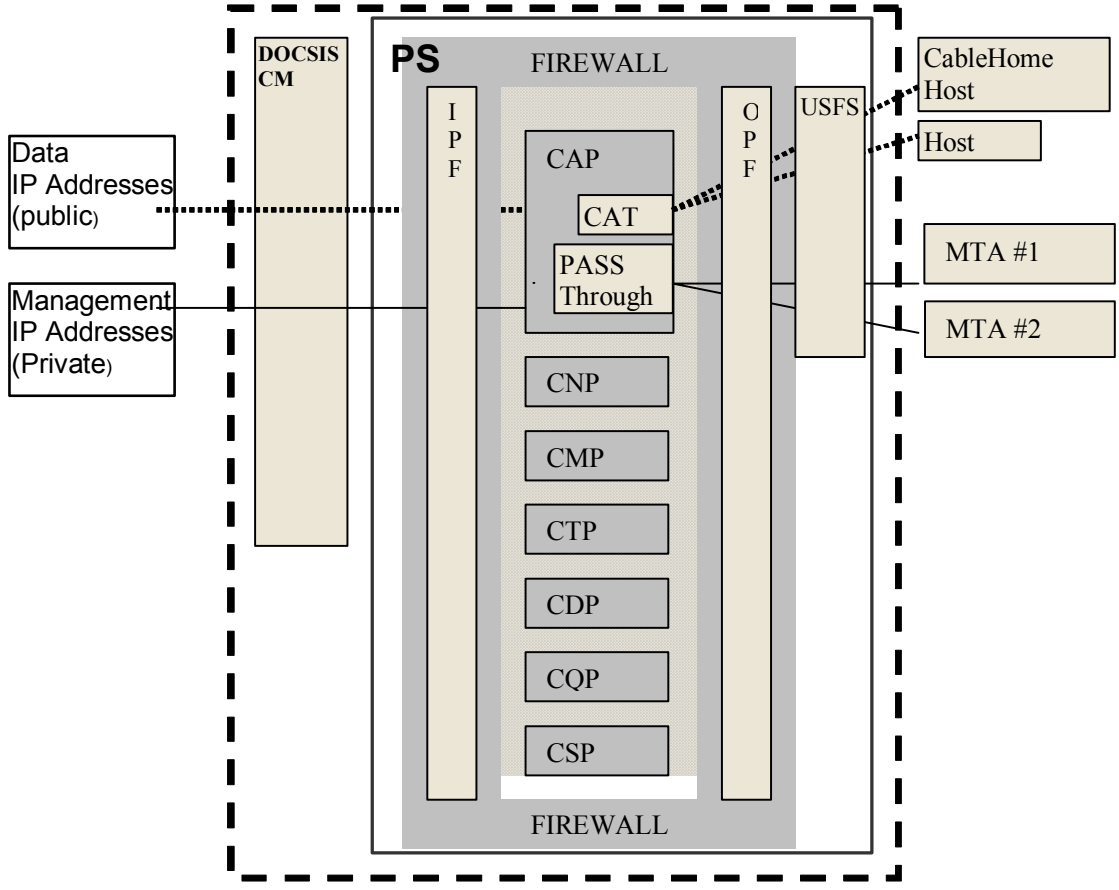

Figure 11-4 Firewall Functionality inside the PS

The following filtering definitions are used:

- ALLOW means “let the packet through”.
- DENY means to “drop the packet”.
- NAT/NAPT (CH CAT) packets will be translated from the LAN, while packets returning from the WAN to the LAN will be recognized as such and will undergo reverse NAT/NAPT. The firewall filters will be applied in conjunction with the correct source or destination address on the LAN.

The firewall's inbound and outbound packet filters MUST exhibit the following behavior:

- The firewall MUST deny non-LAN IP address initiated traffic, unless there is a rule to explicitly allow the traffic. Non-LAN IP addresses are addresses that are not in the LAN-Trans address list or the LAN-Pass address list.
- The firewall MUST allow all LAN IP address (does not include the PS WAN-Man or PS Server Router IP address) initiated traffic, unless there is a rule to explicitly deny the traffic.
- The firewall MUST deny replayed packets from either the LAN or the WAN.
- The firewall MUST create a “state” for all allowed packets initiating a session. A packet will either be accepted because there is a static rule to allow packets of that criteria, or there is a state that implies that the packet will be allowed through as a result of an ongoing allowed session.
- The PS SHOULD NOT allow TCP outbound traffic prior to establishing a TCP session (i.e., prior to completing a 3-way TCP handshake).
- Packets with one of the following IP Options: LSRR (Loose-Source-Route), SSRR (Strict-Source-Route), RR (Record-Route) MUST be denied.

There are many types of network attacks that the firewall can filter. Many methods and tools are used to attack various devices on a network. The list is very long and changes faster then any current published document can claim. This Recommendation calls out some the well known attacks for general security consideration. The firewall SHOULD protect against port or network scanning launched from LAN or WAN. The firewall SHOULD protect against floods of packets and malformed packets. The firewall SHOULD protect against the following list of denial of service attacks: “Ping of Death”, “Teardrop”, “Bonk”, “Nestea”, “SYN Flood”, “LAND Attack”, “IP Spoofing”, “Smurf Attack”, “WinNuke”, and any high-frequency messaging originated by LAN IP Devices, such as BP_Init or DHCP DISCOVER messages.

11.6.4.4.2 Filter Criteria

The default is to deny traffic initiated from WAN IP addresses, the PS WAN-Man IP address, or the PS Server Router IP address. Therefore, the rulesets and default policy are built to allow particular traffic for these addresses. The default is to allow traffic from LAN IP addresses unless explicitly set to deny, therefore, the rulesets are built to deny particular traffic to these addresses. This section does not specify all the expected filtering capabilities, but lists a minimum set of criteria which is expanded by specified MIB objects. Inbound and outbound packet filters MUST examine traffic to see if a rule will allow the traffic based on the following filtering criteria:

- IP source address
- IP destination address
- IP (“next level”) protocol; e.g., TCP, UDP, ICMP, IPsec AH, IPsec ESP
- TCP or UDP source and destination ports
- Start-of-connection information for TCP packets (i.e., absence of ACK bit) for session tracking
- Sequence number tracking for sessions

The above packet data is used as criteria for matching incoming packets to a specific rule, and hence, arriving at a specific filtering decision (allow/deny). The firewall MUST check the IP source and destination address to see if any rule applies to that address. If the ruleset currently prohibits forwarding traffic to or from an IP address, the firewall MUST deny the packet, unless it needs to be passed due to state.

Note: Filtering against the current policy include more requirements for filtering that must be applied, however, are not considered a part of the built-in filtering criteria.

11.6.4.4.3 Filtering Architecture

The firewall packet filter MUST be able to filter traffic as it enters the PS, with the exception of using the USFS function from the LAN, and provide distinct inbound (WAN-to-LAN), outbound (LAN-to-WAN), and PS packet filters. This firewall MUST have the following attributes:

- filter packets received from the PS WAN interface, e.g. IfIndex = 1, (this is referred to as inbound filtering)
- packets received from the PS LAN interface, e.g. IfIndex = 255, (this is referred to as outbound filtering)
- filter packets originating from within the PS going either to the LAN or WAN
- apply filters only as currently enabled
- inbound and outbound packet filtering precedes the delivery of packets to any of the PS's non-firewall components, except the USFS for packets coming from the LAN
- outbound packet filtering precedes any ASP/SPF processing

The WAN inbound packet filter MUST exhibit the following behavior:

- Default deny; meaning the default behavior of the firewall on inbound packets, that do not have explicit filter rules to allow the packet, is to drop the packet.
- Deny all packets whose source address is in the LAN-Pass or LAN-Trans address realms received from the PS WAN interface, e.g. IfIndex = 1.
- Deny all packets with broadcast or multicast source addresses.

The LAN outbound packet filter MUST exhibit the following behavior:

- Default allow; meaning the default behavior of the firewall on outbound packets, that do not have explicit filter rules to deny the packet, is to allow the packet.
- Reject all packets with broadcast or multicast source addresses.

11.6.4.5 Firewall Event Reporting

The information coming out of the firewall is critical for routine management and monitoring, as well as providing the appropriate events for specified attacks. The events generated by the firewall can be used for intrusion detection, DOS attacks, and any failures or logs related to the firewall system. The analysis of the logs can be quite cumbersome if there are large amounts of data to sort through. Also, if there are too many events sent to the cable operator, it could tie up bandwidth, since there can be many firewalls sending events to the NMS located in the cable operator back office. The cable operator will need to decide which items they wish to turn on to monitor the firewall and how often they would like to receive events. Turning-on event reporting is separate from turning-on the ruleset for the firewall filtering criteria. When the firewall event enable MIB objects have been set to enable the firewall to track defined event types, the firewall will log and send specified event messages as defined in this section and Annex B.

Each of the specified events can be turned on or off by the cable operator through setting a SNMP MIB object through a configuration file, or a SNMP set. It is recommended that SNMPv3 be used to secure SNMP messages containing firewall information.

11.6.4.5.1 Firewall Events

Firewall events allow a cable operator to remotely assess the level of hacker activity and modifications to the firewall on specific PS elements. Event generation is based on management changes to the ruleset, events detected by the firewall as enabled by the ruleset, or TFTP/HTTP events based on downloading. The TFTP/HTTP events for firewall download MUST be sent, as defined by Annex B.

The firewall MUST be capable of logging the following types of events:

> **TYPE 1**: Type 1 MUST log all attempts from both LAN and WAN clients to traverse the firewall that violate the Security Policy when this type is turned on via the cabhSec2FwEventEnable MIB object. This logs all connection attempts that are dropped due to policy violation. An attack is defined as packets (meaning each packet is counted as an attack), that attempt to traverse the firewall and violate the current policy. If enabled, and the threshold is reached, the PS MUST immediately send event 80010201.

TYPE 2: Type 2 MUST log identified Denial of Service attack attempts when this type is turned on, via the cabhSec2FwEventEnable MIB object. A type 2 attack is defined as any attempt that is considered to be disrupting any service, like the flood of duplicate packets (meaning 10 packets are counted as one attempt), or malformed packets or unpermitted connection attempts from the same host, for a multiple number of times. If enabled, and the threshold is exceeded, the PS MUST immediately send event 80010202.

TYPE 3: Type 3 MUST log all changes made to the cabhSec2FwPolicyFileURL or cabhSec2FwPolicyFileCurrentVersion or cabhSec2FwEnable MIB objects when this type is turned on, via the cabhSec2FwEventEnable MIB object. Tracking the changes to the firewall configuration provides valuable feedback to the cable operator for debugging purposes. If enabled and the threshold is exceeded, the PS MUST immediately send event 80010203.

TYPE 4: Type 4 MUST log all failed attempts to modify cabhSec2FwPolicyFileURL and cabhSec2FwEnable MIB objects when this type is turned on, via the cabhSec2FwEventEnable MIB. If enabled and the threshold is exceeded, the PS MUST immediately send event 80010204.

TYPE 5: Type 5 MUST log allowed inbound packets from the WAN when this type is turned on, via the cabhSec2FwEventEnable MIB object. This enables the cable operator to monitor traffic in a scenario where there are signs of detection intrusion or DOS attacks from the WAN side. If enabled and the threshold is exceeded, the PS MUST immediately send event 80010205.

TYPE 6: Type 6 MUST log allowed outbound packets from the LAN when this type is turned on, via the cabhSec2FwEventEnable MIB object. This enables the cable operator to monitor traffic in a scenario where there are signs of attacks coming from a home LAN across the WAN. If enabled and the threshold is exceeded, the PS MUST immediately send event 80010206.

The event types for IPCable2Home are defined for monitoring purposes only. It is up to the individual cable operator to evaluate and execute any necessary response to issues detected and reported by the firewall.

11.6.4.5.2 Firewall Logs

The firewall log information MUST be recorded in the PS for each enabled log type, as specified in Section 11.6.4.5.1. The PS MUST log the specified information unless the cabhSec2FwEventThreshold is set to zero, or the cabhSec2FwEventEnable is set to disable, or the cabhSec2FwEventInterval is set to zero, or the log is full. If the cabhSec2FwEventThreshold is not set to zero, the cabhSec2FwEventEnable is enabled, the cabhSec2FwEventInterval is not set to zero and the log is not full, the PS MUST continue to log events of the enabled type. Once the cabhSec2FwEventLogReset is set to 1 to clear the log, and the cabhSec2FwEventEnable is enabled, the cabhSec2FwEventCount MUST start counting from zero.

The PS, at a minimum, MUST support the logging of 1 Kilobyte of data for each log in volatile memory which will allow about 40 occurrences to be logged without compression. If an event type is enabled, the PS MUST log information required by the event type at a minimum rate of 1 event per every 5 seconds, even while under attack. It is expected that the PS will not consume the majority of its computing resources on logging and when attacks occur, the PS SHOULD be able to pass traffic at a normal rate and otherwise function normally.

11.6.4.5.2.1 Log Data

Logging can pose different problems if not properly done. Logging all events and packets can make the log complex, lengthy, and difficult to understand. It is difficult to sort through a lot of information to look for one item in particular. If logging is limited to only a few types of events, it will not provide enough information to the cable operator to debug intrusions or detect attacks. Note that logs can be sniffed if they are not encrypted. A hacker can use log information to gain insight into the various services running on the PS or LAN Host devices.

IPCable2Home requires a particular set of information to be logged for each type of event that is enabled. The log function MUST log packets of each type according to the rules for that type of event. The requirement for Date and Time assumes that the Date and Time will be as accurate as the last update of the PS clock during the provisioning sequence.

The cabhSec2FwLogTable for the event types 1, 2, 5, & 6 MUST record the following information for each occurrence unless otherwise specified:

- Event Number - MUST be recorded as defined in Annex B, one time, at the start of the log
- Event Priority - MUST be recorded as defined in Annex B, one time, at the start of the log
- Date and Time - when the event occurred:
- MUST consist of the four-digit year, month, and day
- MUST consist of the hour, minute, and second
- Protocol – the protocol indicated in the IP header field (1 = ICMP; 2= IGMP; 6 =TCP; 17= UDP)
- Source IP Address
- Destination IP Address
- Source Port (TCP and UDP)
- Destination Port (TCP and UDP)
- Message Type (ICMP) - [RFC 2474] defines ICMP and when the firewall blocks an ICMP packet the log MUST display a number indicating what type of ICMP message it was. 0 - Echo Reply, 3 - Destination Unreachable, 4 - Source Quench, 5 - Redirect, 8 - Echo Request, 9 - Router Advertisement, 10 - Router Solicitation, 11 - Time Exceeded, 12 - Parameter Problem, 13 - Timestamp Request, 14 - Timestamp Reply, 15 - Information Request, 16 - Information Reply, 17 - Address Mask Request, 18 - Address Mask Reply
- Replay Count - If the data being recorded is a replay attack, the firewall SHOULD NOT record each occurrence of the attack. However, the firewall SHOULD record the number of occurrences up to the threshold value set for the specific type

The cabhSec2FwLogTable for the event type 3 MUST record the following information for each occurrence unless otherwise specified:

- Event Number - MUST be recorded as defined in Annex B, one time, at the start of the log
- Event Priority - MUST be recorded as defined in Annex B, one time, at the start of the log
- Date and Time – when the event occurred:
- MUST consist of the four-digit year, month, and day
- MUST consist of the hour, minute, and second
- Source IP Address
- MIB object changed

The cabhSec2FwLogTable for the event type 4 MUST record the following information for each occurrence unless otherwise specified:

- Event Number - MUST be recorded as defined in Annex B, one time, at the start of the log.
- Event Priority - MUST be recorded as defined in Annex B, one time, at the start of the log
- Date and Time – when the event occurred:
- MUST consist of the four-digit year, month, and day
- MUST consist of the hour, minute, and second
- Source IP Address
- MIB object attempted to be changed
- Statement of failure - Type 4 events MUST state the failure and the reason for failure

11.6.4.6 Applications Through the Firewall

As part of the minimum set of capabilities, the firewall MUST be capable of allowing specified applications, as defined by Annex D, to traverse the PS to reach its intended destination. These applications are allowed, but most are not by default, instead, can be turned on by the cable operator. The firewall applies the current ruleset to the policy to ensure the correct openings are created to support specific traffic between the LAN and WAN, as well as to and from the PS itself.

The firewall policy is applied to the traffic as it attempts to traverse the firewall. The packets are first processed in the firewall prior to being sent to the PS for further processing, or to the destination on the WAN or LAN. The policy is applied to source and destination IP addresses, ports, and time of day. Annex D lists the requirements and

provides more detail.

11.6.4.7 Firewall MIB Objects

The firewall MIB objects consist of a three general groupings: 1) a set to manage the firewall configuration, 2) a set to monitor and log events, and 3) a set to manage the rulesets themselves. The requirements for the firewall MIB objects MUST be used in conjunction with the Security MIB document [Annex E.5].

11.6.4.7.1 Firewall Ruleset Management MIB Objects

The following firewall management objects MUST be implemented in the PS:

cabhSec2FwPolicyFileURL - contains the name of the policy rule set file and the IP address of the TFTP or HTTPS server containing the policy rule set file, in a TFTP or HTTPS URL format. A policy rule set file download is triggered when the value used to SET this MIB is different than the value in the cabhSec2FwPolicySuccessfulFileURL MIB. Refer to Section 7.4.4.2.3 Firewall Configuration File Trigger.

If the download of the Firewall Configuration File is not successful, the PS MUST NOT update the cabhSec2FwPolicySuccessfulFileURL MIB with the same value as the cabhSec2FwPolicyFileURL MIB. In any case, the cabhSec2FwPolicyFileURL MIB object MUST contain the value SET by either the PS Configuration File or by a SNMP SET command. When the PS is reset, the cabhSec2FwPolicyFileURL MIB object MUST be populated with its default value.

CabhSec2FwPolicySuccessfulFileURL - contains the name of the policy rule set file and the IP address of the TFTP server that contained the policy rule set file, in a TFTP or HTTPS URL format, which was used to trigger the last successful download. If a successful download has not yet occurred, this MIB should have a Null value.

cabhSec2FwPolicyFileHash - Defines the SHA-1 digest for the corresponding ruleset file.

cabhSec2FwPolicyFileOperStatus - Contains the operational status of the firewall configuration file download and it MUST contain the following three states:

- inProgress(1) - indicates that a firewall configuration file download is underway.
- complete(2) - indicates that the firewall configuration file has downloaded successfully.
- failed(3) - indicates that the last attempted download of the firewall configuration file failed.

cabhSec2FwPolicyFileCurrentVersion - A label set by the cable operator that can be used to track various versions of configured rulesets. Once the label is set, and it, along with the configured rules are changed, may not accurately reflect the version of configured rules running on the box. This object MUST contain the string “null”, if it has never been configured.

cabhSec2FwEnable - Allows for activation and deactivation of firewall. If this object is set to disabled, the firewall MUST be completely turned off. If this object is set to enable, the firewall MUST be activated immediately without re-booting the PS.

cabhSec2FwClearPreviousRuleset - Allows PS or firewall configuration files to contain a complete firewall configured ruleset, or an incremental to the already established configured ruleset, depending up on its existence in the configuration file. If the PS receives a configuration file with firewall settings, which includes a cabhSec2FwClearPreviousRuleset object setting marked as increment(1), or if this object setting is not included in a configuration file which contains filter settings for the firewall, the PS MUST treat the firewall filter settings in the configuration file as an increment to the configured ruleset. If the PS receives a configuration file with firewall settings which includes a cabhSec2FwClearPreviousRuleset object setting marked as incrementDefault(3), the PS MUST remove all previously configured rules from the configured ruleset, including any rules in the filter schedule table, and increment the newly downloaded rules on top of (i.e., subsequent to) the factory default policy. If the PS receives a configuration file with firewall settings which includes a cabhSec2FwClearPreviousRuleset object setting

marked as complete(2), the PS MUST remove all previously configured rules from the configured ruleset, including any rules in the filter schedule table, before applying the firewall filter settings contained in the configuration file.

If cabhSec2FwClearPreviousRuleset is set to increment(1) using SNMP, the PS MUST treat all of the following firewall filter settings using SNMP as an increment to the configured ruleset. If cabhSec2FwClearPreviousRuleset is set to incrementDefault(3) using SNMP, the PS MUST remove all previously configured rules from the configured ruleset, including any rules in the filter schedule table, and treat all of the following firewall filter settings, using SNMP, as an increment on top of the factory default policy. If cabhSec2FwClearPreviousRuleset is set to complete(2) using SNMP, the PS MUST remove all rules from the configured ruleset, including any rules in the filter schedule table. In this scenario the PS will operate without any configured rules, (e.g. there will be no defined filtering rules, but the firewall will still provide the minimum set of capabilities and architecture, as defined in Section 11.6.4.4.1 and Section 11.6.4.4.3). Default = increment (1)

cabhSec2FwPolicySelection - Allows for selection of the filtering policy for the factory default, or the configured ruleset:

- factoryDefault (1) - indicates the firewall is using the factory default settings. If this object is set to factoryDefault (1), the firewall MUST filter against the specified factory default policy.
- configuredRuleset (2) - indicates the firewall is using the rulesets configured by the cable operator. If this object is set to configuredRuleset (2), the firewall MUST use the last known configured ruleset.

cabhSec2FwEventSetToFactory - Allows the operator to clear all the events currently set in the event table. The PS MUST immediately clear the cabhSec2FwEventControlTable if this object is set to true.

cabhSec2FwEventLastSetToFactory - This object reports the last time the event table was cleared.

11.6.4.7.2 MIB Objects for Firewall Events

The following firewall event objects MUST be implemented in the PS, as defined in the Security MIB and are included in the cabhSec2FwEventControlTable:

cabhSec2FwEventType - Assigns the event type for the table to track. Event types are defined in Section 11.6.4.5.1.

cabhSec2FwEventEnable - Enables or disables counting and logging of firewall events by type as assigned in cabhSec2FwEventType. Logging requirements are defined in the log data section of this document. This is an on/off switch only. If the enable value changes, the PS MUST immediately send the appropriate event (8001010x). If this value is enabled, the firewall MUST log occurrences in the cabhSec2FwLog. The firewall MUST NOT count, send events, or collect log data for attacks when cabhSec2FwEventEnable is disabled. Default = false

cabhSec2FwEventThreshold - Number of attacks to count before sending the appropriate event by type as assigned in cabhSec2FwEventType. If the value is set to zero, the firewall MUST NOT count, send events, or collect log data for this type. Default = 0

cabhSec2FwEventInterval - Indicates the time interval in hours to count and log occurrences of a firewall event type as assigned in cabhSec2FwEventType. This time interval applies as long as the cabhSec2FwEventThreshold object is not exceeded. If the cabhSec2FwEventInterval MIB object has a value of zero, there is no interval assigned and the PS MUST NOT count, send, or log events. Default = 0

cabhSec2FwEventCount - Indicates the current count of attacks, up to the cabhSec2FwEventThreshold value by type as assigned by cabhSec2FwEventType. The firewall MUST start counting attacks from zero each time the cabhSec2FwEventEnable MIB object is enabled, or the cabhSec2FwEventInterval is over, or the cabhSec2FwEventCount equals the cabhSec2FwEventThreshold value. If the number of attacks counted in the cabhSec2FwEventCount equals the threshold set in the cabhSec2FwEventThreshold, prior to the end of the time interval defined by the cabhSec2FwEventInterval object, the PS MUST immediately

send the appropriate event (8001020x). Default = 0

cabhSec2FwEventLogReset - Setting this object to true clears the log table for the specified event type. Reading this object always returns false. Default = false

cabhSec2FwEventLogLastReset - This object reports the last time the log was cleared.

11.6.4.7.3 Firewall Policy MIB Objects

The firewall policy MIB objects provide a way for the cable operator to configure rules that will be used by the firewall to filter traffic. The cable operator can create any configured ruleset needed to filter traffic passing through the firewall on the PS. The firewall filtering policy MIB objects are based on the minimum set of filtering requirements. The firewall's filtering capability is similar to the filters defined in the cable industry CM MIB objects, specified in [RFC 2669]. Therefore, IPCable2Home had adopted some of the filtering objects already defined in [RFC 2669], and add some firewall specific MIB objects within the Security MIB.

Within [RFC 2669], the docsDevFilterIpTable provides the basic filtering properties. The docsDevFilterIpTable contains a sequence, docsDevFilterIpEntry, of MIB object. Each row in the table describes rules associated with IP addresses which is then compared to IP packets traversing the firewall. The template includes source and destination IP addresses (and their associated masks), upper level protocol (e.g. TCP, UDP), as well as the source and destination port ranges. This is the heart of the policy implementation. It is in this MIB table that the policy is defined and constructed. Each packet, inbound or outbound, shall be compared to the enabled policy

IPCable2Home defines a docsDevFilterIPTable extension, cabhSec2FwFilterScheduleTable that provides filter attributes for start time, end time and day of week to the filter settings in the docsDevFilterIPTable entries . This table allows a rule or filter to be enforced via the day of week, (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday), during a start and end time. For example, a parent may request that communications be denied between the WAN and the child's computer for Monday through Friday, 9pm to 7am and on Saturday and Sunday, 10pm to 8am. The firewall MUST NOT associate time restrictions with any filtering policies unless there is an explicit rule to define the time restrictions and it is clearly associated with known IP addresses.

The combination of filters defined in [RFC 2669], and in the Security MIB, allow for any rules to be created based on any combination of source IP address, destination IP address, source port, destination port, time of day, and day of week.

If there is not a match when the PS is comparing each inbound or outbound packet to the rules in the docsDevFilterIpTable, then the PS MUST apply the minimum set of firewall capabilities and architecture, as defined in sections 11.6.4.4.1 and 11.6.4.4.3. The docsDevFilterIpDefault flag defined in [RFC 2669] MUST be ignored

The following MIB objects MUST be implemented from [RFC 2669] to create the FilterIpTable for the filtering rules of the firewall. Unless otherwise noted in this section, the functionality is as specified in [RFC 2669]:

- docsDevFilterIpTable >>DocsDevFilterIpEntry
    - **docsDevFilterIpIndex**
    - consistent with [RFC 2669], the filter with the lowest index is always applied, meaning the filter is checked, then the PS MUST continue checking filters and apply the filter with the highest index in the case of conflicts
    - **docsDevFilterIpStatus**
    - **docsDevFilterIpControl**
    - the PS MUST ignore the setting (3) for policy; IPCable2Home does not use the policy table
    - **docsDevFilterIpIfIndex**
    - to filter for traffic coming in from the WAN, docsDevFilterIpIfIndex; MUST be set to 1
    - to filter for traffic coming in from the LAN, docsDevFilterIpIfIndex; MUST be set to 255
    - **docsDevFilterIpDirection**
    - this variable has no value for the firewall. Therefore, it should not matter what value is set in this object. However, since docsDevFilterIpDirection MUST be set to a value of 1, 2 or 3, set this mib object to both(3), since [RFC 2669] does not have an allowed value to ignore this object

- **docsDevFilterIpBroadcast**
- it is expected that this will always be the default value of false. Therefore, the rule will apply to all traffic
- **docsDevFilterIpSaddr**
- **docsDevFilterIpSmask**
- **docsDevFilterIpDaddr**
- **docsDevFilterIpDmask**
- **docsDevFilterIpProtocol**
- **docsDevFilterIpSourcePortLow**
- **docsDevFilterIpSourcePortHigh**
- **docsDevFilterIpDestPortLow**
- **docsDevFilterIpDestPortHigh**
- **docsDevFilterIpMatches**
- **docsDevFilterIpTos**
- this object can be ignored, it's function is not required.
- **docsDevFilterIpTosMask**
- this object can be ignored, it's function is not required.
- **docsDevFilterIpContinue**
- this object MUST always be set to true so the PS will continue checking filters until all the filters have been checked. Unlike RFC2669, this object MUST NOT trigger a discard until all the filters have been checked and there are no later filters which requires the packet to be accepted.
- **docsDevFilterIpPolicyId**
- this object can be ignored, it's function is not required.

Additionally, the firewall MUST support the following MIB objects as specified in the Security MIB document:

- **cabhSec2FwFilterScheduleStartTime** - The time to begin traffic restrictions as defined within the ruleset.
- **cabhSec2FwFilterScheduleEndTime -** The time to end traffic restrictions as defined within the ruleset.
- **cabhSec2FwFilterScheduleDOW -** The day of the week for which traffic restrictions will apply.

The cabhSec2FwFilterScheduleTable rules for the time and day restrictions are associated with policies as configured in the docsDevFilterIPTable. A packet processed with a date and time stamp within the restricted day and time, as specified by this table, MUST be denied.

11.7 Additional Security MIB Objects in the PS

The firewall MIB objects are described in the firewall section of this document. This section describes the remaining security MIB objects required. The security MIB objects are defined in more detail and MUST be supported as defined in Annex A.

11.7.1 Secure Software Download MIB Objects

Secure software download follows the design as created by J.112 Annex B, and as such, the MIB objects can be re-used in the PS just as the CM uses them. The PKI structure for IPCable2Home is defined separately and therefore some of the certificate MIBs MUST be used as defined by IPCable2Home, not by the J.112 MIBs, as currently written in [draft-ietf-ipcdn-bpiplus-mib-05].

The Standalone PS MUST support the following MIB objects as defined in the CL-SP-MIB-CLABDEF-I03-030411 [Annex E.6]:

- **clabCVCRootCACert** - Code Verification Root CA used for CVC validation
- **clabCVCCACert** - Code Verification CA used for CVC validation
- **clabMfgCVCCert** - Manufacturer Code Verification Certificate used to store the Mfg CVC Cert

The Standalone PS MUST support the following software download MIB objects defined in [draft-ietf-ipcdn-bpiplus-mib-05]:

- **docsBpi2CodeDownloadGroup** - Collection of objects that provide authenticated software download support. The docsBpi2CodeDownloadGroup includes:
    - **docsBpi2CodeDownloadStatusCode** - Indicates the result of the latest configuration file CVC verification, SNMP CVC verification, or code file verification.
    - **docsBpi2CodeDownloadStatusString** - Additional information to the status code.
    - **docsBpi2CodeMfgOrgName** - The device manufacturer's organizationName.
    - **docsBpi2CodeMfgCodeAccessStart** - The device manufacturer's current codeAccessStart value referenced to Greenwich Mean Time (GMT).
    - **docsBpi2CodeMfgCvcAccessStart** - The device manufacturer's current cvcAccessStart value referenced to Greenwich Mean Time (GMT).
    - **docsBpi2CodeCoSignerOrgName** - The Co-Signer's organizationName.
    - **docsBpi2CodeCoSignerCodeAccessStart** - The co-signer's current codeAccessStart value referenced to Greenwich Mean Time (GMT).
    - **docsBpi2CodeCoSignerCvcAccessStart** - The co-signer's current cvcAccessStart value referenced to Greenwich Mean Time (GMT).
    - **docsBpi2CodeCvcUpdate** - Triggers the device to verify the CVC and update the cvcAccessStart value.

11.7.2 Security Configuration File MIB Objects

The PS MUST support the following configuration file download MIB object as defined in the Security MIB:

cabhPsDevProvConfigHash - SHA-1 [FIPS 186] hash of the entire content of the configuration file, taken as a byte string.

11.7.3 Security Service Provider MIB Objects

The PS MUST support the following service provider authentication MIB object as defined in the Security MIB:

clabSrvcPrvdrRootCACert - The Service Provider Root CA used to validate certificates of devices on the service provider's network.

11.7.4 PS Certificate MIB Objects

The PS MUST support the following PS Certificate MIB object as defined in the Security MIB:

cabhSecCertPsCert - The X.509 DER-encoded PS certificate used to provide secure identity of the PS.

11.7.5 Kerberos MIB Objects

The needs of Kerberos within IPCable2Home is a subset of the functionality required by IPCablecom. The following MIB objects are required for IPCable2Home and the PS MUST support these MIB objects, as defined in the Security MIB:

- cabhSecKerbPKINITGracePeriod - The number of minutes prior to current ticket expiration for the PS to initiate a request with the KDC for a new ticket.
- cabhSecKerbTGSGracePeriod - The number of minutes prior to current ticket expiration for the PS to initiate a request with the KDC for a new ticket.
- cabhSecKerbUnsolicitedKeyMaxTimeout -The maximum timeout value for the AP Req/Rep exchange.
- cabhSecKerbUnsolicitedKeyMaxRetries - The maximum number of retries the PS is allowed to attempt AP Req/Rep negotiation

11.8 Secure Software Download for the PS

11.8.1 Goals of Secure Software Download

Secure Software Download goals include the following:

- The cable operator can securely load code into the PS as needed.
- The cable operator can manage secure downloads with various configuration policies.
- The security of the download will provide integrity, authentication, and if possible, encryption.
- The PS will only download images appropriate for the device.

11.8.2 Secure Software Download Design Guidelines

Table 11-19 IPCable2Home Security System Design Guidelines

Reference	Security System Design Guidelines
SEC13	The cable operator will have the ability to securely download software images to the PS element.

11.8.3 Secure Software Download System Description

Secure software download ensures that only a software image can be downloaded to the PS if the image is created by the same manufacturer. It also ensures that the image has not been modified since the manufacturer signed the code image. The image can also be signed by a Certification Testing Laboratory , as a co-signer, to guarantee that the image has been certified. For additional security on the download process, the cable operator can optionally sign any image as a co-signer to ensure that only images will be loaded into the PS that the cable operator has approved. The control mechanism for secure software download is to insert the code verification certificates (CVCs) into the configuration file which match the CVCs on the code image to be downloaded. After the PS has received CVC(s) in the configuration file, the PS is enabled to download the new code image when triggered via configuration file, or SNMP Set.

11.8.4 Secure Software Download Requirements

A Standalone PS Element MUST be capable of remotely downloading a software image over the network. As described in Section 6.3.3.2.4.9, secure software download to an Embedded PS is controlled by the cable modem. The new software image would allow the cable operator to improve performance, accommodate new functions and features, correct design deficiencies, and to allow a migration path for IPCable2Home devices as the IPCable2Home evolves. The software download capability MUST allow the functionality of the PS element to be changed without requiring that cable system personnel physically visit and reconfigure each unit. The Standalone PS secure software download process addresses the following primary system requirements:

- The mechanism used for software download MUST be TFTP file transfer.
- The software download MUST be initiated in one of two ways: 1) An SNMP set request issued by the NMS to the docsDevSwAdminStatus; 2) via the PS element's configuration file. If the Software Upgrade File Name in the configuration file does not match the current software image of the device, the PS element MUST request the specified file via TFTP from the Software Server.
- The PS element MUST verify that the downloaded software image is appropriate for itself. If the downloaded software image is appropriate, the PS element MUST write the new software image to non-volatile storage. Once the file transfer is completed successfully, the device MUST restart itself with the new code image.
- If the PS element is unable to complete the file transfer for any reason, the PS element MUST remain capable of accepting new software downloads (without operator or user interaction), even if power or connectivity is interrupted between attempts.
- The PS element MUST log software download failures and can report failures asynchronously to the network manager.
- Where software has been upgraded to meet a new version of this Recommendation, then it is critical

that the software MUST work with the previous version in order to allow a gradual transition of units on the network.

- The PS element MUST authenticate the downloaded software image.
- The PS element MUST verify that the downloaded code has not been altered from the original form in which it was provided by the trusted source.
- The software download process MUST provide a cable operator with mechanisms to upgrade or downgrade the code version of the IPCable2Home elements.
- The software download process MUST provide options for a cable operator to dictate their own download policies.
- The code file manufacturer MUST apply a Code Verification Signature (CVS) over the code image and any other authenticated attributes as defined in this specification for the PKCS#7 structure digital signature to the code file; the private key used to apply the signature MUST be bound to a public key certificate that chains up to the CVC root. The manufacturer's signature authenticates the source and integrity of the code file.
- A Co-Signer (cable operator or CTL) MAY counter sign the code file in addition to the manufacturer's signature.
- The PS element MUST be able to process a PKCS#7 digital signature and a [ITU-T X.509] certificate as defined in Section 11.8.4.1.1 and Section 11.3.4.1.1, respectively.
- (Optional): The PS element SHOULD be able to update the CVC Root CA Certificate stored in the device.
- (Optional): The PS element SHOULD be able to replace the Manufacturer CA Certificate(s) stored in the device.
- (Optional): The PS element SHOULD be able to update the CVC CA Certificate stored in the device.
- (Optional): The PS element SHOULD be able to update the Service Provider Root CA Certificate stored in the device.

The optional downloading of the Service Provider Root CA Certificate, CVC Root CA Certificate, CVC CA Certificate, and/or the Manufacturer CA Certificate, as a part of the Code File, are clearly separated from the code image and the other parameters in the code download file. It is possible to change the Service Provider Root CA Certificate, CVC Root CA Certificate, CVC CA Certificate, and/or the Manufacturer CA Certificate, understood by the PS element, by including the new certificates in the code image. Inclusion of the Manufacturer CVC Certificate and/or a co-signer CVC and corresponding CVS, permits the PS element to verify that the code image has not been altered since the Service Provider Root CA Certificate, CVC Root CA Certificate, CVC CA Certificate, and/or the Manufacturer CA Certificate, or SignedData parameters, are appended to the code image.

A IPCable2Home Complaint Residential Gateway device MAY include a cable modem and the PS Element, as separate entities or embedded as defined in the architecture section of this document.

- If the PS Element is embedded with a cable modem, the PS/CM image MUST be a single image, and the software download MUST be performed only by the cable modem.
- If the PS Element is composed of separate standalone entities, the software download for the IPCable2Home elements MUST be performed by the PS Element, as described below in this specification.

11.8.4.1 Code Download File Structure for Secure Software Download

For secure software download, the code download file is a file built using a [RFC 2315] compliant structure that has been defined in a specific format for use with PS Elements. The code file MUST comply with [RFC 2315] and MUST be DER encoded. The code file MUST match the structure shown in Table 11-20.

When certificates are downloaded as a part of the Code File, the certificates MAY be contained in the fields as specified in Table 11-20, and separated from the actual code image contained in the CodeImage field.

Table 11-20 Code File Structure

Code File	Description
PKCS#7 Digital Signature {	
ContentInfo	
ContentType	SignedData
SignedData ()	EXPLICT signed-data content value: includes CVS and [ITU-T X.509] compliant CVSs
} end [RFC 2315] Digital Signature	
SignedContent {	
Download Parameters {	Mandatory TLV format (Type 28). (Length is zero if there is no sub-TLVs).
MfgCACerts ()	Optional TLV for one or more DER-encoded certificate(s) each formatted according to the Manufacturer CA-Certificate TLV format (Type 17).
clabServProvRootCACert ()	Optional TLV for one DER-encoded certificate formatted according to the Service Provider Root CA-Certificate TLV Format (Type 50).
clabCVCRootCACert ()	Optional TLV for one DER-encoded certificate formatted according to the CVC Root CA Certificate TLV Format (Type 51).
clabCVCCACertificate ()	Optional TLV for one DER-encoded certificate formatted according to the CVC CA-Certificate TLV Format (Type 52).
}	
CodeImage ()	Upgrade code image.
} end SignedContent	

11.8.4.1.1 Signed Data

The code download file will contain the information in a [RFC 2315] Signed Data content type as shown in Table 11-21. Though maintaining compliance to [RFC 2315], the structure used has been restricted in format to ease the processing performed by the PS to validate the signature. The [RFC 2315] Signed Data MUST be DER encoded and exactly match the structure shown below, except for any change in order required to DER encode (e.g., the ordering of SET OF attributes). The PS element SHOULD reject the [RFC 2315] signature if the [RFC 2315] Signed Data does not match the DER encoded structure.

Table 11-21 PKCS#7 Signed Data

PKCS#7 Field	Description
Signed Data {	
version	version = 1
DigestAlgorithmIdentifiers	SHA-1
ContentInfo	
ContentType	data (SignedContent is concatenated at the end of the PKCS#7 structure)
certificates {	(CableLabs Code Verification Certification (CVC))
mfgCVC	(REQUIRED for all code files)
co-signerCVC	(OPTIONAL; required for co-signatures)
} *end certificates*	
SignerInfo{	
MfgSignerInfo {	(REQUIRED for all code files)
version	version = 1
issuerAndSerialNumber	
issuerName	
CountryName	US
organizationName	CableLabs
CommonName	CableLabs CVC Root CA
certificateSerialNumber	<Mfg CVC serial number>
digestAlgorithm	SHA-1
AuthenticatedAttributes	
contentType	data (contentType of signedContent)
signingTime	UTCTime (GMT), YYMMDDhhmmssZ
messageDigest	(digest of the content as defined in [PKCS#7])
digestEncryptionAlgorithm	rsaEncryption
EncryptedDigest	
} *end mfg signer info*	
CoSignerInfo {	(OPTIONAL; required for co-signatures)
version	version = 1
issuerAndSerialNumber	
issuerName	
CountryName	US
organizationName	CableLabs
CommonName	CableLabs CVC Root CA
certificateSerialNumber	<CoSigner CVC serial number>
digestAlgorithm	SHA-1
AuthenticatedAttributes	
contentType	data (contentType of signedContent)
signingTime	UTCTime (GMT), YYMMDDhhmmssZ
messageDigest	(digest of the content as defined in [PKCS#7])
digestEncryptionAlgorithm	rsaEncryption
EncryptedDigest	
} *end mso signer info*	
} *end signer info*	
} *end signed data*	

11.8.4.1.2 Signed Content

The signed content field of the code file contains the code image and the download parameters field, which possibly contains the following additional optional items:

- Service Provider Root CA Certificate
- Certification Testing Laboratory (CTL) CVC Root CA Certificate
- CTL CVC CA Certificate
- Manufacturer CA Certificate

The final code image is in a format compatible with the destination PS element. In support of the [RFC 2315] signature requirements, the code content is typed as data; i.e., a simple octet string. The format of the final code image is not specified here and will be defined by each manufacturer according to their requirements.

Each manufacturer SHOULD build their code with additional mechanisms that verify an upgrade code image is compatible with the destination PS element.

If included in the signed content field, a certificate is intended to replace the certificate currently stored in the PS

element. If the code download and installation is successful, the PS element MUST replace its currently stored certificate with the new certificate received in the signed content field. This new certificate will be used for subsequent verification.

11.8.4.1.3 Code Signing Keys

The [RFC 2315] digital signature uses the RSA Encryption Algorithm [PKCS #1] with SHA-1 [FIPS 186]. The PS element MUST be able to verify code file signatures. The public exponent is F_4 (65537 decimal).

11.8.4.1.4 Manufacturer CA Certificate

This Attribute is a string attribute containing an X.509 CA Certificate, as defined in [ITU-T X.509].

Type	Length	Value
17	Variable	X.509 CA Certificate (DER-encoded ASN.1)

11.8.4.1.5 Service Provider Root CA Certificate

This Attribute is a string attribute containing an X.509 Service Provider Root CA Certificate, as defined in [ITU-T X.509]. This certificate must be used by the PS Element in SNMP provisioning mode for mutual authentication.

Type	Length	Value
50	Variable	X.509 CA Certificate (DER-encoded ASN.1)

11.8.4.1.6 CVC Root CA Certificate

This Attribute is a string attribute containing an X.509 CVC Root CA Certificate, as defined in [ITU-T X.509]. This certificate must be used by the standalone PS Element in the secure software downloading process.

Type	Length	Value
51	Variable	X.509 CA Certificate (DER-encoded ASN.1)

11.8.4.1.7 CVC CA Certificate

This Attribute is a string attribute containing an X.509 CVC CA Certificate, as defined in [ITU-T X.509]. This certificate must be used by the standalone PS Element in the secure software downloading process.

Type	Length	Value
52	Variable	X.509 CA Certificate (DER-encoded ASN.1)

11.8.4.2 CVC Format for Secure Software Download

For secure software download, the format used for the CVC is [ITU-T X.509] compliant. However, the X.509 structure has been restricted to ease the processing a PS element does to validate the certificate and extract the public key used to verify the CVS. The CVC MUST be DER encoded and exactly match the structure shown in Table 11-22, except for any change in order required to DER encode (e.g., the ordering of SET OF attributes). The PS element SHOULD reject the CVC if it does not match the DER encoded structure represented in Table 11-22. The DER encoding MUST meet the requirements of Section 11.3.4.2, Public Key Infrastructure (PKI) of this specification.

Table 11-22 X.509 Compliant Code Verification Certificate

X.509 Certificate	Description
Certificate {	
version	2 (i.e., [ITU-T X.509] version 3)
serialNumber	integer, less than or equal to 20-octets (i.e., unique number assigned by the root CA)
signature	SHA-1 RSA, null parameters
issuer	
countryName	US
organizationName	
commonName	CVC Root CA
validity	
notBefore	utcTime (GMT), YYMMDDhhmmssZ (i.e., Time of issue)
notAfter	utcTime (GMT), YYMMDDhhmmssZ
subject	
countryName	<Country Name>
organizationName	<Company Name>
commonName	<Common Name>
subjectPublicKeyInfo	
algorithm	RSA encryption, null parameters
subjectPublicKey	2048-bit modulus
extensions	
KeyUsage	<Key usage>
authorityKeyIdentifier	<Authority key identifier>
signatureAlgorithm	SHA-1 RSA, null parameters
signatureValue	<Signature value>
} end certificate	

11.8.4.2.1 Certificate Revocation

This specification does not require or define the use of certificate revocation lists (CRLs). The PS element is not required to support CRLs. Operators can define and use CRLs to help manage code files provided to them by manufacturers. However, there is a method for revoking certificates based on the validity start date of the certificate. This method requires that an updated CVC be delivered to the PS element with an updated validity start time. Once the CVC is successfully validated, the X.509 validity start time will update the PS element's current value of cvcAccessStart.

11.8.4.3 Code File Access Controls

For secure software download, special control values are included in the code file for the PS element to check before it will validate a code image. The conditions placed on the values of these control parameters MUST be satisfied before the PS element will validate the CVC or the CVS, and accepts the code image.

11.8.4.3.1 Subject Organization Names

The PS element will recognize up to two names, at any one time, that it considers a trusted code-signing agent in the subject field of a code file CVC:

- The device manufacturer: The manufacturer name in the manufacturer's CVC subject field MUST exactly match the manufacturer name stored in the PS element's non-volatile memory by the manufacturer. A manufacturer CVC MUST always be included in the code file.
- A co-signing agent: It is permitted that another trusted organization co-sign code files destined to the device. In most cases, this is the cable operator controlling the current operating domain of the device.

The organization name of the co-signer is communicated to the PS element via a co-signer's CVC in the configuration file when initializing the PS element's code verification process. The co- signer's organization name in the co-signer's CVC subject field MUST exactly match the co-signer's organization name previously received in the co-signer's initialization CVC and stored by the PS element.

The PS element MAY compare organization names using a binary comparison.

11.8.4.3.2 Time Varying Controls

To mitigate the possibility of a PS element receiving a previous code file via a replay attack, the code files include a signing-time value in the PKCS#7 structure that can be used to indicate the time the code image was signed. The PS element MUST keep two UTC time values associated with each code-signing agent. One set MUST always be stored and maintained for the device's manufacturer. Additionally, if the code file is co-signed, the PS element MUST also store and maintain a separate set of time values for the co-signer.

These values are used to control code file access to the PS element by individually controlling the validity of the CVS and the CVC:

- codeAccessStart: a 12-byte UTC time value referenced to Greenwich Mean Time (GMT).
- cvcAccessStart: a 12-byte UTC time value referenced to GMT.

UTCTime values in the CVC MUST be expressed as GMT and MUST include seconds. That is, they MUST be expressed in the following form: YYMMDDhhmmssZ. The year field (YY) MUST be interpreted as follows:

- Where YY is greater than or equal to 50, the year shall be interpreted as 19YY.
- Where YY is less than 50, the year shall be interpreted as 20YY.

These values will always be referenced to Greenwich Mean Time, so the final ASCII character (Z) can be removed when stored by the PS element as codeAccessStart and cvcAccessStart.

The PS element MUST maintain each of these time values in a format that contains equivalent time information and accuracy to the 12 character UTC format (i.e., YYMMDDhhmmss). The PS element MUST accurately compare these stored values with UTC time values delivered to the PS element in a CVC. These requirements are discussed later in this specification.

The values of codeAccessStart and cvcAccessStart corresponding to the PS Element's manufacturer MUST NOT decrease. The value of codeAccessStart and cvcAccessStart, corresponding to the co-signer, MUST NOT decrease as long as the co-signer does not change and the PS element maintains that co-signer's time-varying control values.

11.8.4.4 Code Upgrade Initialization

11.8.4.4.1 Manufacturer Initialization

It is the responsibility of the manufacturer to correctly install the initial code version in the PS Element.

In support of secure software download, values for the Manufacturer's time-varying controls MUST be loaded into the PS Element's non-volatile memory:

- PS Element manufacturer's organizationName
- Manufacturer's time-varying control values:
- codeAccessStart initialization value
- cvcAccessStart initialization value

The organization name of the PS Element manufacturer MUST always be present in the device. The PS Element manufacturer's organizationName MAY be stored in the device's code image. The manufacturer named used for code upgrade is not necessarily the same name used in the Manufacturer CA Certificate.

The time-varying control values, codeAccessStart, and cvcAccessStart, MUST be initialized to a UTCTime compatible with the validity start time of the manufacturer's latest CVC. These time-varying values will be updated periodically under normal operation via manufacturer's CVC's that are received and verified by the PS element.

The Manufacturer MUST initialize the following certificates into the Standalone PS Element’s non-volatile memory:

- Service Provider Root CA Certificate
- CVC Root CA Certificate
- CVC CA Certificate
- Manufacturer CA Certificate
- PS Element Certificate

The Manufacturer MUST initialize the following certificates into the Embedded PS Element's non-volatile memory:

- Service Provider Root CA Certificate
- Manufacturer CA Certificate
- PS Element Certificate

11.8.4.4.2 Network Initialization

In support of code verification, the PS configuration file is used as an authenticated means in which to initialize the code verification process. In the PS element configuration file, the PS element receives configuration settings relevant to code upgrade verification.

The configuration file SHOULD always include the most up-to-date CVC applicable for the destination PS element. When the configuration file is used to initiate a code upgrade, it MUST include a Code Verification Certificate (CVC) to initialize the PS element for accepting code files according to this specification. Regardless of whether a code upgrade is required, a CVC in the configuration file MUST be processed by the PS element. A configuration file MAY contain:

- No CVC - The PS element MUST NOT accept a code file.
- A Manufacturer's CVC only - The PS element MUST verify that the manufacturer's CVC chains up to the CVC Root before accepting a code file. When the PS element's configuration file only contains a valid Manufacturer's CVC, the device will only require a manufacturer signature on the code files. In this case, the PS element MUST NOT accept code files that have been co-signed.
- A Co-Signer's (cable operator or CTL) CVC only - The PS element MUST verify the Co-Signer CV chains up to the CVC Root before accepting a code file. When the PS element's configuration file contains a valid co-signer's CVC, it is used to initialize the device with a co- signer. Once validated, the name of the CVC's subject organizationName will become the code co-signer assigned to the PS element. In order for a PS element to subsequently accept a code image, the co-signer, in addition to the device manufacturer, MUST have signed the code file.
- Both a Manufacturer's CVC and a Co-Signer's CVC - The PS element MUST verify that both CVCs chain up to the CVC Root before accepting a code file.

Before the PS element will enable its ability to upgrade code files on the network, it MUST receive a valid CVC in a configuration file. In addition, when the PS element's configuration file does not contain a valid CVC, and its ability to upgrade code files has been disabled, the PS element MUST reject any information in a CVC subsequently delivered via SNMP.

The organization name of the PS Element manufacturer and the manufacturer's time-varying control values MUST always be present in the PS element. If the PS element is initialized to accept code co-signed by an additional code-signer, the name of the organization and their corresponding time-varying control values MUST be stored and maintained while operational. Space MUST be allocated in the PS element's memory for the following co-signer's control values:

- co-signing agent's organizationName
- co-signer's time-varying control values:
- cvcAccessStart
- codeAccessStart

The manufacturer's set of these values MUST be stored in the PS element's non-volatile memory and not lost when the device's main power source is removed or during a reboot.

When a co-signer is assigned to the PS element, the co-signer's set of CVC values MUST be stored in the PS element's memory. The PS element MAY retain these values in non-volatile memory that will not be lost when the

device's main power source is removed or during a reboot. However, when assigning a PS element a co-signer, the CVC is always in the configuration file. Therefore, the PS element will always receive the co-signer's control values during the initialization phase and is not required to store the co-signer's time- varying control values when main power is lost or during a reboot process.

11.8.4.4.3 CVC Processing

To expedite the delivery of an updated CVC without requiring the PS to process a code upgrade, the CVC MAY be delivered in the configuration file or an SNMP Set message. The format of the CVC is the same whether it is in a code file, configuration file, or SNMP message.

11.8.4.4.3.1 Processing the Configuration File CVC

When a CVC is included in the configuration file, the PS element MUST verify the CVC before accepting any of the code upgrade settings it contains. At receipt of the CVC in the configuration file, the PS element MUST perform the following validation and procedural steps. If any of the following verification checks fail, the PS element MUST immediately halt the CVC verification process and log the error if applicable. If the PS Configuration File does not include a CVC that validates properly, the PS element MUST NOT download the upgrade code files whether triggered by the PS Configuration File, or via SNMP. In addition, if the PS Configuration File does not include a CVC that validates properly, the PS element is not required to process CVC's subsequently delivered, via an SNMP Set, and MUST NOT accept information from a CVC subsequently delivered, via an SNMP Set.

At receipt of the CVC in a configuration file, the PS element MUST:

1. Verify that the extended key usage extension is in the CVC as defined in Section 11.3.4.2.2.2.

2. Check the CVC subject organization name:

    a) If the CVC is a Manufacturer's CVC (Type 32) then:

        1. If, the organizationName is identical to the device's manufacturer name, then this is the manufacturer's CVC. In this case, the PS element MUST verify that the manufacturer's CVC validity start time is greater-than or equal-to the manufacturer's cvcAccessStart value currently held in the PS element.

        2. If, the organizationName is not identical to the device's manufacturer name, then this CVC MUST be rejected and the error logged.

    b) If the CVC is a Co-signer's CVC (Type 33) then:

        1. If, the organizationName is identical to the PS element's current code co-signer, then this is the current co-signer's CVC, and the PS element MUST verify that the validity start time is greater-than or equal-to the co-signer's cvcAccessStart value currently held in the PS element.

        2. If, the organizationName is not identical to the current code co-signer name, then after the CVC has been validated (and registration is complete), this subject organization name will become the PS element's new code co-signer. The PS element MUST NOT accept a code file unless it has been signed by the manufacturer, and co-signed by this code co-signer.

3. Validate the CVC issuer signature using the CTL CVC CA Public Key held by the PS element.

4. Validate the CTLCVC CA signature using the CTL CVC Root CA Public Key held by the PS element. Verification of the signature will authenticate the source and validate trust in the CVC parameters.

5. Update the PS element's current value of cvcAccessStart corresponding to the CVC's subject organizationName (i.e., manufacturer or co-signer) with the validity start time value from the validated CVC. If the validity start time value is greater than the PS element's current value of codeAccessStart, update the PS element's codeAccessStart value with the validity start time value. The PS element SHOULD discard any

remnants of the co-signer CVC.

11.8.4.4.3.2 Processing the SNMP CVC

The PS element MUST process SNMP delivered CVC's when enabled to upgrade code files. Otherwise, all CVC's delivered via SNMP MUST be rejected. When validating the CVC delivered via SNMP, the PS element MUST perform the following validation and procedural steps:

Note: If any of the following verification checks fail, the PS element MUST immediately halt the CVC verification process, log the error if applicable, and remove all remnants of the process to that step.
The PS element MUST:

1. Verify that the extended key usage extension is in the CVC, as defined in Section 11.3.4.2.2.2.

2. Check the CVC subject organization name:

   a) If, the organizationName is identical to the device's manufacturer name, then this is the manufacturer's CVC. In this case, the PS element MUST verify that the manufacturer's CVC validity start time is greater-than the manufacturer's cvcAccessStart value currently held in the PS element.

   b) If, the organizationName is identical to the PS element's current code co-signer, then this is a current co-signer's CVC and the validity start time MUST be greater-than the co-signer's cvcAccessStart value currently held in the PS element.

   c) If, the organizationName is not identical to device's manufacturer or current co-signer's name, then the PS element MUST immediately reject this CVC.

3. Validate the CVC issuer signature using the CTL CVC CA Public Key held by the PS element.

4. Validate the CVC issuer signature using the CTL CVC Root CA Public Key held by the PS element. Verification of the signature will authenticate the certificate and confirm trust in the CVC's validity start time.

5. Update the current value of the subject's cvcAccessStart values with the validated CVC's validity start time value. If the validity start time value is greater than the PS element's current value of codeAccessStart, update the PS element's codeAccessStart value with the validity start value.

11.8.4.5 Code Signing Requirements

11.8.4.5.1 Certificate Authority (CA) Requirements

Code Verification Certificates (CVCs) are signed and issued by the Certification Testing Laboratory (CTL) CVC CA. The CVC MUST be exactly as specified in Section 11.8.4.1.7. The CTL CVC CA MUST NOT sign any CVC unless it is identical to the format specified in Section 11.8.4.1.7. Before signing a CVC, the CTL CVC CA MUST verify that the certificate request is authentic.

The CTLCVC CA will be responsible for registering names of authorized CVC subscribers. CVC Subscribers include PS Element manufacturers and cable operator's that will co-sign code images. It is the responsibility of the CTL CVC CA to guarantee that the organization name of every CVC Subscriber is different. The following guidelines MUST be enforced when assigning organization names for code file co-signers:

- The organization name used to identify itself as a code co-signer agent in a CVC MUST be assigned by .
- The name MUST be a printable string of eight hexadecimal digits that uniquely distinguishes a code-signing agent from all others.
- Each hexadecimal digit in the name MUST be chosen from the character set 0-9 (0x30-0x39) or A- F (0x41-0x46).

- The string consisting of eight 0-digits is not allowed and MUST NOT be used in a CVC.

In any alternate format, all the information MUST be maintained and the original format MUST be reproduced; e.g., as a 32-bit nonzero integer, with an integer value of 0 representing the absence of a code-signer.

11.8.4.5.2 Manufacturer CVC Requirements

To sign their code files, the manufacturer MUST obtain a valid CVC from the CTL CVC CA. All manufacturer code images provided to an cable operator for remote upgrade of a device MUST be signed according to the requirements defined in this specification. When signing a code file, a manufacturer MAY choose not to update the [RFC 2315] signingTime value in the manufacturer's signing information. This specification requires that the [RFC 2315] signingTime value be equal-to or greater-than the CVC's validity start time. If the manufacturer uses a signingTime equal to the CVC's validity start time when signing a series of code files, those code files can be used and re-used. This allows a cable operator to use the code file to upgrade or downgrade the code version for that manufacturer's devices. These code files will be valid until a new CVC is generated and received by the PS element.

11.8.4.5.3 Cable Operator Requirements

When a cable operator receives software upgrade code files from a manufacturer, the cable operator will validate the code image using the CTL CVC CA Public Key. This will allow the cable operator to verify that the code image is as built by the trusted manufacturer. The cable operator can re-verify the code file at any time by repeating the process.

If a cable operator wants to exercise the option of co-signing the code image destined for a device on their network, the cable operator MUST obtain a valid CVC from the CTL CVC CA.

When signing a code file, the cable operator MUST co-sign the file content according to the PKCS#7 signature standard, and include their cable operator CVC, as defined in Section 11.8.4.1.1. IPCable2Home does not require a cable operator to co-sign code files. However, when the cable operator follows all the rules defined in this specification for preparing a code file, the PS element MUST accept it.

11.8.4.6 Triggering Process

Code downloads, regardless of the provisioning mode, can be initiated during the provisioning and registration process via a configuration-file-initiated download, or, during normal operation, using an SNMP-initiated download command. The PS element MUST support both methods.

Note: Prior to triggering a secure software download, appropriate CVC information MUST be included in the configuration file. If the operator decides to use the SNMP-initiated download as a method to trigger a secure software download. It is recommended that CVC information always be present in the configuration file so that a PS element will always have the CVC information initialized when needed. If the operator decides to use the configuration-file-initiated download as a method to trigger secure software download, CVC information is needed to be present in the configuration file at the time the device is rebooted to get the configuration file that will trigger the upgrade.

11.8.4.6.1 SNMP-initiated Software Download

From a network management station:

- Set docsDevSwServer to the address of the TFTP server for software upgrades
- Set docsDevSwFilename to the file pathname of the software upgrade image
- Set docsDevSwAdminStatus to Upgrade-from-mgt. docsDevSwAdminStatus MUST persist across reset/reboots until over-written from an SNMP manager, or via the PS element configuration file.

The default state of docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2} until it is over-written by ignoreProvisioningUpgrade{3}, following a successful SNMP-initiated software upgrade, or otherwise altered by the management station. docsDevSwOperStatus MUST persist across resets to report the outcome of the last software upgrade attempt.

If a PS element suffers a loss of power or resets during SNMP-initiated upgrade, the PS element MUST resume the upgrade without requiring manual intervention, and when the PS element resumes the upgrade process:

- docsDevSwAdminStatus MUST be Upgrade-from-mgt{1}
- docsDevSwFilename MUST be the filename of the software image to be upgraded
- docsDevSwServer MUST be the address of the TFTP server containing the software upgrade image to be upgraded
- docsDevSwOperStatus MUST be inProgress{1}
- docsDevSwCurrentVers MUST be the current version of software that is operating on the device

In case the PS element reaches the maximum number of retries (max retries = 3) resulting from multiple losses of power or resets during an SNMP-initiated upgrade, the PS element's status MUST adhere to the following requirements after it is registered:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade process.
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the device

If a PS element exhausts the required number of TFTP retries by issuing a total of 16 consecutive retries, the PS element MUST fall back to the last known working image, proceed to an operational state, and adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade process
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be failed{4}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the device

After the PS element has completed the SNMP-initiated secure software upgrade, the PS element MUST reboot and become operational with the correct software image. When the device is operational, it MUST adhere to the following requirements:

- set its docsDevSwAdminStatus to ignoreProvisioningUpgrade{3}
- set its docsDevOperStatus to completeFromMgt{3}
- reboot

The PS element MUST properly use ignoreProvisioningUpgrade status to ignore the software upgrade value that can be included in the PS element configuration file. The PS MUST become operational with the correct software image and it MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be ignoreProvisioningUpgrade{3}
- docsDevSwFilename MAY be the filename of the software currently operating on the PS element
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the PS element
- docsDevSwOperStatus MUST be completeFromMgt{3}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the PS element

In the case where PS element successfully downloads (or detects during download), an image that is not intended for the device the:

- DocsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- DocsDevSwFilename MUST be the filename of the software that failed the upgrade
- DocsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- DocsDevSwOperStatus MUST be other{5}

- DocsDevSwCurrentVer MUST be the current version of software that is operating on the device

In the case where PS element determines that the download image is damaged or corrupted, the PS element MUST reject the newly downloaded image. The PS element MAY re-attempt to download if the MAX number of TFTP sequence retries has not been reached. If the PS element chooses not to retry and the MAX number of TFTP sequence retries has not been reached, the PS element MUST fall back to the last known working image and proceed to an operational state, generate an appropriate event notification, as specified in Section 11.8.4.8, and adhere to the following requirements:

- DocsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- DocsDevSwFilename MUST be the filename of the software that failed the upgrade
- DocsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- DocsDevSwOperStatus MUST be other{5}
- DocsDevSwCurrentVer MUST be the current version of software that is operating on the device

In the case where PS element determines that the image is damaged or corrupted, the PS element MUST reject the newly downloaded image. The PS element MAY re-attempt to download the new image if the MAX number of TFTP sequence retries has not been reached. On the 16th consecutive failed software download attempt, the PS element MUST fall back to the last known working image and proceed to an operational state. In this case, the PS element is required to send two notifications; one to notify that the MAX TFTP retry limit has been reached, and another to notify that the image is damaged. Immediately after the PS element reaches the operational state, the PS element MUST adhere to the following requirements:

- DocsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- DocsDevSwFilename MUST be the filename of the software that failed the upgrade
- DocsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- DocsDevSwOperStatus MUST be other{5}
- DocsDevSwCurrentVer MUST be the current version of software that is operating on the device

11.8.4.6.2 Configuration-file-initiated Software Download

The configuration-file-initiated software download is initiated by sending the Software Upgrade File Name in the PS element's configuration file. If the Software Upgrade File Name in the PS element's configuration file does not match the current software image of the device, the PS element MUST request the specified file, via TFTP, from the Software Server.

Note: The Software Server IP Address is a separate parameter. If present, the PS element MUST attempt to download the specified file from this server. If not present, the PS element MUST attempt to download the specified file from the configuration file server.

In a case where the PS element reaches the maximum number of retries (max retries = 3) resulting from multiple loss of powers, or resets during a configuration-file-initiated upgrade, the PS element's status MUST adhere to the following requirements, after it is registered:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade process
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the device

If a PS element exhausts the required number of TFTP retries by issuing a total of 16 consecutive retries, the PS element MUST fall back to the last known working image, proceed to an operational state, and adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade process

- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be failed{4}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the device

After the PS element has completed the configuration-file-initiated secure software upgrade, the PS element MUST reboot and become operational with the correct software image. After the PS element is registered the:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MAY be the filename of the software currently operating on the device
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the device
- docsDevSwOperStatus MUST be completeFromProvisioning{2}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the device

11.8.4.7 Code Verification

For secure software download, the PS element MUST perform the verification checks presented in this section. If any of the verification checks fail, or if any portion of the code file is rejected due to invalid formatting, the PS element MUST immediately halt the download process, log the error if applicable, remove all remnants of the process to that step, and continue to operate with its existing code.

The following verification checks can be made in any order, as long as all of the applicable checks presented in this section are made:

1. The PS element MUST validate the manufacturer's signature information by verifying that the [RFC 2315] signingTime value is:

   a) equal-to or greater-than the manufacturer's codeAccessStart value currently held in the PS element.

   b) equal-to or greater-than the manufacturer's CVC validity start time.

   c) less-than or equal-to the manufacturer's CVC validity end time.

2. The PS element MUST validate the manufacturer's CVC by verifying that the:

   a) CVC subject organizationName is identical to the manufacturer name currently stored in the PS element's memory.

   b) CVC validity start time is equal-to or greater-than the manufacturer's cvcAccessStart value currently held in the PS element.

   c) extended key usage extension is in the CVC as defined in Section 11.3.4.2.2.2.

3. The PS element MUST validate the certificate signature using the CTL CVC CA Public Key held by the PS element. In turn, the CTL CVC CA Certificate signature is validated by the CTL CVC Root CA Public Key held by the PS element. Verification of the signature will authenticate the source of the public code verification key (CVK) and confirm trust in the key.

4. The PS element MUST verify the manufacturer's code file signature:

   a) The PS element MUST perform a new SHA-1 hash over the SignedContent. If the value of the messageDigest doesn't match the new hash, the PS element MUST consider the signature on the code file as invalid.

   b) If the signature does not verify, all components of the code file (including the code image), and any values derived from the verification process, MUST be rejected and SHOULD be immediately discarded.

5. If the manufacturer signature verifies, and a co-signing agent signature is required:

    a) The PS element MUST validate the co-signer's signature information by verifying that the:

        (1) co-signer's signature information is included in the code file.

        (2) [RFC 2315] signingTime value is equal-to or greater-than the corresponding codeAccessStart value currently held in the PS element.

        (3) [RFC 2315] signingTime value is equal-to or greater-than the corresponding CVC validity start time.

        (4) [RFC 2315] signingTime value is less-than or equal-to the corresponding CVC validity end time.

    b) The PS element MUST validate the co-signer's CVC, by verifying that the:

        (1) CVC subject organizationName is identical to the co-signer's organization name currently stored in the PS element's memory.

        (2) CVC validity start time is equal-to or greater-than the cvcAccessStart value currently held in the PS element for the corresponding subject organizationName.

        (3) extended key usage extension is in the CVC as defined in Section 11.3.4.2.2.2.

    c) The PS element MUST validate the certificate signature using the CTL CVC CA Public Key held by the PS element. In turn, the CTL CVC CA certificate signature is validated by the CTL CVC Root CA Public Key held by the PS element. Verification of the signature will authenticate the source of the co-signer's public code verification key (CVK) and confirm trust in the key.

    d) The PS element MUST verify the co-signer's code file signature.

    e) The PS element MUST perform a new SHA-1 hash, over the SignedContent. If the value of the messageDigest doesn't match the new hash, the PS element MUST consider the signature on the code file as invalid.

    f) If the signature does not verify, all components of the code file (including the code image), and any values derived from the verification process, MUST be rejected and SHOULD be immediately discarded.

6. If the manufacturer's, and optionally, the co-signer's signature has verified, the code image can be trusted and installation can proceed. Before installing the code image, all other components of the code file and any values derived from the verification process, except the [RFC 2315] signingTime values and the CVC validity start values, SHOULD be immediately discarded.

7. If the code installation is unsuccessful, the PS element MUST reject the [RFC 2315] signingTime values and CVC validity start values it just received in the code file.

8. When the code installation is successful, the PS element MUST update the manufacturer's time-varying controls with the values from the manufacturer's signature information and CVC:

    a) Update the current value of codeAccessStart with the [RFC 2315] signingTime value

    b) Update the current value cvcAccessStart with the CVC validity start value

9. When the code installation is successful, and if the code file was co-signed, the PS element MUST update the co- signer's time-varying controls with the values from the co-signer's signature information and CVC:

    a) Update the current value of codeAccessStart with the [RFC 2315] signingTime value

b) Update the current value of cvcAccessStart with the CVC validity start value

11.8.4.8 Error Codes

Error codes are defined to reflect the failure states possible during the secure software download code verification process.

1. Improper code file controls:

   a) CVC subject organizationName for manufacturer does not match the PS element's manufacturer name.

   b) CVC subject organizationName for code co-signing agent does not match the PS element's current code co-signing agent.

   c) The manufacturer's [RFC 2315] signingTime value is less-than the codeAccessStart value currently held in the PS element.

   d) The manufacturer's [RFC 2315] validity start time value is less-than the cvcAccessStart value currently held in the PS element.

   e) The manufacturer's CVC validity start time is less-than the cvcAccessStart value currently held in the PS element.

   f) The manufacturer's [RFC 2315] signingTime value is less-than the CVC validity start time.

   g) Missing or improper extended key-usage extension in the manufacturer CVC

   h) The co-signer's [RFC 2315] signingTime value is less-than the codeAccessStart value currently held in the PS element.

   i) The co-signer's [RFC 2315] validity start time value is less-than the cvcAccessStart value currently held in the PS element.

   j) The co-signer's CVC validity start time is less-than the cvcAccessStart value currently held in the PS element.

   k) The co-signer's [RFC 2315] signingTime value is less-than the CVC validity start time.

   l) Missing or improper extended key-usage extension in the co-signer's CVC.

2. Code file manufacturer CVC validation failure.
3. Code file manufacturer CVS validation failure.
4. Code file co-signer CVC validation failure.
5. Code file co-signer CVS validation failure.
6. Improper Configuration File CVC format (e.g., Missing or improper key usage attribute)
7. Configuration File CVC validation failure.
8. Improper SNMP CVC format:

   a) CVC subject organizationName for manufacturer does not match the device's manufacturer name.

   b) CVC subject organizationName for code co-signing agent does not match the PS element's current code

co-signing agent.

c) The CVC validity start time is less-than or equal-to the corresponding subject's cvcAccessStart value currently held in the PS element.

d) Missing or improper key usage attribute.

9. SNMP CVC validation failure.

11.8.4.9 Software Downgrade

The Software Downgrade defines the process of removing the upgraded version of the software image download, thus reverting the Cable Home Device to the exact previous state.

When the PS element receives a code file with a signing-time that is later than the signing-time it has in its memory, the device MUST update its internal memory with the received value.

Because the PS element will not accept code files with an earlier signing-time than this internally stored value, to upgrade a device with a new code file without denying access to past code files, the signer (e.g., the Manufacturer, the cable operator, CTL) can choose not to update the signing-time. In this manner, multiple code files with the same code signing-time allows an operator to freely downgrade a device's code image to a past version (that is, until the CVC is updated). This has a number of advantages for the cable operator, but these advantages will be weighed against the possibilities of a code file replay attack.

Another approach would be to sign the previous code file with a signing-time that is equal to or greater than the signing-time of the last upgrade.

11.9 PS Configuration File Security in DHCP Provisioning Mode

11.9.1 Configuration File Security Infrastructure Goals

The goals for securing the configuration file include:

- Provide an authenticated tunnel between the PS client device and HTTPS server to ensure that configuration files are secured from the cable operator to the PS. An integrity check is automatically included when a message is authenticated.
- Encryption of configuration files while in transport to reduce the possibility of eavesdropping on firewall and PS configuration.
- Reduce the risk of an unauthorized configuration file download to the PS by an unauthorized source.

11.9.2 Configuration File Security System Design Guidelines

Table 11-23 Security System Design Guidelines

Reference	Security System Design Guidelines
Sec14	The cable operator will have the ability to authenticate and optionally encrypt the transport of configuration files for the PS or firewall.

11.9.3 Configuration File Security System Description

In DHCP provisioning mode, the cable operator can choose to turn on security for the configuration file download. Within this section, the term configuration file refers to the PS configuration file, or the firewall configuration file. Security is provided by establishing a TLS session between the PS and the HTTPS server. IPCable2Home requires the PS to understand this security option and to use TLS within the provisioning sequence to provide a secure session between the HTTPS Server and the PS, for the purposes of downloading the PS configuration file, and the

firewall configuration file, in a secure manner. TLS provides authentication and encryption for the session, as configured by the cable operator. The session is torn down prior to sending the Syslog and/or NMS notification provisioning completed message. The configuration file download trigger, management, and contents remain as industry standards when TLS is layered under the HTTPS protocol. IPCable2Home specifies the requirements for an [RFC 2246] complaint TLS session. The TLS options are tightened to create a minimum set of interoperable behavior for the PS. The provisioning flow with HTTP/TLS is described in detail in Section 13.

TLS provides an encrypted and authenticated transport tunnel for any application above TLS in the ISO stack. The HTTP protocol itself is not affected by the TLS layering. The italicized and underlined layers in the stack are encrypted for a standard TLS data packet. The HTTP protocol, which normally sits on TCP, sits directly on TLS.

Table 11-24 TLS Encryption

Configuration File Data (Payload)
HTTP
TLS
TCP
IP
MAC
PHY

11.9.4 Configuration File Security Requirements

The PS MUST implement the Transport Layer Security (TLS) protocol as defined by [RFC 2246], TLS Protocol Version 1.0, with the exceptions as listed in this specification. The exceptions within this specification are intended to simplify the requirements necessary for implementation and testing purposes. Some of the exceptions place a minimum set requirements that already align with other technology used within the cable industry. The requirements placed will ensure the PS shall provide a consistent level of performance for the cable operators. This section also helps remove any ambiguity and define processes which are not defined in the RFCs, but is needed for IPCable2Home purposes. This is especially true in the case of failure handling.

Note: The compression algorithm feature of TLS will not be used.
TLS version 1.0 (SSL3, TLSv1) MUST be supported. Earlier versions of TLS MUST NOT be supported by the PS. The PS MUST reject messages from the server if it attempts to use previous TLS versions.

11.9.4.1 Triggering TLS

To trigger a secure configuration file download in DHCP provisioning mode, the DHCP Ack will contain the IP address of the HTTPS server in the siaddr field. The DHCP Ack will also contain option 72 with the IP address of the HTTPS server. If the IP address in the siaddr field and the first IP address in option 72 match, the PS MUST establish a TLS session with the HTTPS server at the IP address listed in the ack, prior to requesting the configuration file. The PS MUST download the configuration file using HTTP/TLS, if the first IP address in TLV option 72 matches that IP address in siaddr, of the DHCP Ack message. If the PS does not receive a match in the DHCP ack, the PS MUST NOT initiate a TLS session, the requirements in this section are not applicable, and the PS client MUST use DHCP provisioning mode with the specified TFTP download process. The provisioning flow diagram and description table are specified in Section 13. If option 66 is included as well as option 72, and the IP address in option 72 matches the IP address in the siaddr field, the PS MUST initiate a TLS session to the HTTPS server and MUST NOT initiate download from the TFTP server listed in option 66.

If the PS receives, in the PS configuration file, the necessary information to initiate a separate firewall configuration file, as specified in Section 6, the PS MUST determine it is needs to continue the TLS session with the HTTPS server, which delivered the PS configuration file or establish a new TLS session with a different HTTPS server for firewall configuration file download. If the PS is instructed to download a firewall configuration file to a different HTTPS server, the one used to download the PS configuration file, the PS MUST establish a TLS session, as specified by this document, prior to requesting the firewall configuration file.

11.9.4.2 TLS Session Prerequisites

Prior to establishing a TLS session, the PS client MUST synchronize its clock with the TOD server. Details are specified in Section 13.

Additionally, the PS client MUST establish the TCP/IP connection to the HTTPS server prior to sending the TLS ClientHello. Once the configuration file download is complete, the PS MUST close the TCP/IP connection. The PS client MUST use TCP port #443, specified by IANA standards, to connect to the HTTP/TLS server. If the TCP/IP connection cannot be made after 5 attempts, with 30 seconds allowed for each attempt, the PS MUST send event 68002000.

11.9.4.3 TLS Messages

Unless otherwise noted, all the messages are [RFC 2246] compliant.

11.9.4.3.1 ClientHello

The PS client MUST send a ClientHello to the HTTP/TLS server to initiate the TLS Handshake sequence. After the initial ClientHello message has been sent to the HTTP/TLS server, if the TLS session is not established after 5 attempts, with 30 seconds allowed for each attempt, the PS MUST fail the session and send event 68002100.

11.9.4.3.2 PS Processing of the Server Messages

The PS MUST be able to process the server messages, as defined in [RFC 2246], with the following exceptions:

- HelloRequest: The PS MUST ignore HelloRequest messages from a server. This protects the PS from answering rogue requests from HTTPS servers. The HTTP/TLS process can only be initiated if the appropriate DHCP options are configured by the cable operator. This assumes DHCP is trusted, even though it is not secured by IPCable2Home.
- ServerCertificate: The HTTPS server is expected to send its device certificate to the PS within the ServerCertificate message. In addition to the [RFC 2246] requirements for this message, the PS client MUST validate and verify the HTTPS server certificate. If the HTTPS server certificate authentication fails, the TLS session is considered a failure and the PS MUST send event 68002200 with the [RFC 2246] defined error code.

11.9.4.3.3 ClientCertificate

The PS MUST send its PS Element certificate to the HTTPS server within the ClientCertificate message. It is expected the HTTPS server will validate and verify the PS client certificate prior to proceeding with the handshake. If the PS certificate is not successfully authenticated by the server, the PS client MUST treat the received alert message as a fatal alert and send event 68002200 with the appropriate error code from [RFC 2246].

11.9.4.4 TLS Ciphersuites and Compression

Within the ClientHello message, the requested ciphersuite MUST be listed. The required ciphersuite support is a subset of [RFC 2246] to align with the technology already used within the cable industry. The cable operator will need to select the appropriate encryption and authentication algorithm on the HTTPS server to communicate to the PS that meets the security model for that operator. The ciphersuites required in this specification are a subset of those available and the PS can support additional ciphersuites.

The following cryptographic algorithms MUST be support by the PS.

- TLS_NULL_WITH_NULL_NULL
- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

The compression feature of the TLS protocol is not required. Therefore, the PS client MUST use compressionMethod.null, as the compression type.

11.9.4.5 TLS Session Tear Down

If the PS is required to download a separate firewall configuration file immediately after the PS configuration file is downloaded, and the firewall configuration file will be downloaded from the same HTTPS server as the PS configuration file was downloaded from, the TLS session is expected to remain active. The PS MUST ensure the TLS and corresponding TCP/IP session is closed with each HTTPS server after:

The PS configuration file is downloaded, if, and only if there is no firewall configuration file to be downloaded from the same HTTPS server, immediately after the PS configuration file is processed

The firewall configuration file is downloaded and processed

11.9.4.6 TLS Events

[RFC 2246] defines an alert protocol to handle closure and errors for TLS. The TLS alerts and errors MUST be supported and used as defined in [RFC 2246], except the decompression_failure (30) alert will not be used, since compression is not supported. All TLS alerts MUST be recorded by the PS using event 68002200 with the appropriate error code defined in [RFC 2246]. The certificate related errors MUST be treated as fatal, since the PS and HTTP rely on client and server authentication.

If the PS client has not received a message from the HTTP/TLS server in response to any TLS message sent after 5 attempts, with 30 seconds allowed for each attempt, the TLS connection is considered a failure, and the PS MUST send event 68002100.

11.9.4.7 HTTP Download and Events

The HTTP transfer MUST only be initiated after the TLS handshake has been completed. The PS MUST communicate to the HTTP/TLS server using standard HTTP, as defined by [RFC 2616]. The PS client MUST initiate an HTTP version 1.1 request to the server for the PS configuration file, or the firewall configuration file. The PS configuration filename used in the HTTP "GET Request" MUST be the same filename the PS received in the DHCP ack. The firewall configuration filename used in the HTTP "GET Request" MUST be the same filename the PS received in the PS configuration filename, or via SNMP set.

The PS client MUST handle all status messages according to [RFC 2616]. If the PS client receives an HTTP status message indicating that the HTTP download cannot be completed, the PS MUST fail the session and send event 68003000, with the appropriate error code from [RFC 2616]. If the download cannot be completed after 5 attempts, with 240 seconds allowed for each attempt, the PS MUST fail the session and send event 68003100.

Note: A long timeout is given to include the configuration file download, which at times, can unfortunately be slow. Once the configuration file is downloaded successfully, the PS MUST send event 68003200.

11.10 Physical Security

The PS is required to maintain, in its non-volatile memory, keys and other crypto-variables related to network security. The PS MUST deter unauthorized physical access to this cryptographic material.

The level of physical protection of keying material required for the PS is specified in terms of the security levels defined in the FIPS PUBS 140-2, Security Requirements for Cryptographic Modules, standard [FIPS 140-2]. In particular, the PS MUST meet FIPS PUBS 140-2 Security Level 1 requirements.

FIPS PUBS 140-2 Security Level 1 requires minimal physical protection through the use of production-grade enclosures and recommended software practices.

11.11 Cryptographic Algorithms

11.11.1 SHA-1

The PS implementation of SHA-1 MUST use the SHA-1 hash algorithm as defined in [FIPS 180-1].

12 MANAGEMENT PROCESSES

12.1 Introduction/Overview

This section provides examples of processes associated with the use of the tools described in Section 6 (Management Tools), and additional processes that facilitate other required management functions defined in this specification. PS Database access and other PS operations of the IPCable2Home Management Portal (CMP) are described in Section 6. Typical MIB access rules are provided in Section 6.3.3.1.4.2.

Management-related and other descriptive processes are provided for the following scenarios:

- Management Tool Processes
    - CTP Operation
        - Connection Speed Tool
        - Ping Tool
- PS Operation
    - PS Database Access
    - Reconfiguration
        - PS Software Download
        - PS Configuration File Download
- MIB Access
    - VACM Configuration
    - Management Event Messaging Configuration
        - CMP Event Notification Operation
        - CMP Event Throttling and Limiting Operation

12.1.1 Goals

This section is primarily composed of informative text, intended to aid in understanding, and does not contain requirements. The examples describe how the Management Tools are used to accomplish typical management functions. Sequence charts of additional management-related processes (i.e., those not defined in Section 6) are also provided, including management processes or process steps associated with the use of required Management Tools. All processes shown involve interaction of the PS element with Headend systems.

12.2 Management Tool Processes

Management Tool Processes are those associated with the required Management Tools defined in Section 6.

12.2.1 CTP Operation

The IPcable2Home Test Portal (CTP) provides Connection Speed Tool and Ping Tool capabilities, described in Section 6.4.3.1 and Section 6.4.3.2, respectively.

12.2.1.1 Remote Connection Speed Test

The Remote Connection Speed Test can be useful in validating performance levels, identifying possible configuration errors, and determining other performance-oriented characteristics:

1. The Network Management System (NMS) starts the test by initializing the test parameters and setting the

Begin Test flag, via SNMP SET Request.

2. The CMP SNMP Agent updates the PS Database with the test parameters and notifies the CTP to begin the test.

3. The CTP queries the PS database for the test parameters.

4. The CTP issues a burst of UDP packets to port 7 of the specified LAN IP Device. Port 7 is reserved for the echo service.

5. The target LAN IP Device simply echoes the UDP packet payload back to the CTP

6. Once all of the packets have been received, or the test timeout period has expired, the CTP updates the PS Database with the results of the test and sets the Test Complete flag.

7. The NMS verifies that the command is complete by checking Status = complete.

8. The NMS requests the test results via SNMP GET Request.

9. The CMP SNMP agent queries the PS database for the test results and reports them in the SNMP GET Response. If the test has not completed, the test data will indicate the test is still running. The NMS must repeat the SNMP GET Request until the test results indicate the test has completed.

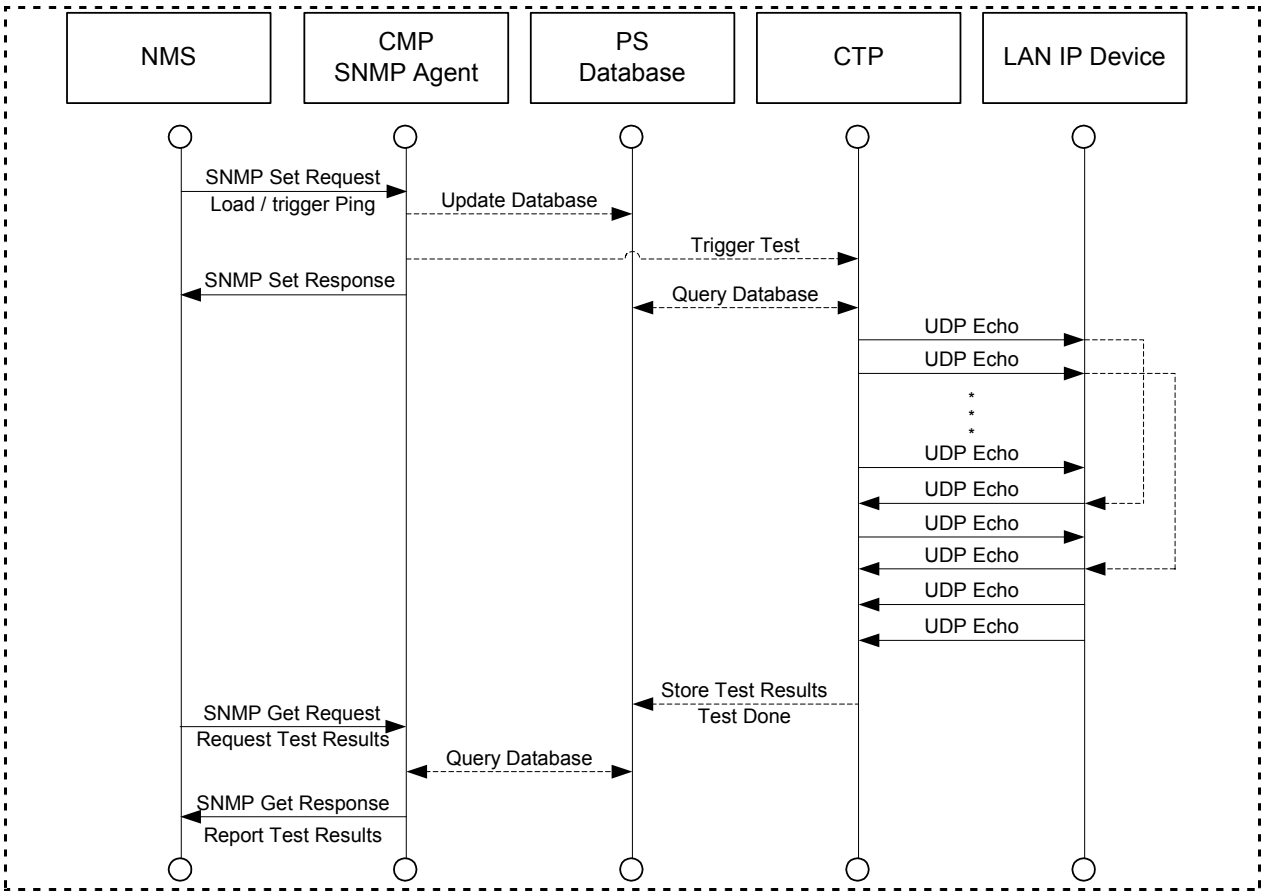

Figure 12-1 Connection Speed Tool Process Sequence Diagram

12.2.1.2 Ping Tool Process

The Ping Tool can be useful in validating connectivity state, performance levels, and identifying possible configuration errors.

1. The NMS starts the test by initializing the test parameters and setting the Begin Test flag, via SNMP SET Request.

2. The CMP SNMP Agent updates the PS Database with the test parameters and notifies the CTP to begin the test.

3. The CTP queries the PS database for the test parameters.

4. The CTP issues an ICMP Echo Request packet to the specified LAN IP Device.

5. The target LAN IP Device responds with an ICMP Echo Response.

6. The CTP updates the PS Database with the results of the test and sets the Test Complete flag.

7. The NMS verifies that the command is complete by checking Status = complete.

8. The NMS requests the test results via SNMP GET Request.

9. The CMP SNMP agent queries the PS database for the test results and reports them in the SNMP GET Response. If the test has not completed, the test data will indicate the test is still running. The NMS must repeat the SNMP GET Request until the test results indicate the test has completed.

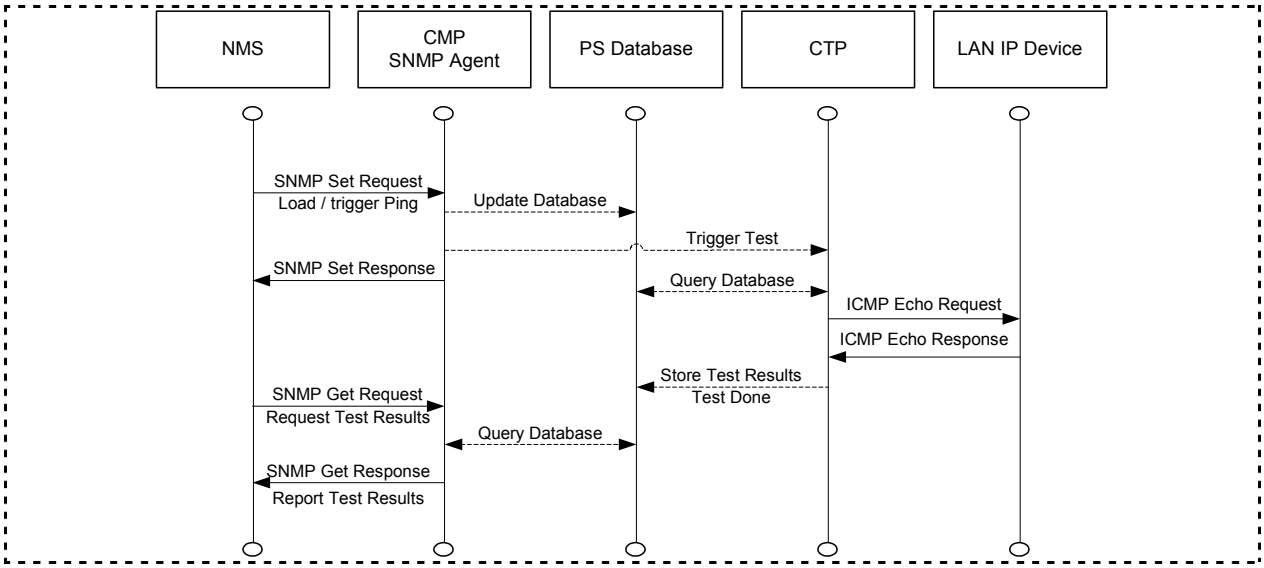

Figure 12-2 Ping Tool Process Sequence Diagram

12.3 PS Operation

The IPCable2Home Management Portal (CMP) provides access to the PS Database via the PS WAN-Man interface, as described in Section 6. The message sequence for a typical PS Database access operation from the PS WAN-Man interface is described below.

12.3.1 PS Database Access

Configuration and management parameters stored in the PS Database are accessed by the NMS via SNMP MIBs. Parameters are retrieved using SNMP Get-Request, Get-Next-Request, and Get-Bulk messages issued by the NMS with the PS WAN-Man address as the destination address. Parameters can be modified and actions (such as the Connection Speed and Ping tools), executed by the NMS issuing SNMP Set-Request messages with the appropriate parameters, to the PS WAN-Man address.

Figure 12-3 describes the management message sequences for a typical PS Database access from the PS WAN-Man interface. The following message sequences assumes that a secure SNMPv3 link has been established:

1. The NMS reads data from the PS database using the SNMP "GET Request". The request lists the specific objects the NMS wants from the database.

2. The CMP SNMP Agent queries the PS Database for the specified parameters.

3. The CMP SNMP reports the data to the NMS with the SNMP "GET Response".

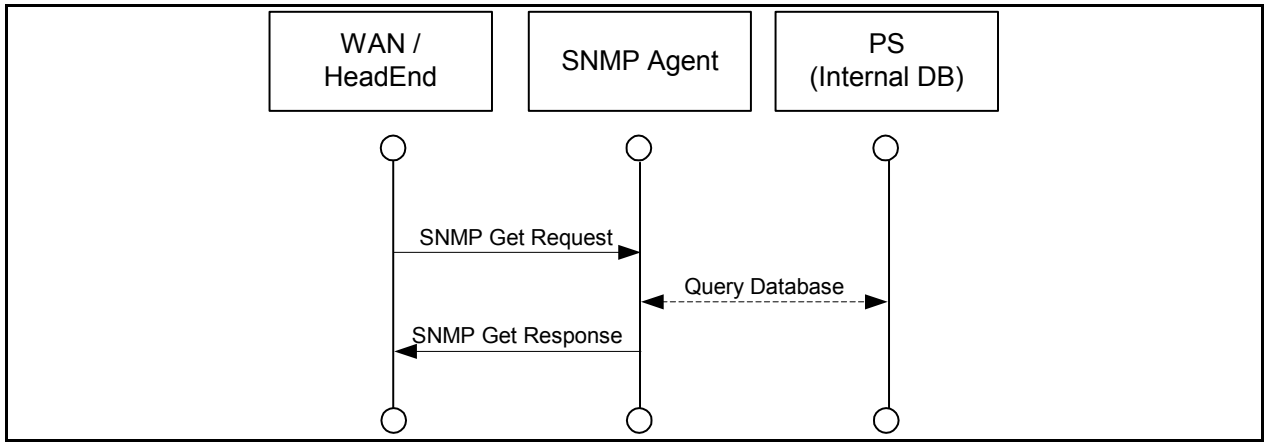

Figure 12-3 PS Database Access from the PS WAN-Man Interface Sequence Diagram

12.3.2 Reconfiguration

12.3.2.1 PS Software Download

Figure 12-4 illustrates a software/firmware download process for a PS in SNMP Provisioning Mode, which is triggered by the NMS. The PS is instructed where to obtain the new software code file. Once download of the code file is complete, the PS will test the image for any corruption that may have occurred during the download. Authentication is performed to verify that the code file can be trusted. Following this step, a system reboot is performed.

Following the reboot, the PS resumes operation on the new software image. The PS may need to be reconfigured after the software upgrade, and the WAN interfaces may need to be provisioned again (not shown). If the PS does not accept the new software image, it will revert back to the prior (backup) software version and report the results to the NMS.

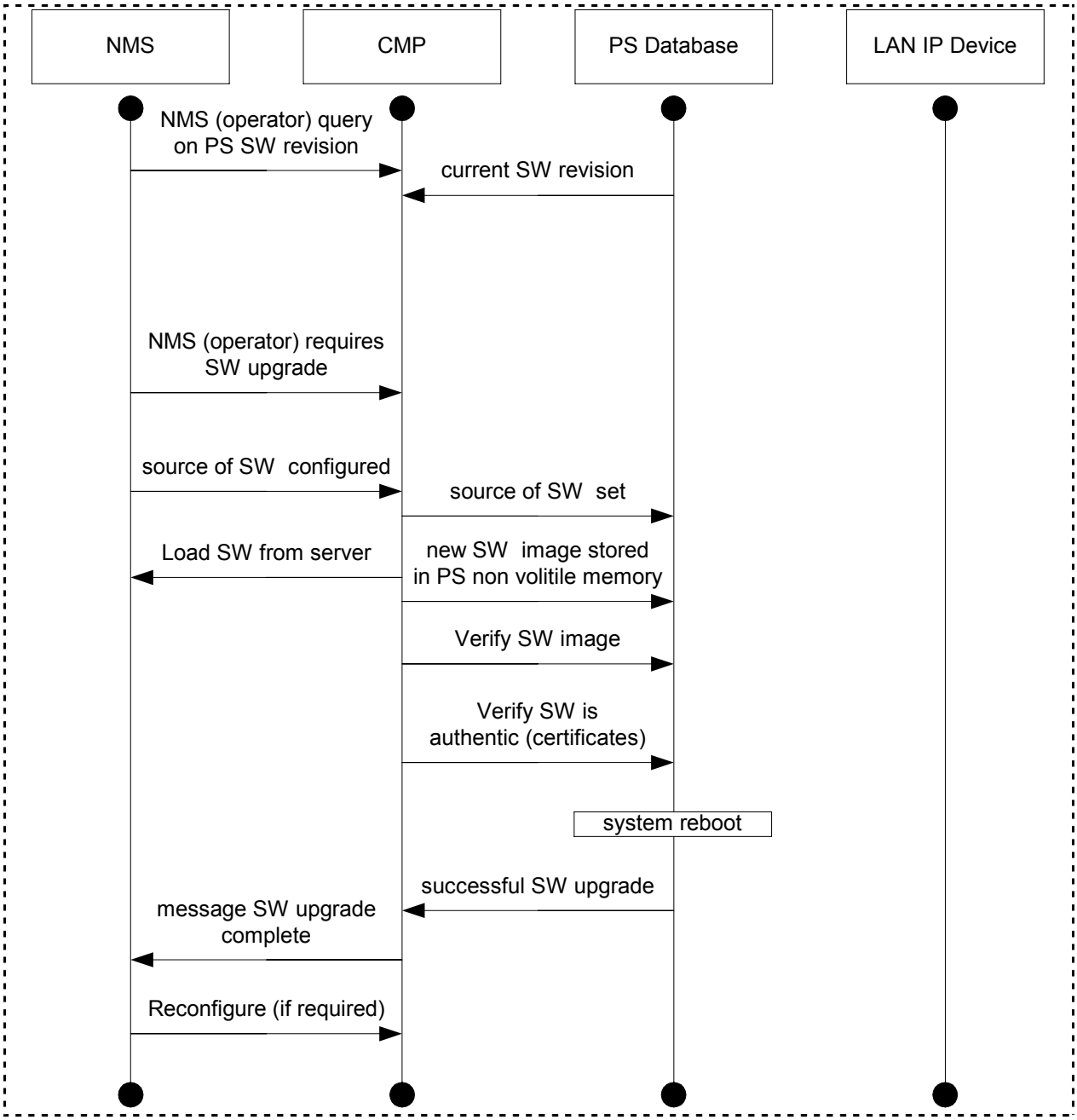

Figure 12-4 PS Software Download Sequence Diagram

12.3.2.2 PS Configuration File Download

Figure 12-5 illustrates a reconfiguration of a PS in SNMP Provisioning Mode, via config file download, and is triggered by the NMS. The configuration file is given to the PS by writing the fileserver and filename into the PS, and triggering the PS to download the file. Once the configuration file is loaded, the commands within it are interpreted. If any of the commands are not understood, or are not applicable, they are skipped and an event is generated. When the PS has completed processing the config file, it will record the number of TLV tuples processed and skipped in the appropriate MIB objects.

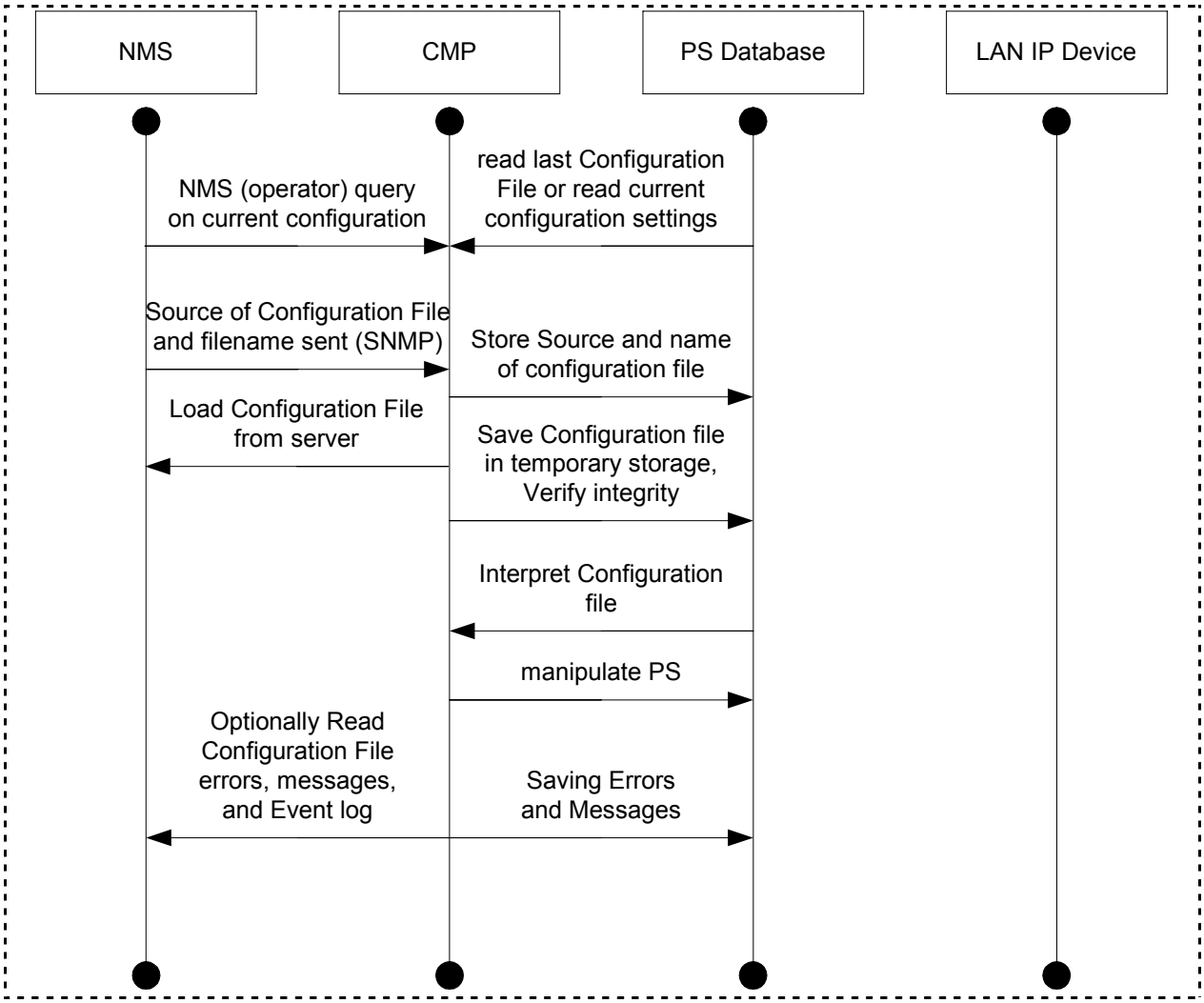

Figure 12-5 PS Reconfiguration (Configuration File Download) Sequence Diagram

12.4 MIB Access

12.4.1 VACM Configuration

IPCable2Home specifies operator control of the IPCable2Home management domain. An example of the configuration of VACM parameters is shown below.

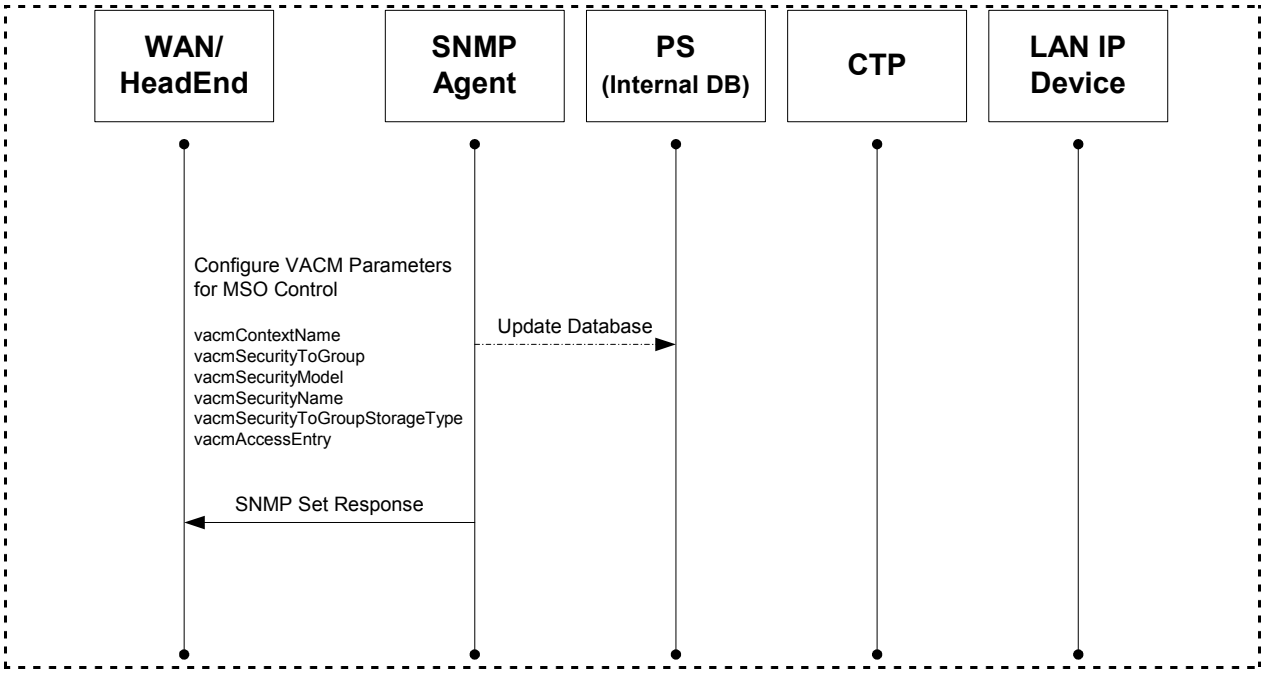

Figure 12-6 PS Configuration (VACM Parameters) Sequence

12.4.2 Management Event Messaging Configuration

12.4.2.1 CMP Event Notification Operation

IPCable2Home events are reported through local event logging, SNMP TRAP, SNMP INFORM messages, and SYSLOG. The event notification mechanism can be set or modified by the NMS, by issuing an SNMP Set-Request message to the PS WAN-Man address.

Figure 12-7 illustrates configuring the PS database to store events in local log files. Local log events are of two types: local non-volatile and local volatile. The NMS will read the content of the local log and write that content to the Headend event logging system. A PS reboot causes only the volatile events to be cleared from the PS database. Nonvolatile events persist across reboots.

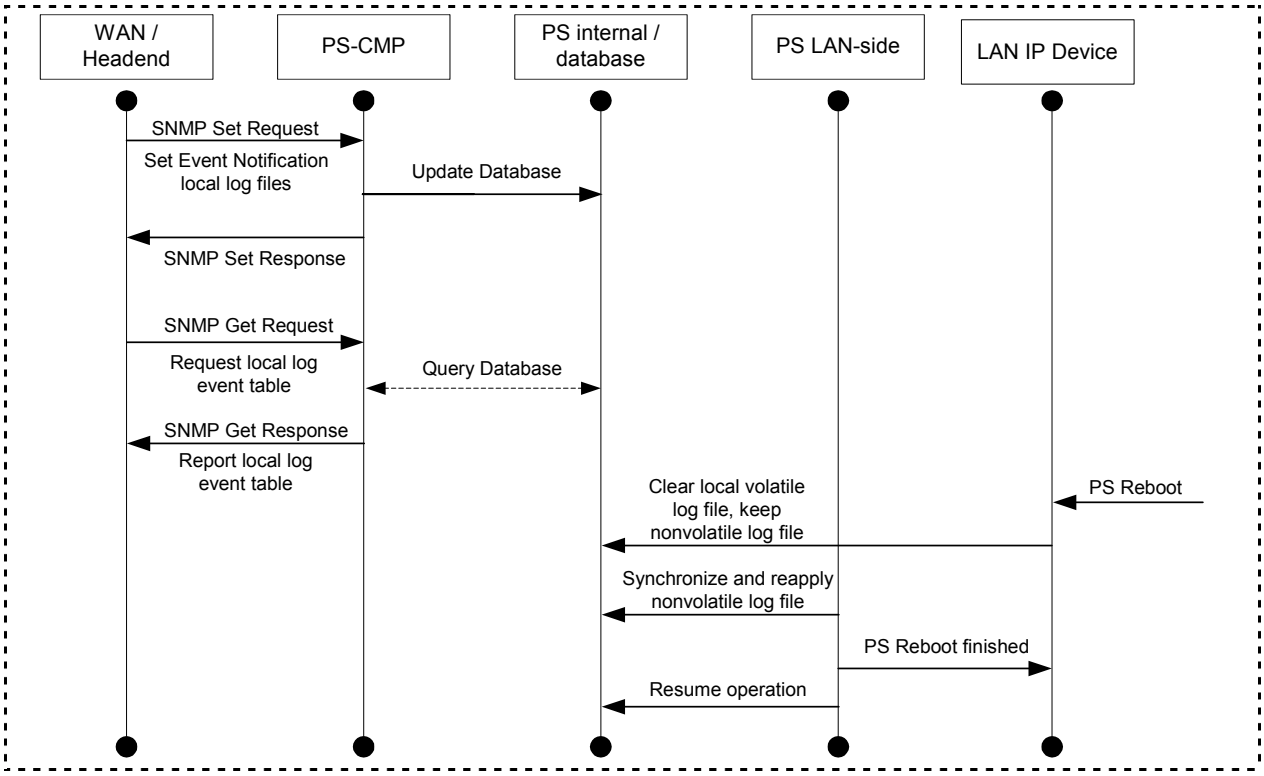

Figure 12-7 PS Configuration (Event Control) Sequence

Figure 12-8 illustrates the download of a configuration file for a PS in SNMP Provisioning Mode. This process is triggered via an SNMP Set Request. The PS must verify this file before accepting it. In the example, a TLV error exists and is reported. Since the event notification is set to the SNMP TRAP mode, the address of the TRAP server is retrieved from the PS database and the event is sent to that TRAP server.

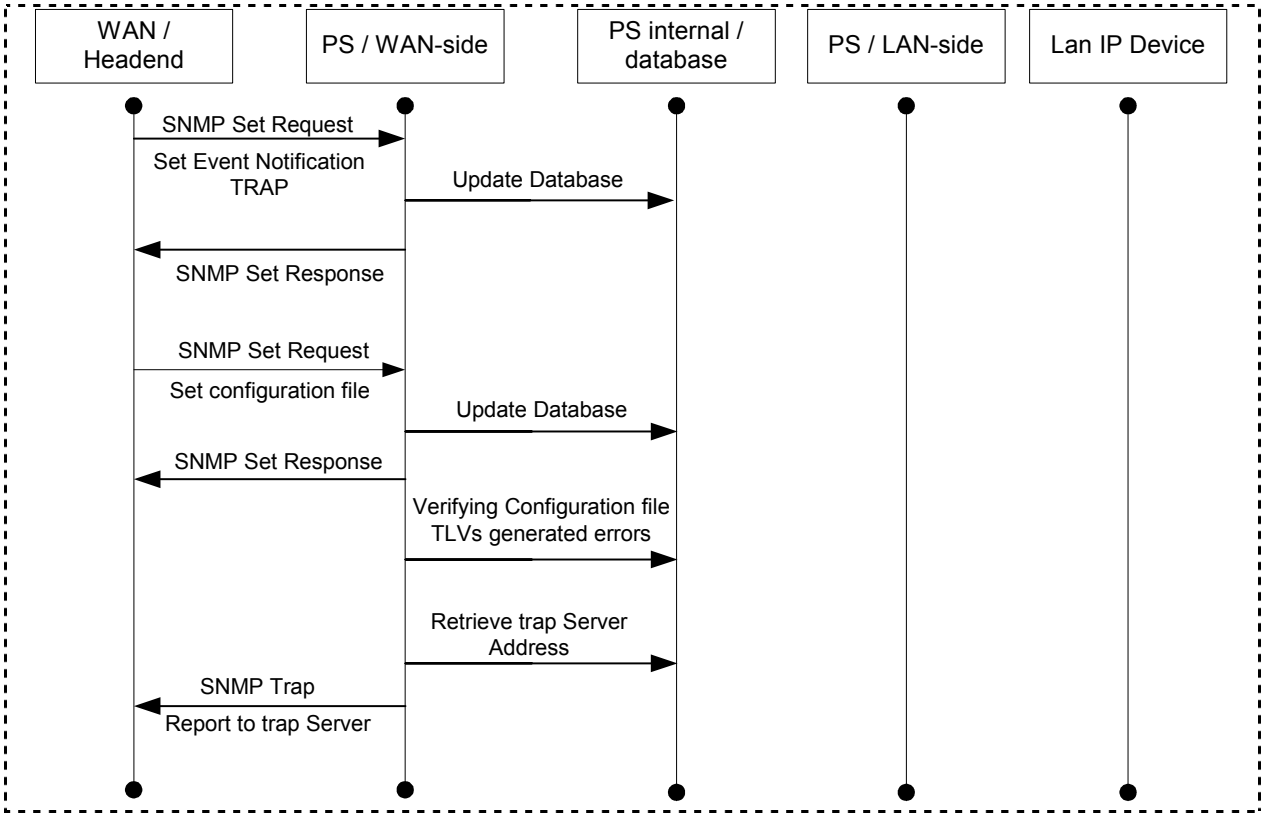

Figure 12-8 PS Configuration File Download (with Invalid TLVs) Sequence

Figure 12-9 illustrates the process of a LAN IP Device trying to obtain an IP address from the local DHCP server (CDS). The CDS function checks the PS database for an available IP address. In this case, the CDS detects that no IP address is available from the address pool, and generates an event to SYSLOG.

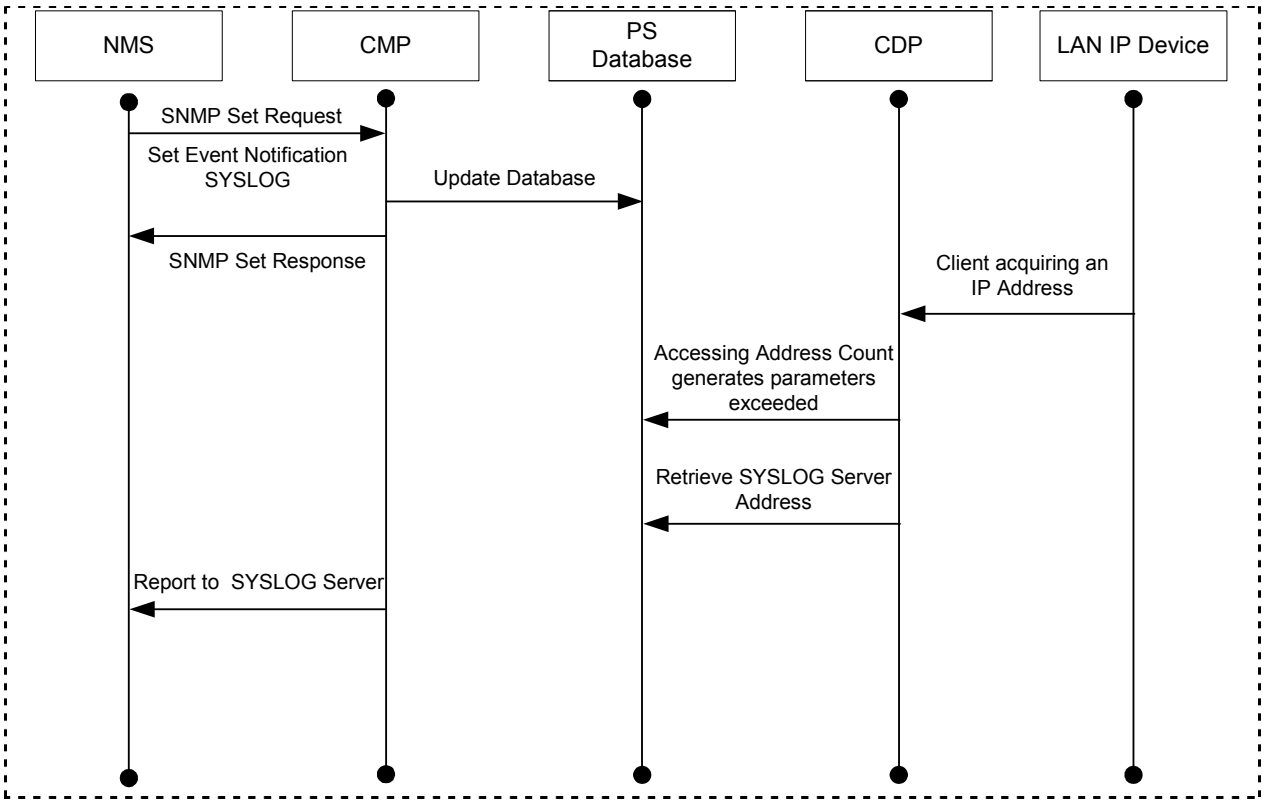

Figure 12-9 Address Acquisition (Request Exceeds Provisioned Count) Sequence

12.4.2.2 Example CMP Event Throttling and Limiting Operation

This Recommendation provides an event throttling mechanism via the CMP functionality of the PS. Event throttling and limiting is very flexible and can include cases in which all events are reported, and cases in which no events are reported to the NMS. Refer to Section 6.3.3.2.4.8 for a description of the CMP Event Throttling and Limiting mechanism.

Figure 12-10 illustrates configuring the PS database to return events via the SNMP INFORM method. Initially, several INFORM messages are written to the local log file and delivered to the NMS. The event throttling mechanism sets the limit of the number of events that can be sent to the NMS within a given time frame. When that limit is reached, the PS will stop sending INFORM messages to the NMS. In order to restart the event notification, the NMS SHOULD re-enable the event reporting.

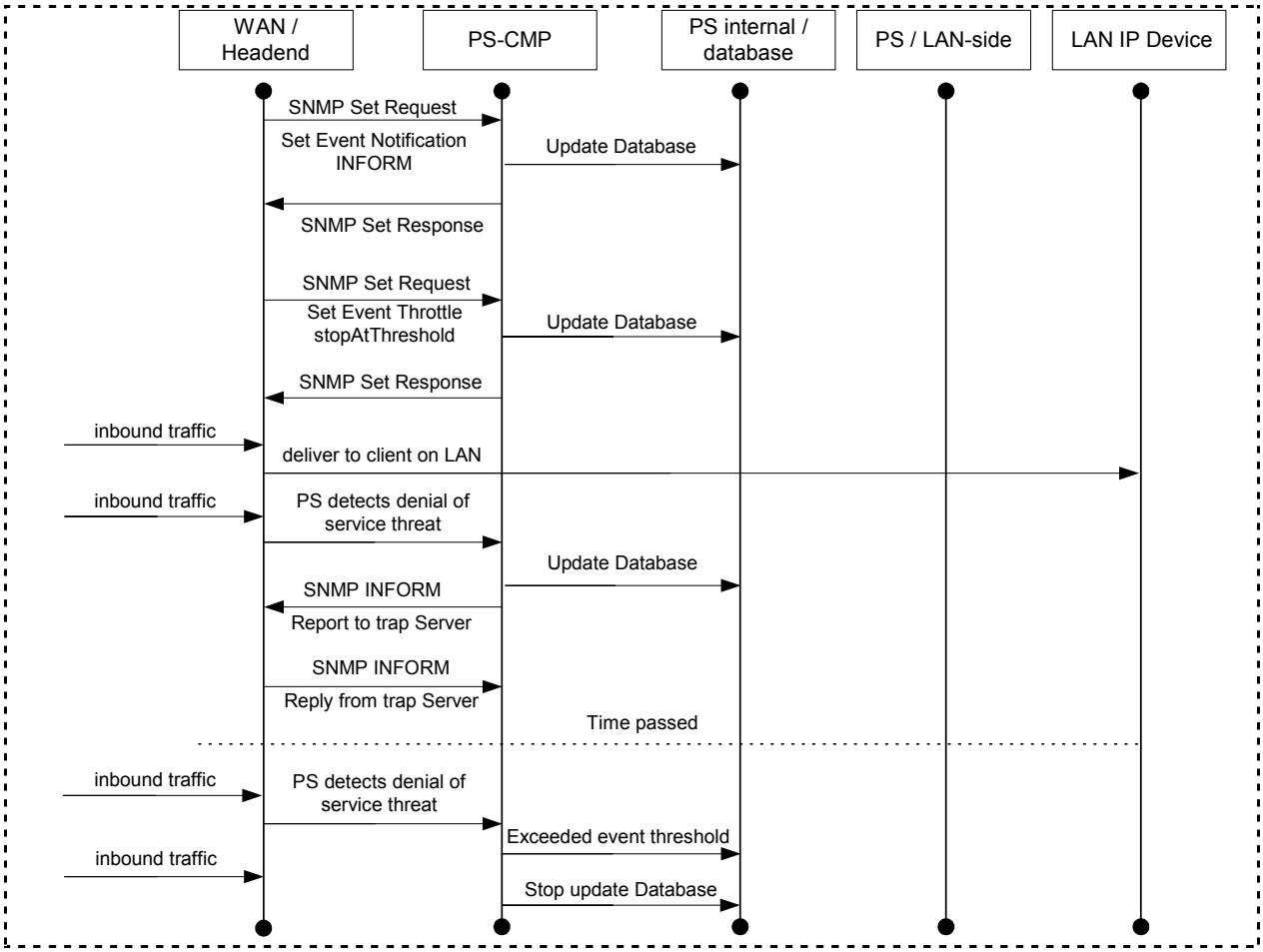

Figure 12-10 CMP Event Throttling and Limiting Operation

13 PROVISIONING PROCESSES

This chapter describes the processes involved when using the Provisioning Tools, described in Section 7, for initial provisioning of LAN IP Device and the PS element Provisioning has the following three tasks:

1. Acquiring network addresses
2. Acquiring server information
3. Secure download and processing of the PS Configuration File

Provisioning processes are described in this section for each of the following relevant cases:

- PS WAN-Man - Provisioning of the PS WAN based management functionality
- PS WAN Data - Provisioning of PS WAN-Data IP addresses to be used for creating CAT Mappings to LAN IP Devices in the LAN-Trans address realm
- LAN IP Device in the LAN-Trans Realm - Provisioning of a LAN IP Device with a translated IP address
- LAN IP Device in the LAN-Pass Realm - Provisioning of a LAN IP Device with an IP address that is passed through to the WAN

Provisioning of the cable modem element of an embedded PS is separate and distinct from IPCable2Home provisioning, and is out of scope for this Recommendation. The reader is referred to CableModem specifications for descriptions of cable modem provisioning.

The functional elements with which the Portal Services element interacts during the provisioning processes listed above are identified in Figure 13-1. The Key Distribution Center (KDC) functional element is shown with a broken outline, since it is used in SNMP Provisioning Mode, but not in DHCP Provisioning Mode. The other functional elements are used in both provisioning modes.

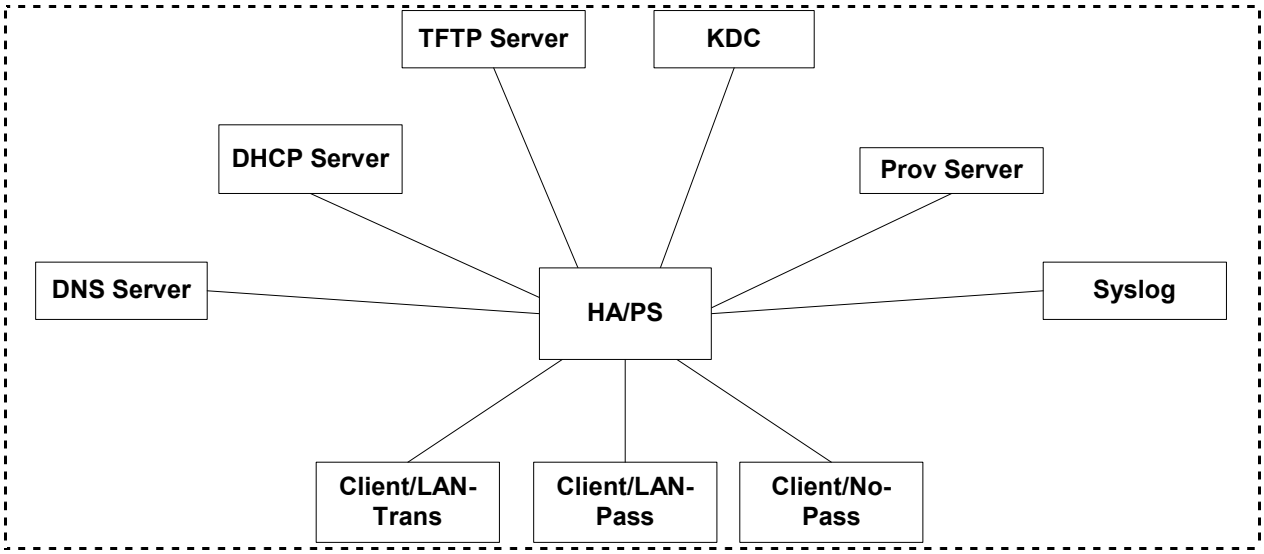

Figure 13-1 IPCable2Home Provisioning Functional Elements

The Trivial File Transfer Protocol (TFTP) server or the HyperText Transfer Protocol (HTTP) server provides access to the PS Configuration File for the PS and follows rules described in [RFC 1350]. The Time of Day (ToD) server provides the means for the PS to acquire the current time in UTC format as described in[[RFC 868]. The Dynamic Host Configuration Protocol (DHCP) server provides the PS with private and/or global IP addresses following [RFC 2131], as well as providing other information via DHCP options in accordance with [RFC 2132]. The Network Management System (NMS) complies with the Simple Network Management Protocol (SNMP) versions SNMPv1, SNMPv2, and SNMPv3 as described in [RFC 2576]. The Key Distribution Center (KDC) manages authorization and encryption keys for establishing trust between networked elements, and implements rules defined in [RFC 1949]. The System Log (SYSLOG) server handles event messages generated by the PS and by LAN IP Devices in the home. The PS implements clients for these cable data network-based servers, and uses these client functions during the provisioning processes described in this section to accomplish the tasks listed at the beginning of this section.

13.1 Provisioning Modes

Section 5.5 and Section 7.2.1 introduce two valid provisioning modes supported by the Portal Services element: DHCP Provisioning Mode and SNMP Provisioning Mode. The PS operates in a third mode, Dormant CableHome Mode, if it is not configured to operate in either of the two valid provisioning modes. In this section the two valid provisioning modes are presented in more detail. Figure 13-2 illustrates a possible event flow for the two provisioning modes and the Dormant CableHome Mode. The key point of Figure 13-2 is the switch used by the PS to determine the mode in which it is to operate.

The PS operates in DHCP Provisioning Mode (DHCP Mode) if the DHCP server in the cable network provides a valid IP address for the TFTP or HTTP server in the DHCP message 'siaddr' field, provides a valid file name for the PS Configuration File in the DHCP message 'file' field, and does NOT provide DHCP option 177 sub-options 3, 6, and 51 to the PS CDC, during the DHCPACK phase of the initialization process. DHCP Provisioning Mode is intended to enable the PS to operate on a DOCSIS 1.0 or a DOCSIS 1.1 infrastructure, with little or no changes to the DOCSIS network.

SNMP Provisioning Mode in the PS is triggered when the DHCP server in the cable network does NOT provide values for 'siaddr' and 'file', and when the cable network DHCP server DOES send DHCP option 177 sub-options 3, 6, and 51. SNMP Provisioning Mode is intended to enable the PS to take advantage of advanced features of a PacketCable infrastructure.

The PS defaults to Dormant CableHome Mode if it receives none of the fields or sub-options defined as triggers for DHCP Provisioning Mode and for SNMP Provisioning Mode, or if it receives an invalid combination of the fields and sub-options.

The PS defaults to Dormant CableHome Mode if it receives none of the fields or sub-options defined as triggers for DHCP Provisioning Mode and for SNMP Provisioning Mode, or if it receives an invalid combination of the fields and sub-options.

Not all error conditions are shown in Figure 13-2. Refer to Section 7.2.2 for a description of PS behavior in the event of incorrect Provisioning Mode decision criteria.

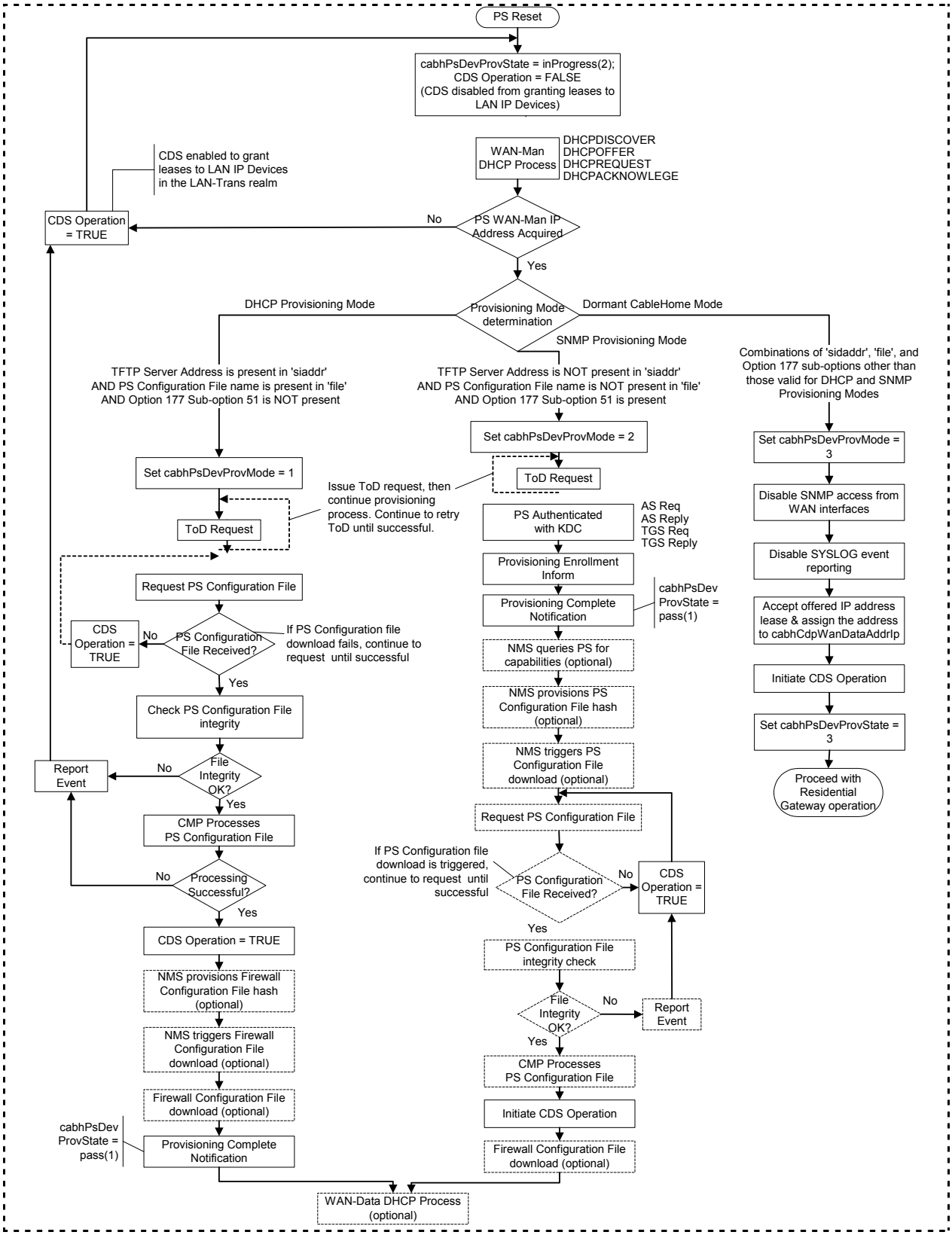

Figure 13-2 IPCable2Home Provisioning Modes

13.2 Process for Provisioning the PS for Management: DHCP Provisioning Mode

The PS requests, from the Headend provisioning system, an IP address to be used for the exchange of management

messages between the NMS and the PS. The PS parses the DHCP message returned in the DHCP OFFER and makes a determination about the provisioning mode in which it is to operate (ref. Section 7.3.3.2.4). Section 7.3.3.2.3.2 describes three WAN Address Modes supported for the acquisition of IP addresses by the PS from the DHCP server in the cable network.

If the PS makes the determination that it is to operate in DHCP Provisioning Mode, it will use the PS Configuration File information passed in the DHCP message as a trigger to download the PS Configuration File, as described in Section 7.3. PS Configuration File download is a requirement for the PS operating in DHCP Provisioning Mode, but is optional for the PS operating in SNMP Provisioning Mode.

In DHCP Provisioning Mode, the PS (CMP) defaults to using NmAccess mode for management message exchange with the NMS, but the NMS can optionally configure the CMP for Coexistence Mode. These management messaging modes are described in Section 6.3.3.

Figure 13-3 and Figure 13-1 describe the sequence of messages needed to initialize a PS operating in DHCP Provisioning Mode. The process for provisioning for management of a PS operating in DHCP Provisioning Mode is the same for the PS embedded with a DOCSIS cable modem, as it is for the stand-alone PS. The provisioning for the Embedded PS MUST NOT occur before the cable modem provisioning process. The stand-alone PS management provisioning SHOULD occur immediately after power-up/reset.

The optional process of downloading a Firewall Configuration File is shown with shading in Figure 13-3.

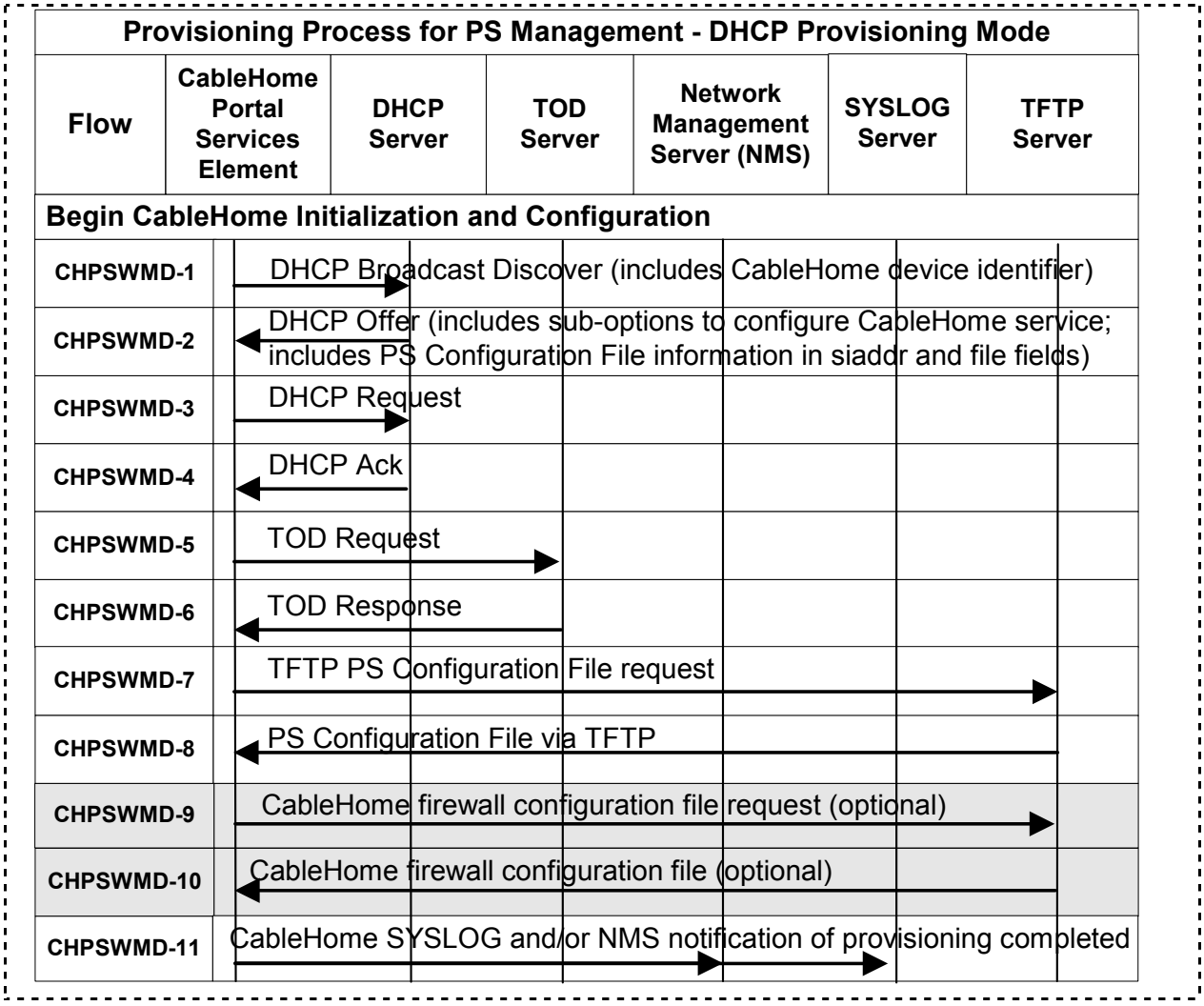

Figure 13-3 Provisioning Process for PS Management - DHCP Provisioning Mode

Table 13-1 describes the individual messages CHPSWMD-1 - CHPSWMD-12 shown in Figure 13-3.

Table 13-1 Flow Descriptions for PS WAN-Man Provisioning Process for DHCP Provisioning Mode

Flow Step	PS WAN-Man Provisioning: DHCP Provisioning Mode	Normal Sequence	Failure Sequence
CHPSWMD-1	DHCP Broadcast Discover The CDP (CDC) sends a broadcast DHCP DISCOVER message to acquire the WAN-Man IP address as described in Section 7.3.3.2.4. The DHCP DISCOVER broadcast by the CDP (CDC) includes mandatory options listed in Table 7-10 CDC DHCP Options in DISCOVER and REQYEST Messages. The PS sets cabhPsDevProvState to status 'inProgress' (2) when the CDC sends a broadcast DHCP DISCOVER.	Begin provisioning sequence.	If unsuccessful per DHCP protocol report an error and continue to retry DHCP Broadcast Discover until successful (return to step CHPSWMD-1). If unsuccessful on the first attempt to acquire a WAN-Man IP address, the PS initiates operation of the CDS as specified in Section 7.3.3.2.4.
CHPSWMD-2	DHCP OFFER	CHPSWMD-2 MUST occur after CHPSWMD-1 completion.	If failure per DHCP protocol return to CHPSWMD-1 and report an error.
CHPSWMD-3	DHCP REQUEST The CDP MUST send the appropriate DHCP server a DHCP REQUEST message to accept the DHCP OFFER.	CHPSWMD-3 MUST occur after CHPSWMD-2 completion.	If failure per DHCP protocol return to CHPSWMD-1 and report an error.
CHPSWMD-4	DHCP ACK The DHCP server sends the CDP a DHCP ACK message which contains the IPv4 address of the PS. The PS modifies cabhPsDevProvMode based on information received in the DHCP ACK (ref.: Section 7.3.3.2.4). The PS stores the Time of Day server address in cabhPsDevTimeServerAddr. The PS modifies cabhPsDevProvMode based on information received in the DHCP ACK (ref.: Section 7.3.3.2.4).	CHPSWMD-4 MUST occur after CHPSWMD-3 completion.	If failure per DHCP protocol return to CHPSWMD-1 and report an error.
CHPSWMD-5	Time of Day (TOD) Request per [[RFC 868] The PS issues a ToD Request to the Time Server identified in Option 4 of the DHCP ACK message.	CHPSWMD-5 MUST occur after CHPSWMD-4 completion.	Continue with CHPSWMD-6.
CHPSWMD-6	TOD Response The ToD server is expected to reply with the current time in UTC format.	CHPSWMD-6 MUST occur after CHPSWMD-5 completion.	Continue with CHPSWMD-7, report an error, and return to CHPSWMD-5 (continue to retry TOD until successful).
CHPSWMD-7	TFTP Request The PS operating in DHCP Provisioning Mode sends the TFTP Server a TFTP Get Request to request the specified configuration data file as described in Section 7.4.4.	CHPSWMD-7 MUST occur after CHPSWMD-5 completion. CHPSWMD-7 MAY occur before CHPSWMD-6 completion.	Continue to CHPSWMD-8.

CHPSWMD-8	TFTP server sends PS Configuration File After the PS Configuration File is received, the hash is checked. Refer to Section 7.4.4.1. The PS Configuration File is then processed. Refer to Section 7.4.4 for PS Configuration File contents. Optionally, the IP Address of the firewall Configuration File TFTP server, the firewall Configuration File filename and the hash of the firewall Configuration File are included in the PS Configuration File if there is a firewall Configuration File to be loaded, and this is the method selected to specify it.	CHPSWM-8 MUST occur after CHPSWM-7 completion.	If the TFTP download fails, report an error and return to CHPSWMD- 7 (continue to retry PS Configuration File download). If processing of the PS Configuration File produces an error, continue with CHPSWMD-9 and report the error as an event. If the Provisioning Timer expires before PS Configuration File is successfully downloaded, the PS MUST report an error and return to CHPSWMD- 1.
CHPSWMD-9	TFTP Request - Firewall Configuration File (optional) If the PS receives Firewall Configuration File information (Firewall TFTP server and Firewall Configuration File name) in the PS Configuration File, the PS sends the Firewall Configuration TFTP Server a TFTP Get Request to request a Firewall Configuration File (see Section 11.6.4.2). If the PS does not receive Firewall Configuration File information in the PS Configuration file, the PS provisioning process (DHCP Provisioning Mode) MUST skip steps CHPSWMD-9 and CHPSMWD-10 and continue with step CHPSWMD-11.	If CHPSWMD-9 occurs, it MUST occur after CHPSWMD-8 completion.	If TFTP fails, continue with PS operation but report an error and continue to retry CHPSWMD-9.
CHPSWMD-10	TFTP server sends firewall configuration file (optional) If step CHPSWMD-9 occurs, the TFTP Server sends the PS a TFTP Response containing the requested file. After the firewall configuration file is received the hash of the configuration file is calculated and compared to the value received in the PS Configuration File. The file is then processed. Refer to Section 11.6.4.	CHPSWMD-10 MUST occur after CHPSWMD-9 completion.	If the TFTP fails, continue with PS operation but report an error and continue to retry CHPSWMD-9. If processing of the firewall configuration file produces an error, continue and report the error as an event.

CHPSWMD-11	Provisioning Complete If requested by the provisioning system the PS is required to inform the provisioning system of the status of PS provisioning. The provisioning system could request the PS to send a SYSLOG message or an SNMP trap, or both. If the PS successfully completes all required steps from CHPSWMD-1 through CHPSWMD-10 and the PS received a SYSLOG server address in the DHCP OFFER, the PS MUST send a provisioning complete message to the SYSLOG server with provisioning state set to PASS. If the PS successfully completes all required provisioning steps from CHPSWMD-1 through CHPSWMD-10 and the PS received valid parameters the Notification Receiver, the PS MUST send a provisioning complete notification (cabhPsDevInitTrap) with appropriate parameters to the Notification Receiver. If the PS provisioning timer expires before all required steps from CHPSWMD-1 through CHPSWMD-10 are completed and the PS received a SYSLOG server address in the DHCP OFFER, the PS MUST send a provisioning complete message to the SYSLOG server with provisioning state set to FAIL. If the PS provisioning timer expires before all required steps from CHPSWMD-1 through CHPSWMD-10 are completed and the PS received valid parameters for the Notification Receiver, the PS MUST send a provisioning failed notification (cabhPsDevInitTrap) to the Notification Receiver. The PS MUST update the value of cabhPsDevProvState with status 'pass' (1) when provisioning flow steps CHPSWMD-1 through CHPSWMD-11 complete successfully. The PS MUST update the value of cabhPsDevProvState with status 'fail' (3) and report an event indicating provisioning process failure if the PS Provisioning Timer expires before the value of cabhPsDevProvState is updated with status 'pass'.	CHPSWMD-11 MUST occur after CHPSWMD-10 completion.	If the SNMP trap fails, the provisioning server may not know the provisioning process has completed unless it polls the cabhPsProvState object.

13.3 Process for Provisioning the PS for Management: DHCP Provisioning Mode with HTTP/TLS

The PS requests from the Headend provisioning system, an IP address to be used for the exchange of management messages between the NMS and the PS. The PS parses the DHCP message returned in the DHCP OFFER and makes a determination about the provisioning mode in which it is to operate (ref. Section 7.3.3.2.4). Section 7.3.3.2.3.2 describes three WAN Address Modes supported for the acquisition of IP addresses by the PS from the DHCP server in the cable network.

If the PS makes the determination that it is to operate in DHCP Provisioning Mode, it will use the PS Configuration File information passed in the DHCP message, as a trigger to download the PS Configuration File. If DHCP option code 72 is present in the DHCP ACK message, and if its contents match the IP address in the siaddr field, the download will occur, using HTTP over TLS, as specified in Section 11.9.

In DHCP Provisioning Mode, the PS (CMP) defaults to using NmAccessTable mode for management message exchange with the NMS, but the NMS can optionally configure the CMP for Coexistence Mode. These management messaging modes are described in Section 6.3.3.

Figure 13-4 and Table 13-2 describe the sequence of messages needed to initialize a PS operating in DHCP Provisioning Mode with HTTP/TLS. The process for provisioning and management of the PS operating in DHCP Provisioning Mode is the same for the PS embedded with a DOCSIS cable modem as it is for the stand-alone PS. The provisioning for the Embedded PS MUST NOT occur before the cable modem provisioning process. The stand-alone PS management provisioning SHOULD occur immediately after power-up/reset.

The optional process of downloading a Firewall Configuration File is shown with shading in Figure 13-4.

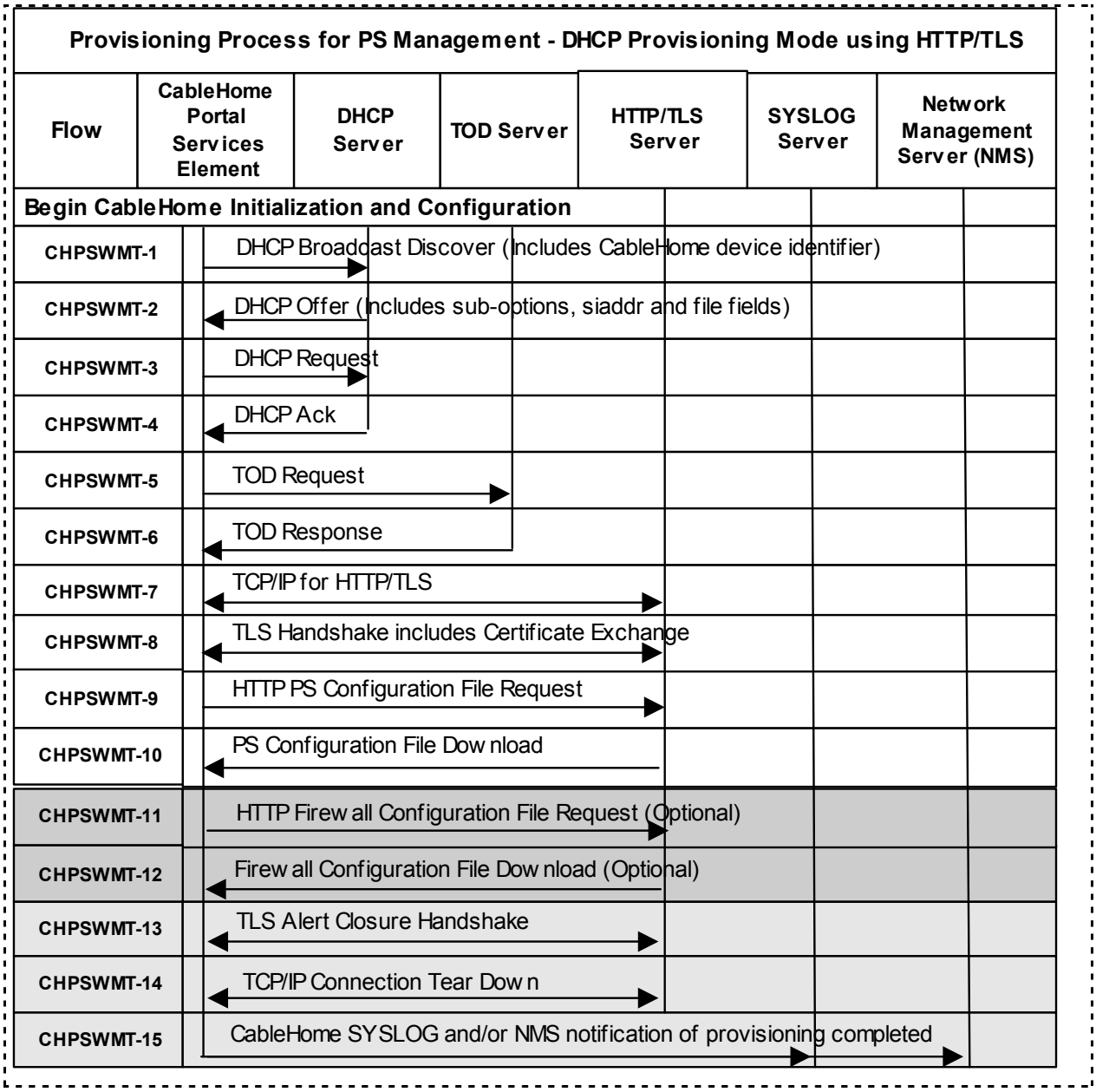

Figure 13-4 Provisioning Process DHCP Provisioning Mode using HTTP/TLS

Table 13-2 describes the individual messages CHPSWMT-1 - CHPSWMT-12 shown in Figure 13-3. Refer to Section 11.9 PS Configuration File Security in DHCP Provisioning Mode for more information.

Table 13-2 Flow Descriptions for DHCP Provisioning Mode using HTTP/TLS

Flow Step	PS WAN-Man Provisioning: DHCP Provisioning Mode	Normal Sequence	Failure Sequence
CHPSWMT-1	DHCP Broadcast Discover The CDP (CDC) sends a broadcast DHCP DISCOVER message to acquire the WAN-Man IP address as described in Section 7.3.3.2.4. The DHCP DISCOVER broadcast by the CDP (CDC) includes mandatory options listed in Table 7-10, CDC DHCP Options in DISCOVER and REQUEST messages. The PS sets cabhPsDevProvState to status 'inProgress' (2) when the CDC sends a broadcast DHCP DISCOVER.	Begin provisioning sequence.	If unsuccessful per DHCP protocol report an error and continue to retry DHCP Broadcast Discover until successful (return to step CHPSWMT-1). If unsuccessful on the first attempt to acquire a WAN-Man IP address, the PS initiates operation of the CDS as specified in Section 7.3.3.2.4.
CHPSWMT-2	DHCP OFFER	CHPSWMT-2 MUST occur after CHPSWMT-1 completion.	If failure per DHCP protocol return to CHPSWMT-1 and report an error.
CHPSWMT-3	DHCP REQUEST The CDP sends the appropriate DHCP server a DHCP REQUEST message to accept the DHCP OFFER.	CHPSWMT-3 MUST occur after CHPSWMT-2 completion.	If failure per DHCP protocol return to CHPSWMT-1 and report an error.
CHPSWMT-4	DHCP ACK The DHCP server sends the CDP a DHCP ACK message which contains the IPv4 address of the PS. The PS stores the Time of Day server address in cabhPsDevTimeServerAddr. If the IP address in the siaddr field of the DHCP ACK matches the first IP address in option 72, the PS initiates a TLS session and downloads the configuration file from the HTTP server. The PS modifies cabhPsDevProvMode based on information received in the DHCP ACK. Refer to Table 11-9 PS Configuration File Security in DHCP Provisioning Mode.	CHPSWMT-4 MUST occur after CHPSWMT-3 completion.	If failure per DHCP protocol return to CHPSWMT-1 and report an error.
CHPSWMT-5	Time of Day (TOD) Request per [[RFC 868] The PS synchronizes its time with the time server selected from DHCP Option 4 (Time Server Option) in the DHCP ACK. Refer to Section 7.5.4 Time of Day Client Function Requirements.	CHPSWMT-5 MUST occur after CHPSWMT-4 completion.	Continue with CHPSWMT-6.
CHPSWMT-6	TOD Response The ToD server is expected to reply with the current time in UTC format.	CHPSWMT-6 MUST occur after CHPSWMT-5 completion.	Report an error, and return to CHPSWMT-5 (continue to retry TOD until successful).

CHPSWMT-7	TCP/IP Setup The PS operating in DHCP Provisioning Mode establishes. a TCP/IP session to exchange HTTP messages with the HTTP server in the cable operator's provisioning system.	CHPSWMT-7 MUST occur after CHPSWMT-5 completion. CHPSWMT-7 MAY occur before CHPSWMT-6 completion.	If failure per TCP/IP retry per the specification. If retries all fail return to CHPSWMT-1 and report an error.
CHPSWMT-8	TLS Handshake The PS operating in DHCP Provisioning Mode establishes a TLS session with the HTTPS server.	CHPSWMT-8 MUST occur after CHPSWMT-7 completion.	If failure for TLS retry per the specification. If retries all fail return to CHPSWMT-1 and report an error.
CHPSWMT-9	HTTP Configuration File Request The PS operating in DHCP Provisioning Mode requests the configuration file from the HTTP server.	CHPSWMT-9 MUST occur after CHPSWMT-8 completion.	If failure for HTTP retry per the specification. If retries all fail return to CHPSWMT-1 and report an error.
CHPSWMT-10	HTTPS server sends PS Configuration File The PS Configuration File is processed. Refer to Section 7.4.4 for PS Configuration File contents. Optionally, the IP Address of the firewall Configuration File HTTP server and the firewall Configuration File filename of the firewall Configuration File are included in the PS Configuration File.	CHPSWMT-10 MUST occur after CHPSWMT-9 completion.	If the HTTP download fails, report an error and return to CHPSWMT- 9 (continue to retry PS Config File download). If processing of the PS Config File produces an error, continue with CHPSWMT-13 and report the error as an event. If the Provisioning Timer expires before PS Config File is successfully downloaded, the PS MUST report an error and return to CHPSWMT- 1.
CHPSWMT-11	HTTP Request - Firewall Configuration File (Optional) If the PS receives Firewall Configuration File information (Firewall TFTP server and Firewall Configuration File name) in the PS Configuration File, the PS requests the Firewall Configuration File from the HTTP Server. If the PS does not receive Firewall Configuration File information in the PS Configuration file, the PS provisioning process (DHCP Provisioning Mode) MUST skip steps CHPSWMT-11 and CHPSMWT-12 and continue with step CHPSWMT-13.	If CHPSWMT-11 occurs, it MUST occur after CHPSWMT-10 completion.	If HTTP fails, continue with PS operation but report an error and continue to retry CHPSWMT-13.

CHPSWMT-12	HTTP server sends firewall configuration file (Optional) If step CHPSWMD-11 occurs, the HTTP Server sends the PS a HTTP Response containing the requested firewall configuration file.	CHPSWMT-12 MUST occur after CHPSWMT-11 completion.	If the HTTP fails, continue with PS operation but report an error and continue to retry CHPSWMD-11. If processing of the firewall configuration file produces an error, continue and report the error as an event.
CHPSWMT-13	TLS Alert Closure Handshake The PS MUST tear down the TLS session immediately prior to sending the provisioning complete message.	CHPSWMT-13 MUST occur after CHPSWMT-12 completion	Continue to CHPSWMT-14. If failure for HTTP retry per the specification. If retries all fail report an error.
CHPSWMT-14	TCP/IP Tear Down The TCP/IP session between the PS and the HTTP Server is torn down.	CHPSWMT-14 MUST occur after CHPSWMT-13 completion.	If the TCP/IP tear down fails report an error. Continue to 15.
CHPSWMT-15	Provisioning Complete If requested by the provisioning system the PS is required to inform the provisioning system of the status of PS provisioning. The provisioning system could request the PS to send a SYSLOG message or an SNMP trap, or both. If the PS successfully completes all required steps from CHPSWMT-1 through CHPSWMT-14 and the PS received a SYSLOG server address in the DHCP OFFER, the PS MUST send a provisioning complete message to the SYSLOG server with provisioning state set to PASS. If the PS successfully completes all required provisioning steps from CHPSWMT-1 through CHPSWMT-12 and the PS received valid parameters for docsDevNmAccessGroup identifying the Trap Receiver (docsDevNmAccessIP) and configuring the provisioning complete trap (cabhPsDevInitTrap) for 'read only with Traps' (set docsDevNmAccess control to '4'. Refer to [RFC 2669]), the PS MUST send a provisioning complete trap (cabhPsDevInitTrap) with appropriate parameters to the Trap Receiver. If the PS provisioning timer expires before all required steps from CHPSWMD-1 through CHPSWMD-14 are completed and the PS received a SYSLOG server address in the DHCP OFFER, the PS MUST send a provisioning complete message to the SYSLOG server with provisioning state set to FAIL. If the PS provisioning timer expires before all required steps from CHPSWMD-1 through CHPSWMD-14 are completed and the PS received valid parameters for docsDevNmAccessGroup identifying the Trap Receiver (docsDevNmAccessIP) and configuring the provisioning complete trap (cabhPsDevInitTrap) for 'read only with Traps' (set docsDevNmAccess control to '4'. Refer to [RFC 2669].), the PS MUST send a provisioning failed trap (cabhPsDevInitRetryTrap) to	CHPSWMT-15 MUST occur after CHPSWMT-14 completion.	If the SNMP trap fails, the provisioning server may not know the provisioning process has completed unless it polls the cabhPsDevProvState object.

	the Trap receiver. The PS MUST update the value of cabhPsDevProvState with status 'pass' (1) when provisioning flow steps CHPSWMD-1 through CHPSWMD-14 complete successfully. The PS MUST update the value of cabhPsDevProvState with status 'fail' (3) and report an event indicating provisioning process failure if the PS Provisioning Timer expires before the value of cabhPsDevProvState is updated with status 'pass'.		

13.4 Provisioning the PS for Management: SNMP Provisioning Mode

The PS requests a WAN-Man network address from the Headend DHCP server to be used for the exchange of management messages between the PS management functions and the cable network NMS. If the PS determines based on the procedure described in Section 7.3.3.2.4 that it is to operate in SNMP Provisioning Mode, the PS will secure its management messages using SNMPv3, following the authentication procedure described in Section 11.3.2.

The cable network NMS may optionally instruct the PS (CMP) operating in SNMP Provisioning Mode to download a PS Configuration File from the TFTP server. Notification of completion of the provisioning process is provided through the Event Reporting process described in Section 6.3.3.2. The PS will operate without a PS Configuration File if it is not triggered to download the file.

Figure 13-5 illustrates message flows that are to be used to accomplish the provisioning of the PS when it operates in SNMP Provisioning Mode. The provisioning process for the PS WAN-Man interface is the same for the Embedded PS as it is for the Stand-alone PS. The Standalone PS provisioning SHOULD occur immediately after power-up/reset.

The provisioning process for the WAN-Man interface of a PS operating in SNMP Provisioning Mode MUST occur via the sequence depicted in Figure 13-5 and described in detail in Table 13-3. Optional steps are shown with a shaded background in Figure 13-5. These optional steps may be done immediately following step CHPSWMS-13, at a later time, or not at all.

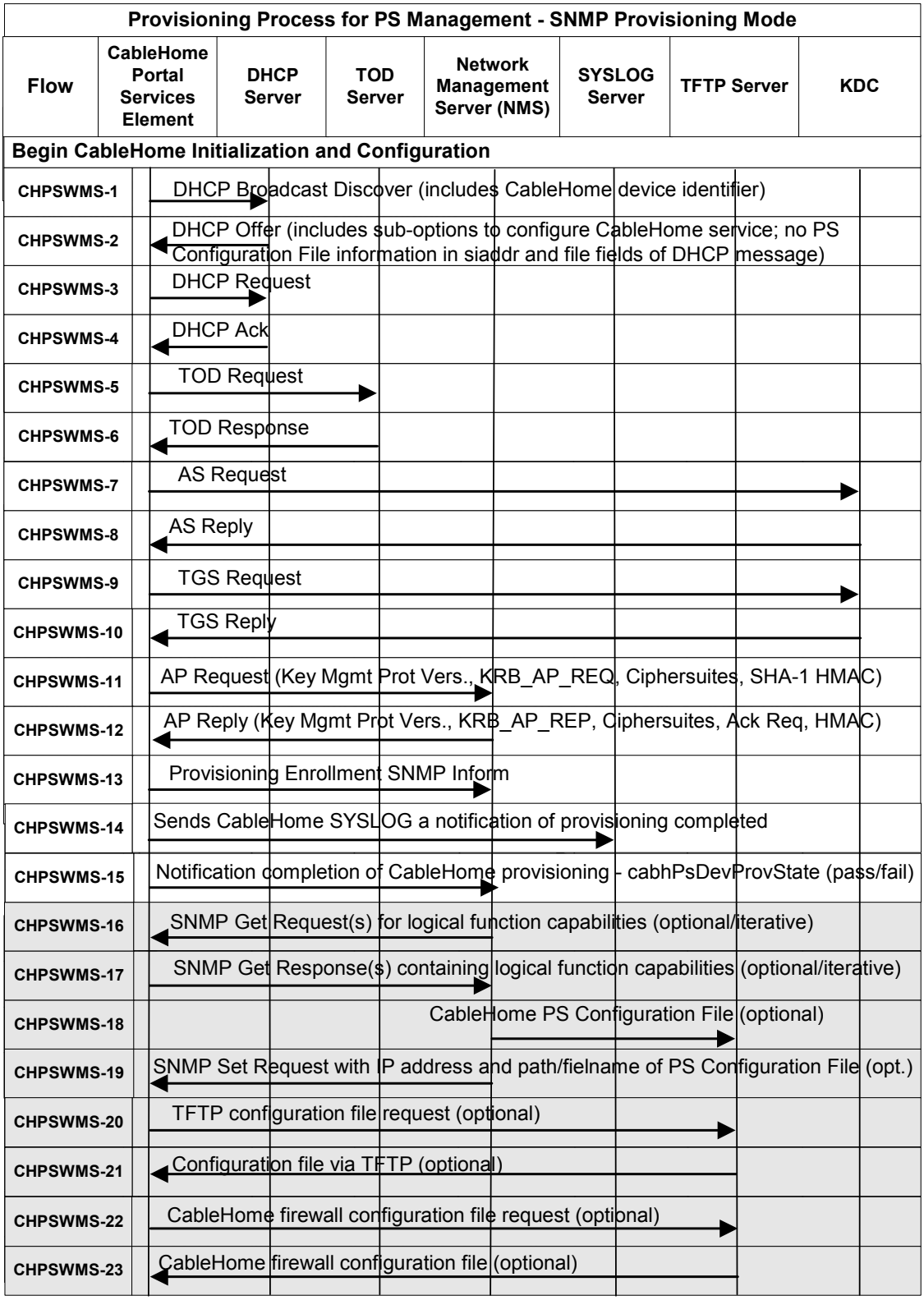

Figure 13-5 Provisioning Process for PS Management - SNMP Provisioning Mode

Table 13-3 describes the individual steps of the provisioning process depicted in Figure 13-5.

Table 13-3 Flow Descriptions for PS WAN-Man Provisioning Process for SNMP Provisioning Mode

Flow Step	PS WAN-Man Provisioning: SNMP Provisioning Mode	Normal Sequence	Failure Sequence
CHPSWMS- 1	DHCP Broadcast Discover The CDP (CDC) broadcasts a DHCP DISCOVER message to acquire the WAN-Man IP address as described in Section 7.3.3.2.4 CDC Requirements. The DHCP DISCOVER broadcast by the CDC includes mandatory options listed in Table 10, CDC DHCP Options in DISCOVER and REQUEST messages.. The PS starts monitoring time elapsed AND sets cabhPsDevProvState to status 'inProgress' (2) when the CDC broadcasts its initial DHCP DISCOVER message.	Begin provisioning sequence.	If failure per DHCP protocol report an error and continue to retry DHCP Broadcast Discover until successful (return to CHPSWMS-1). If the first attempt to acquire an address lease from the cable operator's DHCP server fails, initiate operation of the CDS as specified in Section 7.3.3.2.4 CDC Requirements.
CHPSWMS- 2	DHCP OFFER	CHPSWMS-2 MUST occur after CHPSWMS-1 completion.	If failure per DHCP protocol return to CHPSWMS-1 and report an error.
CHPSWMS- 3	DHCP REQUEST The CDP sends to the appropriate DHCP server a DHCP REQUEST message to accept the DHCP OFFER.	CHPSWMS-3 MUST occur after CHPSWMS-2 completion.	If failure per DHCP protocol return to CHPSWMS-1.
CHPSWMS- 4	DHCP ACK The DHCP server sends the CDC a DHCP ACK message which contains the IPv4 address of the PS WAN-Man Interface and is expected to include the IPCable2Home option code 177 with sub-options 3, 6, & 51 AND no PS configuration file information in the siaddr and file fields of the DHCP message. The PS modifies cabhPsDevProvMode based on information received in the DHCP ACK (ref.: Section 7.2.3.3). The PS stores the Time of Day server address in cabhPsDevTimeServerAddr.	CHPSWMS-4 MUST occur after CHPSWMS-3 completion.	If failure per DHCP protocol return to CHPSWMS-1 and report an error.
CHPSWMS- 5	Time of Day (TOD) Request per [[RFC 868] The PS issues a ToD Request message to the Time Server identified in the DHCP Option 4 of the DHCP ACK message.	CHPSWMS-5 MUST occur after CHPSWMS-4 completion.	Continue with CHPSWMS-6.
CHPSWMS- 6	TOD Response The ToD server is expected to reply with the current time in UTC format.	CHPSWMS-6 MUST occur after CHPSWMS-5 completion.	Report an error, and return to CHPSWMS-5 (continue to retry TOD until successful).
CHPSWMS- 7	AS Request[1] The PS sends the AS Request message to the MSO IPCable2Home KDC provided in DHCP Option 177 suboption 51, to request a Kerberos ticket.	CHPSWMS-7 MUST occur after CHPSWMS-6 completion.	Return to CHPSWMS-1. PS initiates operation of CDS.

CHPSWMS- 8	AS Reply The AS Reply Message is received from the MSO IPCable2Home KDC containing the Kerberos ticket	CHPSWMS-8 MUST occur after CHPSWMS-7 completion.	Return to CHPSWMS-1. PS initiates operation of CDS.
CHPSWMS- 9	TGS Request If the PS obtained a Ticket Granting Ticket (TGT) during step CHPSWMS-8, the PS sends the TGS Request message to the MSO KDC server whose address was passed to the PS (CDC) in DHCP Option 177 sub-option 51.	CHPSWMS-9 MUST occur after CHPSWMS-8 completion.	Return to CHPSWMS-1. PS initiates operation of CDS.
CHPSWMS- 10	TGS Reply The TGS Reply message containing the ticket is received from the MSO IPCable2Home KDC.	CHPSWMS-10 MUST occur after CHPSWMS-9 completion.	Return to CHPSWMS-1. PS initiates operation of CDS.
CHPSWMS- 11	AP Request The PS sends the AP Request message to the NMS (SNMP manager) to request keying information for SNMPv3, as described in Section 11.3 PS Authentication Infrastructure.	CHPSWMS-11 MUST occur after CHPSWMS-10 completion.	Return to CHPSWMS-1. PS initiates operation of CDS.
CHPSWMS- 12	AP reply The AP Reply message is received from the NMS containing the keying information for SNMPv3.Note: The PS MUST establish SNMPv3 keys AND populate the associated SNMPv3 tables before it sends an SNMPv3 Inform message. The keys and tables are established using the information in the AP Reply. Refer to Section 11.3 PS Authentication Infrastructure.	CHPSWMS-12 MUST occur after CHPSWMS-11 completion.	Return to CHPSWMS-1. PS initiates operation of CDS.
CHPSWMS- 13	SNMP Inform After the PS operating in SNMP Provisioning Mode establishes SNMPv3 keys, it MUST send an SNMPv3 INFORM (cabhPsDevProvEnrollTrap) requesting enrollment to the SNMP ENTITY whose IP address was provided in Option 177 suboption 3, in the DHCP ACK message.	CHPSWMS-13 MUST occur after CHPSWMS-12 completion.	Return to CHPSWMS-1.
CHPSWMS- 14	SYSLOG message If the PS received a SYSLOG server address in the DHCP ACK, the PS MUST send the SYSLOG a “provisioning complete” message. This notification will include the pass-fail result of the provisioning operation. The general format of this message is defined in Table B-1 Defined Events for IPCable2Home, Event ID 73001100 (see Message Notes and Details).	CHPSWMS-14 MUST occur after CHPSWMS-13 completion.	

CHPSWMS- 15	SNMP Inform The PS MUST send the NMS an SNMP INFORM (cabhPsDevInitTrap) containing a “provisioning complete” notification. FAIL occurs when the Configuration File processing fails. Otherwise the provisioning state is PASS. The PS MUST update the value of cabhPsDevProvState with status 'pass' (1) when provisioning flow steps CHPSWMS-1 through CHPSWMS-15 complete successfully. The PS MUST update the value of cabhPsDevProvState with status 'fail' (3) and report an event indicating provisioning process failure if the PS Provisioning Timer expires before the value of cabhPsDevProvState is updated with status 'pass'.	CHPSWMS-15 MUST occur after CHPSWMS-14 completion.	If the PS does not receive a response to the Provisioning Complete inform, the PS MUST retry to send the cabhPsDevInitTrap inform, for a total of 5 attempts, at an interval of 10 seconds. If all 5 attempts to send the cabhPsDevInitTrap fail, the PS MUST re-start the initialization process: return to CHPSWMS-1 and report an error.
Optional Steps			
CHPSWMS- 16	SNMP Get If any additional device capabilities are needed by the provisioning system, the provisioning system requests these from the PS via SNMPv3 Get Requests. Iterative: The NMS sends the PS one or more SNMPv3 GET requests to obtain any needed PS capability information. The Provisioning Application may use a GETBulk request to obtain several pieces of information in a single message.	CHPSWMS-16 is not expected to occur before CHPSWMS-15 completion.	Return to CHPSWMS- 1.
CHPSWMS- 17	SNMP Get Response Iterative: The PS replies to the NMS Get-request or Get-bulk request messages with a Get Response for each Get Request. After all the Gets, or the GetBulk, finish, the NMS sends the requested data to the provisioning application	If CHPSWMS-16 occurs, CHPSWMS-17 MUST occur after CHPSWMS-16 completes.	N/A
CHPSWMS- 18	Configuration File Create Optional: The provisioning system uses information from PS provisioning steps CHPSWMS-16 and CHPSWMS-17 to create a PS configuration file. The provisioning system runs a hash on the contents of the configuration file. The hash is sent to the PS in the next step.	If CHPSWMS-17 occurs, CHPSWMS-18 MUST occur after CHPSWMS-17 completes.	N/A
CHPSWMS- 19	SNMP Set The provisioning system might instruct the NMS to send an SNMP Set message to the PS containing the IP Address of the TFTP server, the PS Configuration File filename and the hash of the configuration file as described in Section 7.4.4.1 Configuration File Format Requirements (SNMP Provisioning Mode). Optionally, the IP Address of the Firewall Configuration File TFTP server, the Firewall Configuration File filename and the hash of the firewall Configuration File are included in the SNMP set if there is a firewall Configuration File to	If CHPSWMS-18 occurs, CHPSWMS-19 MUST occur after CHPSWMS-18 completes.	Return to CHPSWMS- 1 if the set was received, but there was a processing error.

	be loaded, and this method is selected to specify it.		
CHPSWMS- 20	TFTP Request If the NMS triggers the PS to download a PS Configuration File as described in Section 7.4.4.1, the PS sends the TFTP Server a TFTP Get Request to request the specified PS Configuration File.	If CHPSWMS-19 occurs, CHPSWMS-20 MUST occur after CHPSWMS-19 completes.	Continue with CHPSWMS-19.
CHPSWMS- 21	TFTP server sends Configuration File After the PS receives the PS Configuration File, the PS calculates the hash of the PS Configuration File and compares it to the value received in step CHPSWMS-19. The PS then processes the PS Configuration File. Optionally, the IP Address of the Firewall Configuration File TFTP server, the Firewall Configuration File filename and the hash of the firewall configuration file are included in the PS Configuration File if there is a firewall Configuration File to be loaded, and this is the method selected to specify it.	If CHPSWMS-20 occurs, CHPSWMS-21 occurs after CHPSWMS-20 completes.	If the TFTP download fails, report an error, proceed to CHPSWMS-22, and continue to retry CHPSWMS-20 (continue to retry PS Configuration File download). If processing of the Configuration File produces an error, continue and report the error as an event.
CHPSWMS- 22	TFTP Request - Firewall Configuration File (optional) The PS sends the Firewall Configuration TFTP Server a TFTP Get Request to request the specified firewall configuration data file.	If CHPSWMS-22 occurs, it MUST occur after CHPSWMS-21 completes.	Return to CHPSWMS- 1.
CHPSWMS- 23	TFTP server sends Firewall Configuration File The TFTP Server sends the PS a TFTP Response containing the requested file. After the PS receives the Firewall Configuration File the PS calculates the hash of the Firewall Configuration File and compares it to the value received in step CHPSWMS-21. The file is then processed. Refer to Section for description of PS configuration file contents.	If CHPSWMS-22 occurs, CHPSWMS-23 MUST occur after CHPSWMS-22 completes.	If the TFTP download fails, continue with PS operation but report an error and continue to retry CHPSWMS-22. If processing of the firewall configuration file produces an error, continue and report the error as an event.

Notes to Table 13-3:

1. Steps CHPSWMS-5-CHPSWMS-8 are optional in some cases. Refer to Section 11 for details.
2. The SNMP Get and following SNMP Get Response operations are optional, depending on whether additional information is required to form a PS Configuration File, and also depending on whether a PS Configuration File is needed.

13.4.1 PS WAN-Man Configuration File Download

The PS operating in SNMP Provisioning Mode might contain sufficient factory default information to provide for operation of either or both LAN and WAN sides without a PS Configuration File being downloaded. If the PS is operating in SNMP Provisioning Mode the NMS might trigger the download of a PS Configuration File for initial provisioning to replace the factory defaults or to provide additional information.

The firewall Configuration File contains information to provision the firewall function. The indication to download a firewall Configuration File will come in either the PS Configuration File or via an SNMP Set during initialization.

13.4.2 PS Provisioning Timer

A provisioning timer is provided to ensure that the PS will continue to cycle through the provisioning process should any operation not complete. The timer object, cabhPsDevProvTimer, has a default initialization of 5 minutes.

13.4.3 Provisioning Enrollment/Provisioning Complete Informs

For the PS operating in SNMP Provisioning Mode only, the provisioning enrollment inform (cabhPsDevProvEnrollTrap) enables the Provisioning Server to determine that the PS is ready for the PS Configuration File.

In either DHCP Provisioning Mode or SNMP Provisioning Mode, the provisioning complete trap (cabhPsDevInitTrap) indicates whether the provisioning sequence has completed successfully or not.

13.4.4 SYSLOG Provisioning

The syslog server IP address MUST be provisioned through the DHCP process. The syslog event will not be sent if the syslog server IP address is not configured.

13.4.5 Provisioning State and Error Reporting

As indicated in Table 13-1 and Table 13-3, failure of the steps in the provisioning process generally results in the process restarting at the first step, CHPSWMD-1 or CHPSWMS-1.

13.5 PS WAN-Data Provisioning Process

The PS requests zero or more WAN-Data network address(es) from the DHCP server in the cable network to be used for the exchange of data between elements connected to the Internet and LAN IP Devices.

There is no difference in PS WAN-Data operation between the DHCP and SNMP Provisioning Modes.

The following diagrams illustrate the message flows that are to be used to accomplish the provisioning of PS WAN-Data addresses. The provisioning process for the PS WAN-Data addresses is the same for the PS embedded with a DOCSIS cable modem as it is for the stand-alone PS.

If the provisioning process for the PS WAN-Data address(es) occurs, it MUST follow the sequence depicted in Table 13-6 and described in detail in Table 13-4.

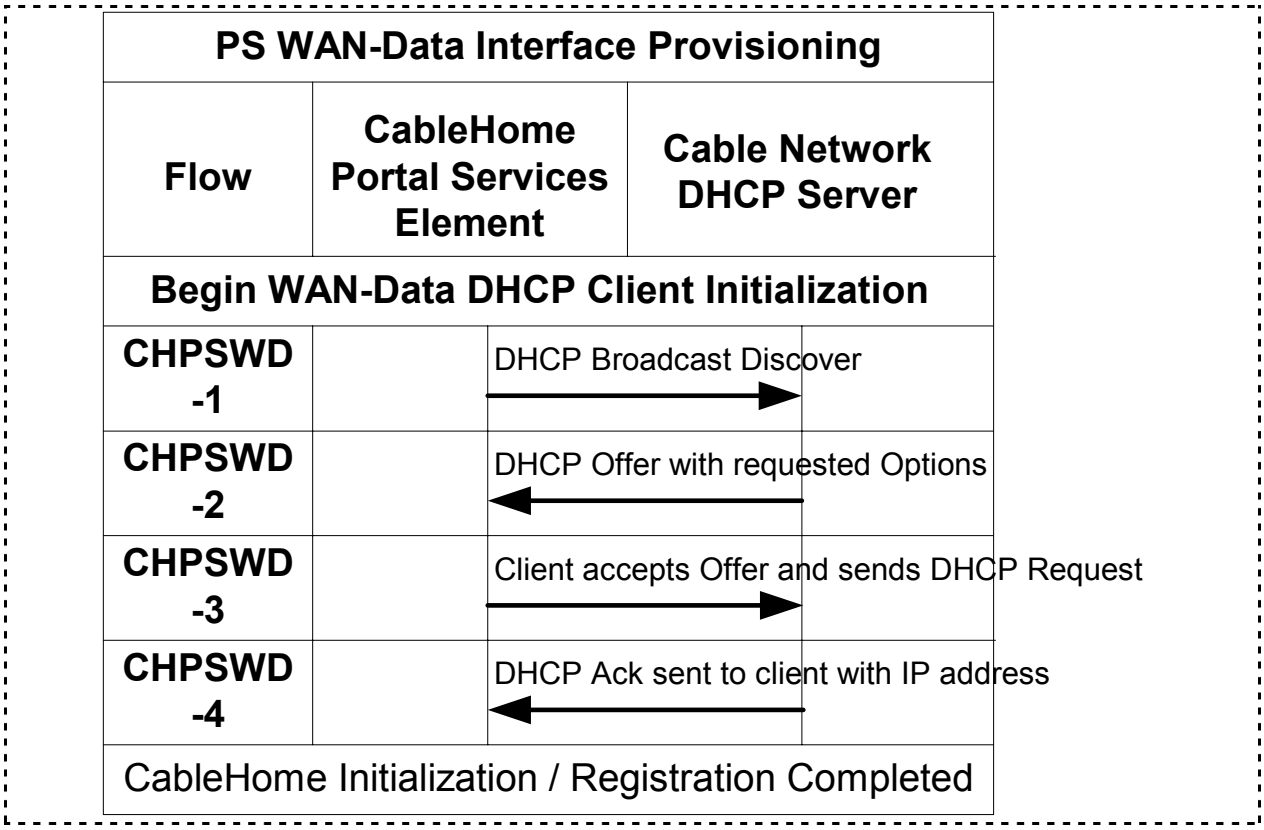

Figure 13-6 PS WAN-Data Provisioning Process

Table 13-4 Flow Descriptions for PS WAN-Data Provisioning Process

Flow Step	PS WAN-Data Address Provisioning	Normal Sequence	Failure Sequence
CHPSWD-1	DHCP Broadcast Discover The PS broadcasts a DHCP DISCOVER message including the mandatory options listed in Table 10, CDC DHCP Options in DISCOVER and REQUEST messages..	Proceed to CHPSWD-2.	If failure per DHCP protocol repeat CHPSWD-1.
CHPSWD-2	DHCP OFFER The DHCP Server at the Headend receives the DHCP DISCOVER packet, assigns an IP address from the WAN- Data pool, builds a DHCP OFFER packet, and transmits the DHCP OFFER to the DHCP Relay Agent [RFC 3046] in the CMTS.	Proceed to CHPSWD-3.	If failure, the client will time out per DHCP protocol and CHPSWD-1 will be repeated.
CHPSWD-3	DHCP REQUEST The CDP sends a DHCP REQUEST message to the selected DHCP server to accept the DHCP OFFER in accordance with client requirements of [RFC 2131].	CHPSWD-3 MUST occur after CHPSWD-2 completion.	If failure per DHCP protocol return to CHPSWD-1.
CHPSWD-4	DHCP ACK The DHCP server sends the CDP a DHCP ACK message which contains the IPv4 address for the PS WAN Data interface.	CHPSWD-4 MUST occur after CHPSWD-3 completion. Provisioning complete with completion of CHPSWD-4.	If failure per DHCP protocol return to CHPSWD-1.

13.6 Provisioning Process: BP in the LAN-Trans Realm

Boundary Point (BP) logical elements are required to implement two protocols used during their provisioning process: DHCP [RFC 2131] and BP_Init messaging, defined in Section 6.5.3.2 MBP LAN Messaging Function.

The CDP (CDS) function of the PS element responds to DHCP messages issued by BPs in the LAN-Pass realm according to the requirements defined in Section 7.3.3.1.4 CDS Function Requirements. The PS CMP function responds to the BP_Init message received from BPs. This is described in Section 6.3.3.4 CMP LAN Messaging Function.

This section describes the provisioning process for the case where the NMS has provisioned the PS to operate in C-NAT or C-NAPT Primary Packet Handling mode (see Section 8). There is no difference in LAN-Trans realm BP provisioning process between the DHCP and SNMP Provisioning Modes.

The provisioning process for a BP in the LAN-Trans realm MUST occur via the sequence depicted in Figure 13-7 and described in detail in Table 13-5.

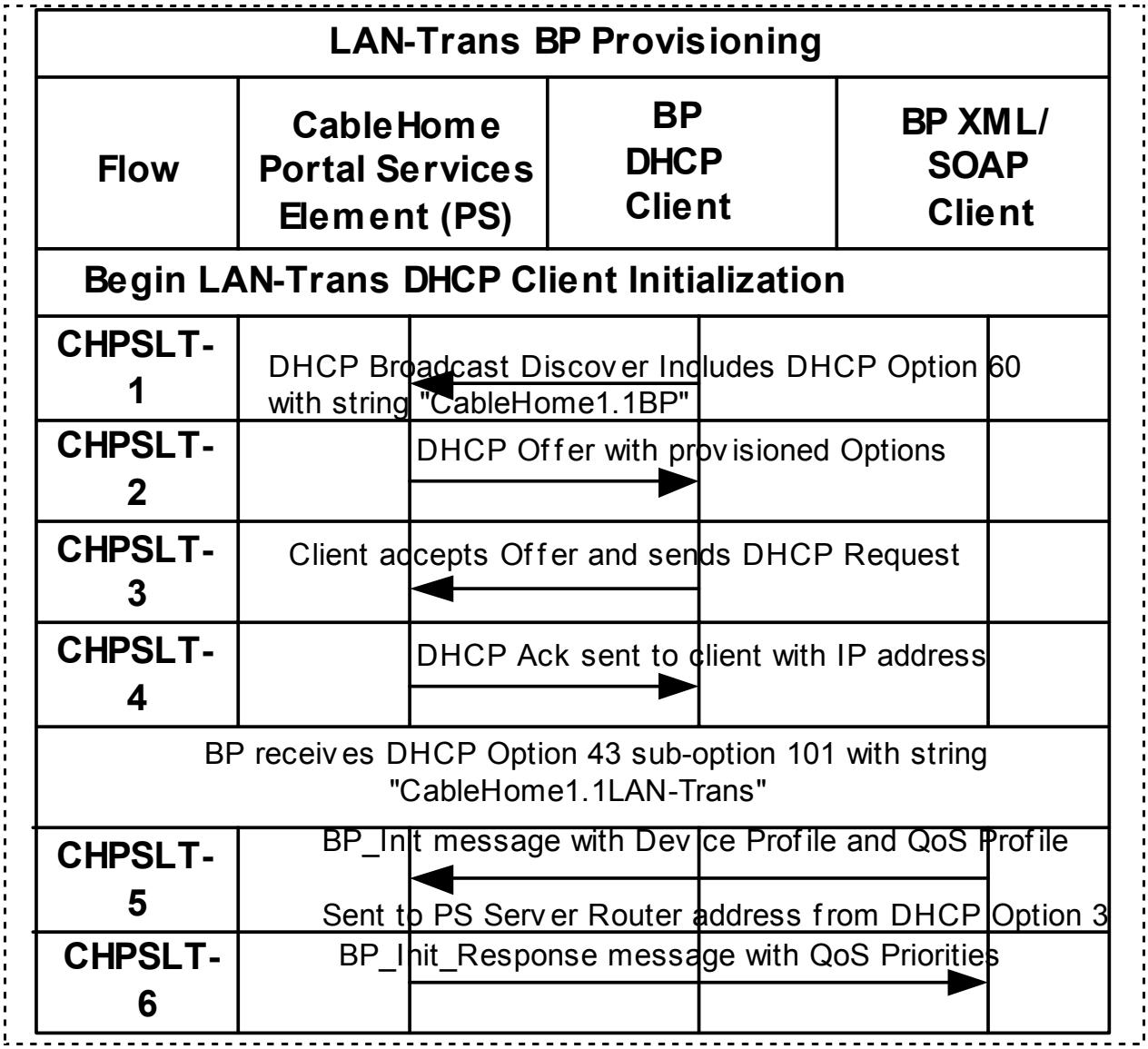

Figure 13-7 Provisioning Process for a BP in the LAN-Trans Realm

Table 13-5 Flow Descriptions for LAN-Trans BP Provisioning Process

Flow Step	Client LAN-Trans Address Provisioning	Normal Sequence	Failure Sequence
CHBPLT-1	DHCP Broadcast Discover The DHCP Client[5] sends a broadcast DHCP DISCOVER message on its local LAN.[6] The BP is required to include DHCP Option 60 containing string "CableHome1.1BP"	Proceed to CHBPLT-2.	If failure per DHCP protocol repeat CHBPLT -1.
CHBPLT-2	DHCP Offer The PS receives the DHCP DISCOVER message on its LAN interface and examines the chaddr field. If: - there is a LAN-Trans address available, and - there is no administrative consideration which motivates denying the LAN-Trans address to the client; then the PS MUST send a DHCP OFFER message to the client to offer it the LAN-Trans address as either unicast or link-specific broadcast (according to the BROADCAST bit of the flags field of the DHCP DISCOVER). If the DHCP discover included DHCP Option 60 containing the string "CableHome1.1BP" the PS is required to include DHCP Option 43 sub-option 101 containing the string "CableHome1.1LANTrans" in the DHCP Offer message.	Proceed to CHBPLT-3.	If failure, the client will time out per DHCP protocol and CHBPLT -1 will be repeated.
CHBPLT-3	DHCP Request The LAN IP Device's DHCP client receives the DHCPOFFER message. When a LAN IP Device's DHCP client wishes to accept a DCHP OFFER, it is expected that it will format and send a DHCP REQUEST packet using link-specific broadcast.[7]	Proceed to CHBPLT-4.	If failure, the client will time out per DHCP protocol and CHBPLT -1 will be repeated.
CHBPLT-4	DHCP ACK The PS receives the DHCP REQUEST on its LAN interface. If the indicated LAN-Trans address is still assignable, the PS MUST then send DHCP ACK to the client as either unicast or link-specific broadcast (according to the BROADCAST bit of the flags field of the DHCP REQUEST). The DHCP ACK includes DHCP Option 43 sub-option 101 with the string "CableHome1.1LAN-Trans". This is an indication to the BP that it is in the LAN-Trans address realm and received the PS Server Router IP address in DHCP Option 3. The BP is therefore required to send its BP_Init messages to the PS Server Router IP address.	Proceed to CHBPLT-5.	If failure, the client will time out per DHCP protocol and CHBPLT -1 will be repeated.
CHBPLT-5	BP_Init The BP sends a BP_Init SOAP/XML message with its Device and QoS Profiles to the PS Server Router IP address.	Proceed to CHBPLT-6.	If the BP does not receive BP_Init_Response, it retries BP_Init for a total of three attempts
CHBPLT-6	BP_Init_Response The PS sends a BP_Init_Response SOAP/XML message to the BP.	Provisioning Complete.	

[5] 1. If the client is aware of its previous IP address (e.g., following reboot), it may omit the DHCPDISCOVER and proceed with step 3.

[6] 1. If the client is located on a non-broadcast network it is expected to unicast the message to the DHCP Server.

[7] 1. If the client is located on a non-broadcast network it is expected that it will unicast the message to the PS.

13.7 Provisioning Process: LAN IP Device in the LAN-Pass Realm

Some home LAN applications will not function properly with a translated network address. To accommodate these applications the PS is enabled to operate in Passthrough (transparent bridging) mode. As described in Section 8.3.3.1 Packet Handling Modes, bridging occurs when the cable network NMS sets the Primary Packet-handling mode (cabhCapPrimaryMode) to Passthrough, or by writing individual LAN IP Device MAC addresses into the Passthrough Table (cabhCapPassthroughTable). Figure 13-8 describes the process for the request and assignment of a network address to LAN IP Devices for which the PS has been pre-provisioned to bridge traffic. When the PS has been configured to bridge traffic for a LAN IP Device, DHCP DISCOVERs and DHCP REQUESTs issued by that LAN IP Device will be served by the cable network DHCP server, not by the CDS.

A non-IPCable2Home compliant LAN IP Device is assumed to implement a DHCP client and request an IP address lease using DHCP [RFC 2131]. A IPCable2Home compliant LAN IP Device, i.e., one that implements BP functionality defined in this specification, is required to implement a DHCP client and request an IP address lease via DHCP. The BP logical element of a IPCable2Home compliant LAN IP Device is also required to exchange BP_Init messaging with the PS, as described in Section 6.5.3.2 MBP LAN Messaging Function. This section describes the required BP messaging. The DHCP messaging assumed to occur between a non-compliant LAN IP Device and a DHCP server will typically follow the first four steps of the required BP DHCP messaging. However, a non-compliant LAN IP Device is not likely to include the DHCP Option 61 string "CableHome 1.1 BP <*hardware address*>".

The provisioning process for a BP in the LAN-Pass realm is required to occur via the sequence depicted in Figure 13-8 and described in detail in Table 13-6.

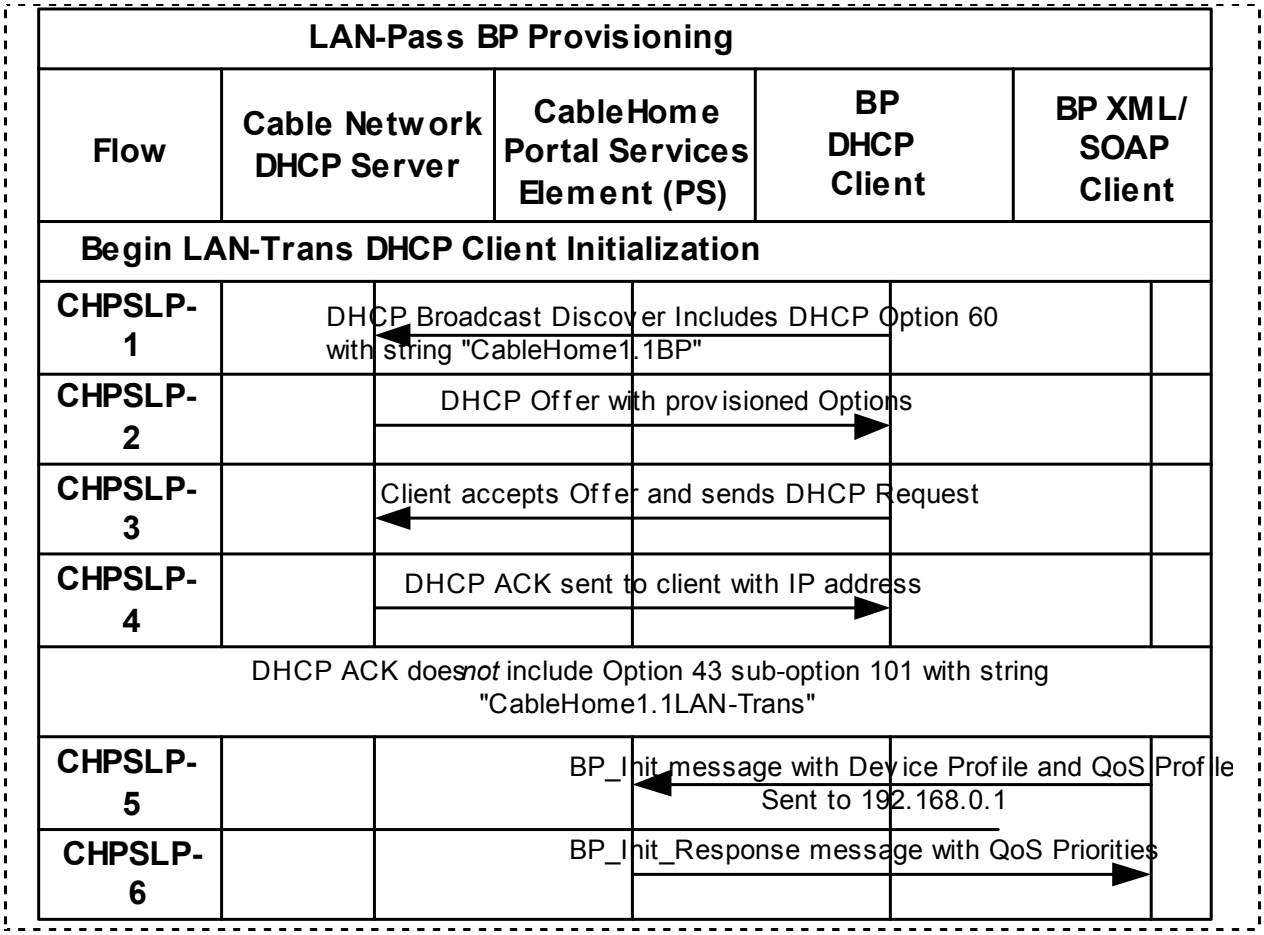

Figure 13-8 Provisioning Process for BP in the LAN-Pass Realm

Table 13-6 Flow Descriptions for LAN-Pass BP Provisioning Process

Flow Step	Client Pass Thru Address Provisioning	Normal Sequence	Failure Sequence
CHBPLP-1	DHCP Broadcast Discover The BP or non-IPCable2Home compliant LAN IP Device broadcasts a DHCP DISCOVER message on its local LAN.[8] The PS receives the broadcast DHCP DISCOVER packet on its LAN interface and is required to transparently bridge the packet to the WAN interface without changing the content of the packet. Refer to Section 8.3.4 CAP Requirements.	Proceed to CHBPLP -2.	If failure per DHCP protocol repeat CHBPLP -1.
CHBPLP-2	The DHCP Server in the cable operator's network receives the DHCP DISCOVER packet and assigns an externally addressable IP address and other options, builds a DHCP OFFER packet, and transmits the DHCP OFFER to the LAN IP Device. The PS is required to transparently bridge the DHCP OFFER from its WAN interface to its LAN interface without changing the content of the IP packet. Refer to Section 8.3.4 CAP Requirements.	Proceed to CHBPLP -3.	If failure, the LAN IP Device will time out per DHCP protocol and CHBPLP-1 will be repeated.
CHBPLP-3	DHCP REQUEST The LAN IP Device receives the DHCP OFFER and issues a DHCP REQUEST message. The PS is required to transparently bridge the DHCP REQUEST from its LAN interface to its WAN interface without changing the content of the IP packet. Refer to Section 8.3.4 CAP Requirements.	Proceed to CHBPLP -4.	If failure per DHCP protocol repeat CHBPLP -1.
CHBPLP-4	The DHCP server in the cable operator's network receives the DHCP REQUEST and sends the DHCP ACK to the LAN IP Device with the LAN IP Device's IPv4 address. The PS is required to transparently bridge the DHCP ACK from its WAN interface to its LAN interface without changing the content of the IP packet. Refer to Section 8.3.4 CAP Requirements.The DHCP ACK is expected to not include DHCP Option 43 sub-option 101 with the string "CableHome 1.1 LAN-Trans". This signals the BP that it is in the LAN-Pass address realm and did not receive the PS Server Router address in DHCP Option 3, so it is required to send its BP_Init messages to the "well known" PS IP address 192.168.0.1. Refer to Section 6.5.3.2 MBP LAN Messaging Function.	Proceed to CHBPLP-5.	If failure, the LAN IP Device will time out per DHCP protocol and CHBPLP -1 will be repeated.
CHBPLP-5	BP_Init The BP sends a BP_Init SOAP/XML message with its Device and QoS Profiles to the PS.	Proceed to CHBPLP-6.	If the BP does not receive BP_Init_Response, it retries BP_Init for a total of three attempts
CHBPLP-6	BP_Init_Response The PS sends a BP_Init_Response SOAP/XML message to the BP.	Provisioning complete.	

[8] 1. If the client is located on a non-broadcast network it must unicast the message to the DHCP Server or DHCP Relay Agent [RFC 3046] in the cable network

Annex A MIB Objects

This annex lists all required MIB objects, as indicated in Section 6.3.3.1.4.1, *SNMP Protocol Requirements* and Section 6.3.3.1.4.7, IP*Cable2Home MIB Requirements*, and indicates requirement for persistence of each listed object.
The term 'persistent' as it applies to this Annex is defined below:

> *Persistent*: The requirement for the PS to retain the value of a configurable (by the manager or by the PS itself) MIB object across a PS reboot or reset.

For MIB objects with entry 'Yes' in the Persistent column, the object's value immediately following a PS reboot or reset, MUST be the same as its value immediately preceding the reboot or reset.
For MIB objects with entry 'No' in the Persistent column, the object's value MUST be set to its factory default value (DEFVAL) or, if it has no default value, it MUST be set to zero or null as appropriate, immediately following a PS reboot or reset.
For MIB objects with entry "-" in the Persistent column, one of the following apply:

- the value of the object immediately following PS reboot, or reset is left to vendor implementation because there is no specific requirement for its value following PS reboot or reset, or
- the value of the object is deterministic, based upon the MIB description. (the object's value is fixed or can be derived from known values after the PS reboot or reset)

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
mib-2[RFC 1213]			
system			
sysDescr	read-only	-	N/A
sysObjectID	read-only	-	N/A
sysUpTime	read-only	-	N/A
sysContact	read-write	Yes	1
sysName	read-write	Yes	1
sysLocation	read-write	Yes	1
sysServices	read-only	-	N/A
interfaces [RFC 2863]			
ifNumber	read-only	-	N/A
ifTable/ifEntry			
ifIndex	read-only	-	N/A
ifDescr	read-only	-	N/A
ifType	read-only	-	N/A
ifMtu	read-only	-	N/A
ifSpeed	read-only	-	N/A
ifPhysAddress	read-only	-	N/A
ifAdminStatus	read-write	No	N/A
ifOperStatus	read-only	-	N/A
ifLastChange	read-only	-	N/A
ifInOctets	read-only	-	N/A

ifInUcastPkts	read-only	-	N/A
ifInDiscards	read-only	-	N/A
ifInErrors	read-only	-	N/A
ifInUnknownProtos	read-only	-	N/A
ifOutOoctets	read-only	-	N/A
ifOutUcastPkts	read-only	-	N/A
ifOutDiscards	read-only	-	N/A
ifOutErrors	read-only	-	N/A
ip [RFC 2011]			
ipForwarding	read-write	No	N/A
ipDefaultTTL	read-write	No	N/A
ipInReceives	read-only	-	N/A
ipInHdrErrors	read-only	-	N/A
ipInAddrErrors	read-only	-	N/A
ipForwDatagrams	read-only	-	N/A
ipInUnknownProtos	read-only	-	N/A
ipInDiscards	read-only	-	N/A
ipInDelivers	read-only	-	N/A
ipOutRequests	read-only	-	N/A
ipOutDiscards	read-only	-	N/A
ipOutNoRoutes	read-only	-	N/A
ipReasmTimeout	read-only	-	N/A
ipReasmReqds	read-only	-	N/A
ipReasmOKs	read-only	-	N/A
ipReasmFails	read-only	-	N/A
ipFragOKs	read-only	-	N/A
ipFragFails	read-only	-	N/A
ipFragCreates	read-only	-	N/A
ipNetToMediaTable/ipNetToMediaEntry			
ipNetToMediaIfIndex	read-create	No	N/A
ipNetToMediaPhyAddress	read-create	No	N/A
ipNetToMediaNetAddress	read-create	No	N/A
ipNetToMediaType	read-create	No	N/A
icmp			
icmpInMsgs	read-only	-	N/A
icmpInErrors	read-only	-	N/A
icmpInDestUnreachs	read-only	-	N/A
icmpInTimeExcds	read-only	-	N/A
icmpInParmProbs	read-only	-	N/A

icmpInSrcQuenchs	read-only	-	N/A
icmpInRedirects	read-only	-	N/A
icmpInEchos	read-only	-	N/A
icmpInEchosReps	read-only	-	N/A
icmpInTimestamps	read-only	-	N/A
icmpInTimestampsReps	read-only	-	N/A
icmpInAddrMasks	read-only	-	N/A
icmpInAddrMaskReps	read-only	-	N/A
icmpOutMsgs	read-only	-	N/A
icmpOutErrors	read-only	-	N/A
icmpOutDestUnreachs	read-only	-	N/A
icmpOutTimeExcds	read-only	-	N/A
icmpOutParmProbs	read-only	-	N/A
icmpOutSrcQuenchs	read-only	-	N/A
icmpOutRedirects	read-only	-	N/A
icmpOutEchos	read-only	-	N/A
icmpOutEchosReps	read-only	-	N/A
icmpOutTimestamps	read-only	-	N/A
icmpOutTimestampReps	read-only	-	N/A
icmpOutAddrMasks	read-only	-	N/A
icmpOutAddrMaskReps	read-only	-	N/A

udp [RFC 2013]

udpInDatagrams	read-only	-	N/A
udpNoPorts	read-only	-	N/A
udpInErrors	read-only	-	N/A
udpOutDatagrams	read-only	-	N/A
udpTable/udpEntry			
udpLocalAddress	read-only	-	N/A
udpLocalPort	read-only	-	N/A

transmission [draft-ietf-ipcdn-bpiplus-mib-05]

docsIfMib

docsBpi2MIB

docsBpi2MIBObjects

docsBpi2CmObjects

docsBpi2CmCertObjects

docsBpi2CmDeviceCertTable/docsBpi2CmDeviceCertEntry			
docsBpi2CmDeviceCmCert	read-only	-	N/A
docsBpi2CmDeviceManufCert	read-only	-	N/A
docsBpi2CodeDownloadGroup			
docsBpi2CodeDownloadStatusCode	read-only	-	N/A
docsBpi2CodeDownloadStatusString	read-only	-	N/A
docsBpi2CodeMfgOrgName	read-only	-	N/A
docsBpi2CodeMfgCodeAccessStart	read-only	-	N/A
docsBpi2CodeMfgCvcAccessStart	read-only	-	N/A
docsBpi2CodeCoSignerOrgName	read-only	-	N/A
docsBpi2CodeCoSignerCodeAccessStart	read-only	-	N/A
docsBpi2CodeCoSignerCvcAccessStart	read-only	-	N/A
docsBpi2CodeCvcUpdate	read-write	Yes	1
snmp [RFC 3418]			
snmpInPkts	read-only	-	N/A
snmpInBadVersions	read-only	-	N/A
snmpInBadCommunityNames	read-only	-	N/A
snmpInBadCommunityUses	read-only	-	N/A
snmpInASNParseErrs	read-only	-	N/A
snmpEnableAuthenTraps	read-write	No	N/A
snmpSilentDrops	read-only	-	N/A
ifMIB [RFC 2863]			
ifMIBOjects			
ifXTable/ifXEntry			
ifName	read-only	-	N/A
ifInMulticastPkts	read-only	-	N/A
ifInBroadcastPkts	read-only	-	N/A
ifOutMulticastPkts	read-only	-	N/A
ifOutBroadcastPkts	read-only	-	N/A
ifLinkUpDownTrapEnable	read-write	No	N/A
ifHighSpeed	read-only	-	N/A
ifPromiscuousMode	read-write	No	N/A
ifConnectorPresent	read-only	-	N/A
ifAlias	read-write	No	N/A
ifCounterDiscontinuityTime	read-only	-	N/A

ifStackTable/ifStackEntry			
ifStackHigherLayer	read-only	-	N/A
ifStackLowerLayer	read-only	-	N/A
ifStackStatus	read-only	-	N/A

docsDev [RFC 2669]

docsDevMIBObjects

docsDevNmAccessTable/docsDevNmAccessEntry			
docsDevNmAccessIndex	not-accessible	-	N/A
docsDevNmAccessIp	read-create	No	N/A
docsDevNmAccessIpMask	read-create	No	N/A
docsDevNmAccessCommunity	read-create	No	N/A
docsDevNmAccessControl	read-create	No	N/A
docsDevNmAccessInterfaces	read-create	No	N/A
docsDevNmAccessStatus	read-create	No	N/A
docsDevNmAccessTrapVersion	read-create	No	N/A

docsDevSoftware

docsDevSwServer	read-write	Yes	1
docsDevSwFilename	read-write	Yes	1
docsDevSwAdminStatus	read-write	Yes	1
docsDevSwOperStatus	read-only	Yes	1
docsDevSwCurrentVers	read-only	-	N/A

docsDevEvent

docsDevEvControl	read-write	No	N/A
docsDevEvSyslog	read-write	No	N/A
docsDevEvThrottleAdminStatus	read-write	No	N/A
docsDevEvThrottleInhibited	read-only	-	N/A
docsDevEvThrottleThreshold	read-write	No	N/A
docsDevEvThrottleInterval	read-write	No	N/A
docsDevEvControlTable/docsDevEvControlEntry			
docsDevEvPriority	not-accessible	-	N/A
docsDevEvReporting	read-write	No	N/A
docsDevEventTable/docsDevEventEntry			
docsDevEvIndex	not-accessible	-	N/A
docsDevEvFirstTime	read-only	Yes	10
docsDevEvLastTime	read-only	Yes	10
docsDevEvCounts	read-only	Yes	10

docsDevEvLevel	read-only	Yes	10
docsDevEvId	read-only	Yes	10
docsDevEvText	read-only	Yes	10
docsDevFilter			
docsDevFilterIpTable/docsDevFilterIpEntry			
docsDevFilterIpIndex	not-accessible	-	N/A
docsDevFilterIpStatus	read-create	Yes	20
docsDevFilterIpControl	read-create	Yes	20
docsDevFilterIpIfIndex	read-create	Yes	20
docsDevFilterIpDirection	read-create	No	N/A
docsDevFilterIpBroadcast	read-create	No	N/A
docsDevFilterIpSaddr	read-create	Yes	20
docsDevFilterIpSmask	read-create	Yes	20
docsDevFilterIpDaddr	read-create	Yes	20
docsDevFilterIpDmask	read-create	Yes	20
docsDevFilterIpProtocol	read-create	Yes	20
docsDevFilterIpSourcePortLow	read-create	Yes	20
docsDevFilterIpSourcePortHigh	read-create	Yes	20
docsDevFilterIpDestPortLow	read-create	Yes	20
docsDevFilterIpDestPortHigh	read-create	Yes	20
docsDevFilterIpMatches	read-only	-	N/A
docsDevFilterIpTos	read-create	No	N/A
docsDevFilterIpTosMask	read-create	No	N/A
docsDevFilterIpContinue	read-create	No	N/A
docsDevFilterIpPolicyId	read-create	Yes	20

private

enterprises

cableLabs

clabProject

clabProjCableHome

cabhPsDevMib

cabhPsDevBase

cabhPsDevDateTime	read-write	No	N/A

cabhPsDevResetNow	read-write	No	N/A
cabhPsDevSerialNumber	read-only	-	N/A
cabhPsDevHardwareVersion	read-only	-	N/A
cabhPsDevWanManMacAddress	read-only	-	N/A
cabhPsDevWanDataMacAddress	read-only	-	N/A
cabhPsDevTypeIdentifier	read-only	-	N/A
cabhPsDevSetToFactory	read-write	No	N/A
cabhPsDevTodSyncStatus	read-only	-	N/A
cabhPsDevProvMode	read-only	-	N/A

cabhPsDevProv

cabhPsDevProvisioningTimer	read-write	No	N/A
cabhPsDevProvConfigFile	read-write	No	N/A
cabhPsDevProvConfigHash	read-write	No	N/A
cabhPsDevProvConfigFileSize	read-only	-	N/A
cabhPsDevProvConfigFileStatus	read-only	-	N/A
cabhPsDevProvConfigTLVProcessed	read-only	-	N/A
cabhPsDevProvConfigTLVRejected	read-only	-	N/A
cabhPsDevProvSolicitedKeyTimeout	read-write	Yes	1
cabhPsDevProvState	read-only	-	N/A
cabhPsDevProvAuthState	read-only	-	N/A
cabhPsDevTimeServerAddrType	read-only	-	N/A
cabhPsDevTimeServerAddr	read-only	-	N/A

cabhPsDevAttrib

cabhPsDevPsAttrib

cabhPsDevPsDeviceType	read-only	-	N/A
cabhPsDevPsManufacturerURL	read-only	-	N/A
cabhPsDevPsModelURL	read-only	-	N/A
cabhPsDevPsModelUPC	read-only	-	N/A

cabhPsDevAttrib

cabhPsDevBpAttrib

cabhPsDevBpProfileTable/cabhPsDevBpProfileEntry			
cabhPsDevBpIndex	not-accessible	-	N/A
cabhPsDevBpDeviceType	read-only	-	N/A
cabhPsDevBpManufacturer	read-only	-	N/A
cabhPsDevBpManufacturerURL	read-only	-	N/A
cabhPsDevBpSerialNumber	read-only	-	N/A
cabhPsDevBpHardwareVersion	read-only	-	N/A
cabhPsDevBpHardwareOptions	read-only	-	N/A

cabhPsDevBpModelName	read-only	-	N/A
cabhPsDevBpModelNumber	read-only	-	N/A
cabhPsDevBpModelURL	read-only	-	N/A
cabhPsDevBpModelUPC	read-only	-	N/A
cabhPsDevBpModelSoftwareOs	read-only	-	N/A
cabhPsDevBpModelSoftwareVersion	read-only	-	N/A
cabhPsDevBpLanInterface	read-only	-	N/A
cabhPsDevBpNumberInterfacePriorities	read-only	-	N/A
cabhPsDevBpPhysicalLocation	read-only	-	N/A
cabhPsDevBpPhysicalAddress	read-only	-	N/A
cabhPsDevPsStats			
cabhPsDevLanIpTrafficResetCounters	read-write	No	N/A
cabhPsDevLanIpTrafficCountersLastReset	read-only	-	N/A
cabhPsDevLanIpTrafficEnabled	read-write	No	N/A
cabhPsDevLanIpTrafficTable/cabhPsDevLanIpTrafficEntry			
cabhPsDevLanIpTrafficIndex	not-accessible	-	N/A
cabhPsDevLanIpTrafficInetAddressType	read-only	-	N/A
cabhPsDevLanIpTrafficInetAddress	read-only	-	N/A
cabhPsDevLanIpTrafficInOctets	read-only	-	N/A
cabhPsDevLanIpTrafficIpOutOctets	read-only	-	N/A
cabhSecMib			
cabhSec2FwObjects			
cabhSec2FwBase			
cabhSec2FwEnable	read-write	Yes	N/A
cabhSec2FwPolicyFileURL	read-write	No	N/A
cabhSec2FwPolicyFileHash	read-write	No	N/A
cabhSec2FwPolicyFileOperStatus	read-only	-	N/A
cabhSec2FwPolicyFileCurrentVersion	read-write	Yes	N/A
cabhSec2FwClearPreviousRuleset	read-write	No	N/A
cabhSec2FwPolicySelection	read-write	Yes	N/A
cabhSec2FwEventSetToFactory	read-write	Yes	N/A
cabhSec2FwEventLastSetToFactory	read-only	Yes	N/A
cabhSec2FwPolicySuccessfulFileURL	read-only	Yes	1
cabhSec2FwEvent			
cabhSec2FwEventType	not-accessible	-	N/A
cabhSec2FwEventEnable	read-write	No	N/A

cabhSec2FwEventThreshold	read-write	No	N/A
cabhSec2FwEventInterval	read-write	No	N/A
cabhSec2FwEventCount	read-only	-	N/A
cabhSec2FwEventLogReset	read-write	No	N/A
cabhSec2FwLogEntry			
cabhSec2FwLogIndex	not-accessible	-	N/A
cabhSec2FwLogEventType	read-only	-	N/A
cabhSec2FwLogEventPriority	read-only	-	N/A
cabhSec2FwLogEventId	read-only	-	N/A
cabhSec2FwLogTime	read-only	-	N/A
cabhSec2FwLogIpProtocol	read-only	-	N/A
cabhSec2FwLogIpSourceAddr	read-only	-	N/A
cabhSec2FwLogIpDestAddr	read-only	-	N/A
cabhSec2FwLogIpSourcePort	read-only	-	N/A
cabhSec2FwLogIpDestPort	read-only	-	N/A
cabhSec2FwLogMessageType	read-only	-	N/A
cabhSec2FwLogReplayCount	read-only	-	N/A
cabhSec2FwLogMIBPointer	read-only	-	N/A
cabhSec2FwFilter			
cabhSec2FwFilterScheduleTable			
cabhSec2FwFilterScheduleEntry			
cabhSec2FwFilterScheduleIndex	not-accessible	-	N/A
cabhSec2FwFilterScheduleRowStatus	read-create	Yes	1
cabhSec2FwFilterScheduleStartTime	read-create	Yes	1
cabhSec2FwFilterScheduleEndTime	read-create	Yes	1
cabhSec2FwFilterScheduleDOW	read-create	Yes	1
cabhSecCertObjects			
cabhSecCertPsCert	read-only	-	1
cabhSecKerbBase			
cabhSecKerbPKINITGracePeriod	read-write	No	N/A
cabhSecKerbTGSGracePeriod	read-write	No	N/A
cabhSecKerbUnsolicitedKeyMaxTimeout	read-write	No	N/A
cabhSecKerbUnsolicitedKeyMaxRetries	read-write	No	N/A
cabhCapMib			
cabhCapObjects			
cabhCapBase			
cabhCapTcpTimeWait	read-write	No	N/A

cabhCapUdpTimeWait	read-write	No	N/A
cabhCapIcmpTimeWait	read-write	No	N/A
cabhCapPrimaryMode	read-write	No	N/A
cabhCapSetToFactory	read-write	No	N/A
cabhCapMap			
cabhCapMappingTable/cabhCapMappingEntry			
cabhCapMappingIndex	not-accessible	-	N/A
cabhCapMappingWanAddrType	read-create	Yes[1]	16
cabhCapMappingWanAddr	read-create	Yes[1]	16
cabhCapMappingWanPort	read-create	Yes[1]	16
cabhCapMappingLanAddrType	read-create	Yes[1]	16
cabhCapMappingLanAddr	read-create	Yes[1]	16
cabhCapMappingLanPort	read-create	Yes[1]	16
cabhCapMappingMethod	read-only	-	N/A
cabhCapMappingProtocol	read-create	Yes[1]	16
cabhCapMappingRowStatus	read-create	Yes	16
cabhCapPassthroughTable/cabhCapPassthroughEntry			
cabhCapPassthroughIndex	not-accessible	-	N/A
cabhCapPassthroughMACAddr	read-create	Yes	16
cabhCapPassthroughRowStatus	read-create	Yes	16

[1.] cabhCapMappingEntry objects are persistent if provisioned by the NMS and non-persistent if created dynamically based on outbound traffic. Refer to Section 8.3.4.4.

cabhCdpMib

cabhCdpObjects

cabhCdpBase

cabhCdpSetToFactory	read-write	No	N/A
cabhCdpLanTransCurCount	read-only	-	N/A
cabhCdpLanTransThreshold	read-write	No	N/A
cabhCdpLanTransAction	read-write	No	N/A
cabhCdpWanDataIpAddrCount	read-write	No	N/A
cabhCdpAddr			
cabhCdpLanAddrTable/cabhCdpLanAddrEntry			
cabhCdpLanAddrIpType	not-accessible	-	N/A

cabhCdpLanAddrIp	not-accessible	-	N/A
cabhCdpLanAddrClientID	read-create	Yes	16
cabhCdpLanAddrLeaseCreateTime	read-only	-	N/A
cabhCdpLanAddrLeaseExpireTime	read-only	-	N/A
cabhCdpLanAddrMethod	read-only	Yes	16
cabhCdpLanAddrHostName	read-only	Yes	16
cabhCdpLanAddrRowStatus	read-create	Yes	16
cabhCdpWanDataAddrTable/cabhCdpWanDataAddrEntry			
cabhCdpWanDataAddrIndex	not-accessible	-	N/A
cabhCdpWanDataAddrClientId	read-create	No	N/A
cabhCdpWanDataAddrIpType	read-only	-	N/A
cabhCdpWanDataAddrIp	read-only	-	N/A
cabhCdpWanDataAddrRenewalTime	read-only	-	N/A
cabhCdpWanDataAddrRowStatus	read-create	No	N/A
cabhCdpWanDnsServerTable/cabhCdpWanDnsServerEntry			
cabhCdpWanDnsServerOrder	not-accessible	-	N/A
cabhCdpWanDnsServerIpType	read-only	-	N/A
cabhCdpWanDnsServerIp	read-only	-	N/A

cabhCdpServer

cabhCdpLanPoolStartType	read-write	Yes	1
cabhCdpLanPoolStart	read-write	Yes	1
cabhCdpLanPoolEndType	read-write	Yes	1
cabhCdpLanPoolEnd	read-write	Yes	1
cabhCdpServerNetworkNumberType	read-write	Yes	1
cabhCdpServerNetworkNumber	read-write	Yes	1
cabhCdpServerSubnetMaskType	read-write	Yes	1
cabhCdpServerSubnetMask	read-write	Yes	1
cabhCdpServerTimeOffset	read-write	Yes	1
cabhCdpServerRouterType	read-write	Yes	1
cabhCdpServerRouter	read-write	Yes	1
cabhCdpServerDnsAddressType	read-write	Yes	1
cabhCdpServerDnsAddress	read-write	Yes	1
cabhCdpServerSyslogAddressType	read-write	Yes	1
cabhCdpServerSyslogAddress	read-write	Yes	1
cabhCdpServerDomainName	read-write	Yes	1
cabhCdpServerTTL	read-write	Yes	1
cabhCdpServerInterfaceMTU	read-write	Yes	1
cabhCdpServerVendorSpecific	read-write	Yes	1
cabhCdpServerLeaseTime	read-write	Yes	1

cabhCdpServerDhcpAddressType	read-write	Yes	1
cabhCdpServerDhcpAddress	read-write	Yes	1

cabhCtpMib

cabhCtpObjects

cabhCtpBase

cabhCtpSetToFactory	read-write	No	N/A

cabpCtpConnSpeed

cabhCtpConnSrcIpType	read-write	No	N/A
cabhCtpConnSrcIp	read-write	No	N/A
cabhCtpConnDestIpType	read-write	No	N/A
cabhCtpConnDestIp	read-write	No	N/A
cabhCtpConnProto	read-write	No	N/A
cabhCtpConnNumPkts	read-write	No	N/A
cabhCtpConnPktSize	read-write	No	N/A
cabhCtpConnTimeOut	read-write	No	N/A
cabhCtpConnControl	read-write	No	N/A
cabhCtpConnStatus	read-only	-	N/A
cabhCtpConnPktsSent	read-only	-	N/A
cabhCtpConnPktsRecv	read-only	-	N/A
cabhCtpConnRTT	read-only	-	N/A
cabhCtpConnThroughput	read-only	-	N/A

cabhCtpPing

cabhCtpPingSrcIpType	read-write	No	N/A
cabhCtpPingSrcIp	read-write	No	N/A
cabhCtpPingDestIpType	read-write	No	N/A
cabhCtpPingDestIp	read-write	No	N/A
cabhCtpPingNumPkts	read-write	No	N/A
cabhCtpPingPktSize	read-write	No	N/A
cabhCtpPingTimeBetween	read-write	No	N/A
cabhCtpPingTimeOut	read-write	No	N/A
cabhCtpPingControl	read-write	No	N/A
cabhCtpPingStatus	read-only	-	N/A
cabhCtpPingNumSent	read-only	-	N/A
cabhCtpPingNumRecv	read-only	-	N/A
cabhCtpPingAvgRTT	read-only	-	N/A
cabhCtpPingMinRTT	read-only	-	N/A

cabhCtpPingMaxRTT	read-only	-	N/A
cabhCtpPingNumIcmpError	read-only	-	N/A
cabhCtpPingIcmpError	read-only	-	N/A

cabhQosMib

cabhPriorityQosMibObjects

cabhPriorityQosBase

cabhPriorityQosSetToFactory	read-write	No	N/A
cabhPriorityQosLastReset	read-only	No	N/A
cabhPriorityQosMasterTable/ cabhPriorityQosMasterEntry			
cabhPriorityQosMasterApplicationId	not-accessable	-	N/A
cabhPriorityQosMasterDefaultCHPriority	read-create	Yes	16
cabhPriorityQosMasterRowStatus	read-create	Yes	16
cabhPriorityQosBp cabhPriorityQosBpTable/cabhPriorityQosBpEntry			
cabhPriorityQosBpIndex	not-accessible	-	N/A
cabhPriorityQosBpIpAddrType	read-only	-	N/A
cabhPriorityQosBpIpAddr	read-only	-	N/A
cabhPriorityQosBpApplicationId	read-only	-	N/A
cabhPriorityQosBpDefaultCHPriority	read-only	-	N/A
cabhPriorityQosBpDestTable/cabhPriorityQosBp DestEntry			
cabhPriorityQosBpDestIndex	not-accessible	-	N/A
cabhPriorityQosBpDestIpAddrType	read-only	-	N/A
cabhPriorityQosBpDestIpAddr	read-only	-	N/A
cabhPriorityQosBpDestPort	read-only	-	N/A
cabhPriorityQosBpDestIpPortPriority	read-only	-	N/A
cabhPriorityQosPs cabhPriorityQosPsIfAttribTable/cabhPriorityQosP sIfAttribEntry			
cabhPriorityQosPsIfAttribIfNumPriorities	read-only	-	N/A
cabhPriorityQosPsIfAttribIfNumQueues	read-only	-	N/A

experimental

snmpUSMDHObjectsMIB [RFC 2786]

usmDHKeyObjects

usmDHPublicObjects

usmDHParamaters	read-write	No	N/A
usmDHUserKeyTable/usmDHUserKeyEntry			
usmDHUserAuthKeyChange	read-create	No	N/A
usmDHUserOwnAuthKeyChange	read-create	No	N/A
usmDHUserPrivKeyChange	read-create	No	N/A
usmDHUserOwnPrivKeyChange	read-create	No	N/A

usmDHKickstartGroup

usmDHKickstartTable/usmDHKickstartEntry			
usmDHKickstartIndex	not-accessible	-	N/A
usmDHKickstartMyPublic	read-only	-	N/A
usmDHKickstartMgrPublic	read-only	-	N/A
usmDHKickstartSecurityName	read-only	-	N/A

snmpV2

snmpModules

snmpMIB

snmpMIBObjects

snmpSet

snmpSetSerialNo	read-write	No	N/A

snmpFrameworkMIB [RFC 3411]

snmpEngine

snmpEngineID	read-only	Yes	1
snmpEngineBoots	read-only	Yes	1
snmpEngineTime	read-only	-	N/A

snmpEngineMaxMessageSize	read-only	-	N/A
snmpMPDMIB [RFC 3412]			
snmpMPDObjects			
snmpMPDStats			
snmpUnknownSecurityModels	read-only	-	N/A
snmpInvalidMsgs	read-only	-	N/A
snmpUnknownPDUHandlers	read-only	-	N/A
snmpTargetMIB [RFC 3413]			
snmpTargetObjects			
snmpTargetSpinLock	read-write	No	N/A
snmpTargetAddrTable/snmpTargetAddrEntry			
snmpTargetAddrName	not-accessible	-	N/A
snmpTargetAddrTDomain	read-create	No	N/A
snmpTargetAddrTAddress	read-create	No	N/A
snmpTargetAddrTimeout	read-create	No	N/A
snmpTargetAddrRetryCount	read-create	No	N/A
snmpTargetAddrTagList	read-create	No	N/A
snmpTargetAddrParams	read-create	No	N/A
snmpTargetAddrStorageType	read-create	No	N/A
snmpTargetAddrRowStatus	read-create	No	N/A
snmpTargetParamsTable/snmpTargetParamsEntry			
snmpTargetParamsName	not-accessible	-	N/A
snmpTargetParamsMPModel	read-create	No	N/A
snmpTargetParamsSecurityModel	read-create	No	N/A
snmpTargetParamsSecurityName	read-create	No	N/A
snmpTargetParamsSecurityLevel	read-create	No	N/A
snmpTargetParamsStorageType	read-create	No	N/A
snmpTargetParamsRowStatus	read-create	No	N/A
snmpUnavailableContexts	read-only	-	N/A
snmpUnknownContexts	read-only	-	N/A
snmpNotificationMIB [RFC 3413]			
snmpNotifyObjects			

snmpNotifyTable/snmpNotifyEntry			
snmpNotifyName	not-accessible	-	N/A
snmpNotifyTag	read-create	No	N/A
snmpNotifyType	read-create	No	N/A
snmpNotifyStorageType	read-create	No	N/A
snmpNotifyRowStatus	read-create	No	N/A
snmpNotifyFilterProfileTable/snmpNotifyFilterProfileEntry			
snmpNotifyFilterProfileName	read-create	No	N/A
snmpNotifyFilterProfileStorType	read-create	No	N/A
snmpNotifyFilterProfileRowStatus	read-create	No	N/A
snmpNotifyFilterTable/snmpNotifyFilterEntry			
snmpNotifyFilterSubtree	not-accessible	-	N/A
snmpNotifyFilterMask	read-create	No	N/A
snmpNotifyFilterType	read-create	No	N/A
snmpNotifyFilterStorageType	read-create	No	N/A
snmpNotifyFilterRowStatus	read-create	No	N/A
snmpUsmMIB [RFC 3414]			
usmStats			
usmStatsUnsupportedSecLevels	read-only	-	N/A
usmStatsNotInTimeWindows	read-only	-	N/A
usmStatsUnknownUserNames	read-only	-	N/A
usmStatsUnknownEngineIDs	read-only	-	N/A
usmStatsWrongDigests	read-only	-	N/A
usmStatsDecryptionErrors	read-only	-	N/A
usmUser			
usmUserSpinLock	read-write	No	N/A
usmUserTable/usmUserEntry			
usmUserEngineID	not-accessible	-	N/A
usmUserName	not-accessible	-	N/A
usmUserSecurityName	read-only	-	N/A
usmUserCloneFrom	read-create	No	N/A
usmUserAuthProtocol	read-create	No	N/A
usmUserAuthKeyChange	read-create	No	N/A
usmUserOwnAuthKeyChange	read-create	No	N/A
usmUserPrivProtocol	read-create	No	N/A
usmUserPrivKeyChange	read-create	No	N/A

usmUserOwnPrivKeyChange	read-create	No	N/A
usmUserPublic	read-create	No	N/A
usmUserStorageType	read-create	No	N/A
usmUserStatus	read-create	No	N/A

SNMP-VIEW-BASED-ACM-MIB [RFC 3415]

snmpVacmMIB

vacmMIBObjects

vacmContextTable/vacmContextEntry			
vacmContextName	read-only	-	N/A
vacmSecurityToGroupTable/vacmSecurityToGroupEntry			
vacmSecurityModel	not-accessible	-	N/A
vacmSecurityName	not-accessible	-	N/A
vacmGroupName	read-create	No	N/A
vacmSecurityToGroupStorageType	read-create	No	N/A
vacmSecurityToGroupStatus	read-create	No	N/A
vacmAccessTable/vacmAccessEntry			
vacmAccessContextPrefix	not-accessible	-	N/A
vacmAccessSecurityModel	not-accessible	-	N/A
vacmAccessSecurityLevel	not-accessible	-	N/A
vacmAccessContextMatch	read-create	No	N/A
vacmAccessReadViewName	read-create	No	N/A
vacmAccessWriteViewName	read-create	No	N/A
vacmAccessNotifyViewName	read-create	No	N/A
vacmAccessStorageType	read-create	No	N/A
vacmAccessStatus	read-create	No	N/A

vacmMIBViews

vacmViewSpinLock	read-write	No	N/A
vacmViewTreeFamilyTable/vacmViewTreeFamilyEntry			
vacmViewTreeFamilyViewName	not-accessible	-	N/A
vacmViewTreeFamilySubtree	not-accessible	-	N/A
vacmViewTreeFamilyMask	read-create	No	N/A
vacmViewTreeFamilyType	read-create	No	N/A
vacmViewTreeFamilyStorageType	read-create	No	N/A

vacmViewTreeFamilyStatus	read-create	No	N/A
snmpCommunityMIB [RFC 2576]			
snmpCommunityMIBObjects			
snmpCommunityTable/snmpCommunityEntry			
snmpCommunityIndex	not-accessible	-	N/A
snmpCommunityName	read-create	No	N/A
snmpCommunitySecurityName	read-create	No	N/A
snmpCommunityContextEngineID	read-create	No	N/A
snmpCommunityContextName	read-create	No	N/A
snmpCommunityTransportTag	read-create	No	N/A
snmpCommunityStorageType	read-create	No	N/A
snmpCommunityStatus	read-create	No	N/A
snmpTargetAddrExtTable/snmpTargetAddrExtEntry			
snmpTargetAddrTMask	read-create	No	N/A
snmpTargetAddrMMS	read-create	No	N/A
clabSecCertObject			
clabSrvcPrvdrRootCACert	read-only	-	N/A
clabCVCRootCACert	read-only	-	N/A
clabCVCCACert	read-only	-	N/A
clabMfgCVCCert	read-only	-	N/A

Annex B Format and Content for Event, SYSLOG and SNMP Trap

The table in this annex summarizes the format and content for local log event entries, syslog messages, and SNMP traps.

Each row in the table specifies an event that the PS must be capable of generating. These events are to be reported by the PS by any or all of the following three means: local event logging as implemented by the local event table in [RFC 2669], SYSLOG, and SNMP trap. The SYSLOG format is specified in Section 6.3.3.2.4.4 of this document and SNMP trap format is defined in this annex, following Table B-1.

The first and second columns of Table B-1 indicate in which stage the event happens. The third column indicates the priority assigned to the event. These priorities are the same as reported in the docsDevEvLevel object in [RFC 2669] and in the LEVEL field of a syslog message.

The fourth column specifies the event text, which is reported in the docsDevEvText object of the [RFC 2669] and the text field of a syslog message. The fifth column provides additional information about the event text of the fourth column. For example, some of the event text fields are constants and some event text fields include variable information. Some of the variables are only required in the SYSLOG, as described in the fifth column. The sixth column specifies the error code set.

The seventh column indicates an unique identification number for the event, which is assigned to the docsDevEvId object and the <eventId> field of a syslog message. The eighth column specifies the SNMP trap, which notifies this event to a SNMP event receiver.

The rules to uniquely generate an event ID from the error code are described in Section 6.3.3.2.4.4. The event IDs in the table are in decimal format.

To better illustrate the table, the following is an example using the first row in the section of Software Upgrade events.

The first and second columns are “SW Upgrade” and “SOFTWARE UPGRADE INIT”. The event priority is “Notice.” The event text is “Software Download INIT - Via NMS”. The fifth column reads “For SYSLOG only, append: MAC addr: <P1> P1 = PS Mac Address”. This is a note about the SYSLOG. That is to say, the syslog text body will be like “Software Download INIT - Via NMS - MAC addr: x1 x2 x3 x4 x5 x6”.

The last column “TRAP NAME” is cabhPsDevSwUpgradeInitTrap, the format for which is given at the end of this annex.

Table B-1 Defined Events for IPCable2Home

PROCESS	SUB-PROCESS	PS PRIORITY	EVENT TEXT	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DHCP Errors before provisioning complete							
Init	CDC	Critical	DHCP FAILED - Discover sent, no offer received		D01.0	68000100	
Init	CDC	Critical	DHCP FAILED - Request sent, No response		D02.0	68000200	
Init	CDC	Critical	DHCP FAILED - Requested Info not supported.		D03.0	68000300	
Init	CDC	Error	DHCP ERROR - Response does not contain ALL the valid fields OR the PS is unable to determine provisioning mode		D03.1	68000301	
Init	CDC	Warning	DHCP ERROR - Unable to obtain all WAN-Data IP addresses the PS was configured to obtain		P02.0	68000302	cabhPsDevCdpWanDataIpTrap
TOD Errors before provisioning complete							

Init	TOD	Warning	TOD Request sent - no response received		D04.1	68000401	cabhPsDevInitTrap
Init	TOD	Warning	TOD Response received - invalid data format		D04.2	68000402	cabhPsDevInitTrap
TFTP Errors before provisioning complete							
Init	TFTP	Error	TFTP failed - Request sent - No Response		D05.0	68000500	cabhPsDevInitTrap (Trap is relevant for SNMP Prov Mode only.)
Init	TFTP	Error	TFTP failed - configuration file NOT FOUND	For SYSLOG only: append: File name = <P1> P1 = requested file name	D06.0	68000600	cabhPsDevInitTrap (Trap is relevant for SNMP Prov Mode only.)
Init	TFTP	Error	TFTP Failed - OUT OF ORDER packets		D07.0	68000700	cabhPsDevInitTrap (Trap is relevant for SNMP Prov Mode only.)
Init	TFTP	Error	TFTP file complete - but failed SHA-1 hash check	For SYSLOG only: append: File name = <P1> P1 = filename of TFTP file	D08.0	68000800	cabhPsDevInitTrap (Trap is relevant for SNMP Prov Mode only.)

Init	TFTP	Error	TFTP Failed Exceeded maximum number of retries	For Syslog only: append: Retry limit = <P1> P1 = maximum number of retries	D09.0	68000900	cabhPsDevInitTrap (Trap is relevant for SNMP Prov Mode only.)
TFTP Success							
Init	TFTP	Notice	TFTP success		D10.0	68001000	
TLS							
Init	TCP/IP	Critical	PS failed to connect to HTTP/TLS server		D20.0	68002000	
Init	TLS	Critical	TLS Connection timed out and maximum number of retries exceeded		D21.0	68002100	
Init	TLS	Critical	TLS FATAL ERROR <P1>	P1= Error code from [RFC 2246]	D22.0	68002200	
HTTP							
Init	HTTP	Critical	Configuration File Download failed, but will retry. HTTP Error. <P1>	P1= Status codes from [RFC 2616]	D30.0	68003000	
Init	HTTP	Critical	Configuration file download failed. Due to connection timed out and maximum number of retries. Operation aborted.		D31.0	68003100	

Init	HTTP	Critical	Secure Configuration file download successfully completed.		D32.0	68003200	
TLV Parsing							
Init	TLV PARSING	Warning	TLV-28 - unrecognized OID		I401.0	73040100	cabhPsDevInitTLVUnknownTrap
Init	TLV PARSING	Warning	Unknown TLV <P1>	For SYSLOG only, <P1> = the complete TLV in hexadecimal	I401.1	73040101	cabhPsDevInitTLVUnknownTrap
Init	TLV PARSING	Error	Invalid TLV Format/contents <P1>	For SYSLOG only, <P1> = the complete TLV in hexadecimal	I401.2	73040102	
Provisioning							
Init	Provisioning Complete	Notice	Provisioning complete	For SYSLOG only, append MAC Addr: <P1>. P1 = PS MAC address	I11.0	73001100	cabhPsDevInitTrap

SW UPGRADE INIT*							
SW Upgrade	SW UPGRADE INIT	Notice	SW Download INIT - Via NMS	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E101.0	69010100	cabhPsDevSwUpgradeInitTrap

SW Upgrade	SW UPGRADE INIT	Notice	SW Download INIT - Via Config file <P1>	P1 = CM config file name For SYSLOG only, append: SW file: <P2> - SW server: < P3>. P2 = SW file name and P3 = Tftp server IP address	E102.0	69010200	cabhPsDevSwUpgradeInitTrap
SW UPGRADE GENERAL FAILURE*							
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW Upgrade Failed during download - Max retry exceed (3)	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E103.0	69010300	cabhPsDevSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW Upgrade Failed Before Download - Server not Present	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E104.0	69010400	cabhPsDevSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW upgrade Failed before download - File not Present	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = TFTP server IP address	E105.0	69010500	cabhPsDevSwUpgradeFailTrap

SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW upgrade Failed before download - TFTP Max Retry Exceeded	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = TFTP server IP address	E106.0	69010600	cabhPsDevSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW upgrade Failed after download - Incompatible SW file	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E107.0	69010700	cabhPsDevSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	SW upgrade Failed after download - SW File corruption	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = TFTP server IP address	E108.0	69010800	cabhPsDevSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Disruption during SW download - Power Failure	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E109.0	69010900	cabhPsDevSwUpgradeFailTrap
SW UPGRADE SUCCESS*							

SW Upgrade	SW UPGRADE SUCCESS	Notice	SW download Successful - Via NMS	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E111.0	69011100	cabhPsDevSwUpgradeSuccessTrap
SW Upgrade	SW UPGRADE SUCCESS	Notice	SW download Successful - Via Config file	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E112.0	69011200	cabhPsDevSwUpgradeSuccessTrap
DHCP failure after provisioning complete							
DHCP	CDC	Error	DHCP RENEW sent - No response		D101.0	68010100	cabhPsDevDHCPFailTrap
DHCP	CDC	Error	DHCP REBIND sent - No response		D102.0	68010200	cabhPsDevDHCPFailTrap
DHCP	CDC	Error	DHCP RENEW sent - Invalid DHCP option		D103.0	68010300	cabhPsDevDHCPFailTrap
DHCP	CDC	Error	DHCP REBIND sent - Invalid DHCP option		D104.0	68010400	cabhPsDevDHCPFailTrap
TOD failure after provisioning complete							

TOD	TOD	Warning	TOD Request sent - no response received		D04.3	68000403	cabhPsDevTODFailTrap
TOD	TOD	Warning	TOD Response received - invalid data format		D04.4	68000404	cabhPsDevTODFailTrap
VERIFICATION OF CODE FILE							
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Improper Code File Controls	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E201.0	69020100	cabhPsDevSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Code File Manufacturer CVC Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E202.0	69020200	cabhPsDevSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Code File Manufacturer CVS Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E203.0	69020300	cabhPsDevSwUpgradeFailTrap

SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Code File Co-Signer CVC Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E204.0	69020400	cabhPsDevSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error	Code File Co-Signer CVS Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E205.0	69020500	cabhPsDevSwUpgradeFailTrap
VERIFICATION OF CVC							
SW Upgrade	VERIFICATION OF CVC	Error	Improper Configuration File CVC Format - TFTP Server: <P1> - Config File: <P2>	P1 = TFTP Server IP Address P2 = Config File Name	E206.0	69020600	cabhPsDevSwUpgradeCVCFailTrap
SW Upgrade	VERIFICATION OF CVC	Error	Configuration File CVC Validation Failure - TFTP Server: <P1> - Config File: <P2>	P1 = TFTP Server IP AddressP2 = Config File Name	E207.0	69020700	cabhPsDevSwUpgradeCVCFailTrap

SW Upgrade	VERIFICATION OF CVC	Error	Improper SNMP CVC Format - Snmp manager: <P1>	P1= IP Address of SNMP Manager	E208.0	69020800	cabhPsDevSwUpgradeCVCFailTrap
SW Upgrade	VERIFICATION OF CVC	Error	SNMP CVC Validation Failure - Snmp manager: <P1>	P1=IP Addr of SNMP manager	E209.0	69020900	cabhPsDevSwUpgradeCVCFailTrap
CDP Events							
CDP	CDS	Notice	Attempt to allocate more LAN TRANS IP addresses than allowed		P01.0	80000100	cabhPsDevCDPThresholdTrap
CDP	CDS	Notice	Unable to provision DHCP LAN client- IP address pool exhausted		P03.0	80000300	cabhPsDevCdpLanIpPoolTrap
CSP Events							
CSP	Firewall	Notice	Firewall Type 1 Enabled <P1> MIB Value	P1=value of cabhSecFwEventType1 Enable	P101.1	80010101	cabhPsDevCSPTrap
CSP	Firewall	Notice	Firewall Type 2 Enabled <P1>MIB Value	P1=value of cabhSecFwEventType2 Enable	P101.2	80010102	cabhPsDevCSPTrap
CSP	Firewall	Notice	Firewall Type 3 Enabled <P1>MIB Value	P1=value of cabhSecFwEventType3 Enable	P101.3	80010103	cabhPsDevCSPTrap
CSP	Firewall	Notice	Firewall Type 4 Enabled <P1>MIB Value	P1=value of cabhSecFwEventType4 Enable	P101.4	80010104	cabhPsDevCSPTrap

CSP	Firewall	Notice	Firewall Type 5 Enabled <P1>MIB Value	P1=value of cabhSecFw EventType5 Enable	P101.5	80010105	cabhPsDevCSPTrap
CSP	Firewall	Notice	Firewall Type 6 Enabled <P1>MIB Value	P1=value of cabhSecFw EventType6 Enable	P101.6	80010106	cabhPsDevCSPTrap
CSP	Firewall	Warning	Firewall Type 1 event threshold exceeded		P102.1	80010201	cabhPsDevCSPTrap
CSP	Firewall	Warning	Firewall Type 2 event threshold exceeded		P102.2	80010202	cabhPsDevCSPTrap
CSP	Firewall	Warning	Firewall Type 3 event threshold exceeded		P102.3	80010203	cabhPsDevCSPTrap
CSP	Firewall	Warning	Firewall Type 4 event threshold exceeded		P102.4	80010204	cabhPsDevCSPTrap
CSP	Firewall	Warning	Firewall Type 5 event threshold exceeded		P102.5	80010205	cabhPsDevCSPTrap
CSP	Firewall	Warning	Firewall Type 6 event threshold exceeded		P102.6	80010206	cabhPsDevCSPTrap
CSP	Firewall TFTP	Critical	TFTP download of firewall policy file failed: request sent, no response	P1 = requested firewall policy file URL	P130.0	80013000	cabhPsDevCSPTrap
CSP	Firewall TFTP	Critical	TFTP failed - firewall policy file not found	P1 = requested firewall policy file URL	P131.0	80013100	cabhPsDevCSPTrap

CSP	Firewall TFTP	Critical	TFTP failed - invalid firewall policy file	P1 = requested firewall policy file URL	P132.0	80013200	cabhPsDevCSPTrap
CSP	Firewall TFTP	Critical	Firewall policy file download complete but failed SHA-1 has check	P1 = requested firewall policy file URL, P2 = firewall policy file has value	P133.0	80013300	cabhPsDevCSPTrap
CSP	Firewall TFTP	Critical	Firewall policy file download exceeded maximum allowable number of TFTP retries	P1 = requested firewall policy file URL	P134.0	80013400	cabhPsDevCSPTrap
CSP	Firewall TFTP	Notice	Firewall policy file TFTP download success	P1 = requested firewall policy file URL For SYSLOG only: append: Retry limit = <P2> P2 = maximum allowable number of retry attempts	P135.0	80013500	cabhPsDevCSPTrap
CAP Events							
CAP	C-NAT	Warning	CAP unable to make C-NAT mapping. No WAN-data IP address available		P201.0	80020100	cabhPsDevCAPTrap
CAP	C-NAPT	Warning	CAP unable to make C-NAPT mapping. No WAN IP address		P250.0	80025000	cabhPsDevCAPTrap

			available				
CTP Events							
CTP	Connection Speed Tool	Notice	Connection Speed Tool test completed successfully	P1 = IP address of source P2 = IP address of destination P3 = protocol P4 = throughput	P301.0	80030100	cabhPsDevCtpTrap
CTP	Connection Speed Tool	Notice	Connection Speed Tool test timed out	P1 = IP address of source P2 = IP address of destination P3 = protocol P4 = value of timer (millisec)	P302.0	80030200	cabhPsDevCtpTrap
CTP	Connection Speed Tool	Notice	Connection Speed Tool test aborted	P1 = IP address of source P2 = IP address of destination P3 = protocol P4 = value of timer (millisec)	P303.0	80030300	cabhPsDevCtpTrap
CTP	Ping Tool	Notice	Ping Tool test completed successfully	P1 = IP address of source P2 = IP address of destination P3 = average round trip time	P320.0	80032000	cabhPsDevCtpTrap

CTP	Ping Tool	Notice	Ping Tool test timed out	P1 = IP address of source P2 = IP address of destination P3 = number of requests sent P4 = number of responses received	P321.0	80032100	cabhPsDevCtpTrap
CTP	Ping Tool	Notice	Ping Tool test aborted	P1 = IP address of source P2 = IP address of destination P3 = number of requests sent P4 = number of responses received	P322.0	80032200	cabhPsDevCtpTrap

Notes to Table B-1:

Software upgrade (secure software download) events apply to stand-alone Portal Services only. Software upgrade is controlled by the DOCSIS cable modem in an embedded PS, so software upgrade event reporting is managed by the cable modem in an embedded PS. For more information, refer to Section 11.3.6.4.1 Software Download Into Embedded or Standalone PS Elements.

B.1 Trap Descriptions

All specified traps are defined in the PS DEV MIB specification, [Annex E.4].

Annex C Security Threats & Preventative Measures

When developing a security technology, it is important to understand what the primary threats for a given application or environment. This information can then be used to select the most effective security tools and technologies for protection and prevention against malicious attacks.

The following primary home networking security threats to subscribers and system operators have been identified:

Theft of Service: Theft of service comes in two forms; unauthorized access to cable services and unauthorized duplication of service content.

Unauthorized access involves a subscriber or 3rd party (such as a neighbor) having access to cable services for which they have not paid. Devices could be "cloned" or modified to appear as a qualified device on the subscriber's home network. This could also degrade service delivery performance as these devices consume additional transport resources on the HFC and home networks.

Unauthorized duplication usually involves a subscriber or 3rd party (such as a neighbor) making illegal copies of service content. In some cases, these copies are distributed to other consumers without the approval of the operator or content provider.

Denial of Service (DOS) Attacks: Denial of service attacks can occur when a 3rd party entity (attacker, disgruntled customer, etc.) disrupts the normal communication and delivery of services between operators and their subscribers. Offending data transmissions coming from what appears to be a valid device/ source, could be injected into the home network and severely degrade its normal functions. These offending data transmissions could also extend to the operator's HFC network causing performance problems there.

Service Confidentiality: The service confidentiality threat involves a 3rd party (neighbors, attacker, etc.) monitoring/receiving information about a subscriber and the services they use. This could result in passwords or device configuration information being stolen, allowing attackers to gain further access to a subscriber's network resources and confidential files/data.

There are a number of different methods that can be used to prevent the home network security threats mentioned above. Unfortunately, one method cannot prevent them all, but a combination may be the best line of defense. The following preventative measures can be used:

Authentication: Authentication involves the verification that the sending and receiving entities are as claimed. This includes the service source, the receiving device, and the subscriber.

Authentication helps prevent theft of service by validating end devices and users, but it does not prevent content from being illegally copied or, prevent unauthorized access by 3rd parties who are monitoring the link. It does do a good job at preventing DOS attacks because traffic can be rejected if it does not come from a valid source. By itself, authentication does not provide any service confidentiality support, encryption must be used.

Copy Protection: Copy protection methods limit the ability of a receiving device to make unauthorized copies of service content.

Copy protection helps prevent theft of service by limiting how many copies can be made, but it does not prevent unauthorized access to services. It also does not prevent DOS or service confidentiality protection. In general, this preventive measure is implemented at higher application layers.

Data Encryption: Data encryption prevents the unauthorized disclosure/access of data.

Data encryption does an excellent job at providing data confidentiality and protection against theft of service. Encryption prevents making data unable to read without the correct decrypting key. However, it does not validate the source/receiving entities and it does not provide copy protection after the data has been decrypted. It also does not prevent DOS attacks.

Firewall: Firewall applications prevent network traffic from passing from one domain to another, unless it meets certain criteria set by the subscriber or operator. In home networks, firewalls are typically located on residential gateway devices that connect the HFC network to the home network.

A firewall application helps prevent DOS attacks and confidentiality attacks from the wide-area network (WAN) side of the firewall, but it does not prevent these kind of attacks coming from the home network side of the firewall. It also does not provide theft of service protection.

Management Message Security: This method of prevention involves authentication and encryption of network management messages only. Network management messages are used for device configuration, network monitoring/control, service provisioning, and Quality of Service (QoS) reservations.

Management message security provides a good mechanism to prevent DOS attacks by authenticating and encrypting management messages. Subscriber 's personal and network configuration information is also protected from confidentiality attacks, but service content is not. Also, management message security does not prevent theft of service content by unauthorized entities.

Annex D Applications Through CAT and Firewall

In the normal operation of address translation and firewall functionality, a number of protocols and applications may be prohibited from working as expected. Firewalls may purposely filter out certain applications and protocols for security purposes. The firewall policy can be explicitly set by the cable operator to allow as many ports to be opened as needed by the customer without opening ports that are not needed for communication between the LAN and WAN. Limiting the open ports and session initiation between the LAN and WAN may provide protection to the home LAN from attacks. If the ports are not allowed to be opened by the firewall policy, an attacker can not use these ports to attack the LAN. The purpose of this Annex is to provide a minimum level of support for commonly used applications under specific scenarios, and to assist the cable operator with common port configuration.

[RFC 3280], Network Address Translator (NAT)-Friendly Application Design Guidelines, outlines a number of guidelines for creating applications in such a manner that they will not be compromised when running in the presence of Network Address Translation functionality. It is strongly recommended that developers of applications that will run within a IPCable2Home environment adhere to these guidelines.

The existence of NAT and Firewall functionality are known to disrupt a number of protocols and applications when the end nodes/hosts are not in the same address realm and must traverse an IP Network Address Translator (NAT/CAT) and/or Firewall enroute to bridge the realms. In many cases, the CAT and Firewall can not provide the application and protocol transparency desired without the assistance of an Application Level Gateway (ALG). This Recommendation assumes an ALG is implemented in the Residential Gateway that enables applications listed within this Annex to work through the CAT.

Applications though the firewall are described in terms of protocol, specific port numbers, LAN-WAN relationship scenarios and addressing realms. The protocols are divided into two tables; one table is to list the protocols which can be managed by policy alone and is labeled Applications Requiring Firewall Policy Exclusively; the second table is to list the protocols which can only be managed with the combination of policy and ALGs and is labeled Applications Requiring Firewall Policy and an ALG.

According to the policy specified within Section 11 of this document, the tables contain information comments for the reader to be able to map the required applications to those with particular policy requirements for IPCable2Home and IPCablecom. IPCable2Home requires factory default settings for the ports to be opened through the firewall for normal Residential gateway operations. The items marked with IPCablecom in the comments column will be included, in addition to the factory defaults in enable IPCablecom through the firewall. The firewall settings to enable IPCablecom are listed in the comments column of each table and are specified within Section 11 in the configuration file section.

In addition to the specified applications, the PS SHOULD support online gaming applications through the CAT and firewall. Online gaming is considered a typical user application. However, this Recommendation does not specify games, as gaming is a dynamic industry and the online game ports depend upon the current popularity of particular games.

D.1 Relationship Scenarios

The specific scenarios may define the number of hosts communicating with each other through the PS, along with the requirements for each protocol and application. Each application/protocol and specific scenario requires support of the CH CAT and firewall to function correctly. The scenarios include an “xxx to xxx” definition that indicates the number of LAN hosts communicating to WAN hosts (ex. “One to Many” defines One LAN host communicating with Many WAN hosts concurrently.). These scenarios include:

- “One to One” relationship for a single instance
- “One to One” relationship for multiple instances (the number of required instances may be identified)
- “One to Many” relationship for a single instance
- “One to Many” relationship for multiple instances (the number of required instances may be identified)
- “Many to One” relationship for a single instance
- “Many to One” relationship for multiple instances (the number of required instances will be identified if necessary)

Note: The "Many to Many" scenario will be the same as a "One to One" relationship for multiple instances, a "One to Many" relationship for multiple instances, and/or a "Many to One" relationship for multiple instances.

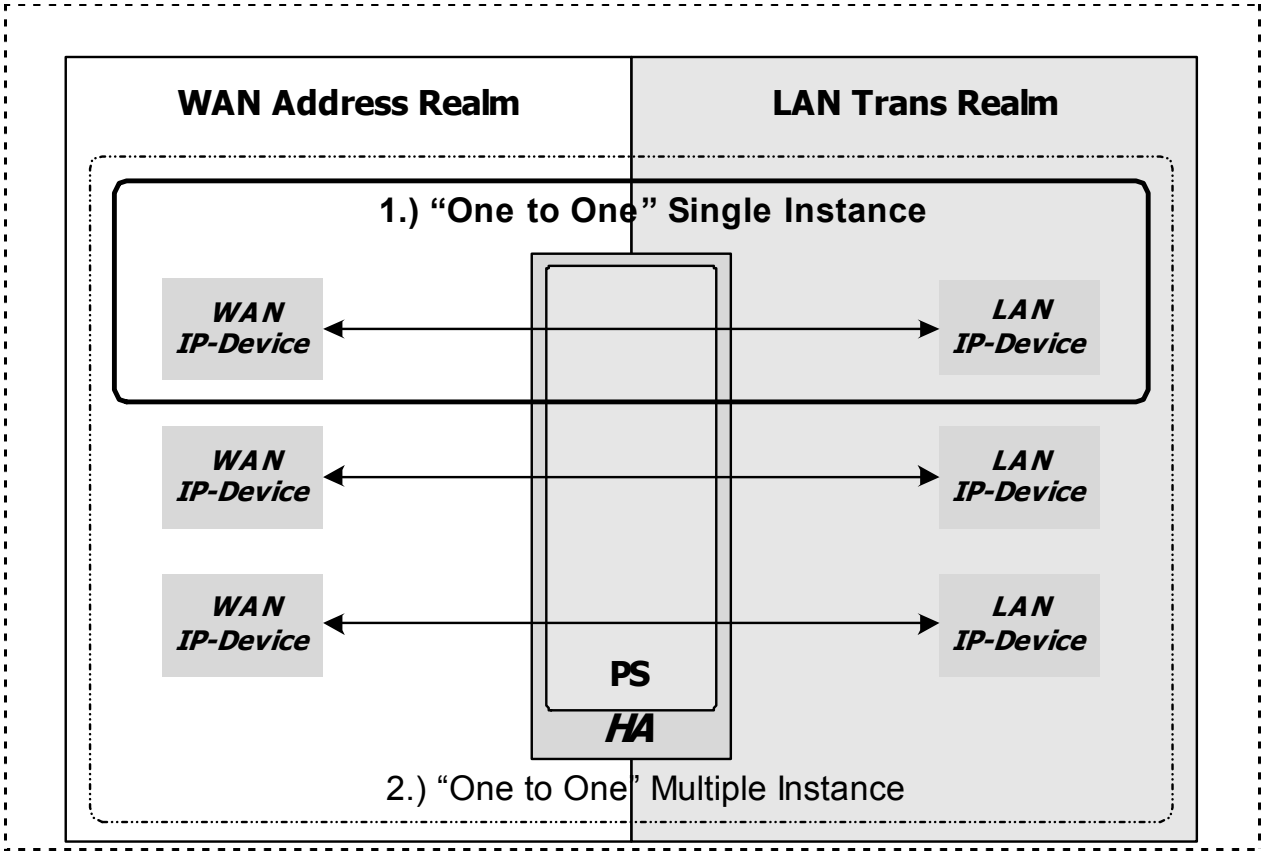

Figure D-1 "One to One" Scenarios

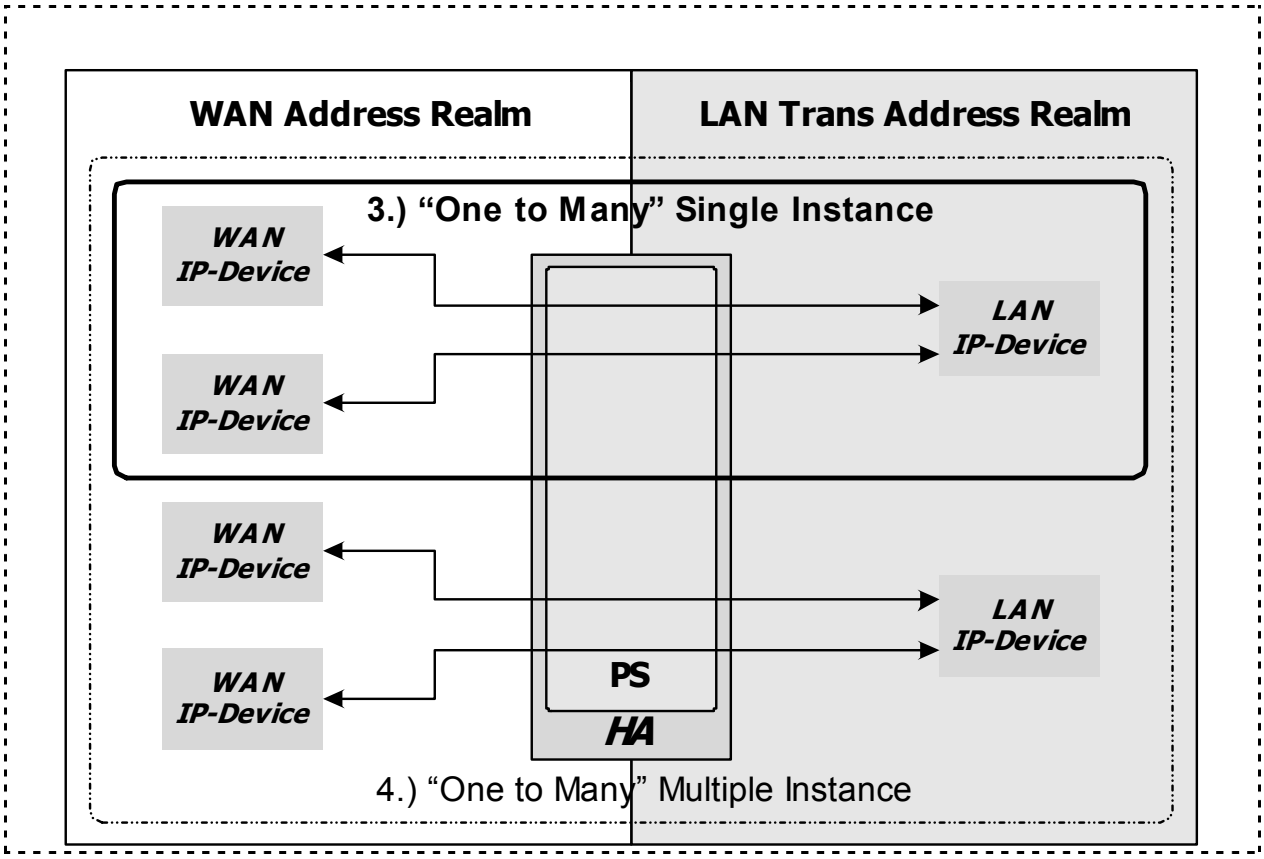

Figure D-2 "One to Many" Scenarios

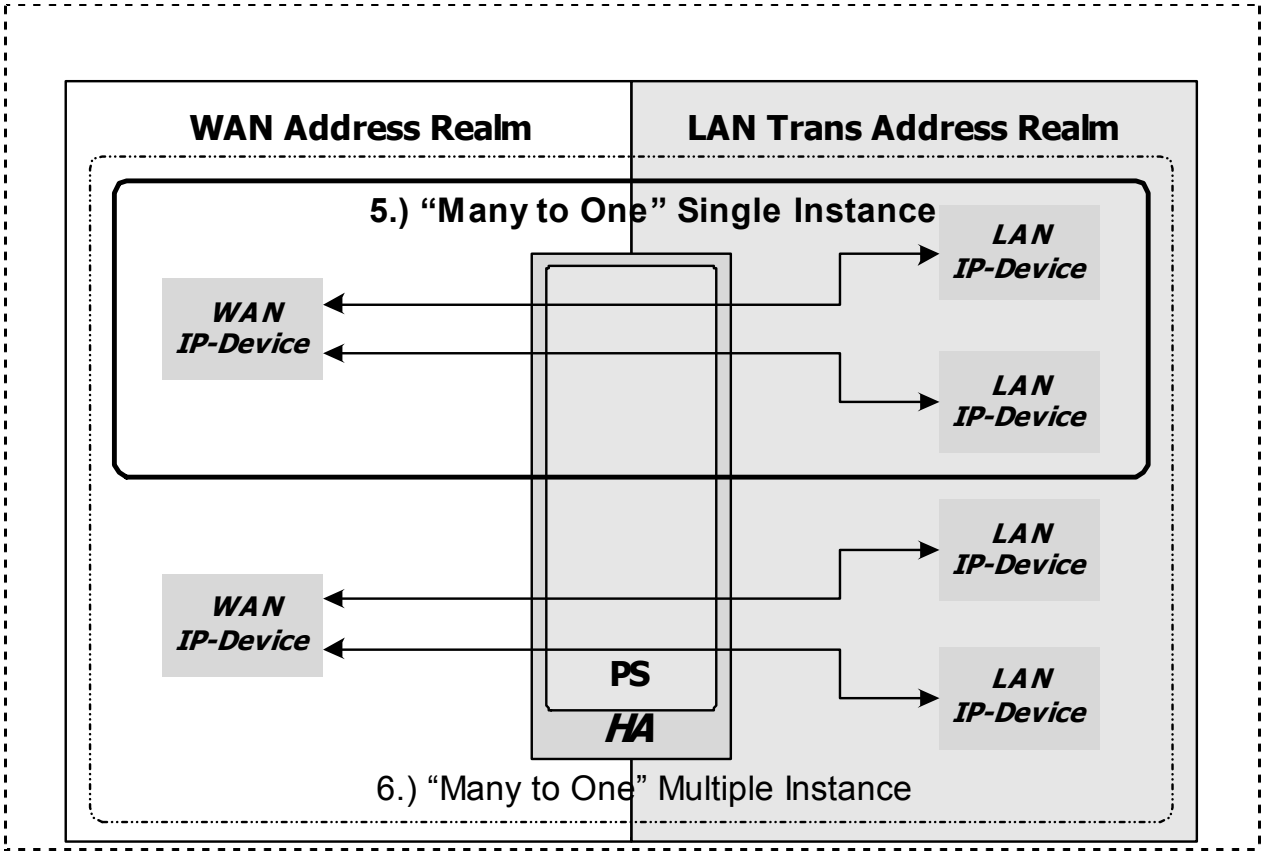

Figure D-3 "Many to One" Scenarios

D.2 Applications Requiring Firewall Policy Exclusively

Table D-1 and Table D-2 identify the applications and protocols that MUST be supported through the CAT and Firewall. This does not preclude the support of additional applications and protocols. A CAT/Firewall that can support these applications and protocols will be able to support most other applications and protocols that do not embed address, port, or other information affected by network address translation, and do not negotiate inbound sessions.

The following list of protocols and applications in Table D-1 MUST work through CAT and Firewall implementations. The firewall MUST NOT begin operations until after the provisioning complete message is sent by the PS, therefore the protocols needed for the PS to provision are not noted in this table.

Note: Applications only requiring Firewall policy configuration exclusively MUST be supported in all six (6) relationship scenarios unless noted in the comments column.

Table D-1 Protocols required to work through CAT and CH Firewall

Application / Protocol	Ports	Comments
AOL IM	TCP/5190, 5191, 5192, 5193 & 13784	Internet Default
CU-SeeMe	TCP/7648, 7649; UDP/7648, 7649, 24032	
DHCP		Internet Default
DNS	UDP/53	IPCablecom and IPCable2Home
FTPS	989 & 990	
HTTP	TCP/80	Internet Default
HTTPS	TCP/443	Internet Default
IGMP and IP Multicast		CH 1.0 Annex requirement
imap	143	
imap3	220	
IPSec	IKE > UDP/500 - ESP > raw IP/50	IKE key exchange, Tunnel mode, one to one single instance (CAT support key) IKE key exchange, Transport mode, one to one single instance (Passthrough mode) IPCablecom & LAN Peer Passthrough mode
IRC	TCP/6665-6669	
Kerberos	1293	IPCablecom and IPCable2Home PS Address Realm
L2TP	UDP/1701	
MediaPlayer (Windows)	TCP/ 80;1755	
Microsoft Messenger	3330 - 3332	Internet Default mcs-calypsoicf 3330 mcs-messaging 3331 mcs-mailsvr 3332
MGCP	2427, 2727	IPCablecom
Peer to Peer (eDonkey)	TCP/4662 UDP/4665	eDonkey

Peer to Peer (FastTrack P2P Protocol)	TCP/1214	KaZaA, Grokster, etc.
Peer to Peer (Gnutella P2P Protocol)	TCP/6346	Gnutella, LimeWire, BearShare, Morpheus, etc.
Peer to Peer (WinMX)	TCP/6699 UDP/6257	WinMX
PING ICMP Echo Request	raw IP/1	IPCable2Home
POP3	TCP/110	Internet Default
PPTP	Control Port > TCP/1723 & GRE > raw IP/47	
RealAudio/RealMedia	TCP: 80;443;554	
RSVP		IPCablecom
RTSP	TCP/554	
RTCP		IPCablecom
RTP		IPCablecom
SMTP	TCP/25	Internet Default
SNMP	TCP/161 UDP/161	IPCable2Home PS Address Realm and IPCablecom
SNMP trap	TCP/162 UDP/162	IPCable2Home PS Address Realm and IPCablecom
SSH	TCP/22 UDP/22	Internet Default
Syslog	UDP/514	IPCable2Home PS Address Realm and IPCablecom
Telnet	UDP/23	Outbound session requests. Internet Default
TFTP	UDP/69	IPCablecom
Traceroute	raw IP/1	Internet Default Reply from all hops between source and destination must be supported
Yahoo Messenger	TCP: 5050, 80 or any available	Internet Default

Note: Some port numbers listed in this section were previously unassigned by IANA, but have been recently assigned and now belong to another application. RTP & Quicktime both list 6970 - 6999, IANA has now assigned 6998 & 6999 to iatp-highpri and iatp-normalpri. IPCable2Home makes no attempt to correct his conflict.

D.3 Application Requiring Firewall Policy and an ALG

There are many cases in which the CAT and Firewall can not provide the application and protocol transparency desired. Since CAT modifies end node addresses (within the IP header of a packet) en-route, some applications are unable to function through the CAT without the assistance of an ALG.. Where possible, application specific ALGs MUST be used in conjunction with CAT and the appropriate Firewall policy to provide the desired application level transparency. The function of an ALG is application specific, so a list of applications, protocols and the scenarios that MUST be supported is found below.

Table D-2 Apps requiring Firewall policy and an ALG

Application / Protocol	Ports	(1) One to One Single	(2) One to One Multi	(3) One to Many Single	(4) One to Many Multi	(5) Many to One Single	(6) Many to One Multi	Comments
FTP	20/tcp, 21/tcp	X	X	X	X	X	X	
Microsoft Netmeeting (H.323)	TCP/389 ILS 522 ULS 1503 T.120 1720 call setup 1731 audio call ctrl Dynamic TCP call control Dynamic UDP 1024-65535 RTP over UDP	X	X	X	X	X	X	
MSN Messenger (H.323)	1863/tcp	X	X	X	X	X	X	Internet Default
Net2Phone	6801/udp (also calls for opening 2 additional unspecified ports UDPPORT=6801 UDPPORT=XXXX TCPPORT=XXXX The Network Administrator needs to make sure UDPPORT 6801 is open. For the other UDPPORT and TCPPORT, the administrator can use anything in the range from 1 - 30000.)	X	X	X	X			
Quicktime 5	RTSP/TCP/554 RTP/UDP 6970-6999	X	X	X	X	X	X	Supporting Quicktime without an ALG via port 80 provides less than optimal performance
Window Messenger (SIP)		X	X					Available on Windows XP only

Annex E MIBS

E.1 IPCable2Home Addressing Portal (CAP) MIB requirement.

Requirements

The CableHome™ CAP MIB MUST be implemented as defined below.

```
CABH-CAP-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32          FROM SNMPv2-SMI
    TEXTUAL-CONVENTION,
    TruthValue,
    RowStatus,
    PhysAddress         FROM SNMPv2-TC
    OBJECT-GROUP,
    MODULE-COMPLIANCE   FROM SNMPv2-CONF
    InetAddressType,
    InetAddress,
    InetPortNumber            FROM INET-ADDRESS-MIB
    clabProjCableHome   FROM CLAB-DEF-MIB;

--============================================================================
--
--    History:
--
--    Date        Modified by Reason
--    04/05/02                Issued I01
--    09/20/02                Issued I02
--    04/11/03                Issued I03
--
--============================================================================

cabhCapMib MODULE-IDENTITY
    LAST-UPDATED    "200304110000Z" --April 11, 2003
    ORGANIZATION    "CableLabs Broadband Access Department"
    CONTACT-INFO
            "Kevin Luehrs
            Postal: Cable Television Laboratories, Inc.
                  400 Centennial Parkway
                  Louisville, Colorado 80027-1266
                              U.S.A.
            Phone:  +1 303-661-9100
            Fax:    +1 303-661-9199
            E-mail: k.luehrs@cablelabs.com"
    DESCRIPTION
           "This MIB module supplies the basic management objects
           for the CableHome Addressing Portal (CAP) portion of
           the PS database.


        Acknowledgements:
        Roy Spitzer           -     Consultant to CableLabs
        Mike Mannette   -     Consultant to Cable Labs
        Randy Dunton    -     Intel
        Dmitrii Loukianov     -     Intel
        Itay Sherman    -     Texas Instruments
        Chris Zacker    -     Broadcom
```

```
        Rick Vetter           -     Consultant to Cable Labs
        John Bevilacqua -     YAS"
    ::=  { clabProjCableHome 3 }

-- Textual conventions

CabhCapPacketMode ::= TEXTUAL-CONVENTION
      STATUS      current
      DESCRIPTION
            "The data type established when
            a binding/mapping is established."
      SYNTAX      INTEGER {
            napt        (1),  -- NAT with port translation
            nat         (2),  -- Basic NAT
            passthrough (3)   -- Pass Through External Address
            }


cabhCapObjects      OBJECT IDENTIFIER ::= { cabhCapMib 1 }
cabhCapBase         OBJECT IDENTIFIER ::= { cabhCapObjects 1 }
cabhCapMap          OBJECT IDENTIFIER ::= { cabhCapObjects 2 }


--=================================================================
--
--    General CAP Parameters
--
--=================================================================

cabhCapTcpTimeWait OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
    "This object is the maximum inactivity time to wait before assuming
    TCP session is terminated. It has no relation to the TCP session
    TIME_WAIT state referred to in [RFC793]"
    DEFVAL { 300 }
    ::= { cabhCapBase 1 }


cabhCapUdpTimeWait OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS    "seconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
       "The inactivity time to wait before destroying
       CAP mappings for UDP."
      DEFVAL { 300 }  -- 5 minutes
      ::={ cabhCapBase 2 }

cabhCapIcmpTimeWait OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS "seconds"
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
       "The inactivity time to wait before destroying
       CAP mappings for ICMP."
      DEFVAL { 300 }  --  5 minutes
      ::= { cabhCapBase 3 }
```

```
cabhCapPrimaryMode OBJECT-TYPE
    SYNTAX  CabhCapPacketMode
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
    "The Primary Packet Handling Mode to be used."
    DEFVAL { napt }
    ::= { cabhCapBase 4 }

cabhCapSetToFactory OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Reading this object always returns false(2). When the
       cabhCapSetToFactory object is set to true(1), the PS must
       take the following actions:

1.    Clear all entries in the cabhCapMappingTable and
      cabhCapPassthroughTable.
2.    Reset the following objects to their factory default values:
               cabhCapTcpTimeWait,
               cabhCapUdpTimeWait,
               cabhCapIcmpTimeWait,
               cabhCapPrimaryMode"

     ::= { cabhCapBase 5 }

--=================================================================
--
--    cabhCapMappingTable (CAP Mapping Table)
--
--    The cabhCapMappingTable contains the info for all CAP mappings.
--
--=================================================================

cabhCapMappingTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CabhCapMappingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "This table contains IP address mappings between private network
addresses, or network addresses and port numbers/ICMP sequence numbers,
assigned to devices on the subscriber's home LAN, and network
addresses, or network addresses and port numbers/ICMP sequence number,
assigned by the cable operator, presumed to be on a separate subnetwork
than the private IP addresses. The CAP Mapping Table is used by the
CableHome Address Portal (CAP) function of the PS to make packet
forwarding decisions."
    ::= {  cabhCapMap 1 }

cabhCapMappingEntry OBJECT-TYPE
    SYNTAX      CabhCapMappingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "List of the private IP (LAN) address - to - cable operator
       assigned IP (WAN) address mappings stored in the PS and
       used by the PS to make packet forwarding decisions."
    INDEX { cabhCapMappingIndex }
    ::= { cabhCapMappingTable 1 }

      CabhCapMappingEntry ::= SEQUENCE {
      cabhCapMappingIndex           INTEGER,
      cabhCapMappingWanAddrType           InetAddressType,
```

```
      cabhCapMappingWanAddr         InetAddress,
      cabhCapMappingWanPort         InetPortNumber,
      cabhCapMappingLanAddrType           InetAddressType,
      cabhCapMappingLanAddr         InetAddress,
      cabhCapMappingLanPort         InetPortNumber,
      cabhCapMappingMethod          INTEGER,
      cabhCapMappingProtocol        INTEGER,
      cabhCapMappingRowStatus             RowStatus
      }

cabhCapMappingIndex    OBJECT-TYPE
    SYNTAX              INTEGER   (1..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The Index into the CAP Mapping Table."
    ::= { cabhCapMappingEntry 1 }

    cabhCapMappingWanAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The IP address type assigned on the WAN side"
    DEFVAL { ipv4 }
    ::= { cabhCapMappingEntry 2 }

cabhCapMappingWanAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
    "The IP address assigned by the cable operator's address (DHCP)
    server, and comprising the WAN-side IP address of the CAP
    Mapping tuple. This object is populated either dynamically by
    LAN-to-WAN outbound traffic or statically by the cable operator."
    ::= { cabhCapMappingEntry 3 }

cabhCapMappingWanPort OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The TCP/UDP port number or ICMP sequence number on the WAN
       side.  A port number of 0 indicates a NAT mapping.  A
       non-zero port number indicates an NAPT mapping."
        DEFVAL { 0 }
    ::= { cabhCapMappingEntry 4 }

cabhCapMappingLanAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The IP address type assigned on the LAN side."
    DEFVAL { ipv4 }
    ::= { cabhCapMappingEntry 5 }

cabhCapMappingLanAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The IP address assigned by the DHCP server function of the
        PS (CableHome DHCP Server, CDS), and comprising the
```

```
        LAN-side IP address of the CAP Mapping tuple.
        This object is populated either dynamically as a result of LAN
        -to-WAN outbound traffic or statically by the cable operator."
    ::= { cabhCapMappingEntry 6 }

cabhCapMappingLanPort OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
    "The TCP/UDP port number or ICMP sequence number on the LAN
    side. A port number/sequence number of 0 indicates a NAT mapping.
    A non-zero port number/sequence number indicates an NAPT mapping."
    DEFVAL { 0 }
    ::= { cabhCapMappingEntry 7 }

cabhCapMappingMethod OBJECT-TYPE
    SYNTAX      INTEGER {
            static   (1),
            dynamic (2)
            }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "Indicates how this mapping was created. Static means that it
       was provisioned, and dynamic means that it was handled by the
       PS itself."
    ::= { cabhCapMappingEntry 8 }

cabhCapMappingProtocol OBJECT-TYPE
    SYNTAX      INTEGER {
            other       (1),  -- any other protocol; e.g. IGMP
            icmp        (2),
            udp         (3),
            tcp         (4)
                        }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The protocol for this mapping."
    ::= { cabhCapMappingEntry 9 }

cabhCapMappingRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
"The RowStatus interlock for the creation and deletion of a
cabhCapMappingTable entry.  Changing the value of the IP
address or port number columns of the CAP Mapping Table may
have an effect on active traffic, so the PS will prevent modification
of this table's columns and return an inconsistentValue error when
cabhCapMappingRowStatus object is active(1).  The PS must not allow
RowStatus to be set to notInService(2) by a manager.  A newly created
row cannot be set to active(1) until the corresponding instances of
cabhCapMappingWanAddrType, cabhCapMappingWanAddr,
cabhCapMappingLanAddrType, cabhCapMappingLanAddr, and
cabhCapMappingProtocol have been set.  If Primary Packet-handling
Mode is NAPT (cabhCapPrimaryMode is napt(1)), a newly created row
can not be set to active(1) until a non-zero value of
cabhCapMappingWanPort and cabhCapMappingLanPort have been set.
If Primary Packet-handling Mode is NAT (cabhCapPrimaryMode is nat(2)),
a newly created row can not be set to active(1) if a non-zero value of
cabhCapMappingWanPort and cabhCapMappingLanPort have been set."
    ::={ cabhCapMappingEntry 10 }
```

```
--================================================================
--
--    cabhCapPassthroughTable (CAP Passthrough Table)
--
--    The cabhCapPassthroughTable contains the MAC Addresses for all
--    LAN-IP Devices which will be configured as passthrough.
--
--================================================================

cabhCapPassthroughTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CabhCapPassthroughEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "This table contains MAC addresses for LAN-IP Devices which are
    configured as passthrough mode."
    ::= {  cabhCapMap 2 }

cabhCapPassthroughEntry OBJECT-TYPE
    SYNTAX  CabhCapPassthroughEntry
    MAX-ACCESS    not-accessible
    STATUS  current
    DESCRIPTION
    "List of hardware addresses of LAN IP Devices which are configured
    for passthrough mode."
    INDEX {cabhCapPassthroughIndex}
::= {cabhCapPassthroughTable 1}

CabhCapPassthroughEntry::=SEQUENCE {
      cabhCapPassthroughIndex       INTEGER,
      cabhCapPassthroughMacAddr     PhysAddress,
      cabhCapPassthroughRowStatus   RowStatus
      }

cabhCapPassthroughIndex       OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    MAX-ACCESS    not-accessible
    STATUS  current
    DESCRIPTION
    "The index into the CAP Passthrough Table."
    ::= { cabhCapPassthroughEntry 1 }

cabhCapPassthroughMacAddr           OBJECT-TYPE
    SYNTAX  PhysAddress
    MAX-ACCESS    read-create
    STATUS  current
    DESCRIPTION
    "Hardware address of the LAN-IP Device to be configured as
    passthrough mode."
    ::={cabhCapPassthroughEntry 2}

cabhCapPassthroughRowStatus   OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
"The RowStatus interlock for the creation and deletion
of a cabhCapPassthroughTable entry.  Any writable object in each
row can be modified at any time while the row is active(1)."
    ::= { cabhCapPassthroughEntry 3 }
--
-- notification group is for future extension.
--
```

```
cabhCapNotification     OBJECT IDENTIFIER ::= { cabhCapMib 2 0 }
cabhCapConformance      OBJECT IDENTIFIER ::= { cabhCapMib 3 }
cabhCapCompliances      OBJECT IDENTIFIER ::= { cabhCapConformance 1 }
cabhCapGroups           OBJECT IDENTIFIER ::= { cabhCapConformance 2 }

--
--    Notification Group
--


-- compliance statements

cabhCapBasicCompliance MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
    "The compliance statement for devices that implement
    MTA feature."
    MODULE   --cabhCapMib


-- unconditionally mandatory groups

MANDATORY-GROUPS {
    cabhCapGroup
    }

::= { cabhCapCompliances 1 }


cabhCapGroup OBJECT-GROUP
    OBJECTS {
            cabhCapTcpTimeWait,
            cabhCapUdpTimeWait,
            cabhCapIcmpTimeWait,
            cabhCapPrimaryMode,
            cabhCapSetToFactory,
            cabhCapMappingWanAddrType,
            cabhCapMappingWanAddr,
            cabhCapMappingWanPort,
            cabhCapMappingLanAddrType,
            cabhCapMappingLanAddr,
            cabhCapMappingLanPort,
            cabhCapMappingMethod,
            cabhCapMappingProtocol,
            cabhCapMappingRowStatus,
            cabhCapPassthroughMacAddr,
            cabhCapPassthroughRowStatus
            }
    STATUS    current
    DESCRIPTION
        "Group of objects for CableHome CAP MIB."
    ::= { cabhCapGroups 1 }

END
```

E.2 IP Cable2Home DHCP Portal (CDP) MIB requirement.

requirements

The CableHome™ CDP MIB MUST be implemented as defined below.

```
CABH-CDP-MIB DEFINITIONS ::= BEGIN
IMPORTS
   MODULE-IDENTITY,
   OBJECT-TYPE,
   Integer32,
   Unsigned32                  FROM SNMPv2-SMI
   MacAddress,
   TruthValue,
   DateAndTime,
   RowStatus                   FROM SNMPv2-TC
   OBJECT-GROUP,
   MODULE-COMPLIANCE    FROM SNMPv2-CONF
   InetAddressType,
   InetAddress          FROM INET-ADDRESS-MIB
   SnmpAdminString            FROM SNMP-FRAMEWORK-MIB
   clabProjCableHome    FROM CLAB-DEF-MIB;

   --================================================================
   --
   -- History:
   --
   --      Date         Modified by         Reason
   --      04/05/02                               Issued I01
   --      09/20/02                               Issued I02
   --      10/25/02                               IETF I-D revisions
   --      04/11/03                               Issued I03
   --
   --================================================================

   cabhCdpMib MODULE-IDENTITY
       LAST-UPDATED    "200304110000Z"  -- April 11, 2003
    ORGANIZATION    "CableLabs Broadband Access Department"
    CONTACT-INFO
            "Kevin Luehrs
            Postal:     Cable Television Laboratories, Inc.
                  400 Centennial Parkway
                  Louisville, Colorado 80027-1266
                  U.S.A.
            Phone:  +1 303-661-9100
            Fax:    +1 303-661-9199
            E-mail: k.luehrs@cablelabs.com"
    DESCRIPTION
      "This MIB module supplies the basic management objects
for the CableHome DHCP Portal (CDP) portion of the PS database.


            Acknowledgements:
            Roy Spitzer       -     Consultant to CableLabs
            Mike Mannette           -     Consultant to CableLabs
            Randy Dunton            -     Intel
            Dmitrii Loukianov       -     Intel
            Itay Sherman            -     Texas Instruments
            Chris Zacker            -     Broadcom
            Rick Vetter       -     Consultant to CableLabs
            John Bevilacqua         -     YAS"
    ::=  { clabProjCableHome 4 }


cabhCdpObjects          OBJECT IDENTIFIER ::= { cabhCdpMib 1 }
cabhCdpBase             OBJECT IDENTIFIER ::= { cabhCdpObjects 1 }
cabhCdpAddr             OBJECT IDENTIFIER ::= { cabhCdpObjects 2 }
cabhCdpServer                 OBJECT IDENTIFIER ::= { cabhCdpObjects 3 }
```

```
--
--    The following group describes the base objects in the Cable Home
--    DHCP Portal.  The rest of this group deals addresses defined on
--    the LAN side.
--

cabhCdpSetToFactory OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
            "Reading this object always returns false(2). When the
            cabhCdpSetToFactory object is set to true(1), the PS must
            take the following actions:
            1. Clear all cabhCdpLanAddrEntries in the CDP LAN Address
                Table.
            2. The CDS must offer the factory default DHCP options
                at the next lease renewal time.
            3. Reset the following objects to their factory default
                values:
            cabhCdpLanTransThreshold,
            cabhCdpLanTransAction,
            cabhCdpWanDataIpAddrCount,
            cabhCdpLanPoolStartType,
            cabhCdpLanPoolStart,
            cabhCdpLanPoolEndType,
            cabhCdpLanPoolEnd,
            cabhCdpServerNetworkNumberType,
            cabhCdpServerNetworkNumber,
            cabhCdpServerSubnetMaskType,
            cabhCdpServerSubnetMask,
            cabhCdpServerTimeOffset,
            cabhCdpServerRouterType,
            cabhCdpServerRouter,
            cabhCdpServerDnsAddressType,
            cabhCdpServerDnsAddress,
            cabhCdpServerSyslogAddressType,
            cabhCdpServerSyslogAddress,
            cabhCdpServerDomainName,
            cabhCdpServerTTL,
            cabhCdpServerInterfaceMTU,
            cabhCdpServerVendorSpecific,
            cabhCdpServerLeaseTime,
            cabhCdpServerDhcpAddressType,
            cabhCdpServerDhcpAddress,
            cabhCdpServerCommitStatus"
::= { cabhCdpBase 1 }

cabhCdpLanTransCurCount OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The current number of active leases in the
             cabhCdpLanAddrTable (the number of row entries in the
             table that have a cabhCdpLanAddrMethod value of
             reservationActive(2) or dynamicActive (4)).  This count
             does not include expired leases or reservations not
             associated with a current lease."
    ::= { cabhCdpBase 2 }

cabhCdpLanTransThreshold OBJECT-TYPE
    SYNTAX INTEGER (0..65533)
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION
            "The threshold number of LAN-Trans IP addresses allocated
             or assigned above which the PS generates an alarm
             condition. Whenever an attempt is made to allocate a
             LAN-Trans IP address when cabhCdpLanTransCurCount is
             greater than or equal to cabhCdpLanTransThreshold, an
             event is generated. A value of 0 indicates that the CDP
             sets the threshold at the highest number of addresses in
             the LAN address pool."
DEFVAL {0 }
::= { cabhCdpBase 3 }

cabhCdpLanTransAction OBJECT-TYPE
    SYNTAX INTEGER {
    normal (1),
    noAssignment (2)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
            "The action taken when the CDS assigns a LAN-Trans
             address and the number of LAN-Trans addresses assigned
             (cabhCdpLanTransCurCount) is greater than the threshold
             (cabhCdpLanTransThreshold) The actions are as follows:
              normal - assign a LAN-Trans IP address as would
              normally occur if the threshold was not exceeded.
              noAssignment - do not assign a LAN-Trans IP address."
DEFVAL { normal }
::= { cabhCdpBase 4 }

cabhCdpWanDataIpAddrCount OBJECT-TYPE
    SYNTAX INTEGER ( 0..63 )
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
            "This is the number of WAN-Data IP addresses the
             PS's CDC must attempt to acquire via DHCP."
DEFVAL {0 }
::= { cabhCdpBase 5 }
--
--    CDP Address Management Tables
--
--=========================================================================
--
--    cabhCdpLanAddrTable (CDP LAN Address Table)
--
--    The cabhCdpLanAddrTable contains the DHCP parameters
--    for each IP address served to the LAN-Trans realm.
--
--    This table contains a list of entries for the LAN side CDP
--    parameters. These parameters can be set
--    either by the CDP or by the cable operator through the CMP.
--
--=========================================================================

cabhCdpLanAddrTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF CabhCdpLanAddrEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
             "This table is a list of LAN-Trans realm parameters.
             This table has one row entry for each allocated
             LAN-Trans IP address.  Each row must have at least a
             valid cabhCdpLanAddrMethod, a cabhCdpLanAddrIpType, a
             unique cabhCdpLanAddrIp, and a unique
```

```
            cabhCdpLanAddrClientId value.

            Static/Manual address assignment: To create a new DHCP
            address reservation, the NMS creates a row with: an
            index comprised of a new cabhCdpLanAddrIp and its
            cabhCdpLanAddrIpType, a new unique
            cabhCdpLanAddrClientID, (an empty LeaseCreateTime and
            empty LeaseExpireTime,) and a
            cabhCdpLanDataAddrRowStatus of createAndGo(4).  If the
            syntax and values of the new row - indicating a
            reservation - are valid, the PS must set
            cabhCdpLanAddrMethod to reservationInactive(1) and
            cabhCdpLanDataAddrRowStatus to active(1).  When the PS
            grants a lease for a reserved IP, it must set the
            cabhCdpLanAddrMethod object for that row to
            reservationActive(2).  When a lease for a reserved IP
            expires, the PS must set the corresponding row's
            cabhCdpLanAddrMethod object to reservationInactive(1).
            For row entries that represent lease reservations - rows
            in which the cabhCdpLanAddrMethod object has a value of
            either reservationInactive(1) or reservationActive(2) -
            the cabhCdpLanAddrIpType, cabhCdpLanAddrIp,
            cabhCdpLanAddrClientID, cabhCdpLanAddrMethod, and
            cabhCdpLanAddrHostName object values must persist across
            PS reboots.

            Dynamic address assignment:  When the PS grants a lease
            for a non-reserved IP, it must set the
            cabhCdpLanAddrMethod object for that row to
            dynamicActive(4).  When a lease for a non-reserved IP
            expires, the PS must set the corresponding row's
            cabhCdpLanAddrMethod object to dynamicInactive(3).  The
            PS must create new row entries using cabhCdpLanAddrIp
            values that are unique to this table.  If all
            cabhCdpLanAddrIp values in the range defined by
            cabhCdpLanPoolStart and cabhCdpLanPoolEnd are in use in
            this table, the PS may overwrite the
            cabhCdpLanAddrClientId of a row that has a
            cabhCdpLanAddrMethod object with a value of
            dynamicInactive(3) with a new cabhCdpLanAddrClientId
            value and use that cabhCdpLanAddrIp as part of a new
            lease.  For row entries that represent active leases -
            rows in which the cabhCdpLanAddrMethod object has a
            value of dynamicActive(4) - the cabhCdpLanAddrIpType,
            cabhCdpLanAddrIp, cabhCdpLanAddrClientID,
            cabhCdpLanAddrMethod, and cabhCdpLanAddrHostName object
            values must persist across PS reboots."
    ::= {  cabhCdpAddr 1 }

cabhCdpLanAddrEntry OBJECT-TYPE
    SYNTAX      CabhCdpLanAddrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of general parameters pertaining to LAN-Trans IP
         address reservations and leases."
    INDEX { cabhCdpLanAddrIpType, cabhCdpLanAddrIp }
    ::= { cabhCdpLanAddrTable 1 }

CabhCdpLanAddrEntry ::= SEQUENCE {
   cabhCdpLanAddrIpType       InetAddressType,
   cabhCdpLanAddrIp                    InetAddress,
   cabhCdpLanAddrClientID              MacAddress,
   cabhCdpLanAddrLeaseCreateTime         DateAndTime,
   cabhCdpLanAddrLeaseExpireTime         DateAndTime,
```

```
      cabhCdpLanAddrMethod                INTEGER,
      cabhCdpLanAddrHostName              SnmpAdminString,
      cabhCdpLanAddrRowStatus             RowStatus
       }

   cabhCdpLanAddrIpType OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
               "The type of IP address assigned to the LAN IP Device
                 in the LAN-Trans Realm."
       ::= { cabhCdpLanAddrEntry 1 }

   cabhCdpLanAddrIp OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
       "The address assigned to the LAN IP Device. This parameter is
       entered by the CDP when the CDS grants a lease to a LAN IP
       Device in the LAN-Trans realm and creates a row in this table.
       Alternatively, this parameter can be entered by the NMS
       through the CMP, when the NMS creates a new DHCP address
       reservation.  Each cabhCdpLanAddrIp in the table must fall
       within the range of IPs defined inclusively by
       cabhCdpLanPoolStart and cabhCdpLanPoolEnd.  The PS must
       return an inconsistentValue error if the NMS attempts to
       create a row entry with a cabhCdpLanAddrIP value that falls
       outside of this range or is not unique from all existing
       cabhCdpLanAddrIP entries in this table. The address type of
       this object is specified by cabhCdpLanAddrIpType."
       ::= { cabhCdpLanAddrEntry 2 }

   cabhCdpLanAddrClientID OBJECT-TYPE
       SYNTAX           MacAddress
       MAX-ACCESS       read-create
       STATUS                 current
       DESCRIPTION
"The client's (i.e., LAN IP Device's) hardware address as indicated in
 the chaddr field of its DHCP REQUEST message. There is a one-to-one
relationship between the hardware address and the LAN IP Device. This
parameter is entered by the PS (CDP) when the CDS grants a lease to a
LAN IP Device in the LAN-Trans realm and creates a row in this table.
Alternatively this parameter can be created by the NMS through the CMP,
when the NMS creates a new DHCP address reservation by accessing the
cabhCdpLanDataAddrRowStatus object with an index comprised of a unique
cabhCdpLanAddrIp and creating a row with a unique
cabhCdpLanAddrClientID."
       ::= { cabhCdpLanAddrEntry 3 }

   cabhCdpLanAddrLeaseCreateTime OBJECT-TYPE
       SYNTAX      DateAndTime
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the date and time that the LAN-Trans lease was
           created (if it has not yet been renewed) or last renewed."
       ::= { cabhCdpLanAddrEntry 4 }

   cabhCdpLanAddrLeaseExpireTime OBJECT-TYPE
       SYNTAX      DateAndTime
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
```

```
          "This is the date and time that the LAN-trans lease expired
           or will expire."
       ::= { cabhCdpLanAddrEntry 5 }

   cabhCdpLanAddrMethod OBJECT-TYPE
       SYNTAX      INTEGER {
      reservationInactive (1),
      reservationActive (2),
      dynamicInactive (3),
      dynamicActive (4)
      }
      MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The IP allocation method indicated by this row.
           reservationInactive(1) indicates a reserved IP that has
           not yet been leased or that has an expired lease.
           reservationActive(2) indicates a reserved IP that has an
           active lease.  dynamicInactive(3) indicates an IP that was
           once dynamically assigned to a LAN-Trans device but
           currently has an expired lease.  dynamicActive(4)
           indicates an IP that was dynamically assigned to a
           LAN-Trans device that has a current lease."
      ::= { cabhCdpLanAddrEntry 6 }

   cabhCdpLanAddrHostName OBJECT-TYPE
       SYNTAX      SnmpAdminString(SIZE(0..80))
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the Host Name of the LAN IP address, based on DCHP
          option 12."
       ::= { cabhCdpLanAddrEntry 7 }

   cabhCdpLanAddrRowStatus OBJECT-TYPE
       SYNTAX      RowStatus
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
   "The RowStatus interlock for creation and deletion of row entries.
    The PS must not allow the NMS to set RowStatus to notInService(2).
    The PS must assign a RowStatus of notInService(2) to any new row
    entry created with a non-unique, cabhCdpLanAddrClientID value.
    The PS must assign a RowStatus of notReady(3) to any new row
    entry created without a cabhCdpLanAddrClientID.  The PS will
    prevent modification of this table's columns and return an
    inconsistentValue error, if the NMS attempts to make such
    modifications while the RowStatus is active(1)."
       ::= { cabhCdpLanAddrEntry 8 }

--================================================================
--
--    cabhCdpWanDataAddrTable (CDP WAN-Data Address Table)
--
--    The cabhCdpWanDataAddrTable contains the configuration or DHCP
--    parameters for each IP address mapping per WAN-Data IP Address.
--
--================================================================

cabhCdpWanDataAddrTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CabhCdpWanDataAddrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains WAN-Data address realm information."
```

```
    ::= {  cabhCdpAddr 2 }

cabhCdpWanDataAddrEntry OBJECT-TYPE
    SYNTAX      CabhCdpWanDataAddrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of general parameter for CDP WAN-Data address realm."
    INDEX { cabhCdpWanDataAddrIndex }
    ::= { cabhCdpWanDataAddrTable 1 }

CabhCdpWanDataAddrEntry ::= SEQUENCE {
      cabhCdpWanDataAddrIndex       INTEGER,
      cabhCdpWanDataAddrClientId          OCTET STRING,
      cabhCdpWanDataAddrIpType            InetAddressType,
      cabhCdpWanDataAddrIp          InetAddress,
      cabhCdpWanDataAddrRenewalTime Integer32,
      cabhCdpWanDataAddrRowStatus   RowStatus
    }


cabhCdpWanDataAddrIndex OBJECT-TYPE
    SYNTAX      INTEGER (1..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Index into table."
    ::= { cabhCdpWanDataAddrEntry 1 }

cabhCdpWanDataAddrClientId OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (1..80))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
"A unique WAN-Data ClientID used when attempting the acquire a
WAN-Data IP Address via DHCP."
    ::= { cabhCdpWanDataAddrEntry 2 }

cabhCdpWanDataAddrIpType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The address type assigned on the WAN-Data side."
    DEFVAL { ipv4 }
    ::= { cabhCdpWanDataAddrEntry 3 }

cabhCdpWanDataAddrIp OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The address assigned on the WAN-Data side."
    ::= { cabhCdpWanDataAddrEntry 4 }

cabhCdpWanDataAddrRenewalTime OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "This is the time remaining before the lease expires.
       This is based on DHCP Option 51."
    ::= { cabhCdpWanDataAddrEntry 5 }

cabhCdpWanDataAddrRowStatus OBJECT-TYPE
```

```
        SYNTAX      RowStatus
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
           "The RowStatus interlock for creation and deletion of row
           entries.  Any writable object in a row can be modified at
           any time while the row is active(1).  The PS must assign a
           RowStatus of notInService(2) to any new row entry created
           with a cabhCdpWanDataAddrClientId that is not unique within
           this table."

        ::= { cabhCdpWanDataAddrEntry 6 }
--=================================================================
--
-- cabhCdpWanDnsServerTable (CDP WAN DNS Server Table)
--
-- The cabhCdpWanDnsServerTable is a table of 3 cable network
-- and Internet DNS Servers.
--
--=================================================================
cabhCdpWanDnsServerTable OBJECT-TYPE
        SYNTAX SEQUENCE OF CabhCdpWanDnsServerEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
        "This table contains the IP addresses of cable network and
         Internet DNS servers, in the order of preference in which
         the PS's CNP will query them, when it cannot resolve a DNS
         query using local information. Entries in this table are
         updated with the information contained in DHCP Option 6,
         received during both the WAN-Man and WAN-Data IP acquisition
         processes."
::= { cabhCdpAddr 3 }

cabhCdpWanDnsServerEntry OBJECT-TYPE
        SYNTAX CabhCdpWanDnsServerEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
        "List of cable network and Internet DNS servers."
        INDEX { cabhCdpWanDnsServerOrder }
::= { cabhCdpWanDnsServerTable 1 }

CabhCdpWanDnsServerEntry ::= SEQUENCE {
        cabhCdpWanDnsServerOrder INTEGER,
        cabhCdpWanDnsServerIpType InetAddressType,
        cabhCdpWanDnsServerIp InetAddress
        }

cabhCdpWanDnsServerOrder OBJECT-TYPE
        SYNTAX INTEGER {
            primary(1),
            secondary(2),
            tertiary(3)
        }
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
        "The order of preference for cable network and Internet DNS
         servers, as listed in DHCP option 6 (Domain Server). Any
         time the CDC receives valid IP address information within
         DHCP Option 6, as part of lease acquisition or renewal of
         a WAN-Man or WAN-Data IP, it must update this information
         into this table.  As entries in DHCP Option 6 are listed in
         order of preference the highest priority entry in DHCP
```

```
          Option 6 must correspond to the row with a
          cabhCdpWanDnsServerOrder with a value of 1.  If DHCP
          Option 6 contains 2 valid IP addresses, the PS must update
          the rows with cabhCdpWanDnsServerOrder values of 1 and 2.
          If DHCP Option 6 contains 3 valid IP addresses, the PS must
          update rows with cabhCdpWanDnsServerOrder values of 1, 2,
          and 3.  Any DNS server information included in DHCP Option 6
          beyond primary, secondary and tertiary will not be
          represented in this table."
::= { cabhCdpWanDnsServerEntry 1 }

cabhCdpWanDnsServerIpType OBJECT-TYPE
        SYNTAX InetAddressType
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
        "This parameter indicates the IP address type of a WAN DNS
         server."
        DEFVAL { ipv4 }
::= { cabhCdpWanDnsServerEntry 2 }

cabhCdpWanDnsServerIp OBJECT-TYPE
        SYNTAX InetAddress
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
        "This parameter indicates the IP address of a WAN DNS server.
          The type of this address is specified by
          cabhCdpWanDnsServerIpType."
        ::= { cabhCdpWanDnsServerEntry 3 }

--
--    DHCP Server Side (CDS) Option Values for the LAN-Trans realm
--
cabhCdpLanPoolStartType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The Address type of the start of range LAN Trans IP Addresses."
      DEFVAL { ipv4 }
    ::= { cabhCdpServer 1 }

cabhCdpLanPoolStart OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The start of range LAN Trans IP Addresses."
      DEFVAL { 'c0a8000a'h }  -- 192.168.0.10
      -- 192.168.0.0 is the network number
      -- 192.168.0.255 is broadcast
      -- address and 192.168.0.1
      -- is reserved for the router
    ::= { cabhCdpServer 2 }

cabhCdpLanPoolEndType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The Address type of the end of range LAN Trans IP Addresses."
      DEFVAL { ipv4 }
    ::= { cabhCdpServer 3 }
```

```
cabhCdpLanPoolEnd OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The end of range for LAN-Trans IP Addresses."
      DEFVAL { 'c0a800fe'h }  -- 192.168.0.254
    ::= { cabhCdpServer 4 }

cabhCdpServerNetworkNumberType      OBJECT-TYPE
      SYNTAX            InetAddressType
MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
  "The IP address type of the LAN-Trans network number."
DEFVAL  { ipv4 }
::= { cabhCdpServer 5 }

cabhCdpServerNetworkNumber          OBJECT-TYPE
SYNTAX            InetAddress
MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
      "The LAN-Trans network number."
DEFVAL  { 'c0a80000'h }
::= { cabhCdpServer 6 }



cabhCdpServerSubnetMaskType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Type of LAN-Trans Subnet Mask."
    DEFVAL { ipv4 }
    ::= { cabhCdpServer 7 }

cabhCdpServerSubnetMask OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
         "Option value 1 - Value of LAN-Trans Subnet Mask."
      DEFVAL { 'ffffff00'h }  -- 255.255.255.0
    ::= { cabhCdpServer 8 }

cabhCdpServerTimeOffset OBJECT-TYPE
    SYNTAX      Integer32 (-86400..86400) -- 0 to 24 hours (in seconds)
    UNITS         "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "Option value 2 - Value of LAN-Trans Time Offset from
        Coordinated Universal Time (UTC)."
      DEFVAL { 0 }      -- UTC
    ::= { cabhCdpServer 9 }

cabhCdpServerRouterType OBJECT-TYPE
      SYNTAX      InetAddressType
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Type of Address, Router for the LAN-Trans
            address realm."
```

```
      DEFVAL { ipv4 }
    ::= { cabhCdpServer 10 }

cabhCdpServerRouter     OBJECT-TYPE
      SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Option value 3 - Router for the LAN-Trans
            address realm."
      DEFVAL { 'c0a80001'h }  -- 192.168.0.1
    ::= { cabhCdpServer 11 }

cabhCdpServerDnsAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The Type of IP Addresses of the LAN-Trans address realm
       DNS servers."
      DEFVAL { ipv4 }
    ::= { cabhCdpServer 12 }

cabhCdpServerDnsAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The IP Addresses of the LAN-Trans address realm
       DNS servers.  As a default there is only one DNS
       server and it is the address specified in Option
       Value 3 - cabhCdpServerRouter.  Only one address
       is specified."
      DEFVAL { 'c0a80001'h }  -- 192.168.0.1
    ::= { cabhCdpServer 13 }

cabhCdpServerSyslogAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The Type of IP Address of the LAN-Trans SYSLOG servers."
      DEFVAL { ipv4 }
    ::= { cabhCdpServer 14 }

cabhCdpServerSyslogAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The IP Addresses of the LAN-Trans SYSLOG servers.
       As a default there are no SYSLOG Servers.
       The factory defaults contains the indication of
       no Syslog Server value equals (0.0.0.0)."
      DEFVAL { '00000000'h }  -- 0.0.0.0
    ::= { cabhCdpServer 15 }

cabhCdpServerDomainName OBJECT-TYPE
    SYNTAX      SnmpAdminString(SIZE(0..128))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
     "Option value 15 - Domain name of LAN-Trans address realm."
      DEFVAL {""}
      ::= { cabhCdpServer 16 }
```

```
cabhCdpServerTTL  OBJECT-TYPE
      SYNTAX      INTEGER (0..255)
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Option value 23 - LAN-Trans Time to Live."
      DEFVAL  { 64 }
      ::= { cabhCdpServer 17 }

cabhCdpServerInterfaceMTU       OBJECT-TYPE
    SYNTAX      Integer32 (0 | 68..4096)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Option value 26 - LAN-Trans Interface MTU.  If the value
         of this object is 0, the PS must not include this option in
         its DHCP Offer or DHCP Ack messages to LAN IP Devices."
    DEFVAL { 0 }
    ::= { cabhCdpServer 18 }

cabhCdpServerVendorSpecific   OBJECT-TYPE
      SYNTAX      OCTET STRING (SIZE(0..255))
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Option value 43 - Vendor Specific Options."
      DEFVAL  { ''h }
      ::= { cabhCdpServer 19 }

cabhCdpServerLeaseTime  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "seconds"
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"Option value 51 –Lease Time for LAN IP Devices in the LAN-Trans realm
(seconds)."
DEFVAL  { 3600 }

      ::= { cabhCdpServer 20 }

cabhCdpServerDhcpAddressType  OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
    "Option value 54 - Type of LAN-Trans DHCP server IP address."
      DEFVAL { ipv4 }
      ::= { cabhCdpServer 21 }

cabhCdpServerDhcpAddress      OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Option value 54 - LAN-Trans DHCP server IP
            address. It defaults to the router address as
            specified in cabhCdpServerRouter.  Alternatively
            a vendor may want to separate CDS address from
            router address."
      DEFVAL { 'c0a80001'h }        --    192.168.0.1
      ::= { cabhCdpServer 22 }

cabhCdpServerControl OBJECT-TYPE
```

```
    SYNTAX      INTEGER {
      restoreConfig(1)
      commitConfig (2),
      }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"The control for the CDS (DHCP Server) configuration. All changes to
the cabhCdpServer mib objects are reflected when reading the value of
the mib objects; however, those changes are NOT applied to the running
configuration of the CDS until they are successfully committed via use
of the cabhCdpServerControl object.

If changes are made to the cabhCdpServer mib objects which are not
yet successfully committed to the CDS, the cabhCdpServerControl object
can be used to rollback all changes to the last valid CDS configuration
and discard all intermediate changes.

restoreConfig - Setting cabhCdpServerControl to this value will cause
any changes to the cabhCdpServer objects not yet committed be reset to
the values from the current running configuration of the CDS.

commitConfig – Setting cabhCdpServerControl to this value will cause
the CDS to validate and apply the valid cabhCdpServer mib settings to
its running configuration. The cabhCdpServerCommitStatus object will
detail the status of this operation."

      DEFVAL { restoreConfig }
    ::= { cabhCdpServer 23 }


cabhCdpServerCommitStatus OBJECT-TYPE
    SYNTAX      INTEGER {
      commitSucceeded (1),
      commitNeeded  (2),
      commitFailed (3)
      }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
"Indicates the status of committing the current cabhCdpServer mib object
values to the running configuration of  the CDS (DHCP Server).

commitSucceeded -  indicates the current cabhCdpServer mib object
values are valid and have been successfully committed to the running
configuration of the CDS.

commitNeeded -  indicates that the value of one or more objects in
cabhCdpServer mib group have been changed but not yet committed to the
running configuration of the CDS.

commitFailed – indicates the PS was unable to commit the cabhCdpServer
mib object values to the running configuration of the CDS due to
conflicts in those values."

   DEFVAL { commitSucceeded }
   ::= { cabhCdpServer 24 }


--
-- notification group is for future extension.
--

cabhCdpNotification OBJECT IDENTIFIER ::= { cabhCdpMib 2 0 }
cabhCdpConformance  OBJECT IDENTIFIER ::= { cabhCdpMib 3 }
```

```
cabhCdpCompliances  OBJECT IDENTIFIER ::= { cabhCdpConformance 1 }
cabhCdpGroups       OBJECT IDENTIFIER ::= { cabhCdpConformance 2 }

--
--    Notification Group
--


-- compliance statements

cabhCdpBasicCompliance MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
            "The compliance statement for devices that implement
             MTA feature."
    MODULE   --cabhCdpMib



-- unconditionally mandatory groups

    MANDATORY-GROUPS {
            cabhCdpGroup
    }

::= { cabhCdpCompliances 3 }

cabhCdpGroup   OBJECT-GROUP

   OBJECTS {
   cabhCdpSetToFactory,
   cabhCdpLanTransCurCount,
   cabhCdpLanTransThreshold,
   cabhCdpLanTransAction,
   cabhCdpWanDataIpAddrCount,

   cabhCdpLanAddrClientID,
   cabhCdpLanAddrLeaseCreateTime,
   cabhCdpLanAddrLeaseExpireTime,
   cabhCdpLanAddrMethod,
   cabhCdpLanAddrHostName,
   cabhCdpLanAddrRowStatus,

   cabhCdpWanDataAddrClientId,
   cabhCdpWanDataAddrIpType,
   cabhCdpWanDataAddrIp,
   cabhCdpWanDataAddrRenewalTime,
   cabhCdpWanDataAddrRowStatus,

   cabhCdpWanDnsServerIpType,
   cabhCdpWanDnsServerIp,

   cabhCdpLanPoolStartType,
   cabhCdpLanPoolStart,
   cabhCdpLanPoolEndType,
   cabhCdpLanPoolEnd,
   cabhCdpServerNetworkNumberType,
   cabhCdpServerNetworkNumber,
   cabhCdpServerSubnetMaskType,
   cabhCdpServerSubnetMask,
   cabhCdpServerTimeOffset,
   cabhCdpServerRouterType,
   cabhCdpServerRouter,
   cabhCdpServerDnsAddressType,
   cabhCdpServerDnsAddress,
```

```
   cabhCdpServerSyslogAddressType,
   cabhCdpServerSyslogAddress,
   cabhCdpServerDomainName,
   cabhCdpServerTTL,
   cabhCdpServerInterfaceMTU,
   cabhCdpServerVendorSpecific,
   cabhCdpServerLeaseTime,
   cabhCdpServerDhcpAddressType,
   cabhCdpServerDhcpAddress,
   cabhCdpServerControl,
   cabhCdpServerCommitStatus
}
STATUS current
DESCRIPTION
"Group of objects for CableHome CDP MIB."
::= { cabhCdpGroups 1 }

END
```

E.3 IPCable2Home Test Portal (CTP) MIB requirement.

requirements

The CableHome™ CTP MIB MUST be implemented as defined below.

```
CABH-CTP-MIB DEFINITIONS ::= BEGIN
IMPORTS
   MODULE-IDENTITY,
   OBJECT-TYPE          FROM SNMPv2-SMI
   TruthValue,
   TEXTUAL-CONVENTION   FROM SNMPv2-TC
   OBJECT-GROUP,
   MODULE-COMPLIANCE    FROM SNMPv2-CONF
   InetAddressType,
   InetAddress,
   InetAddressIPv4,
   InetAddressIPv6            FROM INET-ADDRESS-MIB
   clabProjCableHome    FROM CLAB-DEF-MIB;

--=================================================================
--
--    History:
--
--    Date        Modified by       Reason
--    04/05/02                      Issued I01
--    09/20/02                      Issued I02
--    04/11/03                      Issued I03
--
--=================================================================

cabhCtpMib MODULE-IDENTITY
    LAST-UPDATED  "200304110000Z" -- April 11, 2003
    ORGANIZATION    "CableLabs Broadband Access Department"
    CONTACT-INFO
            "Kevin Luehrs
            Postal: Cable Television Laboratories, Inc.
                  400 Centennial Parkway
                  Louisville, Colorado 80027-1266
            U.S.A.
```

```
            Phone:  +1 303-661-9100
            Fax:    +1 303-661-9199
            E-mail: k.luehrs@cablelabs.com"
    DESCRIPTION
            "This MIB module defines the diagnostic controls
            offered by the CableHome Test Portal (CTP).

Acknowledgements:
Roy Spitzer       -     Consultant to CableLabs
Mike Mannette     -     Consultant to CableLabs
Randy Dunton      -     Intel
Dmitrii Loukianov -     Intel
      Wes Peters        -     DoBox, Inc.
      Chris Zacker                Broadcom"
    ::=  { clabProjCableHome 5 }

-- Textual conventions


cabhCtpObjects          OBJECT IDENTIFIER ::= { cabhCtpMib 1 }
cabhCtpBase             OBJECT IDENTIFIER ::= { cabhCtpObjects 1 }
cabhCtpConnSpeed        OBJECT IDENTIFIER ::= { cabhCtpObjects 2 }
cabhCtpPing             OBJECT IDENTIFIER ::= { cabhCtpObjects 3 }

--
--    The following group describes the base objects in the Cable Home
--    Management Portal.
--

cabhCtpSetToFactory     OBJECT-TYPE
SYNTAX            TruthValue
MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
"Setting this object to true(1) causes all the tables in the CTP MIB
to be cleared, and all CTP MIB objects with default values set back to
those default values. Reading this object always returns false(2)."
::={cabhCtpBase 1}

--
--    Parameter and results from Connection Speed Command
--

cabhCtpConnSrcIpType    OBJECT-TYPE
      SYNTAX            InetAddressType
      MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
"The IP Address type used as the source address for the Connection
Speed Test."
      DEFVAL { ipv4 }
::= { cabhCtpConnSpeed 1 }

cabhCtpConnSrcIp  OBJECT-TYPE
      SYNTAX            InetAddress
      MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
"The IP Address used as the source address for the Connection
Speed Test. The default value is the value of cabhCdpServerRouter
(192.168.0.1)."
      REFERENCE
            "CableHome Specification Section 6.4.4"
      DEFVAL { 'c0a80001'h }  -- 192.168.0.1
      ::= { cabhCtpConnSpeed 2 }
```

```
cabhCtpConnDestIpType   OBJECT-TYPE
      SYNTAX            InetAddressType
      MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
"The IP Address Type for the CTP Connection Speed Tool destination
address."
      DEFVAL { ipv4 }
::={ cabhCtpConnSpeed 3 }



cabhCtpConnDestIp OBJECT-TYPE
      SYNTAX            InetAddress
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
    "The IP Address used as the destination address for the Connection
    Speed Test."
      ::= { cabhCtpConnSpeed 4 }

cabhCtpConnProto  OBJECT-TYPE
      SYNTAX            INTEGER {
                              udp               (1),
                              tcp               (2)
                              }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The protocol used in the Connection Speed Test. TCP
        testing is optional."
    DEFVAL { udp }
      ::= { cabhCtpConnSpeed 5 }


cabhCtpConnNumPkts      OBJECT-TYPE
    SYNTAX        INTEGER (1..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The number of packets the CTP is to send when triggered to
 execute the Connection Speed Tool."
      DEFVAL { 100 }
      ::= { cabhCtpConnSpeed 6 }

cabhCtpConnPktSize      OBJECT-TYPE
      SYNTAX            INTEGER (64..1518)
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The size of the test frames."
      REFERENCE
            ""
      DEFVAL { 1518 }
      ::= { cabhCtpConnSpeed 7 }

cabhCtpConnTimeOut      OBJECT-TYPE
      SYNTAX            INTEGER (0..600000)             -- Max 10 minutes
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
    "The timeout value for the response.  A value of zero indicates
```

```
    no time out and can be used for TCP only."
      DEFVAL {30000}  -- 30 seconds
    ::= { cabhCtpConnSpeed 8 }

cabhCtpConnControl      OBJECT-TYPE
SYNTAX            INTEGER {
      start(1),
      abort(2)
      }
MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
"The control for the Connection Speed Tool. Setting this object to
start(1) causes the Connection Speed Tool to execute. Setting this
object to abort(2) causes the Connection Speed Tool to stop running.
This parameter should only be set via SNMP."
DEFVAL  {abort }
::={ cabhCtpConnSpeed 9 }



cabhCtpConnStatus OBJECT-TYPE
SYNTAX            INTEGER {
notRun(1),
running(2),
complete(3),
aborted(4),
timedOut(5)
}
MAX-ACCESS  read-only
STATUS            current
DESCRIPTION
   "The status of the Connection Speed Tool."
DEFVAL   { notRun }
::={ cabhCtpConnSpeed 10 }


cabhCtpConnPktsSent     OBJECT-TYPE
      SYNTAX            INTEGER (0..65535)
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The number of packets the CTP sent after it was triggered to
    execute the Connection Speed Tool."
      ::= { cabhCtpConnSpeed 11 }

cabhCtpConnPktsRecv     OBJECT-TYPE
      SYNTAX            INTEGER (0..65535)
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The number of packets the CTP received after it executed the
       Connection Speed Tool."
      ::= { cabhCtpConnSpeed 12 }

cabhCtpConnRTT    OBJECT-TYPE
      SYNTAX      INTEGER (0..600000)
      UNITS       "millisec"
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The resulting round trip time for the set of
        packets sent to and received from the target LAN IP Device."
      ::= { cabhCtpConnSpeed 13 }
```

```
cabhCtpConnThroughput   OBJECT-TYPE
      SYNTAX            INTEGER (0..65535)
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The average round-trip throughput measured in
            kilobits per second."
      ::= { cabhCtpConnSpeed 14 }



--
--    Parameters and Results for Ping Command
--

cabhCtpPingSrcIpType    OBJECT-TYPE
SYNTAX            InetAddressType
MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
  "The IP Address Type for CTP Ping Tool source address."
DEFVAL { ipv4 }
::={ cabhCtpPing 1 }

cabhCtpPingSrcIp  OBJECT-TYPE
      SYNTAX            InetAddress
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
    "The IP Address used as the source address for the Ping Test. The
    default value is the value of CabhCdpServerRouter (192.168.0.1)."
    REFERENCE
             "CableHome 1.0 Specification Section 6.4.4"
    DEFVAL { 'c0a80001'h }
      ::= { cabhCtpPing 2 }

cabhCtpPingDestIpType   OBJECT-TYPE
SYNTAX            InetAddressType
MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
  "The IP Address Type for the CTP Ping Tool destination address."
DEFVAL { ipv4 }
::={ cabhCtpPing 3 }

cabhCtpPingDestIp OBJECT-TYPE
      SYNTAX            InetAddress
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
    "The Destination IP Address used as the destination address for
    the Ping Test."
      ::= { cabhCtpPing 4 }

cabhCtpPingNumPkts      OBJECT-TYPE
      SYNTAX            INTEGER (1..4)
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The number of packets to send to each host."
      DEFVAL {1}
      ::= { cabhCtpPing 5 }

cabhCtpPingPktSize      OBJECT-TYPE
      SYNTAX            INTEGER (64..1518)
```

```
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The size of the test frames."
      DEFVAL {64}
      ::= { cabhCtpPing 6 }

cabhCtpPingTimeBetween  OBJECT-TYPE
      SYNTAX            INTEGER (0..600000)
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The time between sending one ping and the next."
      DEFVAL { 1000 }
    ::= { cabhCtpPing 7 }

cabhCtpPingTimeOut            OBJECT-TYPE
SYNTAX            INTEGER (1..600000)
UNITS       "milliseconds"
MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
"The time out for ping response (ICMP reply) for a single transmitted
ping message (ICMP request)."
DEFVAL  { 1000 } -- 1 second
::={ cabhCtpPing 8 }



cabhCtpPingControl      OBJECT-TYPE
SYNTAX            INTEGER {
      start(1),
      abort(2)
      }
MAX-ACCESS  read-write
STATUS            current
DESCRIPTION
"The control for the Ping Tool. Setting this object to start(1) causes
the Ping Tool to execute. Setting this object to abort(2) causes the
Ping Tool to stop running. This parameter should only be set via SNMP."
DEFVAL  {abort }
::={ cabhCtpPing 9 }


cabhCtpPingStatus OBJECT-TYPE
SYNTAX            INTEGER {
      notRun(1),
      running(2),
complete(3),
aborted(4),
timedOut(5)
      }
MAX-ACCESS  read-only
STATUS            current
DESCRIPTION
   "The status of the Ping Tool."
DEFVAL   { notRun }
::={ cabhCtpPing 10 }


cabhCtpPingNumSent      OBJECT-TYPE
SYNTAX            INTEGER (0..4)
MAX-ACCESS  read-only
STATUS            current
```

```
DESCRIPTION
  "The number of Pings sent"
::={ cabhCtpPing 11 }

cabhCtpPingNumRecv      OBJECT-TYPE
      SYNTAX            INTEGER (0..255)
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of pings received."
      ::= { cabhCtpPing 12 }

cabhCtpPingAvgRTT OBJECT-TYPE
      SYNTAX            INTEGER (0..600000)
      UNITS             "millisec"
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The resulting average of round trip times for acknowledged
    packets."
      ::= { cabhCtpPing 13 }

cabhCtpPingMaxRTT OBJECT-TYPE
      SYNTAX            INTEGER (0..600000)
      UNITS             "millisec"
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The resulting maximum of round trip times for acknowledged
    packets."
      ::= { cabhCtpPing 14 }

cabhCtpPingMinRTT OBJECT-TYPE
      SYNTAX            INTEGER (0..600000)
      UNITS             "millisec"
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The resulting minimum of round trip times for acknowledged
    packets."
      ::= { cabhCtpPing 15 }

cabhCtpPingNumIcmpError OBJECT-TYPE
      SYNTAX            INTEGER (0..255)
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "Number of ICMP errors."
      ::= { cabhCtpPing 16 }

cabhCtpPingIcmpError    OBJECT-TYPE
      SYNTAX            INTEGER (0..255)
      MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The last ICMP error."
      ::= { cabhCtpPing 17 }

--=====================================================================

--
-- notification group is for future extension.
--

cabhCtpNotification OBJECT IDENTIFIER ::= { cabhCtpMib 2 0 }
```

```
cabhCtpConformance  OBJECT IDENTIFIER ::= { cabhCtpMib 3 }
cabhCtpCompliances  OBJECT IDENTIFIER ::= { cabhCtpConformance 1 }
cabhCtpGroups       OBJECT IDENTIFIER ::= { cabhCtpConformance 2 }

--
--    Notification Group
--

-- compliance statements

cabhCtpBasicCompliance MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
            "The compliance statement for devices that implement
             Portal Service feature."
    MODULE   --cabhCtpMib



-- unconditionally mandatory groups

    MANDATORY-GROUPS {
            cabhCtpGroup
    }

::= { cabhCtpCompliances 3 }


cabhCtpGroup OBJECT-GROUP
    OBJECTS {

         cabhCtpSetToFactory,
         cabhCtpConnSrcIpType,
         cabhCtpConnSrcIp,
         cabhCtpConnDestIpType,
         cabhCtpConnDestIp,
         cabhCtpConnProto,
         cabhCtpConnNumPkts,
         cabhCtpConnPktSize,
         cabhCtpConnTimeOut,
         cabhCtpConnControl,
         cabhCtpConnStatus,
         cabhCtpConnPktsSent,
         cabhCtpConnPktsRecv,
         cabhCtpConnRTT,
         cabhCtpConnThroughput,

         cabhCtpPingSrcIpType,
         cabhCtpPingSrcIp,
         cabhCtpPingDestIpType,
         cabhCtpPingDestIp,
         cabhCtpPingNumPkts,
         cabhCtpPingPktSize,
         cabhCtpPingTimeBetween,
         cabhCtpPingTimeOut,
         cabhCtpPingControl,
         cabhCtpPingStatus,
         cabhCtpPingNumSent,
         cabhCtpPingNumRecv,
         cabhCtpPingAvgRTT,
         cabhCtpPingMinRTT,
         cabhCtpPingMaxRTT,
         cabhCtpPingNumIcmpError,
         cabhCtpPingIcmpError
      }
```

```
    STATUS    current
    DESCRIPTION
        "Group of objects for CableHome CTP MIB."
    ::= { cabhCtpGroups 1 }

END
```

E.4 IPCable2Home Portal Services Device (PSDev) MIB requirement.

Requirements

The CableHome™ PSDev MIB MUST be implemented as defined below.

```
CABH-PS-DEV-MIB DEFINITIONS ::= BEGIN
IMPORTS
   MODULE-IDENTITY,
   OBJECT-TYPE,
   Integer32,
   NOTIFICATION-TYPE          FROM SNMPv2-SMI
   TruthValue,


   PhysAddress,
   DateAndTime,
   TEXTUAL-CONVENTION         FROM SNMPv2-TC
   SnmpAdminString                  FROM SNMP-FRAMEWORK-MIB
   OBJECT-GROUP,
   MODULE-COMPLIANCE,
   NOTIFICATION-GROUP         FROM SNMPv2-CONF

   InetAddressType,
   InetAddress                FROM INET-ADDRESS-MIB

   docsDevSwCurrentVers,
   docsDevEvLevel,
   docsDevEvId,
   docsDevEvText,
   docsDevSwFilename,
   docsDevSwServer                  FROM DOCS-CABLE-DEVICE-MIB -- RFC2669

   cabhCdpServerDhcpAddress,
   cabhCdpWanDataAddrClientId,
   cabhCdpLanTransThreshold,
   cabhCdpLanTransCurCount          FROM CABH-CDP-MIB

   clabProjCableHome          FROM CLAB-DEF-MIB;

--================================================================
--
--    History:
--
--    Date        Modified by       Reason
--    04/05/02                      Issued I01
--    09/20/02                      Issued I02
--    04/11/03                      Issued I03
--
--================================================================


cabhPsDevMib MODULE-IDENTITY
    LAST-UPDATED    "200304110000Z"-- April 11, 2003
    ORGANIZATION    "CableLabs Broadband Access Department"
    CONTACT-INFO
            "Kevin Luehrs
            Postal: Cable Television Laboratories, Inc.
                  400 Centennial Parkway
                  Louisville, Colorado 80027-1266
                        U.S.A.
            Phone:  +1 303-661-9100
            Fax:    +1 303-661-9199
            E-mail: k.luehrs@cablelabs.com"
```

```
    DESCRIPTION
            "This MIB module supplies the basic management objects
             for the PS Device.  The PS device parameter describe
      general PS Device attributes and behavior characteristics.
      Most the PS Device MIB is need for configuration download.

             Acknowledgements:
             Roy Spitzer     - Consultant to CableLabs
            Mike Mannette          - Consultant to CableLabs
             Itay Sherman          - Texas Instruments
             Chris Zacker          - Broadcom
             Rick Vetter     - Consultant to CableLabs "
    ::=  { clabProjCableHome 1 }

-- Textual conventions
      X509Certificate ::= TEXTUAL-CONVENTION
             STATUS current
             DESCRIPTION
"An X509 digital certificate encoded as an ASN.1 DER object."
             SYNTAX OCTET STRING (SIZE (0..4096))


cabhPsDevMibObjects    OBJECT IDENTIFIER ::= { cabhPsDevMib 1 }
cabhPsDevBase       OBJECT IDENTIFIER ::= { cabhPsDevMibObjects 1 }
cabhPsDevProv    OBJECT IDENTIFIER ::= { cabhPsDevMibObjects 2 }

--
-- The following group describes the base objects in the PS.
-- These are device based parameters.
--

cabhPsDevDateTime OBJECT-TYPE
        SYNTAX      DateAndTime
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The date and time, with optional timezone
             information."
    ::= { cabhPsDevBase 1 }

cabhPsDevResetNow       OBJECT-TYPE
SYNTAX                  TruthValue
MAX-ACCESS        read-write
STATUS                  current
DESCRIPTION
"Setting this object to true(1) causes the stand-alone or embedded
PS device to reboot. Device code initializes as if starting from a
power-on reset. The CMP ensures that MIB object values persist as
specified in Appendix I of the CableHome 1.0 specification. Reading
this object always returns false(2)."
::= { cabhPsDevBase 2 }

cabhPsDevSerialNumber OBJECT-TYPE
SYNTAX      SnmpAdminString (SIZE (0..128))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The manufacturer's serial number for this PS.  This parameter
    is manufacturer provided and is stored in non-volatile memory."
::= { cabhPsDevBase 3 }

cabhPsDevHardwareVersion OBJECT-TYPE
SYNTAX      SnmpAdminString (SIZE (0..48))
MAX-ACCESS  read-only
STATUS      current
```

```
DESCRIPTION
    "The manufacturer's hardware version for this PS.  This
    parameter is manufacturer provided and is stored in non-volatile
    memory."
::= { cabhPsDevBase 4 }

cabhPsDevWanManMacAddress  OBJECT-TYPE
SYNTAX      PhysAddress
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The PS WAN-MAN MAC address. This is the PS hardware address to be
used by the CDC to uniquely identify the PS to the cable data network
DHCP server for the acquisition of an IP address to be used for
management messaging between the cable network NMS and the CMP"


::= { cabhPsDevBase 5 }

cabhPsDevWanDataMacAddress  OBJECT-TYPE
      SYNTAX      PhysAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
"The PS WAN-Data MAC address. The PS could have multiple WAN-Data
Interfaces, which share the same hardware address.  The client
identifiers will be unique so that each may be assigned
a different, unique IP address."


    ::= { cabhPsDevBase 6 }

cabhPsDevTypeIdentifier    OBJECT-TYPE
      SYNTAX      SnmpAdminString
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
"This is a copy of the device type identifier used in the
DHCP option 60 exchanged between the PS and the
DHCP server."
      ::= { cabhPsDevBase 7 }

cabhPsDevSetToFactory    OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
"Setting this object to true(1) sets all PsDev MIB objects
to the factory default values. Reading this object always
returns false(2)."

    ::= { cabhPsDevBase 8 }

cabhPsDevWanManClientId OBJECT-TYPE
SYNTAX                  OCTET STRING (SIZE (1..80))
MAX-ACCESS              read-write
STATUS                  deprecated
DESCRIPTION
"This is the client ID used for WAN-MAN DHCP
requests. The default value is the 6 byte MAC address."
::= { cabhPsDevBase 9 }

cabhPsDevTodSyncStatus OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
          "This object indicates whether the PS was able to
          successfully synchronize with the Time of Day (ToD)
          Server in the cable network. The PS sets this object
          to true(1) if the PS successfully synchronizes its time
          with the ToD server. The PS sets this object to
          false(2) if the PS does not successfully synchronize
          with the ToD server"
    DEFVAL { false }
::= { cabhPsDevBase 10 }

cabhPsDevProvMode       OBJECT-TYPE
      SYNTAX            INTEGER
      {
            dhcpmode(1),
            snmpmode(2),
            dormantCHmode(3)
      }
      MAX-ACCESS  read-only
      STATUS            current
      DESCRIPTION
"This object indicates the provisioning mode in which the
PS is operating. If the PS is operating in DHCP Provisioning
Mode as described in the CableHome 1.0 specification, the PS
sets this object to dhcpmode(1). If the PS is operating in SNMP
Provisioning Mode, the PS sets this object to snmpmode(2).If
the PS is not configured to operate in either dhcpmode or
snmpmode it will fall back to Dormant CableHome Mode
dormantCHmode(3)."
::={ cabhPsDevBase 11 }


--
--    The following group defines Provisioning Specific parameters
--

cabhPsDevProvisioningTimer OBJECT-TYPE
      SYNTAX      INTEGER (0..16383)
      UNITS       "minutes"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
"This object enables the user to set the duration of the provisioning
timeout timer. The value is in minutes. Setting the timer
to 0 disables it. The default value for the timer is 5."
      DEFVAL {5}
      ::=  {cabhPsDevProv 1}

cabhPsDevProvConfigFile OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..128))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"The URL of the TFTP host for downloading provisioning
and configuration parameters to this device. Returns NULL if the
server address is unknown."
    ::= { cabhPsDevProv 2 }

cabhPsDevProvConfigHash OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(20))
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Hash of the contents of the PS config file, which is calculated by
```

```
the NMS and sent to the PS. For the SHA-1 authentication algorithm
the hash length is 160 bits. This hash value is encoded in binary
format."
      ::= { cabhPsDevProv 3 }

cabhPsDevProvConfigFileSize OBJECT-TYPE
      SYNTAX       Integer32
      UNITS       "bytes"
      MAX-ACCESS  read-only
      STATUS       current
      DESCRIPTION
            "Size of the configuration file."
      ::={ cabhPsDevProv 4 }

cabhPsDevProvConfigFileStatus OBJECT-TYPE
      SYNTAX            INTEGER
      {
             idle      (1),
             busy    (2)
      }
      MAX-ACCESS  read-only
      STATUS            current
      DESCRIPTION
"This object indicates the current status
of the configuration file download process. It is
provided to indicate to the management entity
that the PS will reject PS Configuration File triggers
(set request to cabhPsDevProvConfigFile) when busy."
      ::={ cabhPsDevProv 5 }

cabhPsDevProvConfigTLVProcessed OBJECT-TYPE
      SYNTAX            INTEGER (0..16383)
      MAX-ACCESS  read-only
      STATUS            current
      DESCRIPTION
            "Number of TLVs processed in config file."
      ::={ cabhPsDevProv 6 }

cabhPsDevProvConfigTLVRejected OBJECT-TYPE
      SYNTAX            INTEGER (0..16383)
      MAX-ACCESS  read-only
      STATUS            current
      DESCRIPTION
            "Number of TLVs rejected in config file."
      ::={ cabhPsDevProv 7 }

cabhPsDevProvSolicitedKeyTimeout OBJECT-TYPE
   SYNTAX      Integer32 (15..600)
   UNITS          "seconds"
   MAX-ACCESS  read-write
   STATUS      current
      DESCRIPTION
"This timeout applies only when the Provisioning Server initiated
key management (with a Wake Up message) for SNMPv3.  It is the
period during which the PS will save a number (inside the
sequence number field) from the sent out AP Request and wait for the
matching AP Reply from the Provisioning Server."
      DEFVAL { 120 }
      ::= { cabhPsDevProv 8 }


cabhPsDevProvState    OBJECT-TYPE
    SYNTAX      INTEGER
    {
            pass              (1),
```

```
            inProgress  (2),
            fail              (3)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object indicates the completion state of the
     initialization process. Pass or Fail states occur after
     completion of the initialization flow. InProgress occurs
     from PS initialization start to PS initialization end."
    ::= { cabhPsDevProv 9 }

cabhPsDevProvAuthState    OBJECT-TYPE
    SYNTAX      INTEGER
    {
            accepted    (1),
            rejected          (2)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object indicates the authentication state
            of the configuration file."
    ::= { cabhPsDevProv 10 }

cabhPsDevProvCorrelationId    OBJECT-TYPE
    SYNTAX              Integer32
    MAX-ACCESS    read-only
    STATUS              deprecated
    DESCRIPTION
    "Random value generated by the PS for use in registration
    authorization.  It is for use only in the PS initialization
    messages and for PS configuration file download. This value
    appears in both cabhPsDevProvisioningStatus and
    cabhPsDevProvisioningEnrollmentReport informs to verify the
    instance of loading the configuration file."
    ::= { cabhPsDevProv 11 }

cabhPsDevTimeServerAddrType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
       "The IP address type of the Time server (RFC-868).  IP version 4
       is typically used."
    ::= { cabhPsDevProv 12 }

cabhPsDevTimeServerAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The IP address of the Time server (RFC-868). Returns
             0.0.0.0 if the time server IP address is unknown."
    ::= { cabhPsDevProv 13 }

--
-- notification group is for future extension.
--

cabhPsNotification OBJECT IDENTIFIER ::= { cabhPsDevMib 2 0 }
cabhPsConformance  OBJECT IDENTIFIER ::= { cabhPsDevMib 3 }
cabhPsCompliances  OBJECT IDENTIFIER ::= { cabhPsConformance 1 }
cabhPsGroups       OBJECT IDENTIFIER ::= { cabhPsConformance 2 }
```

```
--
--    Notification Group
--

cabhPsDevInitTLVUnknownTrap     NOTIFICATION-TYPE
   OBJECTS   {
      docsDevEvLevel,
      docsDevEvId,
      docsDevEvText,
      cabhPsDevWanManMacAddress
   }
   STATUS  current
   DESCRIPTION
"Event due to detection of unknown TLV during the TLV parsing process.
The values of docsDevEvLevel, docsDevId, and docsDevEvText are from the
entry which logs this event in the docsDevEventTable.  The value of
cabhPsDevWanManMacAddress indicates the Wan-Man MAC address of the PS.
This part of the information is uniform across all PS Traps."
   ::= { cabhPsNotification 1 }

cabhPsDevInitTrap  NOTIFICATION-TYPE
   OBJECTS {
         docsDevEvLevel,
         docsDevEvId,
         docsDevEvText,
         cabhPsDevWanManMacAddress,
         cabhPsDevProvConfigFile,
         cabhPsDevProvConfigTLVProcessed,
         cabhPsDevProvConfigTLVRejected
   }
   STATUS      current
   DESCRIPTION
           "This inform is issued to confirm the successful completion
           of the CableHome provisioning process."
   ::= { cabhPsNotification 2 }

cabhPsDevInitRetryTrap  NOTIFICATION-TYPE
   OBJECTS  {
         docsDevEvLevel,
         docsDevEvId,
         docsDevEvText,
      cabhPsDevWanManMacAddress
   }
   STATUS      current
   DESCRIPTION
"An event to report a failure happened during the initialization
process and detected in the PS. "
   ::= { cabhPsNotification 3 }

cabhPsDevDHCPFailTrap NOTIFICATION-TYPE
   OBJECTS {
       docsDevEvLevel,
       docsDevEvId,
       docsDevEvText,
       cabhPsDevWanManMacAddress,
       cabhCdpServerDhcpAddress
   }
   STATUS  current
   DESCRIPTION
           "An event to report the failure of a DHCP server.
           The value of  cabhCdpServerDhcpAddressis the IP address
           of the DHCP server."
   ::= { cabhPsNotification 4 }
```

```
cabhPsDevSwUpgradeInitTrap NOTIFICATION-TYPE
   OBJECTS {
       docsDevEvLevel,
       docsDevEvId,
       docsDevEvText,
       cabhPsDevWanManMacAddress,
       docsDevSwFilename,
       docsDevSwServer
   }
   STATUS  current
   DESCRIPTION
           "An event to report a software upgrade initiated
           event. The values of docsDevSwFilename, and
           docsDevSwServer indicate the software image name
           and the server IP address the image is from."
   ::= { cabhPsNotification 5 }


cabhPsDevSwUpgradeFailTrap NOTIFICATION-TYPE
   OBJECTS {
       docsDevEvLevel,
       docsDevEvId,
       docsDevEvText,
       cabhPsDevWanManMacAddress,
       docsDevSwFilename,
       docsDevSwServer
   }
   STATUS  current
   DESCRIPTION
           "An event to report the failure of a software upgrade
           attempt. The values of docsDevSwFilename, and
           docsDevSwServer indicate the software image name
           and the server IP address the image is from."
   ::= { cabhPsNotification 6 }


cabhPsDevSwUpgradeSuccessTrap NOTIFICATION-TYPE
   OBJECTS {
       docsDevEvLevel,
       docsDevEvId,
       docsDevEvText,
       cabhPsDevWanManMacAddress,
       docsDevSwFilename,
       docsDevSwServer
   }
   STATUS  current
   DESCRIPTION
           "An event to report the Software upgrade success event.
           The values of docsDevSwFilename, and
           docsDevSwServer indicate the software image name
           and the server IP address the image is from."
   ::= { cabhPsNotification 7 }


cabhPsDevSwUpgradeCVCFailTrap NOTIFICATION-TYPE
   OBJECTS {
       docsDevEvLevel,
       docsDevEvId,
       docsDevEvText,
       cabhPsDevWanManMacAddress
   }
   STATUS  current
   DESCRIPTION
           "An event to report the failure of the verification
           of code file happened during a secure software upgrade
```

```
            attempt."
    ::= { cabhPsNotification 8 }


cabhPsDevTODFailTrap NOTIFICATION-TYPE
    OBJECTS {
        docsDevEvLevel,
        docsDevEvId,
        docsDevEvText,
        cabhPsDevTimeServerAddr,
        cabhPsDevWanManMacAddress
    }
    STATUS  current
    DESCRIPTION
         "An event to report the failure of a time of day server.
         The value of cabhPsDevTimeServerAddr indicates the server IP
         address."
    ::= { cabhPsNotification 9 }

cabhPsDevCdpWanDataIpTrap  NOTIFICATION-TYPE
    OBJECTS {
        docsDevEvLevel,
        docsDevEvId,
        docsDevEvText,
        cabhCdpWanDataAddrClientId,
        cabhPsDevWanManMacAddress
    }
    STATUS      current
    DESCRIPTION
"An event to report the failure of PS to obtain all needed WAN-Data Ip
Addresses. cabhCdpWanDataAddrClientId indicates the ClientId for which
the failure occurred."
    ::= { cabhPsNotification 10 }


cabhPsDevCdpThresholdTrap  NOTIFICATION-TYPE
    OBJECTS {
        docsDevEvLevel,
        docsDevEvId,
        docsDevEvText,
        cabhPsDevWanManMacAddress,
        cabhCdpLanTransThreshold
    }
    STATUS      current
    DESCRIPTION
"An event to report that the Lan-Trans threshold has been exceeded."
    ::= { cabhPsNotification 11 }

cabhPsDevCspTrap  NOTIFICATION-TYPE
    OBJECTS {
        docsDevEvLevel,
        docsDevEvId,
        docsDevEvText,
        cabhPsDevWanManMacAddress
    }
    STATUS      current
    DESCRIPTION
            "To report an event with the CableHome Security Portal."
    ::= { cabhPsNotification 12 }

cabhPsDevCapTrap  NOTIFICATION-TYPE
    OBJECTS {
        docsDevEvLevel,
        docsDevEvId,
        docsDevEvText,
```

```
        cabhPsDevWanManMacAddress
    }
    STATUS      current
    DESCRIPTION
            "To report an event with the CableHome Address Portal."
    ::= { cabhPsNotification 13 }

cabhPsDevCtpTrap  NOTIFICATION-TYPE
    OBJECTS {
        docsDevEvLevel,
        docsDevEvId,
        docsDevEvText,
        cabhPsDevWanManMacAddress
    }
    STATUS      current
    DESCRIPTION
            "To report an event with the CableHome Test Portal."
    ::= { cabhPsNotification 14 }

cabhPsDevProvEnrollTrap  NOTIFICATION-TYPE
    OBJECTS {
        cabhPsDevHardwareVersion,
        docsDevSwCurrentVers,
        cabhPsDevTypeIdentifier,
        cabhPsDevWanManMacAddress,
        cabhPsDevProvCorrelationId
    }
    STATUS      current
    DESCRIPTION
            "This inform is issued to initiate the CableHome
               process provisioning."
    REFERENCE
          "Inform as defined in RFC 1902"
    ::= { cabhPsNotification 15 }

cabhPsDevCdpLanIpPoolTrap NOTIFICATION-TYPE
      OBJECTS {
      docsDevEvLevel, docsDevEvId, docsDevEvText,
      cabhPsDevWanManMacAddress,
      cabhCdpLanTransCurCount
      }
      STATUS current
      DESCRIPTION
"An event to report that the pool of IP addresses for LAN clients, as
defined by cabh CdpLanPoolStart and cabhCdpLanPoolEnd, is exhausted "

      ::= { cabhPsNotification 16}

-- compliance statements

cabhPsBasicCompliance MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
             "The compliance statement for devices that implement
              PS feature."
    MODULE   --cabhPsMib




-- unconditionally mandatory groups

    MANDATORY-GROUPS {
```

```
            cabhPsGroup
    }

::= { cabhPsCompliances 1}


cabhPsGroup OBJECT-GROUP
    OBJECTS {
         cabhPsDevDateTime,
         cabhPsDevResetNow,
         cabhPsDevSerialNumber,
         cabhPsDevHardwareVersion,
         cabhPsDevWanManMacAddress,
         cabhPsDevWanDataMacAddress,
         cabhPsDevTypeIdentifier,
       cabhPsDevSetToFactory,
         cabhPsDevWanManClientId,
       cabhPsDevTodSyncStatus,
         cabhPsDevProvMode,

         cabhPsDevProvisioningTimer,
       cabhPsDevProvConfigFile,
       cabhPsDevProvConfigHash,
       cabhPsDevProvConfigFileSize,
       cabhPsDevProvConfigFileStatus,
       cabhPsDevProvConfigTLVProcessed,
       cabhPsDevProvConfigTLVRejected,
       cabhPsDevProvSolicitedKeyTimeout,
       cabhPsDevProvState,
       cabhPsDevProvAuthState,
         cabhPsDevProvCorrelationId,
         cabhPsDevTimeServerAddrType,
         cabhPsDevTimeServerAddr

    }
    STATUS    current
    DESCRIPTION
        "Group of objects for CableHome PS MIB."
    ::= { cabhPsGroups 1 }

cabhPsNotificationGroup       NOTIFICATION-GROUP
      NOTIFICATIONS {
        cabhPsDevInitTLVUnknownTrap,
        cabhPsDevInitTrap,
        cabhPsDevInitRetryTrap,
        cabhPsDevDHCPFailTrap,
        cabhPsDevSwUpgradeInitTrap,
        cabhPsDevSwUpgradeFailTrap,
        cabhPsDevSwUpgradeSuccessTrap,
        cabhPsDevSwUpgradeCVCFailTrap,
        cabhPsDevTODFailTrap,
        cabhPsDevCdpWanDataIpTrap,
        cabhPsDevCdpThresholdTrap,
        cabhPsDevCspTrap,
        cabhPsDevCapTrap,
        cabhPsDevCtpTrap,
        cabhPsDevProvEnrollTrap,
        cabhPsDevCdpLanIpPoolTrap
      }
      STATUS      current
      DESCRIPTION
         "These notifications indicate change in status of the Portal
         Services set of functions in a device complying with
         CableLabs CableHome(tm) specifications."
      ::= { cabhPsGroups 2 }
```

```
END
```

E.5 IPCable2Home Security (SEC) MIB requirement

requirements

The CableHome™ sec MIB MUST be implemented as defined below.

```
CABH-SEC-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
      Unsigned32,
      BITS,
      OBJECT-TYPE       FROM SNMPv2-SMI
      TruthValue,
      DisplayString,
      TimeStamp         FROM SNMPv2-TC
      OBJECT-GROUP,
      MODULE-COMPLIANCE FROM SNMPv2-CONF
      InetAddressIPv4   FROM INET-ADDRESS-MIB
      SnmpAdminString   FROM SNMP-FRAMEWORK-MIB -- RFC2571
      X509Certificate   FROM DOCS-BPI2-MIB
      clabProjCableHome FROM CLAB-DEF-MIB;

--=================================================================
--
--    History:
--
--    Date        Modified by       Reason
--    04/05/02                      Issued I01
--    09/20/02                      Issued I02
--    04/11/03                      Issued I03
--
--=================================================================

cabhSecMib MODULE-IDENTITY
    LAST-UPDATED    "200304110000Z" --April 11, 2003
    ORGANIZATION    "CableLabs Broadband Access Department"
    CONTACT-INFO
            "Kevin Luehrs
            Postal:     Cable Television Laboratories, Inc.
                  400 Centennial Parkway
                  Louisville, Colorado 80027-1266
                        U.S.A.
            Phone:  +1 303-661-9100
            Fax:    +1 303-661-9199
            E-mail: k.luehrs@cablelabs.com"
    DESCRIPTION
            "This MIB module supplies the basic management objects
            for the Security Portal Services.

            Acknowledgements:
            Roy Spitzer -  Consultant to CableLabs
            Chris Zacker      -  Broadcom Visiting Engineer"
    ::=  { clabProjCableHome 2 }

-- Textual conventions


cabhSecFwObjects  OBJECT IDENTIFIER ::= { cabhSecMib 1 }
cabhSecFwBase           OBJECT IDENTIFIER ::= { cabhSecFwObjects 1 }
cabhSecFwLogCtl   OBJECT IDENTIFIER ::= { cabhSecFwObjects 2 }
cabhSecCertObjects      OBJECT IDENTIFIER ::= { cabhSecMib 2 }
--
```

```
-- The following group describes the base objects in the Cable Home
-- Firewall.
--

cabhSecFwPolicyFileEnable OBJECT-TYPE
      SYNTAX      INTEGER {
            enable            (1),
            disable     (2)
            }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "This parameter indicates whether or not to enable the firewall
      functionality."
      DEFVAL {enable}
    ::= { cabhSecFwBase 1 }

cabhSecFwPolicyFileURL OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"This object contains the name and IP address of the policy rule set
file in a TFTP URL format. Once this object has been updated, it will
trigger the file download."

::= { cabhSecFwBase 2 }



cabhSecFwPolicyFileHash OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(20))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
"Hash of the contents of the rules set file, calculated and sent to the
PS prior to sending the rules set file.  For the SHA-1 authentication
algorithm the length of the hash is 160 bits. This hash value is
encoded in binary format."

::= { cabhSecFwBase 3 }


cabhSecFwPolicyFileOperStatus OBJECT-TYPE
SYNTAX      INTEGER {
inProgress(1),
complete (2),
completeFromMgt(3)  --- deprecated,
failed(4)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"inProgress(1) indicates that a TFTP download is underway,
complete (2) indicates that the firewall
configuration file downloaded and configured successfully,
completeFromMgt(3) This state is deprecated.
failed(4) indicates that the last attempted download failed
ordinarily due to TFTP timeout."
::= { cabhSecFwBase 4 }


cabhSecFwPolicyFileCurrentVersion OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
    "The rule set version currently operating in the PS device.
    This object should be in the syntax used by the individual
    vendor to identify software versions.  Any PS element MUST
    return a string descriptive of the current rule set file load.
    If this is not applicable, this object MUST contain an empty
    string."
      ::= { cabhSecFwBase 5 }

--
--    Firewall log parameters
--



cabhSecFwEventType1Enable OBJECT-TYPE
SYNTAX      INTEGER {
            enable (1), -- log event
            disable (2) -- do not log event
            }
MAX-ACCESS read-write
STATUS      current
DESCRIPTION
"This object enables or disables logging of type 1 firewall event
messages. Type 1 event messages report attempts from both private
and public clients to traverse the firewall that violate the Security
Policy."

DEFVAL { disable }
::= { cabhSecFwLogCtl 1 }

cabhSecFwEventType2Enable OBJECT-TYPE
SYNTAX      INTEGER {
            enable (1), -- log event
            disable (2) -- do not log event
            }
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This object enables or disables logging of type 2 firewall event
messages. Type 2 event messages report identified Denial of Service
attack attempts."

DEFVAL { disable }
::= { cabhSecFwLogCtl 2 }

cabhSecFwEventType3Enable OBJECT-TYPE
SYNTAX INTEGER {
enable (1), -- log event
disable (2) -- do not log event
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Enables or disables logging of type 3 firewall event messages. Type 3
event messages report changes made to the following firewall management
parameters: cabhSecFwPolicyFileURL, cabhSecFwPolicyFileCurrentVersion,
cabhSecFwPolicyFileEnable"

DEFVAL { disable }
::= { cabhSecFwLogCtl 3 }

cabhSecFwEventAttackAlertThreshold  OBJECT-TYPE
      SYNTAX      INTEGER     (0..65535)
```

```
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
"If the number of type 1 or 2 hacker attacks exceeds this
threshold in the period define by cabhSecFwEventAttackAlertPeriod, a
firewall message event MUST be logged with priority level 4."
DEFVAL { 65535 }
      ::= { cabhSecFwLogCtl 4 }


cabhSecFwEventAttackAlertPeriod OBJECT-TYPE
SYNTAX      INTEGER (0..65535)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Indicates the period to be used (in hours) for the
cabhSecFwEventAttackAlertThreshold. This MIB variable should always
keep track of the last  x hours of events meaning that if the variable
is set to track events for 10 hours then when the 11th hour is reached,
the 1st hour of events is deleted from the tracking log. A default
value is set to zero, meaning zero time, so that this MIB variable will
not track any events unless configured."
DEFVAL {0}

::= { cabhSecFwLogCtl 5 }



cabhSecCertPsCert OBJECT-TYPE
SYNTAX            X509Certificate
MAX-ACCESS  read-only
STATUS            current
DESCRIPTION
"The X509 DER-encoded PS certificate."
REFERENCE
"CableLabs CableHome 1.0 Specification version I01 (CH-SP-I01-020405)
 Section 11.3 Requirements (security requirements)"
::= { cabhSecCertObjects 1 }


--
-- notification group is for future extension.
--

cabhSecNotification OBJECT IDENTIFIER ::= { cabhSecMib 3 0 }
cabhSecConformance  OBJECT IDENTIFIER ::= { cabhSecMib 4 }
cabhSecCompliances  OBJECT IDENTIFIER ::= { cabhSecConformance 1 }
cabhSecGroups       OBJECT IDENTIFIER ::= { cabhSecConformance 2 }

--
--    Notification Group
--



-- compliance statements

cabhSecBasicCompliance MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
       "The compliance statement for CableHome Firewall feature."
    MODULE   --cabhSecMib
```

```
-- unconditionally mandatory groups

    MANDATORY-GROUPS {
            cabhSecGroup
    }

::= { cabhSecCompliances 3 }


cabhSecGroup OBJECT-GROUP
    OBJECTS {
            cabhSecFwPolicyFileEnable,
            cabhSecFwPolicyFileURL,
            cabhSecFwPolicyFileHash,
            cabhSecFwPolicyFileOperStatus,
            cabhSecFwPolicyFileCurrentVersion,

            cabhSecFwEventType1Enable,
            cabhSecFwEventType2Enable,
            cabhSecFwEventType3Enable,
            cabhSecFwEventAttackAlertThreshold,
            cabhSecFwEventAttackAlertPeriod,
            cabhSecCertPsCert
    }
    STATUS    current
    DESCRIPTION
      "Group of object in CableHome Firewall MIB"
    ::= { cabhSecGroups 1 }

END
```

E.6

```
CLAB-DEF-MIB DEFINITIONS ::= BEGIN
IMPORTS
      MODULE-IDENTITY,
--    X509Certificate          FROM DOCS-BPI2-MIB
    DocsX509ASN1DEREncodedCertificate   FROM DOCS-BPI2-MIB
      enterprises              FROM SNMPv2-SMI;

cableLabs MODULE-IDENTITY
      LAST-UPDATED    "0209200000Z" -- September 20, 2002
    ORGANIZATION    "CableLabs"
    CONTACT-INFO
            "Ralph Brown
            Postal: Cable Television Laboratories, Inc.
                400 Centennial Parkway
                Louisville, Colorado 80027-1266
                U.S.A.
            Phone:  +1 303-661-9100
            Fax:    +1 303-661-9199
            E-mail: r.brown@cablelabs.com"
    DESCRIPTION
            "This MIB module supplies the basic management object categories for
Cable
Labs."

    ::= { enterprises 4491 }

clabFunction            OBJECT IDENTIFIER ::= { cableLabs 1 }
clabFuncMib2            OBJECT IDENTIFIER ::= { clabFunction 1 }
clabFuncProprietary     OBJECT IDENTIFIER ::= { clabFunction 2 }
clabProject             OBJECT IDENTIFIER ::= { cableLabs 2 }
clabProjDocsis          OBJECT IDENTIFIER ::= { clabProject 1 }
clabProjPacketCable     OBJECT IDENTIFIER ::= { clabProject 2 }
clabProjCableHome       OBJECT IDENTIFIER ::= { clabProject 4 }
clabSecurity            OBJECT IDENTIFIER ::= { cableLabs 3}

clabSecCertObject OBJECT IDENTIFIER ::= { clabSecurity 1 }

clabSrvcPrvdrRootCACert       OBJECT-TYPE
--    SYNTAX            X509Certificate
    SYNTAX         DocsX509ASN1DEREncodedCertificate
    MAX-ACCESS          read-only
    STATUS        current
    DESCRIPTION
            "The X509 DER-encoded CableLabs Service Provider Root CA
Certificate."
    REFERENCE
    "CableLabs CableHome Specification Section 11"
    ::= { clabSecCertObject 1 }

clabCVCRootCACert             OBJECT-TYPE
    SYNTAX        DocsX509ASN1DEREncodedCertificate
    MAX-ACCESS          read-only
    STATUS        current
    DESCRIPTION
            "The X509 DER-encoded CableLabs CVC Root CA Certificate."
    REFERENCE
    "CableLabs CableHome Specification Section 11 for Standalone PS Elements
only"
    ::= { clabSecCertObject 2 }

clabCVCCACert           OBJECT-TYPE
    SYNTAX        DocsX509ASN1DEREncodedCertificate
```

```
    MAX-ACCESS          read-only
    STATUS        current
    DESCRIPTION
            "The X509 DER-encoded CableLabs CVC CA Certificate."
    REFERENCE
    "CableLabs CableHome Specification Section 11 for Standalone PS Elements
only"
    ::= { clabSecCertObject 3 }

clabMfgCVCCert          OBJECT-TYPE
    SYNTAX        DocsX509ASN1DEREncodedCertificate
    MAX-ACCESS          read-only
    STATUS        current
    DESCRIPTION
            "The X509 DER-encoded Manufacturer CVC Certificate."
    REFERENCE
    "CableLabs CableHome Specification Section 11 for Standalone PS Elements
only"
    ::= { clabSecCertObject 4 }

END
```

E.7 IPCable2Home QoS Portal (CQP) MIB requirements.

Requirements

The CableHome™ CQP MIBs MUST be implemented as defined below.

```
CABH-QOS-MIB DEFINITIONS ::= BEGIN
   IMPORTS
      MODULE-IDENTITY,
      OBJECT-TYPE,
      Unsigned32
                              FROM SNMPv2-SMI

      TruthValue,
      RowStatus               FROM SNMPv2-TC

      OBJECT-GROUP,
      MODULE-COMPLIANCE       FROM SNMPv2-CONF

      InetPortNumber,
      InetAddressType,
      InetAddress             FROM INET-ADDRESS-MIB

      ifIndex                 FROM IF-MIB

-- CL specs releases before RFC
      clabProjCableHome          FROM CLAB-DEF-MIB;

   cabhQosMib MODULE-IDENTITY
       LAST-UPDATED    "200303010000Z" -- March 1, 2003
       ORGANIZATION    "CableLabs Broadband Access Department"
       CONTACT-INFO
            "Kevin Luehrs
             Postal: Cable Television Laboratories, Inc.
             400 Centennial Parkway
             Louisville, Colorado 80027-1266

             U.S.A.
             Phone:  +1 303-661-9100
             Fax:    +1 303-661-9199
             E-mail: k.luehrs@cablelabs.com; mibs@cablelabs.com"
       DESCRIPTION
               "This MIB module supplies parameters for the
                configuration and monitoring of CableHome
                prioritized QoS capability."
       REVISION "200303010000Z" -- March 1, 2003
       DESCRIPTION
               "Initial version, published as RFC xxxx."
                -- RFC editor to assign xxxx
--    ::= { mib-2 xx }
-- xx to be assigned by IANA
-- CL specs releases before RFC
    ::=  { clabProjCableHome 6 }

   -- Textual conventions

   cabhQosMibObjects         OBJECT IDENTIFIER ::= { cabhQosMib 1}
   cabhPriorityQosMibObjects OBJECT IDENTIFIER ::= { cabhQosMibObjects 1 }
   cabhPriorityQosBase       OBJECT IDENTIFIER ::= { cabhPriorityQosMibObjects 1 }
   cabhPriorityQosBp         OBJECT IDENTIFIER ::= { cabhPriorityQosMibObjects 2 }
   cabhPriorityQosPs         OBJECT IDENTIFIER ::= { cabhPriorityQosMibObjects 3 }

   -- future parametric QOS
   -- cabhParamQosMibObjects  OBJECT IDENTIFIER ::= { cabhQosMibObjects 2 }
```

```
--=================================================================
--
--  Application Priority Master Table
--
--  The cabhPriorityQosMasterTable contains the list of
--  application priorities provisioned by the cable operator.
--  Applications are identified by the IANA "well-known" port
--  numbers  assigned to them.
--
--=================================================================
cabhPriorityQosMasterTable  OBJECT-TYPE
      SYNTAX SEQUENCE OF CabhPriorityQosMasterEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
        "This table contains a list of mappings for Application
         IDs to Default CableHome Priorities."
      ::= { cabhPriorityQosBase 1  }

cabhPriorityQosMasterEntry OBJECT-TYPE
        SYNTAX CabhPriorityQosMasterEntry
        MAX-ACCESS   not-accessible
        STATUS    current
        DESCRIPTION
          "An entry for mapping Application IDs to
          Default CableHome Priorities."
   INDEX { cabhPriorityQosMasterApplicationId }
   ::= { cabhPriorityQosMasterTable 1 }


CabhPriorityQosMasterEntry ::= SEQUENCE {
      cabhPriorityQosMasterApplicationId       Unsigned32,
      cabhPriorityQosMasterDefaultCHPriority   Unsigned32,
      cabhPriorityQosMasterRowStatus           RowStatus
     }

cabhPriorityQosMasterApplicationId       OBJECT-TYPE
    SYNTAX                  Unsigned32 (1..65535)
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
     "The IANA well-known port number identifying an application."
    ::= { cabhPriorityQosMasterEntry 1 }


cabhPriorityQosMasterDefaultCHPriority       OBJECT-TYPE
    SYNTAX                  Unsigned32 (0..7)
    MAX-ACCESS              read-create
    STATUS                  current
    DESCRIPTION
     "The Qos priority assigned to the application."
    ::= { cabhPriorityQosMasterEntry 2 }

cabhPriorityQosMasterRowStatus  OBJECT-TYPE
    SYNTAX                  RowStatus
    MAX-ACCESS              read-create
    STATUS                  current
    DESCRIPTION
      "The Row Status interlock for creation and deletion
       of row entries. The PS MUST NOT allow the NMS to
       set RowStatus to notInService(2). The PS MUST assign a
       RowStatus of notReady(3) to any new row created
       without a valid value for both entries. The PS will
       prevent modification of this table's columns and return
       an inconsistentValue error if the NMS attempts to make
       such modifications while RowStatus is active(1)."
    ::= { cabhPriorityQosMasterEntry 3 }



-- =================================================================
--
--  SetToFactory Object
```

```
   --
   -- This object is used to clear some of the QoS MIB tables
   --
   -- ====================================================================

cabhPriorityQosSetToFactory OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
         "Reading this object alwyas returns false(2). When this object is
          set to true(1), the PS MUST  clear all the entries in the
          cabhPriorityQosBpTable and cabhPriorityQosBpDestTable."

       ::= { cabhPriorityQosBase 2 }


   --====================================================================
   --
   --  BP Application Priority Table
   --
   --  The cabhPriorityQosBpTable contains the list of
   --  BPs, the applications implemented on each, and the priority
   --  assigned to each application.
   --
   --====================================================================

   cabhPriorityQosBpTable OBJECT-TYPE
      SYNTAX SEQUENCE OF CabhPriorityQosBpEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
         "This table contains the priorities for each of the
          discovered CableHome Host (BP) applications
          and related data."
            ::= {cabhPriorityQosBp 1}

   cabhPriorityQosBpEntry OBJECT-TYPE
      SYNTAX      CabhPriorityQosBpEntry
      MAX-ACCESS     not-accessible
      STATUS      current
      DESCRIPTION
         "List of all the discovered applications on a BP
          and their priorities identified by the PS."
      INDEX { cabhPriorityQosMasterApplicationId,
              cabhPriorityQosBpIpAddrType, cabhPriorityQosBpIpAddr }
      ::= { cabhPriorityQosBpTable 1 }

   CabhPriorityQosBpEntry ::= SEQUENCE {
         cabhPriorityQosBpIpAddrType         InetAddressType,
         cabhPriorityQosBpIpAddr             InetAddress,
         cabhPriorityQosBpApplicationId      Unsigned32,
         cabhPriorityQosBpDefaultCHPriority  Unsigned32,
         cabhPriorityQosBpIndex              Unsigned32
         }

   cabhPriorityQosBpIpAddrType   OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The type of the IP address assigned to a particular
          BP element."
      ::= { cabhPriorityQosBpEntry 1 }


   cabhPriorityQosBpIpAddr    OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The IP address assigned to a particular BP element."
      ::= { cabhPriorityQosBpEntry 2 }
```

```
cabhPriorityQosBpApplicationId   OBJECT-TYPE
   SYNTAX      Unsigned32 (1..65535)
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "The IANA well-known port number assigned to a
       particular application implemented on the
       CableHome Host device in which this BP resides."
   ::= { cabhPriorityQosBpEntry 3 }


cabhPriorityQosBpDefaultCHPriority  OBJECT-TYPE
   SYNTAX      Unsigned32 (0..7)
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "The priority assigned to a particular application
       implemented on CableHome Host device in which this
       BP resides. The PS populates this entry according
       to the Application Priority Master Table."
   ::= { cabhPriorityQosBpEntry 4 }

cabhPriorityQosBpIndex  OBJECT-TYPE
   SYNTAX      Unsigned32 (1..65535)
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
     "The unique identifier for a particular row in the
      BP Application Priority Table. This identifier is
      used as an index into the 'nested' Destination
      Priority Table."
        ::= { cabhPriorityQosBpEntry 5 }


--================================================================
--
--  Destination Priority Table
--
--  The cabhPriorityQosDestListTable contains the list of
--  provisioned destinations (IP address and port number) to
--  which a BP can send traffic with a special Qos
--  priority. Any application listed in the BP Application
--  Priority Table can be provisioned with a destination specific
--  priority in this table.
--
--================================================================

cabhPriorityQosBpDestTable OBJECT-TYPE
   SYNTAX SEQUENCE OF CabhPriorityQosBpDestEntry
   MAX-ACCESS     not-accessible
   STATUS      current
   DESCRIPTION
      "This table contains the priorities based on
       sessions established by a BP, identified by
       destination IP address and port number. It
       is indexed with a unique identifier for rows
       in the BP Application Priority Table
       (cabhPriorityQoSBpTable."
       ::= {cabhPriorityQosBp 2}

cabhPriorityQosBpDestEntry OBJECT-TYPE
   SYNTAX      CabhPriorityQosBpDestEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
      "List of Destination IP addresses and port numbers
       for an application to which special Qos
       priority is provisioned."
   INDEX { cabhPriorityQosBpIndex, cabhPriorityQosBpDestIndex }
   ::= { cabhPriorityQosBpDestTable 1 }

CabhPriorityQosBpDestEntry ::= SEQUENCE {
```

```
    cabhPriorityQosBpDestIndex            Unsigned32,
    cabhPriorityQosBpDestIpAddrType       InetAddressType,
    cabhPriorityQosBpDestIpAddr           InetAddress,
    cabhPriorityQosBpDestPort             InetPortNumber,
    cabhPriorityQosBpDestIpPortPriority   Unsigned32
      }


cabhPriorityQosBpDestIndex OBJECT-TYPE
   SYNTAX      Unsigned32 (1..65535)
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
     "The locally unique index into the Destination
      Priority Table."
   ::= { cabhPriorityQosBpDestEntry 1 }

cabhPriorityQosBpDestIpAddrType     OBJECT-TYPE
   SYNTAX      InetAddressType
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "The type of the Destination IP Address."
   ::= { cabhPriorityQosBpDestEntry 2 }

cabhPriorityQosBpDestIpAddr   OBJECT-TYPE
   SYNTAX      InetAddress
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
     "The Destination IP address of the device to which
      an application-session is established by a BP and
      a special Qos priority is provisioned."
   ::= { cabhPriorityQosBpDestEntry 3 }

cabhPriorityQosBpDestPort     OBJECT-TYPE
   SYNTAX      InetPortNumber
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
     "The port number on a IP device to which
      an application-session is established by a BP and
      a special Qos priority is provisioned."
   ::= { cabhPriorityQosBpDestEntry 4 }

cabhPriorityQosBpDestIpPortPriority    OBJECT-TYPE
   SYNTAX      Unsigned32 (0..7)
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
     "The Qos priority assigned to a particular
      application-session (identified by destination IP
      and Port) on a BP."
   ::= { cabhPriorityQosBpDestEntry 5 }


--================================================================
--
--  PS Interface Attributes Table
--
--  The cabhPriorityQosPsIfAttribTable contains the number of
--  media access priorities and number of queues associated with
--  each LAN interface in the Residential Gateway.
--
--================================================================

cabhPriorityQosPsIfAttribTable   OBJECT-TYPE
        SYNTAX SEQUENCE OF CabhPriorityQosPsIfAttribEntry
        MAX-ACCESS   not-accessible
        STATUS    current
        DESCRIPTION
         "This table contains the number of media
          access priorities and number of queues associated
```

```
          with each LAN interface in the Residential Gateway."
         ::= { cabhPriorityQosPs 1  }

cabhPriorityQosPsIfAttribEntry      OBJECT-TYPE
        SYNTAX    CabhPriorityQosPsIfAttribEntry
        MAX-ACCESS   not-accessible
        STATUS    current
        DESCRIPTION
          "Number of media access priorities and number
           of queues for each LAN interface in the
           Residential Gateway. This table applies only
           to interfaces through which data flows."
        INDEX { ifIndex }
        ::= { cabhPriorityQosPsIfAttribTable 1 }

CabhPriorityQosPsIfAttribEntry ::= SEQUENCE {
   cabhPriorityQosPsIfAttribIfNumPriorities   Unsigned32,
        cabhPriorityQosPsIfAttribIfNumQueues   Unsigned32
}

cabhPriorityQosPsIfAttribIfNumPriorities OBJECT-TYPE
       SYNTAX     Unsigned32 (1..8)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
        "The number of media access priorities supported
         by this LAN interface."
::= { cabhPriorityQosPsIfAttribEntry 1 }

cabhPriorityQosPsIfAttribIfNumQueues OBJECT-TYPE
       SYNTAX     Unsigned32 (1..8)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
        "The number of queues associated with this LAN
         interface."
::= { cabhPriorityQosPsIfAttribEntry 2 }


-- Placeholder for notifications/traps.
--

cabhQosNotification         OBJECT IDENTIFIER  ::= { cabhQosMib 2 }
cabhPriorityQosNotification OBJECT IDENTIFIER  ::= {
cabhQosNotification 1 }



--
-- Conformance definitions
--
cabhQosConformance          OBJECT IDENTIFIER  ::= { cabhQosMib 3 }
cabhPriorityQosConformance  OBJECT IDENTIFIER  ::= {
cabhQosConformance 1 }
cabhPriorityQosGroups       OBJECT IDENTIFIER  ::= {
cabhPriorityQosConformance 1 }
cabhPriorityQosCompliances  OBJECT IDENTIFIER  ::= {
cabhPriorityQosConformance 2 }


-- ==================

-- compliance statements

cabhPriorityQosCompliance MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
      "The compliance statement for devices that implement
       CableHome 1.1 PriorityQos capability."

    MODULE   --cabhPriorityQosMib
```

```
-- unconditionally mandatory groups

    MANDATORY-GROUPS {
            cabhPriorityQosGroup
    }

::= { cabhPriorityQosCompliances 1}


cabhPriorityQosGroup OBJECT-GROUP
    OBJECTS {
    cabhPriorityQosMasterDefaultCHPriority,
    cabhPriorityQosMasterRowStatus,
    cabhPriorityQosSetToFactory,
    cabhPriorityQosBpIpAddrType,
    cabhPriorityQosBpIpAddr,
    cabhPriorityQosBpApplicationId,
    cabhPriorityQosBpDefaultCHPriority,
    cabhPriorityQosBpIndex,
    cabhPriorityQosBpDestIpAddrType,
    cabhPriorityQosBpDestIpAddr,
    cabhPriorityQosBpDestPort,
    cabhPriorityQosBpDestIpPortPriority,
    cabhPriorityQosPsIfAttribIfNumPriorities,
    cabhPriorityQosPsIfAttribIfNumQueues
    }
    STATUS    current
    DESCRIPTION
        "Group of objects for CableHome Application
         Priority MIB."
    ::= { cabhPriorityQosGroups 1 }

END
```

Appendix I Media Access Priority Mapping Examples

This Recommendation defines a prioritized QoS system in which traffic over the shared media is prioritized based on the assigned packet priority. Since different shared media technologies support varying numbers of media access priorities, IPCable2Home defines a mapping scheme to translate Generic IPCable2Home Priorities to a set of values called IPCable2Home Media Access Priorities. IPCable2Home Media Access Priority values describe the level of preference that a packet should get when accessing the shared media. The number of preference levels correspond to the available number of media access priorities supported by a given media technology. The higher the IPCable2Home Media Access priority value for the packet, the higher the preference it should get to access the shared media. IPCable2Home Media Access Priority mapping is separate and distinct from native media access priority mappings defined for the shared media technologies. These native mappings are accomplished at layer-2 of each device. Therefore, regardless of the shared media technology, the packets must be given the desired relative preferential access to the shared media, as required by the IPCable2Home Media Access Priority mapping. Table I-1, Table I-2, and Table I-3 provide mapping examples for a few of the shared media access technologies.

I.1 Ethernet

Ethernet does not provide differentiation between packets and hence only supports one priority.

Table I-1 Ethernet Mappings

Generic IPCable2Home Priority	IPCable2Home Media Access Priority mapping	Native Ethernet Media Access priority mapping
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0

As shown in Table I-1, no special mapping adjustments are required.

I.2 HomePlug

HomePlug supports 4 media access priorities.

Table I-2 HomePlug Mappings

Generic IPCable2Home Priority	IPCable2Home Media Access Priority mapping	Native HomePlug Media Access priority mapping
0	0	1
1	0	0
2	1	0
3	1	1
4	2	2
5	2	2
6	3	3
7	3	3

As shown in Table I-2, HomePlug mapping gives channel access preference to Generic IPCable2Home Priority 0, relative to Generic IPCable2Home Priorities 1 and 2. However, IPCable2Home Media Access Priority mapping requires that Generic IPCable2Home Priority 2 be given higher access, relative to Generic IPCable2Home Priorities 0 and 1, and Generic IPCable2Home Priorities 0 and 1 be given equal access rights. Hence, the vendor must insure that the packets are given the desired relative preferential access to the shared media as required by the IPCable2Home Media Access Priority mapping.

I.3 HomePNA

HomePNA supports 8 media access priorities.

Table I-3 HomePNA Mappings

Generic IPCable2Home Priority	IPCable2Home Media Access Priority mapping	Native HomePNA Media Access priority mapping
0	0	2
1	1	0
2	2	1
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7

As shown in Table I-3, HomePNA mapping gives channel access preference to Generic IPCable2Home Priority 0 relative to Generic IPCable2Home Priorities 1 and 2. However, IPCable2Home Media Access Priority mapping requires that Generic IPCable2Home Priority 2 be given higher access relative to Generic IPCable2Home Priorities 0 and 1, and Generic IPCable2Home Priority 1 be given higher access relative to Generic IPCable2Home Priority 0. Hence, the vendor must insure that the packets are given the desired relative preferential access to the shared media as required by the IPCable2Home Media Access Priority mapping.