INTERNATIONAL TELECOMMUNICATION UNION

ITU-T J.191

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(03/2004)

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

Cable modems

IP feature package to enhance cable modems

ITU-T Recommendation J.191

ITU-T Recommendation J.191

IP feature package to enhance cable modems

Summary

This Recommendation provides a set of IP-based features that may be added to a cable modem or incorporated into a stand-alone device, that will enable cable operators to provide an additional set of enhanced services to their customers including support for IPCablecom Quality of Service (QoS), enhanced security, additional management and provisioning features, and improved addressing and packet handling.

Source

ITU-T Recommendation J.191 was approved on 15 March 2004 by ITU-T Study Group 9 (2001-2004) under the ITU-T Recommendation A.8 procedure.

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

CONTENTS

ITU-T Recommendation J.191

IP feature package to enhance cable modems

1 Scope

This Recommendation provides a set of IP-based features that may be added to a cable modem or incorporated into a stand-alone device, that will enable cable operators to provide an additional set of enhanced services to their customers including support for IPCablecom Quality of Service (QoS), enhanced security, additional management and provisioning features, and improved addressing and packet handling. This Recommendation implements the IPCable2Home Domain defined in ITU-T Rec. J.190.

2 References

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

2.1 Normative references

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

– ITU-T Recommendation J.161 (2001), *Audio codec requirements for the provision of bidirectional audio service over cable television networks using cable modems*.

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

– ITU-T Recommendation J.170 (2002), *IPCablecom security specification*.

– ITU-T Recommendation X.509 (2000) | ISO/IEC 9594-8:2001, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

– ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.

– ISO/IEC 15802-3:1998 (ANSI/IEEE Std 802.1D), *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common Specifications – Part 3: Media access control (MAC) bridges*.

– FIPS 140-2-2001, *Security Requirements for Cryptographic Modules*.

– FIPS 180-2-2002, *Secure hash standard*.

– FIPS 186-2-2000, *Digital signature standard (DSS)*.

– IETF RFC 768 (1980), *User Datagram Protocol (UDP)*.

– IETF RFC 792 (1981), *Internet Control Message Protocol, DARPA Internet Program, Protocol specification*.

– IETF RFC 868 (1983), *Time Protocol*.

- IETF RFC 1034 (1987), *Domain Names – Concepts and Facilities*.
- IETF RFC 1035 (1987), *Domain Names – Implementation and Specification*.
- IETF RFC 1122 (1989), *Requirements for Internet Hosts – Communication layers*.
- IETF RFC 1157 (1990), *A Simple Network Management Protocol (SNMP)*.
- IETF RFC 1350 (1992), *The TFTP Protocol (Revision 2)*.
- IETF RFC 1901 (1996), *Introduction to community-based SNMPv2*.
- IETF RFC 2011 (1996), *SNMPv2 Management Information Base for the Internet Protocol using SMIv2*.
- IETF RFC 2013 (1996), *SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2*.
- IETF RFC 2131 (1997), *Dynamic Host Configuration Protocol*.
- IETF RFC 2132 (1997), *DHCP Options and BOOTP Vendor Extensions*.
- IETF RFC 2233 (1997), *The Interfaces Group MIB using SMIv2*.
- IETF RFC 2236 (1997), *Internet Group Management Protocol, Version 2*.
- IETF RFC 2315 (1998), *PKCS #7: Cryptographic Message Syntax Version 1.5*.
- IETF RFC 2437 (1998), *PKCS #1: RSA Cryptography Specifications Version 2.0*.
- IETF RFC 2576 (2000), *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*.
- IETF RFC 2578 (1999), *Structure of Management Information Version 2 (SMIv2)*.
- IETF RFC 2579 (1999), *Textual Conventions for SMIv2*.
- IETF RFC 2580 (1999), *Conformance Statements for SMIv2*.
- IETF RFC 2669 (1999), *DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems*.
- IETF RFC 2670 (1999), *Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces*.
- IETF RFC 2786 (2000), *Diffie-Helman USM Key Management Information Base and Textual Convention*.
- IETF RFC 2863 (2000), *The Interfaces Group MIB*.
- IETF RFC 3022 (2001), *Traditional IP Network Address Translator (Traditional NAT)*.
- IETF RFC 3280 (2002), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.
- IETF RFC 3291 (2002), *Textual Conventions for Internet Network Addresses*.
- IETF RFC 3396 (2002), *Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)*.
- IETF RFC 3412 (2002), *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*.
- IETF RFC 3413 (2002), *Simple Network Management Protocol (SNMP) applications*.
- IETF RFC 3414 (2002), *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*.

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

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

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

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

2.2 Informative references

– ITU-T Recommendation J.190 (2002), *Architecture of MediaHomeNet that supports cable based services*.

– IETF RFC 347 (1972), *Echo Process*.

– IETF RFC 1949 (1996), *Scalable Multicast Key Distribution*.

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

– IETF RFC 2979 (2000), *Behavior of and Requirements for Internet Firewalls*.

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

– draft-ietf-ipcdn-bpiplus-mib-12 INTERNET DRAFT – *DOCSIS Baseline Privacy Plus MIB – Management Information Base for DOCSIS Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus*, October 2003.

– ICSA, Inc.: Firewall Buyer's Guide, 1998, *www.icsalabs.com*.

3 Terms and definitions

This Recommendation defines the following terms:

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

3.2 Call Management Server (CMS): [IPCablecom] Controls the audio connections. Also called a Call Agent in MGCP/SGCP terminology.

3.3 dynamic Quality of Service (DQoS): [IPCablecom] Assigned on the fly for each communication depending on the QoS requested.

3.4 Embedded Multimedia Terminal Adapter (E-MTA): [IPCablecom] A single node that contains both an MTA and a cable modem.

3.5 IP enhanced cable modem: A cable modem that has been enhanced by the addition of the IP features of this Recommendation.

3.6 Portal Service (PS): A functional element that provides management and translation functions between the HFC and Home.

3.7 LAN IP device: A LAN IP Device is representative of a typical IP device expected to reside in the home, and that contains a TCP/IP stack as well as a DHCP client.

3.8 pass-through: A sub-function of the CAP, the Pass-through function bridges packets on the WAN-Data side of the CAP to the LAN-Pass side unchanged.

3.9 Stand-alone Multimedia Terminal Adapter (S-MTA): A single node that contains an MTA and a non-DOCSIS MAC (e.g., Ethernet).

4 Abbreviations, acronyms and conventions

4.1 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

ASP	Application-Specific Proxy
CA	Certificate Authority
CAP	Cable Address Portal
CAT	Cable Address Translation
CDC	Cable DHCP Client
CDP	Cable DHCP Portal
CM	Cable Modem
CMP	Cable Management Portal
CMS	Call Management Server
CMTS	Cable Modem Termination System
C-NAPT	Cable Network Address and Portal Translation
C-NAT	Cable Network Address Translation
CNP	Cable Naming Portal
CQoS	Cable Quality of Service
CQP	Cable QoS Portal
CRL	Certificate Revocation List
CSP	Cable Security Portal
CTP	Cable Testing Portal
CVC	Code Verification Certificate
CVS	Code Verification Signature
CxP	Cable PS Sub-function
DER	Distinguished Encoding Rules
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DOCSIS	Data-Over-Cable Service Interface Specification
DQoS	Dynamic Quality of Service (IPCablecom)
E-MTA	Embedded Multimedia Terminal Adapter
FTP	File Transfer Protocol
FW	Firewall
GMT	Greenwich Mean Time
HEX	Hexadecimal

HFC	Hybrid Fibre Coax
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
IP	Internet Protocol
KDC	Key Distribution Centre
LAN-Pass	Pass-through LAN address
LAN-Trans	Translated LAN address
MAC	Media Access Control
MGCP	Media Gateway Control Protocol
MIB	Management Information Base
MTA	Multimedia Terminal Adapter
NAPT	Network Address and Portal Translation
NAT	Network Address Translation
NCS	Network-based Call Signalling
NMS	Network Management System
OID	Object Identifier
OSI	Open Systems Interconnection
OSS	Operations Support System
PDU	Protocol Data Unit
PING	Packet Inter-Network Grouper
PKI	Public Key Infrastructure
PKINIT	Public-Key Cryptography for Initial Authentication
PS	Portal Service
PS WAN-Man	Portal Service element WAN management interface
PS WAN-Data	Portal Service element WAN data interface
QoS	Quality of Service
RFC	Request for Comments
RSA	Rivest, Shamir, Adleman
SHA-1	Secure Hash Algorithm 1
S-MTA	Stand-alone 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
TLV	Type-Length-Value

UDP	User Datagram Protocol
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

If this Recommendation is implemented, the key words "MUST" and "SHALL" as well as "REQUIRED" are to be interpreted as indicating a mandatory aspect of this Recommendation. The key words indicating a certain level of significance of a particular requirement that are used throughout this Recommendation are summarized as follows.

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

5 Reference architecture

This Recommendation provides a set of IP-based features that may be added to a Cable Modem, or implemented in a stand-alone device, that will enable cable operators to provide an additional set of enhanced services to their customers. These IP-based features reside in a logical element called the Portal Service (PS or just Portal). A device that contains these enhanced features is referred to as a Residential Gateway, which is an implementation of IPCable2Home as described in ITU-T Rec. J.190.

Major areas and features are as follows:

– Management and provisioning:
  - remote management and configuration of the residential gateway device;
  - simple residential gateway management proxy for IP-based home devices;
  - hands off provisioning for residential gateway devices.

– Addressing and packet handling:
  - one-to-many address translation for home devices;
  - one-to-one address translation for home devices;
  - non translated addressing for home devices (for NAT phobic applications);
  - HFC traffic protection from in-home device intra-communications;
  - home addressing support during HFC outage;
  - simple DNS server in the residential gateway.
– Quality of Service (QoS):
  - residential gateway device transparent bridging functionality for IPCablecom QoS messaging from/to IPCablecom compliant applications.
– Security:
  - residential gateway device authentication;
  - secure residential gateway management messages;
  - secure download of configuration and software files;
  - secure QoS on the HFC link;
  - remote residential gateway firewall management.

Communication across the WAN and LAN is IPv4 based, leveraging specific protocols defined throughout the remainder of this Recommendation. Compliant devices MUST implement version 4 of the Internet Protocol suite (IPv4).

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

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

5.1 Logical reference architecture

As shown in Figure 5-1, this clause introduces the logical concepts of the IPCable2Home domain, logical elements, and the Home Access (HA) device class.

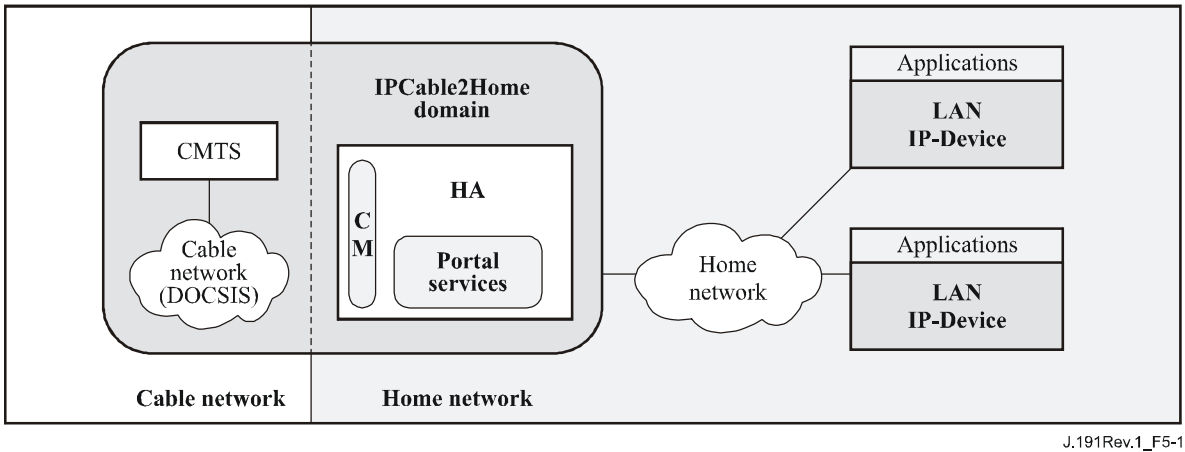

Figure 5-1/J.191 – Key logical concepts

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 compliant. Elements that reside within the IPCable2Home domain (i.e., compliant elements) are directly manageable by operators.

5.1.2 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 IPCable2Home compliant messages. IPCable2Home logical elements operate at the network 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 manage and deliver services over IPCable2Home networks. This Recommendation defines a single logical entity known as the Portal Service (PS) element.

5.1.2.1 Portal Services (PS)

A portal is a logical element that provides in-premise and aggregated security, management, provisioning, and addressing services. Three portal service sets of functions are defined. They are the management set of functions, the Quality of Service (QoS) set of functions and the security set of functions. The PS logical element forms the foundation of the logical reference architecture.

5.1.3 Device classes

The architecture framework also uses the concept of device classes to lend tangible context to the logical elements and combinations of these logical elements. The IPCable2Home concept of device class places no restrictions on physical devices or combinations of logical elements within physical devices. Device classes provide an informative way of depicting collections of logical elements but are not considered definitive or restrictive.

In IPCable2Home, the HA device class represents the physical location of the PS logical element and it enables the network elements within the IPCable2Home domain to interact with LAN IP Devices. The HA device has a single Cable Modem RF-compliant interface, a single PS logical element, and may have zero or more LAN IP interfaces.

This Recommendation also refers to LAN IP Devices. 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.

5.1.3.1 Embedded PS and stand-alone PS

The two primary components possible within a Residential Gateway, the DOCSIS Cable Modem (CM) and the Portal Services (PS) element may use shared or independent hardware and software resources. It is this resource sharing between the CM and PS that distinguishes the stand-alone PS from an Embedded PS.

A stand-alone PS MUST NOT share hardware or software components with a CM. The separation of the CM from the stand-alone 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 stand-alone PS.

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

Figure 5-2 illustrates both the stand-alone and Embedded PS. In both of these cases, the combination of a CM and a PS is considered to embody the concept of the HA device. In other words, an HA consists of a single CM and a single PS. This one-PS-per-CM assumption holds true even for stand-alone PS, i.e., it is assumed that only one stand-alone PS connects to a CM.

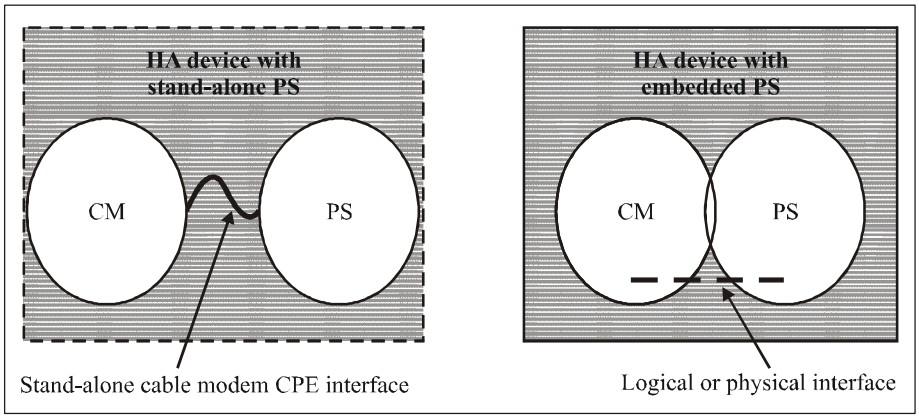

Figure 5-2/J.191 – Stand-alone and Embedded PS

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).

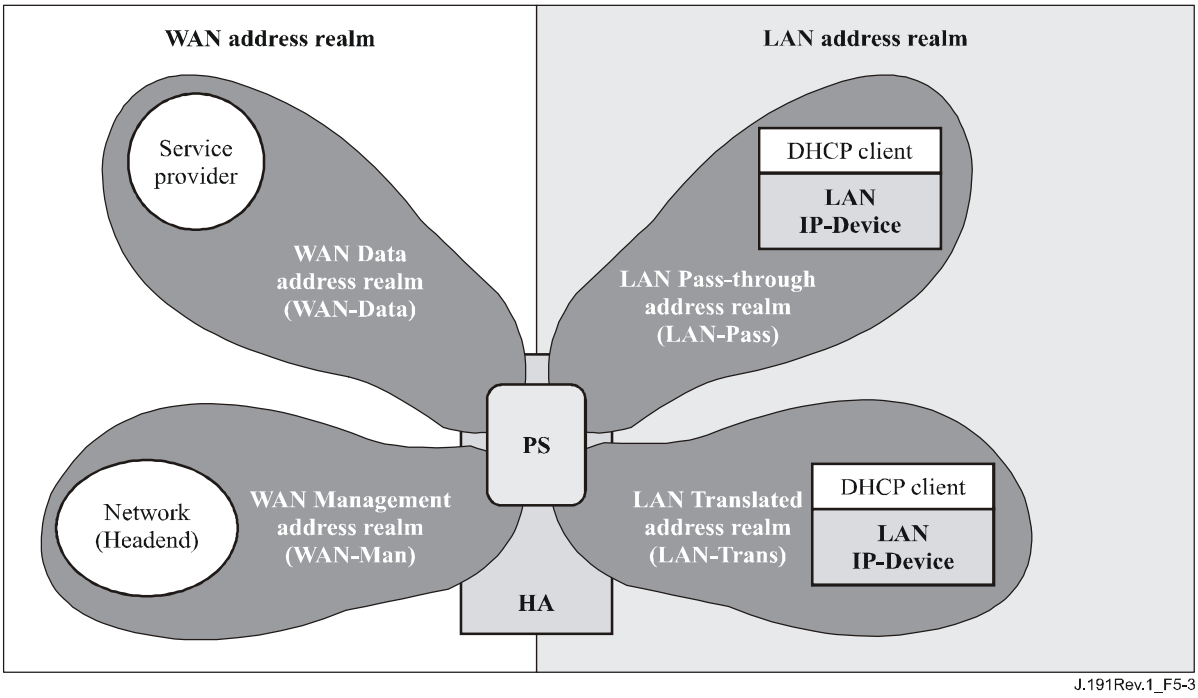

Figure 5-3/J.191 – Address realms

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 Pass-through 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 LAN IP Devices 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 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 Pass-through Address Realm (LAN-Pass) is intended to carry subscriber application traffic, such as traffic between LAN IP Devices and Internet hosts, on the home network, the 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 Pass-through 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 use translated addresses from the LAN Translated Address Realm (LAN-Trans).

Physical LAN interfaces in the PS are assigned an index in accordance with the Interfaces Group MIB [RFC 2233] as described in 6.3.8. A virtual LAN interface aggregating the physical LAN interfaces is also defined for the PS in 6.3.8. 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 services (layer-3 and above) defined for IPCable2Home. These Functions are located within the PS, LAN IP Devices, and the Headend. There are IPCable2Home Functions for each of the major specification areas: Provisioning and Management, Security, and Quality of Service. The Functions for Provisioning and Management, Security, and QoS are briefly introduced in the following three clauses.

5.2.1 Management functions

To support the provisioning and management of IP LAN-Devices within the home, three Management Function classes are defined:

- Management Server Functions;
- Management Client Functions;
- Management Portal Functions.

Several of the Management Server Functions reside within the Headend (HE). Management Client Functions are typically found within LAN IP Devices. Management Portal Functions are located within the PS logical element and may include server-like, client-like, and relay-like functionality to aggregate and translate messages between the Headend and LAN IP Devices. Examples of Management Server, PS and Client functions are introduced in Tables 5-1, 5-2, and 5-3 and are illustrated in Figure 5-4.

Table 5-1/J.191 – Management server function description

Management server functions	Description
Headend DHCP Server	The DHCP server is a Headend component that provides address information for the WAN-Man and WAN-Data address realms to the PS.
Headend Management Messaging Server	The Headend management messaging, download, event notification servers including protocols such as SNMP, SYSLOG, and TFTP.

Table 5-2/J.191 – Management and provisioning portal function description

Management portal functions	Description
Cable Address Portal (CAP)	Within the PS, the CAP interconnects the WAN and LAN address realms for data traffic (see CAT/Pass-through).
Cable Address Translation (CAT)	A sub-function of the CAP, a CAT translates addresses on the WAN-Data side of the CAP to addresses within a single logical subnet on the LAN-Trans side.
Pass-through	A sub-function of the CAP, the Pass-through function bridges packets on the WAN-Data side of the CAP to the LAN-Pass side unchanged.
Cable Management Portal (CMP)	The function that provides an interface between the operator and the PS-database.
Cable 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.
Cable Naming Portal (CNP)	The CNP provides a simple DNS service for LAN IP Devices requiring naming services.
Cable Testing Portal (CTP)	The CTP provides a remote means to initiate pings and loopbacks within the LAN.

Table 5-3/J.191 – Management client function description

Management client functions	Description
LAN IP Device DHCP Client	The Cable DHCP client function is an in-home component used during the LAN IP Device provisioning process to dynamically request IP addresses and other logical element configuration information.
LAN IP Device Loopback responder	Within LAN IP Device, the loopback responder loops data sourced from the CTP loopback function back to the CTP loopback function.

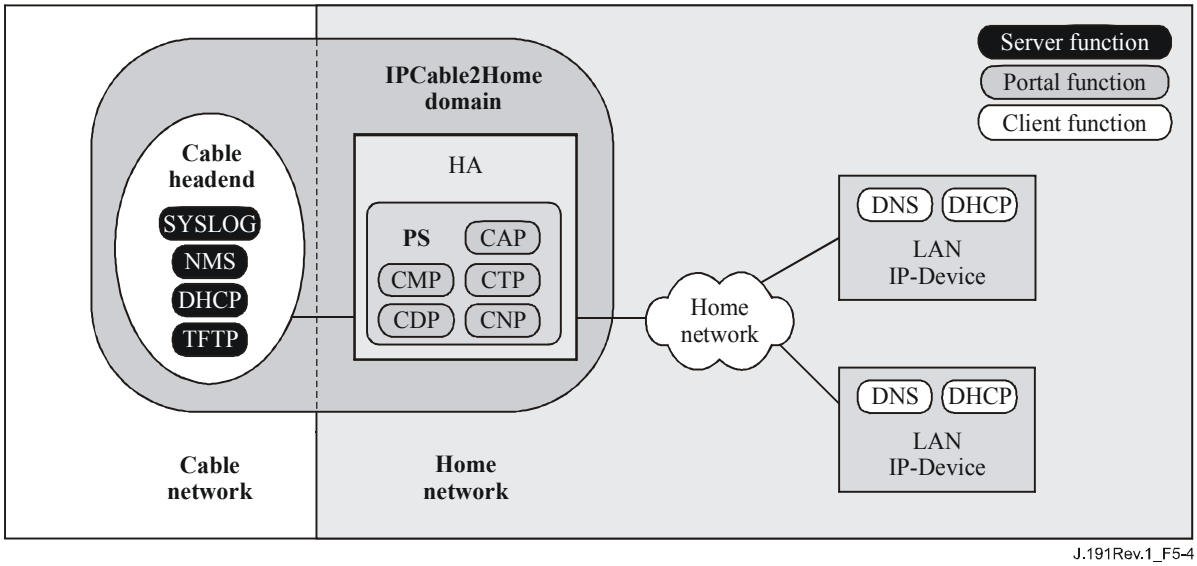

Figure 5-4/J.191 – Management elements

5.2.2 Security functions

To support the IPCable2Home security requirements, two classes of Security Functions are defined:

- Security Server Functions (Kerberos, Key Distribution Centre);
- Security Portal Functions.

Security Server Functions reside within the Headend (HE), and the Security Portal Functions consist of client-like functions residing within the PS. Examples of Security Server and Security Portal functions are introduced in Tables 5-4 and 5-5, and are illustrated in Figure 5-5.

Table 5-4/J.191 – Security portal function description

Security portal functions	Description
Cable 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 defined by IPCable2Home. 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/J.191 – Security server function description

Security server functions	Description
Headend KDC Servers	The Headend KDC servers provide security services to the CSP and include functions that participate in the authentication and key exchange processes defined by IPCable2Home.

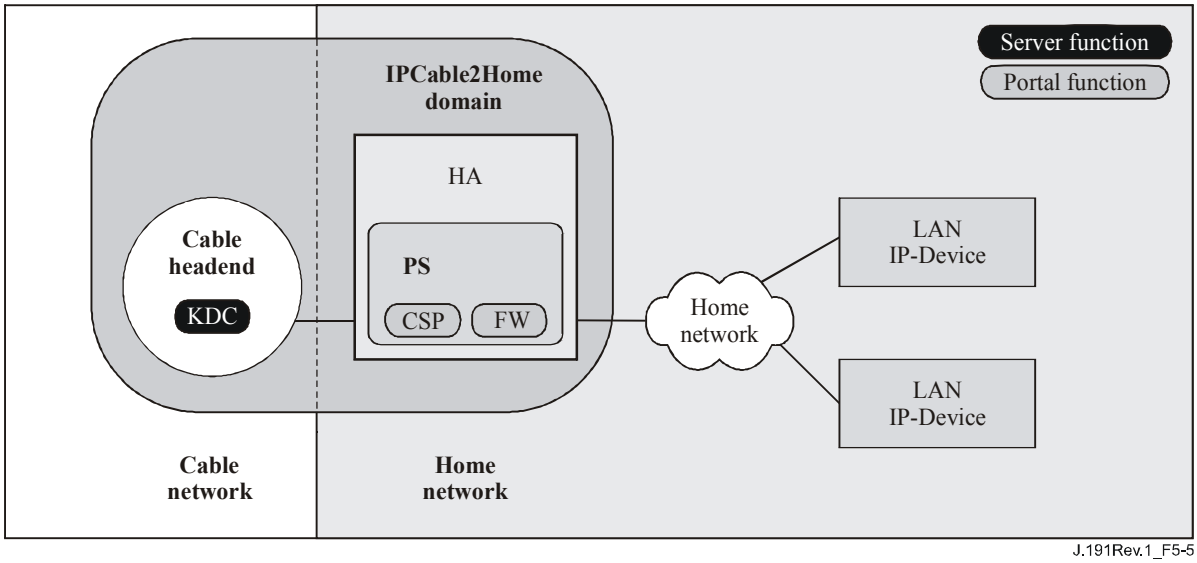

Figure 5-5/J.191 – Security elements

5.2.3 QoS functions

The QoS architecture is composed of a single PS-based functional entity known as the IPCable2Home QoS Portal (CQP). The CQP provides transparent bridging for QoS messaging between IPCablecom applications and the IPCablecom QoS infrastructure on the cable network.

5.3 IPCable2Home messaging interface model

The communication between the functions in IPCable2Home network elements and LAN-IP-Devices occurs on messaging interfaces. The types of messaging interfaces are differentiated by the elements that are involved in the communication. The messaging interfaces are illustrated in Figure 5-6.

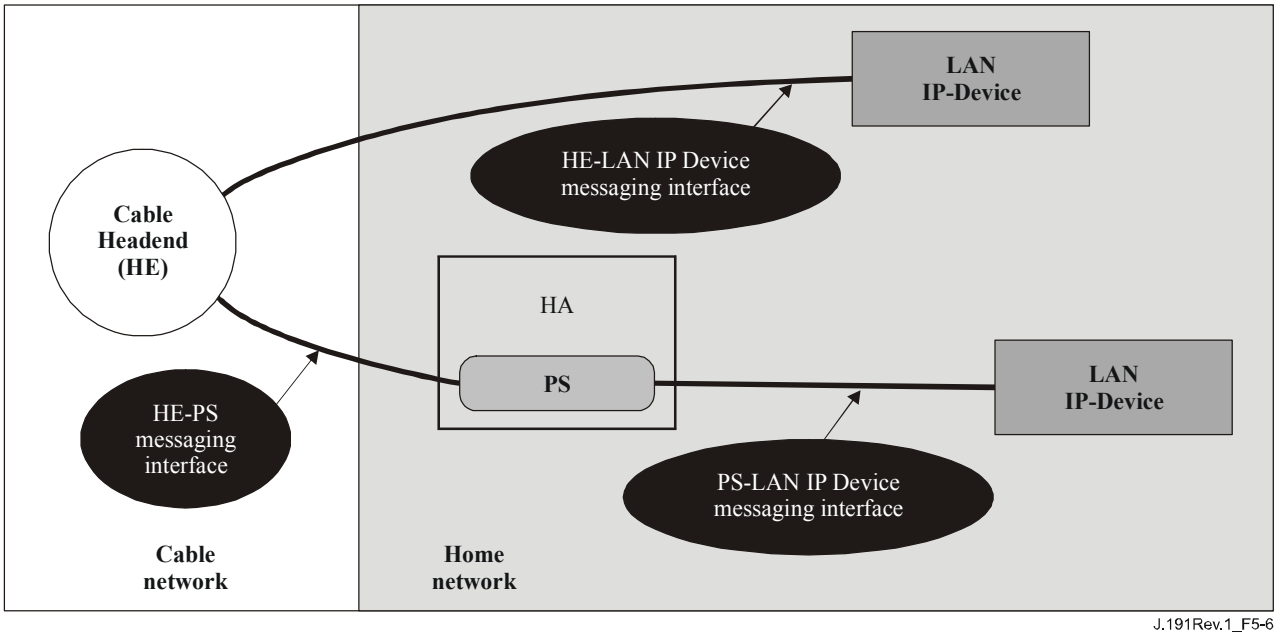

Figure 5-6/J.191 – Reference interfaces

The IPCable2Home Messaging interfaces are summarized in Table 5-6.

Table 5-6/J.191 – Valid interface paths for each functionality

Functionality	Protocol	Interface		
		HE-PS	HE-LAN IP Device	PS-LAN IP Device
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	This Recommendation This Recommendation	Unspecified	Unspecified
Event Notification	SNMP SYSLOG	This Recommendation This Recommendation	Unspecified	Unspecified
QoS	IPCablecom QoS Protocols	Unspecified	IPCablecom	Unspecified
Security (key distribution)	Kerberos	This Recommendation	Unspecified	Unspecified
Security (authentication)	Kerberos	This Recommendation	Unspecified	Unspecified
Ping	ICMP	This Recommendation	Unspecified	This Recommendation
Loopback/Echo	UDP/TCP	Unspecified	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 PS functions (CAP, CDP, CMP, etc.). These functions must have 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 elements.

Figure 5-7 shows the relationship between the database and the PS functions, Table 5-7 describes the typical information associated with each of these functions. Figure 5-8 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.

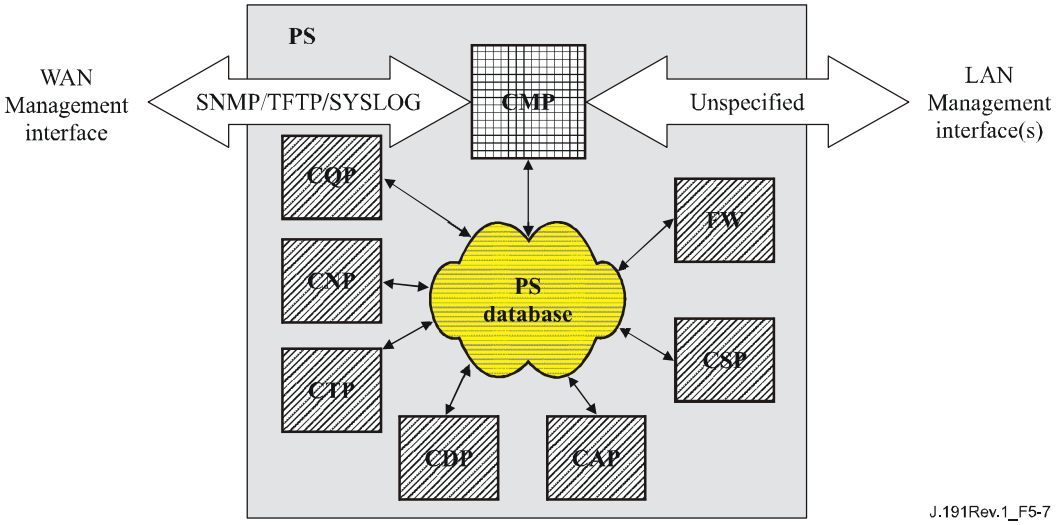

Figure 5-7/J.191 – PS function and database relationship

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 IPCable2Home Functions within the PS.

Table 5-7/J.191 – 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 management functions
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 behaviour of the Firewall (rule set) and firewall logging
Event information	Information associated with the local log for all general events, traps, etc.

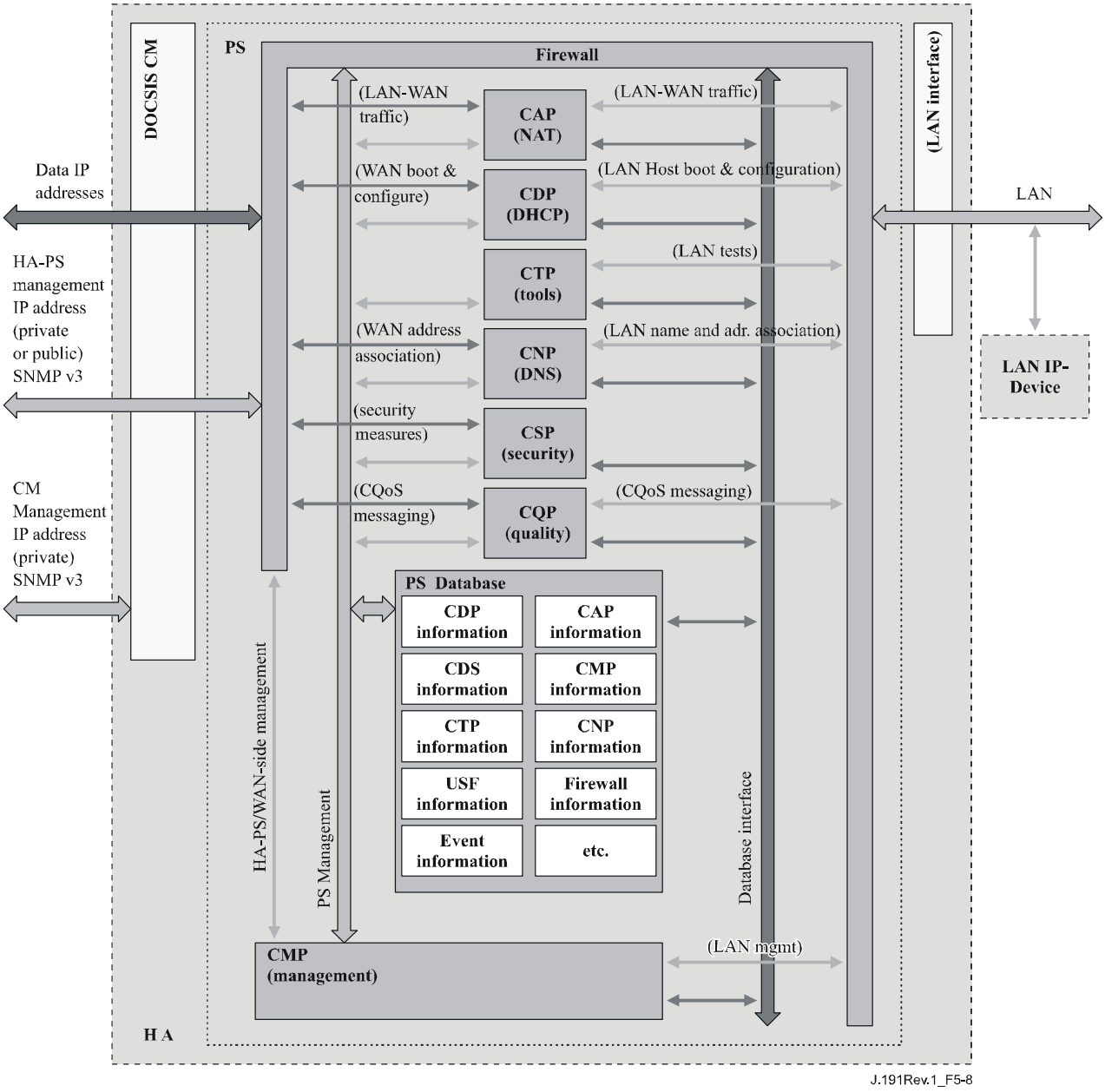

Figure 5-8/J.191 – PS database detailed example implementation

The PS is 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 WAN side network elements, and 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 System (DNS) manager. 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 support the DHCP server in the LAN-Trans realm and 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 Pass-through 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 IPCablecom Quality of Service (QoS) through the PS. The CQP, acting as a transparent bridge, forwards IPCablecom compliant QoS messaging between IPCablecom applications and the IPCablecom QoS infrastructure.

5.5 IPCable2Home operational models

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 Cable Modem infrastructure, and within an Extended IPCable2Home infrastructure. The Extended IPCable2Home infrastructure builds upon the Cable Modem infrastructures to enable additional services, and incorporates a number of capabilities that are similar to those within an 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 does not receive the information needed to make a provisioning mode determination, it will operate with reduced functionality in Dormant CableHome Mode.

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

- NmAccess Mode;
- Coexistence Mode.

Figure 5-9 illustrates the various PS operational modes along with the associated triggers for each. See 6.3.6.1.1.

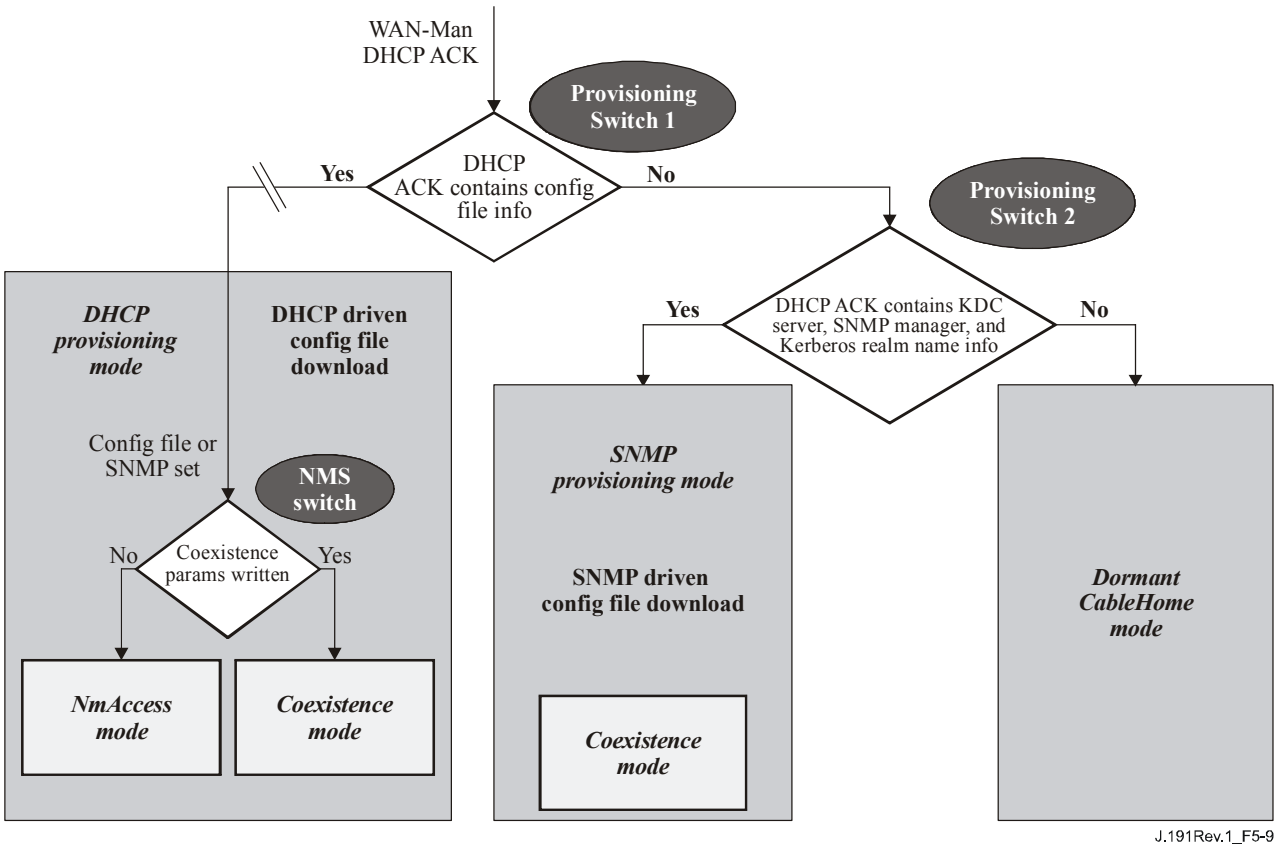

Figure 5-9/J.191 – PS operational modes

If PS Configuration File information (server location and file name) is provided, and Kerberos server information is not provided, to the PS in the DHCP ACK issued by the cable network DHCP server, the PS will operate in DHCP Provisioning Mode. When in DHCP Provisioning Mode, the PS may operate in one of two Network Management Modes (NmAccess and Coexistence). Within DHCP Provisioning Mode, the PS will operate in NmAccess Network Management Mode by default, but can be configured by the NMS to operate in Coexistence Mode.

If Kerberos server information is provided, and PS Configuration File information is not provided, to the PS in the DHCP ACK issued by the cable network DHCP server, the PS will operate in SNMP Provisioning Mode. When operating in the SNMP Provisioning Mode, information and triggers for PS Configuration File download are provided by the NMS via SNMP messaging. As opposed to the DHCP Provisioning Mode, the network management behaviour does not vary within this mode.

If the wrong combination of Kerberos server information and PS Configuration File information is provided to the PS in the DHCP ACK issued by the cable network DHCP server, the PS will default to operation in Dormant CableHome mode. In Dormant CableHome Mode, the PS will use locally stored configuration parameters. If the PS has never been provisioned, it will operate with factory default parameters.

Table 5-8 describes the infrastructures within which each PS mode is intended to operate.

Table 5-8/J.191 – PS infrastructures

Mode	Capability directly effected	Intended infrastructure
SNMP Provisioning Mode	Configuration file download.	Extended IPCable2Home Infrastructure
DHCP Provisioning Mode	Configuration file download.	DOCSIS 1.0 and 1.1 Infrastructures
DHCP Provisioning Mode: NmAccess Mode	SNMP version used between NMS and PS	DOCSIS 1.0 Infrastructure (SNMP v1/v2)
DHCP Provisioning Mode: SNMP Coexistence Mode	SNMP version used between NMS and PS	DOCSIS 1.1 and Extended IPCable2Home Infrastructures (SNMP v3)
Dormant CableHome Mode	SNMP manageability from WAN interface	Any cable network infrastructure that does not support CableHome provisioning and management.

5.6 IPCable2Home physical interfaces

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, which include the Radio Frequency Interface (RFI) as described by ITU-T Rec. J.112 (or ITU-T Rec. J.122) for the Embedded PS case, and other WAN Networking Interfaces, intended for WAN connection, in the stand-alone PS case.
- LAN Networking Interfaces for connection to LAN IP Devices.
- Hardware Test Interfaces, such as JTAG and other proprietary approaches, which are part of the silicon and do not 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 labelled 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. Access to PS functions MUST only be allowed via interfaces specifically prescribed by 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 premises 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 Management Tools provide the cable operator with functionality to monitor and configure the Portal Services (PS) element, as well as to perform remote diagnostics on LAN IP Devices. This clause describes and specifies requirements for these capabilities.

6.1.1 Goals

The goals for the Management Tools include:

- Provide cable operators with visibility to LAN IP Devices.
- Provide cable operators with 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 in the LAN-Trans address realm.
- Provide cable operators with access, via the MIBs, to internal data in the PS element and enable the cable operator to monitor specified parameters and to configure or reconfigure 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 network management environment include:

- IPCable2Home-compliant devices implement the Internet Protocol (IPv4) suite of protocols.
- SNMP is used for the exchange of management messages between the cable network NMS and the PS in the HA 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 Portal Services element.
- LAN IP Devices implement a DHCP client.
- Information acquired through the exchange of DHCP DISCOVER, DHCP REQUEST, and DHCP OFFER messages exchanged between the PS and LAN IP Devices, and information available from the PS database (see 5.4) through the Interfaces Group MIB are sufficient to provide the cable operator with desired knowledge about LAN IP Devices.
- The PS element 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 IPCable2Home Management Tools system design guidelines are listed in Table 6-1. This list provided guidance for the development of the management tools specifications.

Table 6-1/J.191 – Management tools system design guidelines

Reference	Management tools system design guidelines
Mgmt 1	The PS will implement SNMPv1/v2c/v3 to provide access to internal Portal Services data.
Mgmt 2	The PS will be capable of issuing an ICMP Ping command to any specified LAN IP Device in the LAN-Trans realm at the direction of the cable network NMS and store results in the PS Database. Remote Ping test results are accessible through CTP MIB objects cabhCtpPingStatus, cabhCtpPingNumSent, and cabhCtpPingNumRecv.
Mgmt 3	The PS will be capable of executing a Connection Speed Test with a specified LAN IP Device in the LAN-Trans realm at the direction of the cable network NMS and store results in the PS Database.
Mgmt 4	The PS element will be capable of reporting events.

6.2.2 Management tools system description

As shown in Figure 6-1, the Management Tools architecture consists of the following components:

1) the Cable Management Portal (CMP);
2) the Cable Test Portal (CTP);
3) an Event Reporting mechanism within the CMP; and
4) an SNMP Network Management System (NMS) that is part of the cable network.

The cable network NMS monitors and configures the PS by accessing the PS Database through MIBs specified in 6.3.7. The NMS may also directly communicate with LAN IP Devices in the LAN-Pass realm.

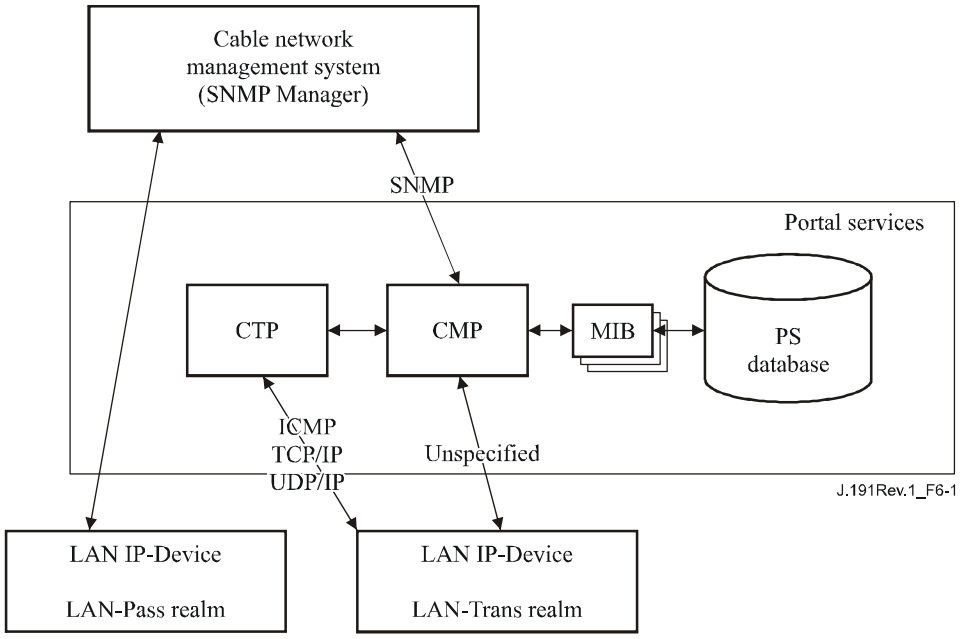

Figure 6-1/J.191 – Management architecture

The CMP and CTP functional elements reside within the PS. The PS logical element may be embedded or stand alone, relative to the cable modem functionality, as described in clause 5.

In both Embedded PS and stand-alone PS cases, from the management perspective, the CM and PS are separate and independent management entities, and no data sharing between CM and PS is implied, except for the case of software image download to an Embedded PS. In the Embedded PS

case, the cable modem's docsDevSoftware objects are accessed to set up, initiate, and monitor the download of a single combined software image. Because of this management independence, the CM and PS MUST 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. The use of a single SNMP agent for Embedded PS case is not precluded.

The Portal Services element supports SNMPv1, SNMPv2c, and SNMPv3 protocols. Clause 5.5 introduced the two provisioning modes supported by a Portal Services element, and clause 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 6.3.3.

6.3 The Cable Management Portal (CMP)

The Cable Management Portal (CMP) exists within the PS. It serves as the hub of Management-control for WAN side management accesses. The CMP aggregates and interconnects management information in the WAN-MAN and LAN-Trans realms because they are not directly accessible to each other.

6.3.1 CMP goals

The goals for the Cable Management Portal include:

- Enable the NMS to view and update Cable Address Portal (CAP) configuration information.
- Enable the NMS to view and update Firewall configuration information.
- Enable Remote Ping for LAN IP Devices in the LAN-Trans realm, via the Cable Test Portal (CTP).
- Enable viewing of LAN IP Device information obtained via the Cable DHCP Portal (CDP).
- Enable viewing of the results of LAN IP Device performance monitoring done by the Cable Test Portal (CTP).
- Enable the NMS to access other PS configuration parameters.
- Process bulk SNMP commands passed from the cable network NMS in a PS Configuration File.
- Facilitate security by providing access to security parameters, and through 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 provided guidance for the specification of CMP functionality.

Table 6-2/J.191 – 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.

Table 6-2/J.191 – CMP system design guidelines

Reference	CMP system design guidelines
CMP 3	The home network will recover gracefully from a power outage, and devices connected to the home network must return to the operational state they were in prior to the outage.
CMP 4	Home network devices will be easy to install and configure for operation, much like a home appliance.

6.3.3 CMP system description

As mentioned previously, the CMP serves as the hub of Management control for WAN side management accesses and it aggregates information for, and interconnects management of WAN Management and LAN network elements.

The CMP works in any of three network management modes.

As described in 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 IPCable2Home-specified MIBs.

As described in 5.5, when in DHCP Provisioning Mode, the PS defaults to operate in NmAccess Table mode, but can be configured by the cable operator to operate in SNMPv3 Coexistence Mode. In NmAccessTable mode, management access is controlled by the NmAccessTable of RFC 2669 and the SNMPv1/v2c protocols are supported. If the PS is 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.

If the PS does not receive SNMP Provisioning Mode or DHCP Provisioning mode decision parameters and falls back to Dormant CableHome mode, it disables SNMP access from its WAN interfaces and responds to any SNMPv1 or SNMPv2c message received through any LAN interface.

Table 6-3 contains definitions for terms that are specific to the CMP.

Table 6-3/J.191 – Definition of terms

Management-control	Read or write access to a set of parameters that control or monitor the behaviour of the PS.
PS Database	A set of parameters that controls or monitors the behaviour 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.

Table 6-3/J.191 – Definition of terms

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 manipulated via Management-control include the firewall policy settings, NMS-configured NAT mappings, remote diagnostic tool initiation and results access, PS status, and LAN address range configuration. As will be illustrated later, the various management messaging interfaces may have access rights to different sets of parameters. It is possible to access the PS database from both the WAN and LAN; however, LAN access is not specified. Figure 6-2 indicates management messaging interfaces:

- NMS-CMP: management message exchange between the cable network NMS and the CMP.
- CMP-LAN IP Device: management message exchange between the CMP and LAN IP Devices in the LAN-Trans realm (not specified by IPCable2Home).
- NMS-LAN IP Device: management message exchange between the cable network NMS and LAN IP Devices in the LAN-Pass realm (not specified by IPCable2Home).
- NMS-LAN IP Device: management message exchange between the cable network NMS and LAN IP Devices in the LAN-Trans realm (provisioned by configuration of the CAP – see 8.3.2). This messaging is not specified by IPCable2Home.

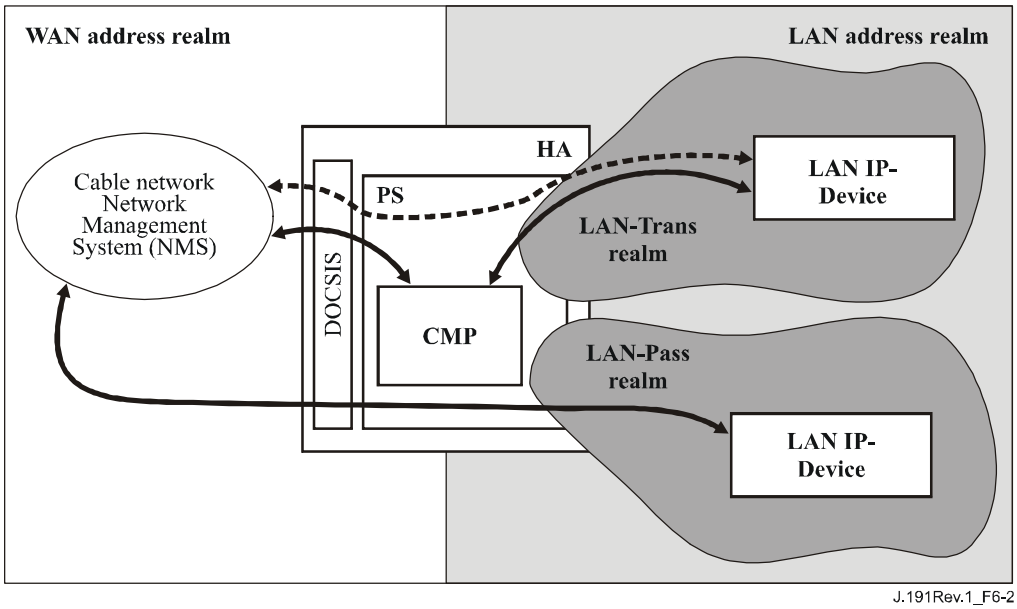

Figure 6-2/J.191 – Management message interfaces

The CMP is primarily a WAN (NMS) accessed and WAN controlled entity. 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.

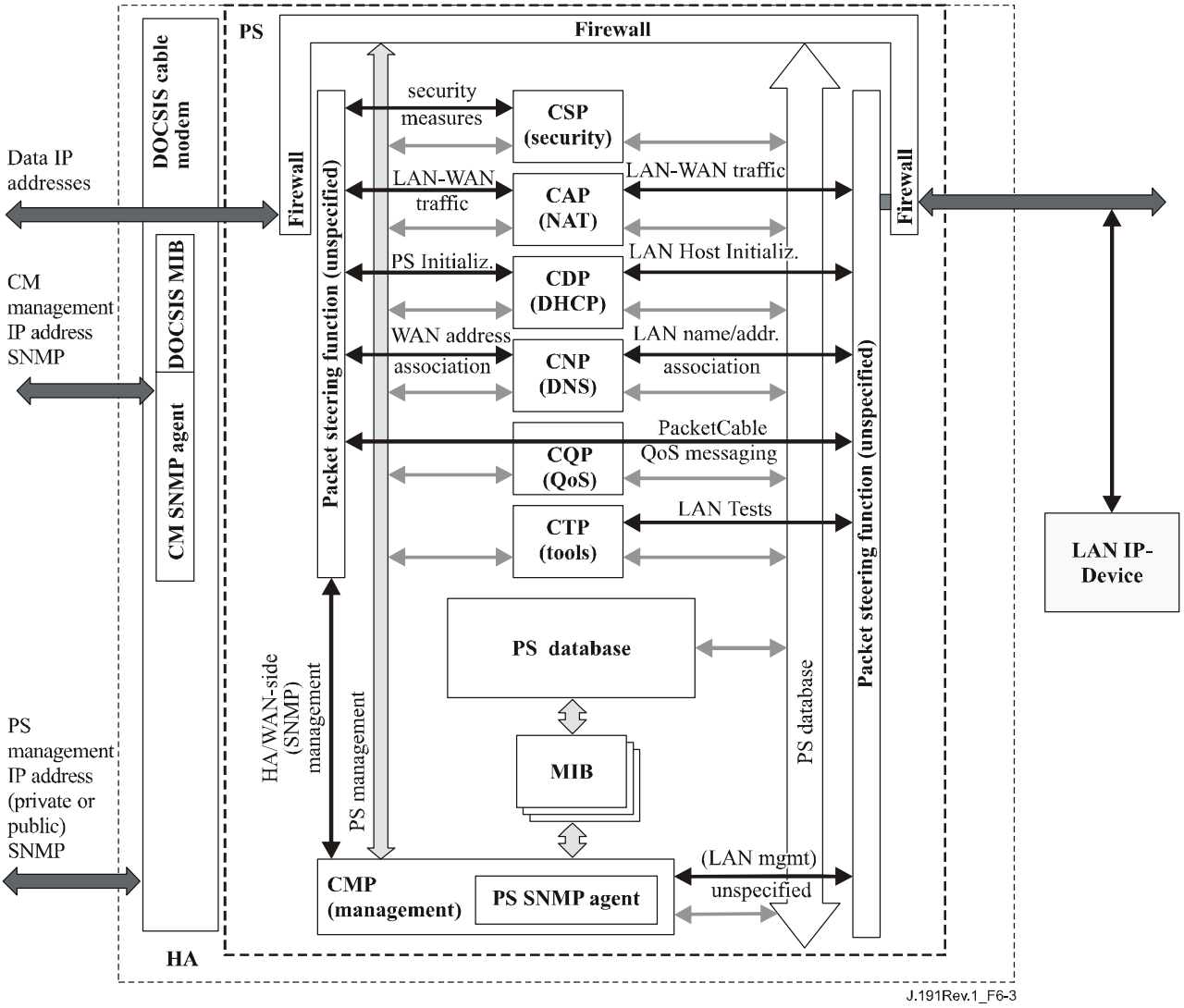

Figure 6-3/J.191 – PS block diagram

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 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.

In addition to the CMP, a NMS operator may directly access or "manage" LAN IP Devices using pass-through addressing between the Headend and the LAN device being managed. However, there are no requirements on LAN IP Devices to respond to any particular protocols, management or otherwise.

6.3.4 General CMP requirements

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 CMP MUST default to using SNMPv1/v2c for management messaging with the NMS and follow rules for NmAccess mode and Coexistence Mode, described in 6.3.6.1.

If the PS is operating in SNMP Provisioning Mode (indicated by a value of '2' in cabhPsDevProvMode), the CMP MUST use SNMPv3 for management messaging with the NMS, following rules described in 6.3.6.2.

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

When the PS is operating in Dormant CableHome mode as described in 5.5 and in 7.2.3.3 and as indicated by a value of '3' in cabhPsDevProvMode, the PS MUST NOT accept or process any SNMP message received through any WAN interface.

When the PS is operating in Dormant CableHome mode as described in 5.5 and in 7.2.3.3 and as indicated by a value of '3' in cabhPsDevProvMode, the PS MUST accept and process SNMP messages received through any LAN interface according to docsDevNmAccessTable settings (see 6.3.6.1) or according to View-based Access Control Model settings (see 6.3.6.3).

The root of MIBs (PSDev MIB, CAP MIB, CDP MIB, CTP MIB, and Security MIB) MUST be (enterprises.4491.2.4).

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 follows:

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 blank space. The separation from one Type/Value pair to the next Type/Value pair is "; " – a semi-colon and a blank space. For instance, a sysDescr of a PS of vendor X, hardware version 5.2, Boot ROM version 1.4, SW version 2.2, and model number X would appear as follows:

any text<<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 in determining 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.

sysServices object MUST return the value "3" (Internet gateway) when queried in a PS Element.

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.

MIB-2 System group objects other than sysDescr, sysObjectID, sysUpTime, sysContact, sysName, sysLocation, and sysServices SHOULD NOT be implemented.

The Interfaces Group MIB [RFC 2863] MUST be implemented in accordance with Annex A and requirements in 6.3.8.

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.

snmpSet group objects other than snmpSetSerialNo SHOULD NOT be implemented.

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

6.3.5 SNMP protocol requirements

The following IETF RFCs MUST be adhered to or implemented as appropriate:

1) A Simple Network Management Protocol [RFC 1157];
2) Introduction to Community-based SNMPv2 [RFC 1901];
3) Protocol Operations for SNMPv2 [RFC 3416];
4) Transport Mappings for SNMPv2 [RFC 3417];
5) Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) [RFC 3418];
6) Introduction to SNMPv3 [RFC 3410];
7) SNMP MIB Framework [RFC 2571];
8) Message Processing and Dispatching for SNMP [RFC 3412];
9) SNMP Applications MIB [RFC 3413];
10) SnmpUSM MIB Group [RFC 3414];
11) SnmpVACM MIB Group [RFC 3415];
12) SNMP Community MIB [RFC 2576];
13) SNMPv2-CONF.

In support of SMIv2, the following IETF RFCs MUST be implemented:

1) Structure of Managed Information Version 2 (SMIv2) [RFC 2578];

2) Textual Conventions for SMIv2 [RFC 2579];

3) Conformance Statements for SMIv2 [RFC 2580].

6.3.6 Network management mode requirements

Clause 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. Clauses 7.2.3.3, 7.3.3.2, and 7.3.3.3 provide additional detail about PS operation in each of the two provisioning modes.

This clause describes rules for the network management modes the PS is required to support. Clause 6.3.6.1 and its subclauses describe network management modes for a PS operating in DHCP Provisioning Mode. Clause 6.3.6.2 and its subclauses describe network management modes for a PS operating in SNMP Provisioning Mode.

6.3.6.1 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 2576 and RFC 3414. 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 following guidelines:

6.3.6.1.1 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) behaviour 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) behaviour of the PS after its network management mode has been configured.

Rules of operation for each of these steps follows:

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 usmDHKickstartTable.

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 docsDevNmAccessTable 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 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 NmAccessTable, for any access to the 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 docsDevNmAccessTable):

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 to be sent to particular NMS"

DEFVAL { Disable SNMPv2trap }

::={docsDevNmAccessEntry 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 11.3.3.1.2. 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 2576, RFC 3412, RFC 3413, RFC 3414, and RFC 3415.
- 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 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 selects 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.

6.3.6.2 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 clause.

6.3.6.2.1 Management views

The management controls defined for IPCable2Home 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 IPCable2Home-specified MIBs, via SNMP from the cable network NMS. A single User is defined by the IPCable2Home specification.

Figure 6-4 illustrates some possible management Views for the PS. A WAN Administrator View (PS Administrator view) and a WAN Administrator User (PS Administrator user) are defined by IPCable2Home. 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 (PS Administrator), following rules defined in RFC 3414 and RFC 3415.

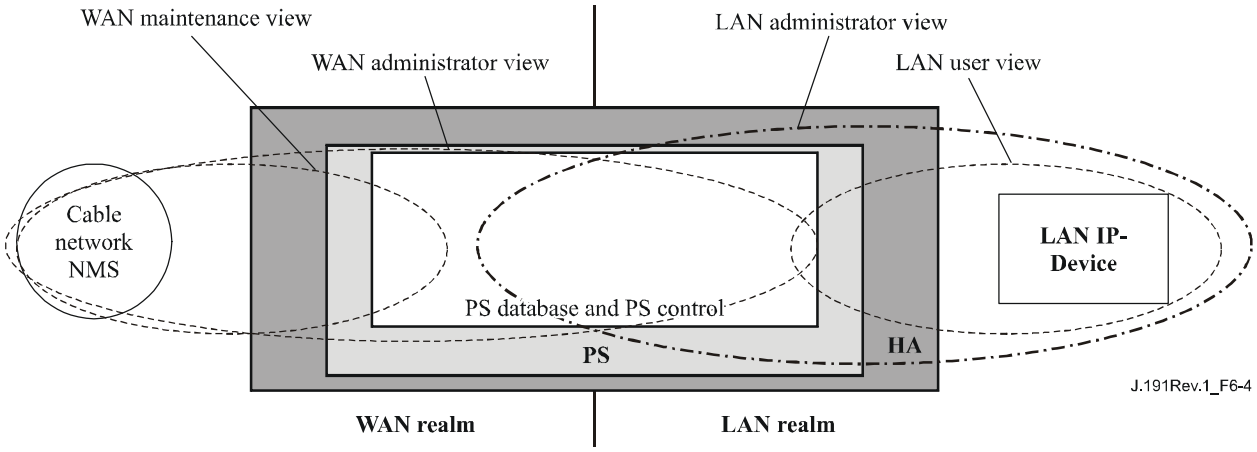

Figure 6-4/J.191 – Management views

Managed parameters defined by IPCable2Home 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 PS Administrator User.

The Ultimate Authorization (PS Administrator 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.

A full VACM implementation requires a set of actions that will tie a "User" to a "Group", and the "Group" to a VACM View, which defines the access. Clause 6.3.6.3 describes how to create these relationships.

The vacmSecurityName is the "User". This security name is tied to the vacmGroupName. Thus, the "User" is tied to a specific Group. The Group is then defined, to specify what security level is used and also what read, write and notify Views are available for this Group. The Views are then specified to show exactly what MIB objects are accessible.

The View-based Access Control Model determines the access rights of a Group, representing zero or more securityNames, which have the same access rights. For a particular context, identified by contextName, to which a Group, identified by groupName, has access using a particular securityModel and securityLevel, that Group's access rights are given by a read-view, a write-view and a notify-view.

The read-view represents the set of object instances authorized for the Group when reading objects. Reading objects occurs when processing a retrieval operation (when handling Read Class PDUs).

The write-view represents the set of object instances authorized for the Group when writing objects. Writing objects occurs when processing a write operation (when handling Write Class PDUs).

The notify-view represents the set of object instances authorized for the Group when sending objects in a notification, such as when sending a notification (when sending Notification Class PDUs).

The PS Administrator View provides full read and write access to all specified MIBs.

Management View requirements are specified in 6.3.6.3.

6.3.6.2.2 WAN-Access Control

SNMP Access Control, per RFC 3415, will be used to control access to 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 IPCable2Home to support WAN Management of the PS. The PS Administrator User access and View are specified in 11.3.3.2.2 and 6.3.6.3. An example sequence of PS Database access from the WAN interface is provided in 12.3.1.

6.3.6.2.3 Security

Security of management messages is provided by SNMPv3. Refer to clause 11 for a detailed description of how SNMPv3 is used. The CMP may use SNMPv3 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 11.3.3.

6.3.6.3 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 SNMPv3 Coexistence Mode, View-based Access Control Model (VACM) MUST be employed as defined by RFC 3415.

CHAdministrator' is the USM user name [RFC 3414] defined in 6.3.4 and 6.3.6.2.1 for the WAN Administrator, which is assumed to be the cable operator. As the Ultimate Authorization for a PS, the WAN Administrator needs to be able to read and write any MIB object and be able to create new users. The view settings for this CHAdministrator user are defined in this clause.

The WAN Administrator View MUST be implemented in a 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	'PS Administrator'
vacmGroupName	'PS Administrator'
vacmSecurityToGroupStorageType	permanent
vacmSecurityToGroupStatus	active

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

PS Administrator Group	
vacmGroupName	'PS Administrator'
vacmAccessContextPrefix	''
vacmAccessSecurityModel	3 (USM)
vacmAccessSecurityLevel	AuthPriv
vacmAccessContextMatch	exact
vacmAccessReadViewName	'PS AdministratorView'

vacmAccessWriteViewName	'PS AdministratorView'
vacmAccessNotifyViewName	'PS AdministratorView'
vacmAccessStorageType	permanent
vacmAccessStatus	active

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

PS AdministratorView subtree 1.3.6.1 (Entire MIB)

6.3.6.4 Mapping of TLV fields into created SNMPv3 table rows

This clause and the following subclauses detail how the *docsisV3 Notification Receiver* Configuration File Element (TLV Type 38) is mapped into SNMPv3 functional tables.

Upon receiving one Type 38 configuration file element, the PS MUST make entries to the following tables in order to cause the desired trap transmission:

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

A PS configuration file MAY contain TLV MIB elements (Type 28) that make entries to any of the 11 tables listed above. These TLV MIB elements are expected to not have index columns that start with the characters "@config" or "@PSconfig".

The tables in this clause show how the fields from the PS Configuration file TLV element (the tags in angle brackets <>) are placed into the SNMPv3 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.

6.3.6.4.1 snmpNotifyTable

Create 2 rows with fixed values, if 1 or more TLV elements are present.

Table 6-4/J.191 – snmpNotifyTable

snmpNotifyTable [RFC 2573] 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)

6.3.6.4.2 snmpTargetAddrTable

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

Table 6-5/J.191 – snmpTargetAddrTable

snmpTargetAddrTable [RFC 2573] 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.1
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)

6.3.6.4.3 snmpTargetAddrExtTable

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

Table 6-6/J.191 – snmpTargetAddrExtTable

snmpTargetAddrExtTable [RFC 2576] SNMP-COMMUNITY-MIB	**New row**
Column Name (* = part of index)	Column Value
* snmpTargetAddrExtName	"@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

6.3.6.4.4 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 6-7/J.191 – snmpTargetParamsTable for <Trap type> 1, 2 or 3

snmpTargetParamsTable [RFC 2573] 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 6-8/J.191 – snmpTargetParamsTable for <Trap type> 4 or 5

snmpTargetParamsTable [RFC 2573] 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)

6.3.6.4.5 snmpNotifyFilterProfileTable

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

Table 6-9/J.191 – snmpNotifyFilterProfileTable

snmpNotifyFilterProfileTable [RFC 2573] 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)

6.3.6.4.6 snmpNotifyFilterTable

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

Table 6-10/J.191 – snmpNotifyFilterTable

snmpNotifyFilterTable [RFC 2573] 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)

6.3.6.4.7 snmpCommunityTable

Create one row with fixed values if 1 or more TLVs are present. This causes SNMPv1 and v2c Notifications to contain the community string in snmpCommunityName.

Table 6-11/J.191 – 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)

6.3.6.4.8 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 6-12/J.191 – usmUserTable

usmUserTable [RFC 2574] 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 TLV element.
usmUserSecurityName	"@PSconfig" When other rows are created, this is replaced with the <Security Name> field from the TLV element.
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)

6.3.6.4.9 vacmSecurityToGroupTable

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

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

Table 6-13/J.191 – vacmSecurityToGroupTable

vacmSecurityToGroupTable [RFC 2575] 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)

6.3.6.4.10 vacmAccessTable

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

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

Table 6-14/J.191 – vacmAccessTable

vacmAccessTable [RFC 2575] SNMP-VIEW-BASED-ACM-MIB	First row	Second row	Third row
Column Name (* = Part of Index)	Column Value	Column Value	Column Value
* vacmGroupName	"@PSconfigV1"	"@PSconfigV2"	"@PSconfigUSM"
* vacmAccessContextPrefix	<Zero length string>	<Zero length string>	<Zero length string>
* vacmAccessSecurityModel	SNMPv1(1)	SNMPv2c(2)	USM(3)
* vacmAccessSecurityLevel	noAuthNoPriv(1)	noAuthNoPriv(1)	noAuthNoPriv(1)
vacmAccessContextMatch	exact(1)	exact(1)	exact(1)
vacmAccessReadViewName	<Zero length octet string>	<Zero length octet string>	<Zero length octet string>
vacmAccessWriteViewName	<Zero length octet string>	<Zero length octet string>	<Zero length octet string>
vacmAccessNotifyViewName	"@PSconfig"	"@PSconfig"	"@PSconfig"
vacmAccessStorageType	volatile	volatile	volatile
vacmAccessStatus	active(1)	active(1)	active(1)

The TLV entries with <Trap Type> set to 4 or 5 and a non-zero length <Security Name> will use the rows created in the vacmAccessTable by the DH Kickstart process.

6.3.6.4.11 vacmViewTreeFamilyTable

Create one row with fixed values if one or more TLVs are present.

This row is used for the TLV entries with <Trap Type> set to 1, 2, or 3 or with a zero length <Security Name>.

Table 6-15/J.191 – vacmViewTreeFamilyTable

vacmViewTreeFamilyTable [RFC 2575] SNMP-VIEW-BASED-ACM-MIB	First row
Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilyViewName	"@PSconfig"
* vacmViewTreeFamilySubtree	1.3
vacmViewTreeFamilyMask	<Default from MIB>
vacmViewTreeFamilyType	included(1)
vacmViewTreeFamilyStorageType	volatile
vacmViewTreeFamilyStatus	active(1)

The TLV entries with <Trap Type> set to 4 or 5 and a non-zero length <Security Name> will use the rows created in the vacmViewTreeFamilyTable by the DH Kickstart process.

6.3.7 MIB requirements

MIB objects listed in Annex A MUST be implemented in an IPCable2Home PS Element. Required MIB objects are from the following MIB documents:

1) Interfaces Group MIB [RFC 2863];
2) DOCSIS Cable Device MIB [RFC 2669];
3) Definition MIB [E.4];
4) Cable PSDev MIB [E.1];
5) Cable CAP MIB [E.6];
6) Cable CDP MIB [E.5];
7) Cable CTP MIB [E.2];
8) Cable Security MIB [E.3];
9) draft-ietf-ipcdn-bpiplus-mib-12;
10) IP MIB (SNMPv2) [RFC 2011];
11) UDP MIB (SNMPv2) [RFC 2013];
12) Diffie-Hellman USM Key [RFC 2786];
13) INET Address MIB [RFC 3291];
14) DOCS IF MIB [RFC 2670];
15) IANA ifType MIB.

In the Embedded PS, the cable modem management entity and PS management entity (CMP) MUST respond to different and independent management IP addresses. Cable Modem and IPCable2Home specify some of the same MIB objects, but if a 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 RFC 2578 snmpv2 subtree, the RFC 3418 SNMP group, the RFC 2011 IP and ICMP group counters, and the RFC 2013 UDP group counters, 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 the Embedded PS, software download of the single image of the combined cable modem software and Portal Services software, is controlled by the cable modem. The docsDevSoftware Group of objects [RFC 2669] MUST NOT be implemented in the Embedded PS, except for the read-only object docsDevSwCurrentVers. I.e., the remainder of this group of objects is only accessible through the cable modem management IP address.

- docsDevSwServer;
- docsDevSwFilename;
- docsDevSwAdminStatus;
- docsDevSwOperStatus.

The docsDevSoftware Group of objects MUST be implemented in a stand-alone PS. Modification of the docsDevSoftware objects (as specified in 11.3.7) by the cable operator for the purpose of downloading the stand-alone PS software image MUST result in proper secure software download operation.

In the Embedded PS, cable modem MIB objects are only 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 IP address, with the exception of the RFC 2578 snmpv2 subtree, the

RFC 3418 SNMP group, the RFC 2011 IP and ICMP group counters, and the RFC 2013 UDP group counters which are allowed to be shared between the CM and PS management entities.

In the Embedded PS, IPCable2Home-specified MIB objects MUST only be visible and accessible when the manager accesses them from the WAN via the PS WAN-Man IP address, or from the LAN via the cabhCdpServerRouter IP address, and are not visible or accessible via the cable modem management IP address, with the exception of the RFC 2578 snmpv2 subtree, the RFC 3418 SNMP group, the RFC 2011 IP and ICMP group counters, and the RFC 2013 UDP group counters which are allowed to be shared between the CM and PS management entities.

The general IPCable2Home MIB hierarchy is illustrated in Figure 6-5. Specific OIDs required for individual MIBs are listed in Annex A.

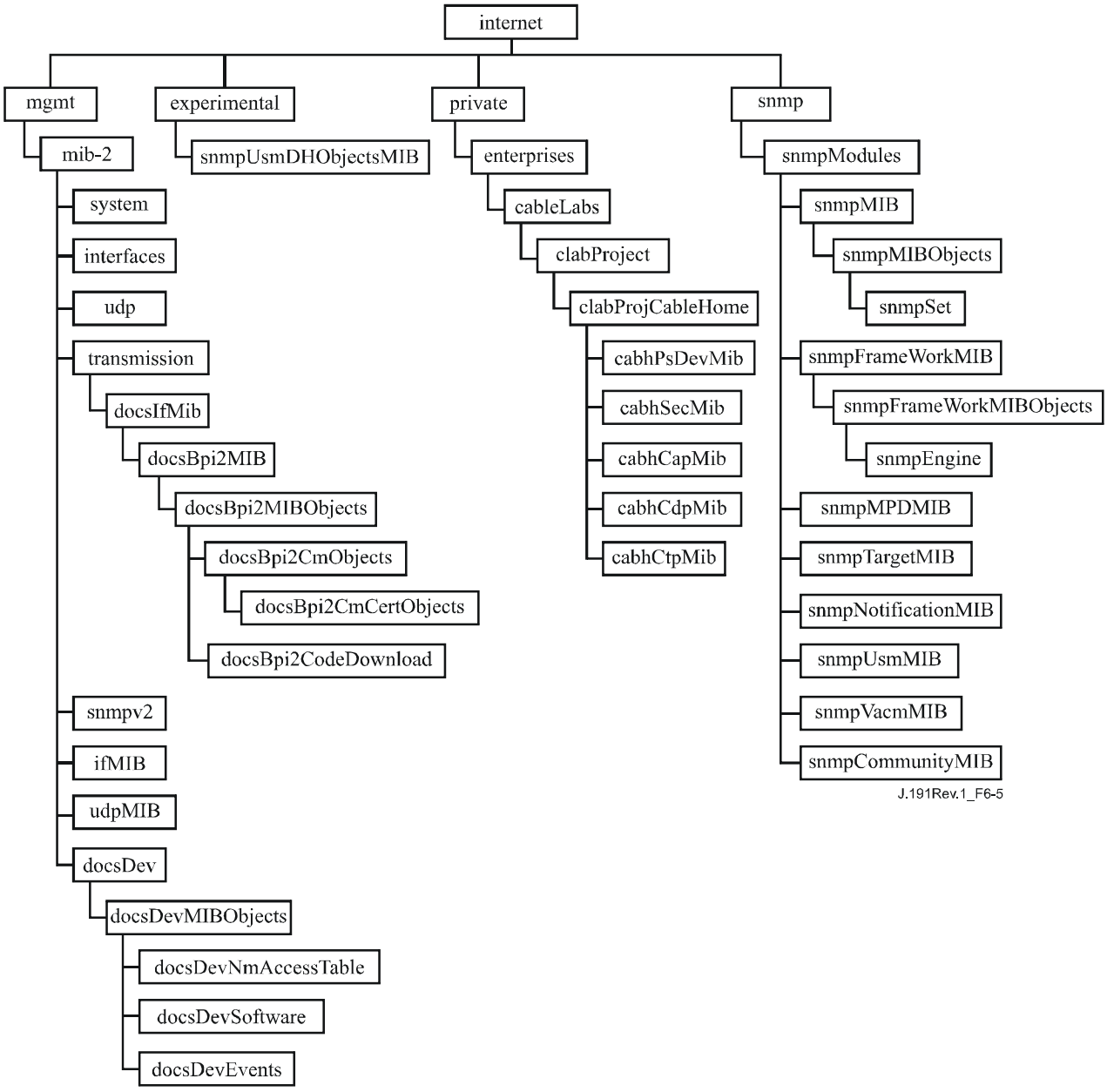

Figure 6-5/J.191 – MIB hierarchy

6.3.8 Interfaces Group MIB requirements

The Interfaces Group MIB 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. 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-MAN interface of the PS element, even if the interface is internal – as exists in the case of an Embedded PS utilizing an integrated chip design.

An instance of ifEntry MUST exist for the WAN-Data interface of the PS element, as long as the interface is part of the PS active configuration, and regardless of whether the interface is external or 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 interfaces MUST be numbered as shown in Table 6-16.

Table 6-16/J.191 – 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 RW.

The ifTable ifType values corresponding to ifIndex 255 MUST be 'Other'. For embedded PSes, the ifTable ifType values corresponding to ifIndex values 1 and 2 MUST be 'Other'. For stand-alone PSes, the ifTable ifType value corresponding to ifIndex values 1 and 2 MUST be the appropriate IANA ifType value.

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 ifStack group MUST be implemented to identify the 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:

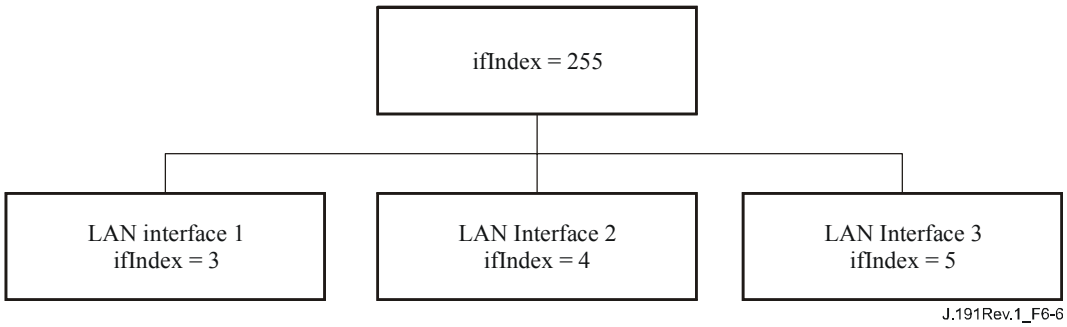

Figure 6-6/J.191 – ifStack implementation example

6.3.9 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[1] and abiding by Table 6-17:

Table 6-17/J.191 – PS ipNetToMediaTable

ipNetToMediaNetAddress	ipNetToMediaPhysAddress	ipNetToMediaIfIndex
WAN-Man IP address	Wan-Man hardware address	1
WAN-Data IP addresses	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

6.4 The Cable Test Portal (CTP)

6.4.1 CTP goals

The goals for the Cable Test Portal include:

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

1 One entry for each for the DHCP, DNS and Router servers' IP addresses will be created only if the three addresses are distinct. In the most typical PS LAN configuration, in which the same IP address is shared by the three servers, only one entry will be displayed in the ipNetToMediaTable.

6.4.2 CTP design guidelines

The IPCable2Home Management Tools system design guidelines are listed in Table 6-18. A number of these guidelines are common with the CMP design guidelines. This list provides guidance for the specification of CTP functionality.

Table 6-18/J.191 – CMP system design guidelines

Reference	CMP 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 (Cable 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 tests are defined based on problems likely to be encountered for IPCable2Home 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 centre and network operations centre to learn more about the connection between the PS element and LAN IP devices in the home.

6.4.3.1 CTP connection speed tool

This function is used to get a rough measure of the throughput performance across the link between the PS and a LAN IP Device. It sends a burst of packets between the PS and the LAN IP Device 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.

The Connection Speed function relies on the LAN IP Devices 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 Pass-through Primary Packet-handling mode and the cable operator can test LAN IP Devices directly from the NMS console if desired. This test feature only works on LAN IP Devices in the LAN Trans address realm that implement the Echo Service function as described in RFC 347.

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

6.4.3.2 CTP ping tool

This function is called to test connectivity between the PS and individual LAN IP 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. 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.

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.

Clause 6.4.4 lists the parameters and responses for the Ping Tool.

Clause 12.2.1.2 details the operation of the Ping Tool.

6.4.4 CTP requirements

6.4.4.1 Connection speed tool

The CTP 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 Cable CTP MIB.

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

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

The Connection Speed Tool MUST NOT generate packets out any WAN Interface.

When the NMS triggers the CTP to initiate the Connection Speed Tool by setting cabhConnControl = start(1), the CTP 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 value of the timer is equal to the value of cabhCtpConnTimeOut before the last bit is received from the target LAN IP Device, 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 OR before the timer is terminated, the CTP MUST set cabhCtpConnStatus = aborted(4) AND report the appropriate event (refer to Annex B – CTP Events).

When the CTP runs the Connection Speed Tool, it MUST determine the average round-trip throughput between the PS and the LAN IP Device 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 payload of the packets transmitted when the Connection Speed Tool is running SHOULD NOT be all zeroes or all ones.

The CTP 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 CTP 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 CTP 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 CTP 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 CTP 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 CTP 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 CTP 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 CTP 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).

6.4.4.2 Ping tool

The CTP 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 Cable CTP MIB.

When the NMS triggers the CTP to initiate the Ping Tool by setting cabhCtpPingControl = start(1), the CTP 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 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 any timeout timer expires, the CTP 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 CTP MUST set cabhCtpPingStatus = aborted(4) AND report the appropriate event (refer to Annex B – CTP Events).

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

When the CTP runs the Ping Tool, it MUST determine the average round-trip time between the PS and the LAN IP 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 runs the Ping Tool, it 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 CTP MUST increment the value of cabhCtpPingNumIcmpError AND log the error in cabhCtpPingIcmpError. The last ICMP error that occurs will overwrite the previous one written.

The payload of the packets transmitted when the Ping Tool is running SHOULD NOT be all zeroes or all ones.

The CTP 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 packet transmitted by the CTP Ping Tool, to the time when the last bit of that packet is received.

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

The Ping Tool MUST NOT generate packets out any WAN Interface.

The CTP 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 CTP 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 Event reporting

The event reporting and control mechanisms used is RFC 2669, which 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.

6.5.1 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.

Event notification by the PS is fully configurable. 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:

- 1: local-non-volatile(0)
- 2: traps(1)
- 3: syslog(2)
- 4: 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 6.5.1.1.

6.5.1.1 Local event logging

The PS MUST maintain a single local-log event table that contains events stored as both local-volatile events and local-non-volatile events. Events stored as local-non-volatile 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 appear in English. 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-non-volatile 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-non-volatile MUST be renumbered beginning with 1.
- The local log MUST then be initialized with the events tagged as local-non-volatile 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-non-volatile, 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-non-volatile.

6.5.1.2 SNMP TRAP and INFORM

The PS MUST support the SNMP Trap PDU as described in RFC 2576. The PS MUST support the SNMP INFORM PDU as described in RFC 2576. 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.

6.5.1.3 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 OR 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 "IPCABLE2HOME" 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 gap 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 gap 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 IPCable2Home 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[IPCABLE2HOME]: <68000402> Time of the day received in invalid format.

The number 68000402 in the given example is the number assigned by IPCable2Home to this particular event.

6.5.2 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 IDs. IPCable2Home standard management events are defined in Annex B. Each management event described in the annex is assigned an IPCable2Home 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 clause.

6.5.2.1 Event priorities

RFC 2669 defines 8 different priority levels and the corresponding reporting mechanism for each level. The standard events specified in this Recommendation 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 the 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 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 non-volatile entries.

Table 6-19/J.191 – Default notification types for event priorities for the PS

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	Standard
3) Critical	Yes	No	No	No	Standard
4) Error	Yes	Yes	Yes	No	Standard
5) Warning	Yes	Yes	Yes	No	Standard
6) Notice	No	Yes	Yes	Yes	Standard
7) Informational	No	No	No	No	Standard 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.

Table 6-20/J.191 – Minimum level of notification type support by event priority in the PS

Event priority	Local-non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-volatile (bit-3)	Note
1) Emergency	Yes	Yes	Yes	Yes	Vendor Specific
2) Alert	Yes	Yes	Yes	Yes	Standard
3) Critical	Yes	Yes	Yes	Yes	Standard
4) Error		Yes	Yes	Yes	Standard
5) Warning		Yes	Yes	Yes	Standard
6) Notice		Yes	Yes	Yes	Standard
7) Informational		Yes	Yes	Yes	Standard and Vendor Specific
8) Debug		Yes	Yes	Yes	Vendor Specific

6.5.2.2 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.

6.5.3 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.

6.5.4 Secure software download event reporting

Table B.1, 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 stand-alone PS, since software upgrade (also referred to as secure software download) for an embedded PS is controlled and managed by the cable modem. Clause 11.3.7.1 defines requirements for secure software download for the two classes of Portal Services elements. The embedded PS, as defined in 5.1.3.1, MUST NOT generate events categorized in Table B.1, 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.

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:

- Cable 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 Provisioning modes

Two provisioning modes are supported. They are referred to as DHCP Provisioning Mode (DHCP Mode) and SNMP Provisioning Mode (SNMP Mode). The two provisioning modes are compared in Table 7-1.

Table 7-1/J.191 – Provisioning modes

	DHCP mode	SNMP mode
PS Configuration File Trigger	Triggered by presence of TFTP server information in DHCP message.	Triggered by NMS via SNMP message.
PS Configuration File Requirement	PS Configuration File download is required.	PS Configuration File download is not required.

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

Clause 13 describes the sequence of events for each of the two Provisioning Modes.

7.1.2 Provisioning architecture

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 LAN IP Devices to satisfy the system design guidelines listed in 7.2.1.

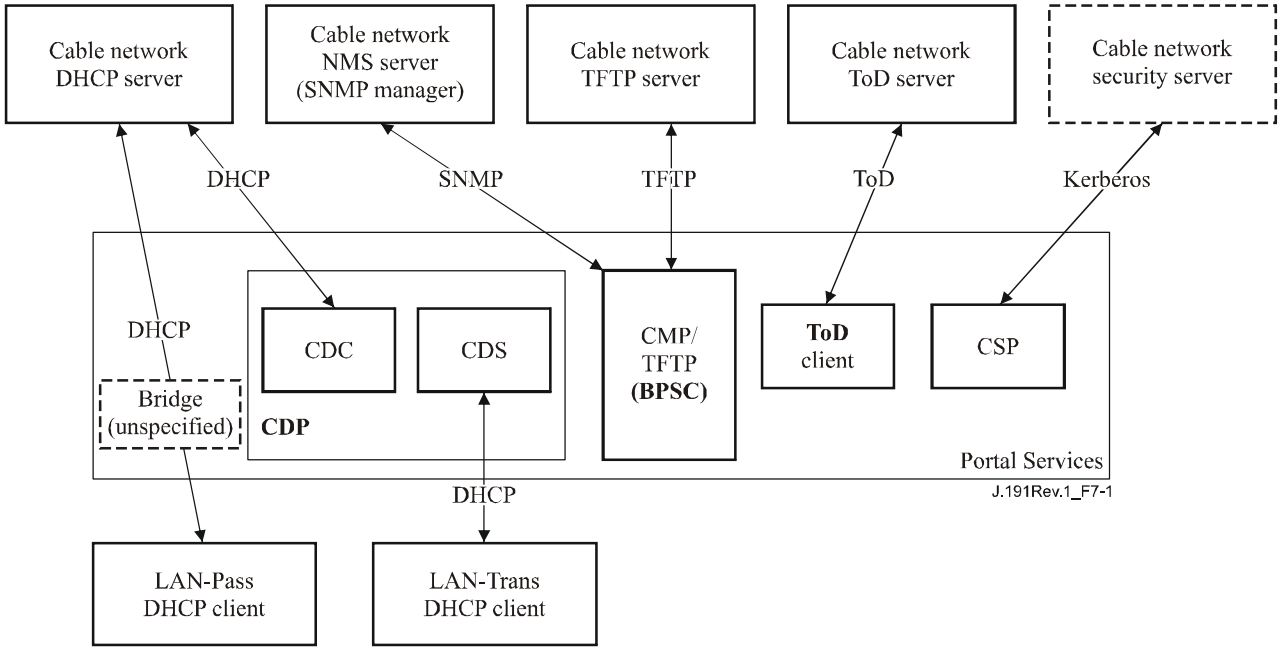

Figure 7-1/J.191 – Provisioning architecture

7.1.3 Goals

The goals of the Cable DHCP Portal include:

- Assign, via DHCP, IP addresses to LAN IP Devices according to rules specified in this clause.
- Acquire, via DHCP, IP addresses for the WAN Interfaces of the Portal Services element according to rules specified in this clause.

The goals of the Bulk PS Configuration tool include:

- Download and process Configuration Files.

The goals of the Time of Day client include:

- Synchronize the Time of Day clock in the PS element with that of the Headend network.

7.1.4 Assumptions

The Cable DHCP Portal operating assumptions include:

1) LAN IP Devices implement a DHCP client as defined by RFC 2131.
2) The cable network provisioning system implements a DHCP server as defined by RFC 2131.
3) 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.
4) LAN IP Devices may support various DHCP Options and BOOTP Vendor Extensions, allowed by RFC 2132.

The Bulk PS Configuration tool operating assumptions include:

- Bulk PS configuration will be accomplished via the download of a PS Configuration File containing one or more parameters.

The Time of Day client operating assumptions include:

- 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 Cable DHCP portal architecture

The IPCable2Home DHCP Portal (CDP) is one of the three provisioning tools introduced in 7.1. This clause describes the System Design Guidelines, System Description, and Requirements pertaining to the CDP.

7.2.1 Cable DHCP portal system design guidelines

The following design guidelines (Table 7-2) drive the capabilities defined for the CDP:

Table 7-2/J.191 – 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	A mechanism will be provided 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).

7.2.2 Cable DHCP portal system description

The Cable DHCP Portal (CDP) is the logical entity that is responsible for IPCable2Home 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 Cable Naming 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 spoofing, 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 clause 8), DHCP Server (CDS), and DNS Server (see clause 9). 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 equal to the value of cabhCdpServerRouter.

As shown in Figure 7-2, the CDP capabilities are embodied by two functional elements residing within the CDP: the Cable DHCP Server (CDS) and the Cable DHCP Client (CDC).

Figure 7-2 also illustrates the interaction between the CDP components and the address realms introduced in clause 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 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 simply bridges DHCP messages between the DHCP server in the cable network and LAN IP Devices in the LAN-Pass realm.

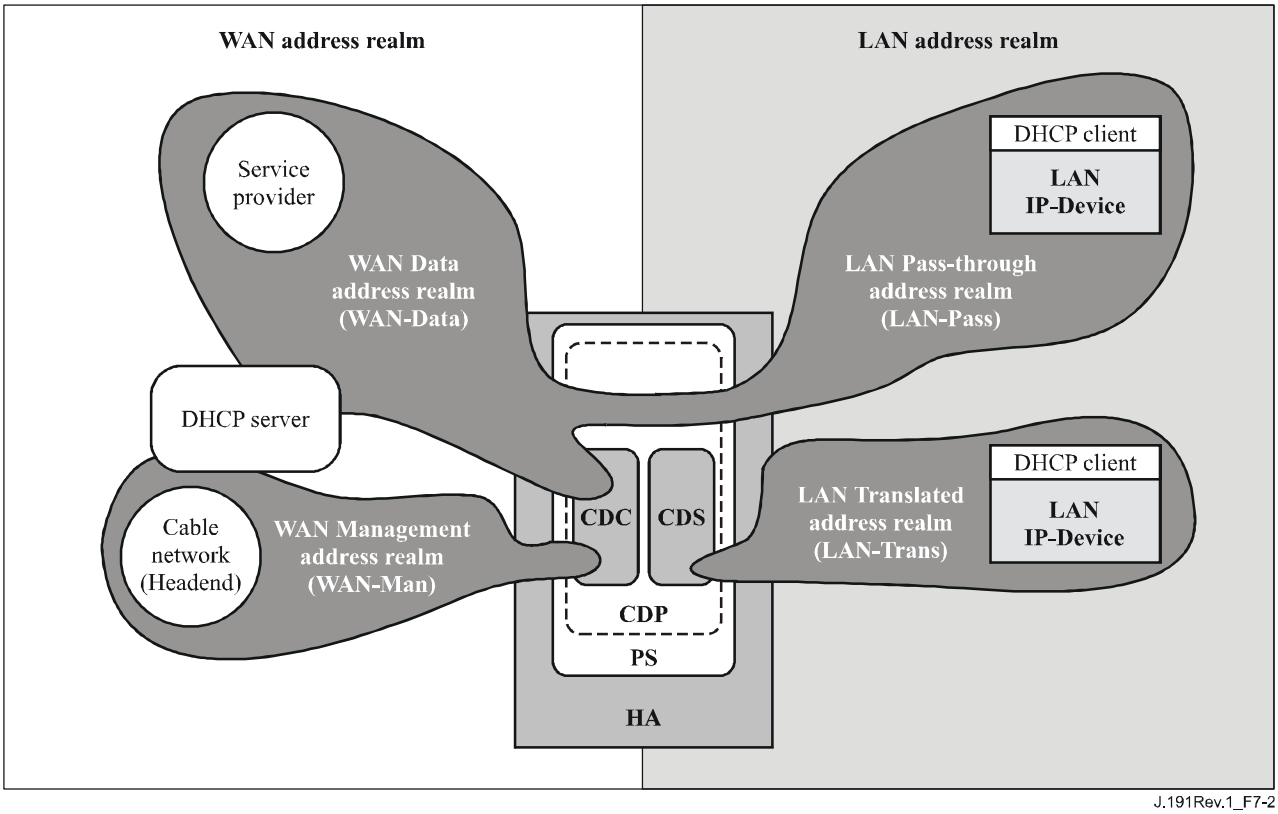

Figure 7-2/J.191 – CDP functions

7.2.2.1 CDS system description

The CDS is a standard DHCP server as defined in RFC 2131, 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 behaviour 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 Pass-through 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 Pass-through Mode or Mixed Bridging/Routing Mode (see 8.2.2.2 for more details). In these cases, DHCP interactions will take place directly between LAN IP Devices and Headend servers, and this Recommendation does not specify the process.

Throughout this Recommendation, the terms Dynamic Allocation and Manual Allocation are used as defined in RFC 2131. 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 2131 section 2.1. The default values of CDS Provisioned DHCP Options, cabhCdpServer objects, are defined (Table 7-2) and the NMS can reset the CDS Provisioned DHCP Options, cabhCdpServer objects, 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:

1) The Client Identifiers RFC 2132 section 9.14 (cabhCdpLanAddrClientID);

2) The LAN IP address assigned to the client (cabhCdpLanAddrIp);

3) An indication that the address was allocated either manually (via the CMP) or dynamically (via the CDP) (cabhCdpLanAddrConfig).

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.

7.2.2.2 CDC 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 Cable DHCP configuration parameters.
- 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 Vendor Class Identifier Option (DHCP Option 60) defines a device class. For IPCable2Home, the Vendor Class Identifier Option will contain the string "IPCable2Home", to identify an IPCable2Home 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 stand-alone, 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. DHCP Option 43 and its suboptions are defined in 7.2.3.3.

Details of the requirements for supporting DHCP Options 60 and 43 are in Tables 7-4 and 7-5. Details related to other optional and mandatory DHCP options are provided in Table 7-6.

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.

7.2.2.2.1 Cable DHCP Client Option 61

The PS element can have one or more WAN IP addresses associated with 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).

7.2.2.2.2 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 Interface. This Address Mode is only applicable when the PS Primary Packet-handling Mode (cabhCapPrimaryMode) is set to Pass-through (refer to 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 non-zero 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 address. 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 operator-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 operator-configured cabhCdpWanDataAddrClientId entries as the value of cabhCdpWanDataIpAddrCount, i.e., the operator 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 operator-configured cabhCdpWanDataAddrClientId entries than the value of cabhCdpWanDataIpAddrCount, i.e., the operator 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 non-zero 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 ($n \leq 0xFF$)

   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.

7.2.3 Cable DHCP portal requirements

7.2.3.1 CDP requirements

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

7.2.3.2 CDS requirements

The CDS behaviour MUST be in accordance with the Server requirements of RFC 2131 section 4.3.

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

CDS 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 CDS MUST be capable of creating, modifying and deleting cabhCdpLanAddrTable entries for devices allocated a LAN-Trans address.

Provisioned CDP LAN Address Management Table (cabhCdpLanAddrTable) entries MUST be retained during a cable outage and MUST persist after a PS power cycle. The CDS 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 CDS 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 CDS 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):

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

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

The CDS 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), a notification MUST be generated as in accordance with the event reporting mechanism defined in 6.5 and Annex B. While the Address Count parameter (cabhCdpLanTransCurCount) exceeds the Address Threshold parameter (cabhCdpLanTransThreshold), the CDS MUST be capable of the following threshold exceeded actions for the next DHCP DISCOVER from the LAN: assign a LAN-Trans address as normal or do not assign an address.

If cabhCdpLanTranCurCount equals or exceeds cabhCdpLanTransThreshold AND a LAN IP Device requests an additional IP address lease, the specific action taken by the CDS MUST be as indicated by the Threshold Exceeded Action (cabhCdpLanTransAction) provisioned parameter.

The CDS MUST assign IP addresses and deliver DHCP configuration parameters listed in Table 7-3 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 CDP 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), it must notify the event in accordance to Annex B and the event reporting mechanism defined in 6.5.

The CDS 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 CableHome CDP MIB objects, including all objects in the cabhCdpLanAddrTable, cabhCdpLanPool objects, cabhCdpServer objects, and cabhCdpLanTrans objects.

The CDS MUST support the DHCP options indicated as mandatory in the CDS Protocol Support column of Table 7-3.

The CDS MUST support offering the default values indicated in the CDS Factory Defaults column of Table 7-3, if the DHCP option has not been provisioned with other values.

If the PS Primary Packet-handling mode (cabhCapPrimaryMode) has been set to Pass-through AND the PS provisioning process has completed (as indicated by cabhPsDevProvState = pass(1)), then the CDS MUST be disabled.

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

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

Table 7-3/J.191 – 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

Table 7-3/J.191 – CDS DHCP options

Option number	Option function	CDS Protocol Support (M)andatory or (O)ptional	CDS factory defaults	MIB object name
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
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

7.2.3.3 CDC requirements

The CDC behaviour MUST be in accordance with the Client requirements of RFC 2131.

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.

PS provisioning complete behaviour is defined in Table 13-1 for DHCP Provisioning Mode, Table 13-2 for SNMP Provisioning Mode.

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

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

If a non-zero 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 PS MUST implement the Vendor Specific Information Option (DHCP Option 43) as specified in Tables 7-5 and 7-6. Details of DHCP Option 43 and its suboptions for CableHome 1.0 are further defined below. The definitions of DHCP Option 43 suboptions MUST conform to requirements imposed by RFC 2132.

The option begins with a type octet with the value of number 43, followed by a length octet. The length octet is followed by the number of octets of data equal to the value of the length octet. The value of the length octet does not include the two octets specifying the tag and length.

DHCP Option 43 in CableHome 1.0 is a compound option. The content of Option 43 is composed of one or more suboptions. Supported DHCP Option 43 suboptions in CableHome 1.0 are: 1, 2, 3, 4, 5, 6, 11, 12, 13, and 14. A suboption begins with a tag octet containing the suboption code, followed by a length octet which indicates the total number of octets of data. The value of the length octet does not include itself or the tag octet. The length octet is followed by "length" octets of suboption data.

The encoding of each Option 43 suboption is defined below. See Tables 7-5 and 7-6 for the intended purpose of each suboption.

The PS MUST encode DHCP Option 43 suboption 1 by the number of octets equal to the value of the length octet of this suboption, with each octet codifying a requested suboption.

The PS MUST encode each of the DHCP Option 43 suboptions 2, 3, 4, 5, 6, 12, 13, and 14 as a character string consisting of characters from the NVT ASCII character set, with no terminating NULL.

A stand-alone PS MUST send DHCP Option 43 suboption 2 containing the character string "SPS" (without the quotation marks).

An embedded PS MUST send DHCP Option 43 suboption 2 containing the character string "EPS" (without the quotation marks).

A stand-alone PS MUST send DHCP Option 43 suboption 3 containing the character string "SPS" (without the quotation marks).

An embedded PS MUST send DHCP Option 43 suboption 3 containing a colon-separated list of all device types in the complete device, including at a minimum the colon-separated character string "ECM:EPS" (without the quotation marks).

If the PS is requesting a PS WAN-Man IP address lease, it MUST send DHCP Option 43 suboption 11 containing the value 0x01, encoded as a binary number, in its DHCP DISCOVER and DHCP REQUEST messages.

If the PS is requesting a PS WAN-Data IP address lease, it MUST send DHCP Option 43 suboption 11 containing the value 0x02, encoded as a binary number, in its DHCP DISCOVER and DHCP REQUEST messages.

Table 7-4 summarizes how the PS is required to set the values for DHCP Option 43, suboption 11 for the WAN interfaces of the PS.

The length limit of suboptions 4, 5, 6, 12, 13, and 14 is each 255 octets. Thus, the total length of Option 43 could exceed 255 octets. If the total number of octets in all DHCP Option 43 suboptions exceeds 255 octets, the PS MUST follow RFC 3396 to split the option into multiple smaller options.

The CDC MUST implement the Vendor Class Identifier Option (DHCP Option 60) as specified in Tables 7-5 and 7-6.

Table 7-4/J.191 – DHCP Option 43, suboption 11 values

Element Id	Description and 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 7-5 describes how the CDC 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 7-5/J.191 – 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	"IPCable2Home"	
CPE Option 43 suboption 1	request suboption vector	List of suboptions (within Option 43) to be returned by server. None defined.
CPE Option 43 suboption 2	"EPS"	Embedded PS
CPE Option 43 suboption 3	"ECM:EPS"	List of embedded devices (Embedded CM and embedded PS)
CPE Option 43 suboption 4	e.g.,"123456"	CM/PS Device serial number
CPE Option 43 suboption 5	e.g.,"v3.2.1"	CM/PS Hardware Version Number
CPE Option 43 suboption 6	e.g.,"v1.0.2"	CM/PS Software Version Number
CPE Option 43 suboption 11	PS WAN-Man (0x01)	Defines that an address is being requested in the PS WAN Management realm
CPE Option 43 suboption 12	e.g., "ABC Inc. CM-PS123…"	CM/PS System Description from sysDescr
CPE Option 43 suboption 13	e.g., "CM-PS123-1.0.2…"	CM/PS Firmware Rev from docsDevSwCurrentVers
CPE Option 43 suboption 14	e.g., "1.2.3…"	Firewall Policy File Version from cabhSecFwPolicyFileCurrentVersion

Table 7-5/J.191 – DHCP options for embedded PS WAN-Man and WAN-Data address requests

DHCP request options	Value	Description
Embedded portal services DHCP request for WAN-Data address		
CPE Option 60	"IPCable2Home"	
CPE Option 43 suboption 1	request suboption vector	List of suboptions (within Option 43) to be returned by server. None defined.
CPE Option 43 suboption 2	"EPS"	Embedded PS
CPE Option 43 suboption 3	"ECM:EPS"	List of embedded devices (Embedded CM and embedded PS)
CPE Option 43 suboption 4	e.g.,"123456"	CM/PS Device serial number
CPE Option 43 suboption 11	PS WAN-Data (0x02)	Defines that an address is being requested in the PS WAN-Data realm

Table 7-6 describes to what the CDC MUST set the contents of Options 60 and 43, when the PS is a stand-alone device.

Table 7-6/J.191 – 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	"IPCable2Home"	
CPE Option 43 suboption 1	Request suboption vector	List of suboptions (within Option 43) to be returned by server. None defined.
CPE Option 43 suboption 2	"SPS"	Stand-alone PS
CPE Option 43 suboption 3	"SPS"	List of Embedded devices (stand-alone PS only)
CPE Option 43 suboption 4	e.g., "123456"	PS Device serial number
CPE Option 43 suboption 5	e.g.,"v3.2.1"	PS Hardware Version Number
CPE Option 43 suboption 6	e.g.,"v1.0.2"	PS Software Version Number
CPE Option 43 suboption 11	PS WAN-Man (0x01)	Defines that an address is being requested in the PS WAN Management realm
CPE Option 43 suboption 12	e.g., "ABC Inc. PS123…"	PS System Description from sysDescr
CPE Option 43 suboption 13	e.g., "PS123-1.0.2…"	PS Firmware Rev from docsDevSwCurrentVers
CPE Option 43 suboption 14	e.g., "1.2.3…"	Firewall Policy File Version from cabhSecFwPolicyFileCurrentVersion
Stand-alone portal services DHCP request for WAN-Data address		
CPE Option 60	"IPCable2Home"	
CPE Option 43 suboption 1	Request suboption vector	List of suboptions (within Option 43) to be returned by server. None defined.

Table 7-6/J.191 – DHCP options for stand-alone PS WAN-Man and WAN-Data address requests

DHCP request options	Value	Description
CPE Option 43 suboption 2	"SPS"	Stand-alone PS
CPE Option 43 suboption 3	"SPS"	List of Embedded devices (Stand-alone PS only)
CPE Option 43 suboption 4	e.g., "123456"	PS Device serial number
CPE Option 43 suboption 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's sysDescr object, see 6.3.4.

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

Table 7-7/J.191 – 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 support a Service Provider's SNMP Entity Address (DHCP Option 177 suboption 3) configured as an IPv4 address.The format of DHCP Option 177 suboption 3 is described below:

The length of DHCP Option 177 suboption 3 MUST be 5 octets. The length octet of DHCP Option 177 suboption 3 MUST be followed by a single octet that indicates the specific address type that follows. The value of the DHCP Option 177 suboption 3 'address type' octet MUST be set to 1 to indicate an IPv4 address. The DHCP Option 177 suboption 3 'address type' octet MUST be followed by 4 octets of IPv4 address.

The PS MUST ignore DHCP Option 177 suboption 3 if its format or contents do not comply with the requirements for this suboption.

Code	Length	Type	Address			
3	5	1	a1	a2	a3	a4

The PS MUST support a Kerberos Realm Name (DHCP Option 177 suboption 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 Centre (KDC) entity. The format of DHCP Option 177 suboption 6 is described below:

The Kerberos realm name provided to the PS in DHCP Option 177 suboption 6 MUST be encoded per the domain style realm name described in [RFC 1510]. The Kerberos realm name provided to the PS in DHCP Option 177 suboption 6 MUST be all capital letters and conform to the syntax described in [RFC 1035] section 3.1. The suboption is encoded as follows:

Code	Length	Kerberos realm name			
6	n	k1	k2	…	kn

The PS MUST ignore DHCP Option 177 suboption 6 if its format or contents do not comply with the requirements for this suboption.

The PS MUST support a Kerberos server IP address (DHCP Option 177 suboption 51). The Kerberos server IP address suboption informs the PS of the network address of one or more Key Distribution Centre servers.

The encoding of the KDC Server Address suboption will adhere to the format of an IPv4 address using the default port. The minimum length for DHCP Option 177 suboption 51 is 4 octets, and the length MUST always be a multiple of 4. If multiple KDC servers are listed in DHCP Option 177 suboption 51, they MUST be listed in decreasing order of priority. The KDC Server Address suboption is encoded as follows:

Code	Length	Address 1				Address 2		
51	N	a1	a2	a3	a4	a1	a2	…

The PS MUST attempt key exchanges with the KDCs in the order listed in DHCP Option 177 suboption 51, until key exchange is successful with one of the KDC or the list is exhausted and key exchanges fail. Refer to 11.3.1, for PS key exchange requirements. The PS MUST ignore DHCP Option 177 suboption 51 if its format or contents do not comply with the requirements for this suboption.

The PS MUST include DHCP Options listed as mandatory in Table 7-8 in DHCP DISCOVER and DHCP REQUEST messages sent to the cable network DHCP server.

Table 7-8/J.191 – 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 7-9 within the DHCP Option 55 (Parameter Request List) [RFC 2132] sent in the DHCP DISCOVER and DHCP REQUEST messages.

Table 7-9/J.191 – 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

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 7.2.3.3), 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 "Fallback" WAN-Data IP address MUST NOT be used when the PS is configured to operate in Pass-through 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 CDC 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 CDC 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 suboption 3, suboption 6, OR suboption 51 (valid combination 1), the PS MUST set cabhPsDevProvMode to '1' (DHCP Mode) and attempt to synchronize time of day with the ToD server as described in 7.4.3.

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 suboption 3, a valid Kerberos realm name in suboption 6, AND a valid IP address (Kerberos server IP address) in suboption 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 '2' (SNMP Mode) 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 clause 11.

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 suboptions 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 aquire 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 suboptions 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 a cable modem to a cable data network that does not support CableHome provisioning (Dormant CableHome mode):

- Disable the SNMP agent (CMP) for WAN interface access. Leave the SNMP agent enabled for messages 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 7.2.2.2.2.
- Terminate the provisioning timer.
- Set the value of cabhPsDevProvMode to dormantCHmode(3).
- Set the value of cabhPsDevProvState to fail(3).
- Enable the CDS.
- Enable CAP and USFS functionality.
- Enable the CNP.
- Enable the Firewall.
- Operate with parameters that have been provisioned in the past, including those from 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 7.2.2.2) 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 CDC 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 7.2.2.2) the CDC 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 CDC MUST continue to retransmit the broadcast DHCP DISCOVER message implementing a randomized exponential backoff algorithm consistent with that described in RFC 2131. The CDC MUST transmit up to 5 DHCP DISCOVER messages (one initial plus 4 retransmission attempts) before resetting the backoff timer value to ZERO and repeating the process.

If the 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 7.4.3, before attempting to download the PS Configuration File.

If the 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 CDC 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 CDC MUST clear all row entries from the cabhCdpWanDnsServerTable.

Until the cabhPsDevProvState MIB has a value of 'pass' (1) indicating that the provisioning process is complete, the PS MUST block incoming traffic on the WAN interface that is not in response to a LAN-to-WAN request from the PS element itself or a LAN IP device. This will help protect against potential hacker attacks during the provisioning process when the PS firewall is disabled.

7.3 Bulk portal services configuration architecture

7.3.1 Bulk portal services configuration system design guidelines

The following system design guidelines drive the capabilities defined for the Bulk PS Configuration tool:

Table 7-10/J.191 – Bulk portal services system design guidelines

Number	Bulk PS Configuration (BPSC) system design guidelines
BPSC 1	It is necessary to provide a mechanism by which the PS can download and process Configuration Files.

7.3.2 Bulk Portal Services Configuration 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. The Bulk PS Configuration tool consists of the following components:

1) The format of the Configuration File.
2) Modes of triggering the download process.
3) Means of authenticating the file.
4) Means of reporting back the status of the PS Configuration File Download and other considerations.

Bulk PS Configuration (BPSC) is a tool that operators can use to change PS 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.

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.

Download of the firewall configuration file uses an analogous procedure as Bulk Portal Services Configuration parameter download. Refer to 11.3.5.2 for a description of the firewall configuration file download procedure.

7.3.3 Bulk portal services configuration 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 7.3.3.2.

MIB object settings passed in the PS Configuration File take precedence over and MUST overwrite existing MIB object settings.

7.3.3.1 Configuration file format requirements

PS configuration data MUST be contained in a file, which is downloaded via TFTP. The PS 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 7-11.

Table 7-11/J.191 – TLV definitions

Type	A single-octet identifier which defines the parameter
Length	One or more octets 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 7.3.3.1.1 for a definition of the pad. Configuration settings are divided into three types:

- Standard Configuration settings which are required to be present;
- Additional or optional IPCable2Home-specified configuration settings which MAY be present;
- Vendor-specific configuration settings.

The PS Configuration File MAY contain many different parameters, but the only parameters that MUST be included in any Portal Services Configuration File is the End of Data Marker (Type 255) and PS MIC (Type 53).

To allow uniform management of Devices conformant to this Recommendation, conformant Devices MUST support a Configuration File that is up to 64K-bytes long.

Each Portal Services element MUST support and a PS Configuration File MAY include configuration parameter Types 0, 9, 10, 21, 28, 32, 33, 34, 38, 43, 53 and 255, which are described in this clause.

The size of the value in the Length field for any configuration parameter included in a Portal Services Configuration File MUST be 2 octets.

The Length value for each Type described in 7.3.3.1.1, 7.3.3.1.2, 7.3.3.1.3, 7.3.3.1.4, 7.3.3.1.5, 7.3.3.1.6, 7.3.3.1.7, and 7.3.3.1.8 is the actual length in octets of the Value field.

7.3.3.1.1 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	–	–

7.3.3.1.2 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

7.3.3.1.3 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 [ITU-T Rec. X.690 | ISO/IEC 8825-1] 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.

7.3.3.1.4 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

7.3.3.1.5 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 1157. 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 clause) 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.

7.3.3.1.6 Manufacturer code verification certificate

The Manufacturer's Code Verification Certificate (M-CVC) for Secure Software Downloading. Refer to 11.3.7.5.2.

Type	Length	Value
32	Variable	Manufacturer CVC (DER-encoded ASN.1)

7.3.3.1.7 Co-signer code verification certificate

The Co-signer's Code Verification Certificate (C-CVC) for Secure Software Downloading. Refer to 11.3.7.5.2.

Type	Length	Value
33	Variable	Co-signer CVC (DER-Encoded ASN.1)

7.3.3.1.8 SNMPv3 Kickstart value

(Refer to B.C.1.2.8 of Annex B to J.112.)

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 6.3.3 and 6.3.6).

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.

7.3.3.1.8.1 SNMPv3 Kickstart security name

Type	Length	Value
34.1	2-16	UTF8 Encoded security name

For the ASCII character set, the UTF8 and the ASCII encodings are identical. Normally, this will be specified as one of the DOCSIS built-in USM users, e.g., "docsisManager", "docsisOperator", "docsisMonitor", "docsisUser".

The security name is NOT zero terminated. This is reported in the usmDHKickStartTable as usmDHKickStartSecurityName and in the usmUserTable as usmUserName and usmUserSecurityName.

7.3.3.1.8.2 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.

7.3.3.1.9 Configuration file element – docsisv3Notification receiver

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 of the TLV config file element. Up to 10 of these elements may be included in the PS Configuration File. Clause 6.3.6.4 provides details about how this configuration file element is mapped into SNMPv3 functional tables.

NOTE – All multi-byte fields of Sub-TLV must be placed in the network byte order.

7.3.3.1.9.1 Sub-TLV 38.1 – IP Address of trap receiver

IP Address of the trap receiver, in binary.

Type	Length	Value
38.1	4	IP address

7.3.3.1.9.2 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 not present, the default value 162 is used.)

7.3.3.1.9.3 Sub-TLV 38.3 – Type of trap sent by the PS

Trap type

Type	Length	Value
38.3	2	Trap type

The following trap type values MUST be recognized:

– 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.

7.3.3.1.9.4 Sub-TLV 38.4 – Timeout

Timeout, in milliseconds, used for sending SNMP inform messages.

Type	Length	Value
38.4	2	0-65535

7.3.3.1.9.5 Sub-TLV 38.5 – Number of retries when sending an inform, after sending the inform the first time

Type	Length	Value
38.5	2	0-65535

7.3.3.1.9.6 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.

If this Sub-TLV is not present, the notification receiver will receive all notifications generated by the SNMP agent.

Filter OID 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.

7.3.3.1.9.7 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 above (If present it must be ignored). If it is not supplied for a Trap type of 4 or 5, then the v3 Notification will be sent in the noAuthNoPriv security level using the security name "@PSconfig" (Note 2).

SecurityName:

> The v3 Security Name to use when sending a v3 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.

NOTE 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.

NOTE 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.

NOTE 3 – Filter OID: SNMP v3 allows the specification of which Trap OIDs 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.

NOTE 4 – The PS Configuration File MAY also contain TLV MIB elements that make entries to any of the 10 tables listed in Note 1. These TLV MIB elements MUST NOT use index columns that start with the characters "@PSconfig".

NOTE 5 – This TLV element MUST be processed only if the PS has entered SNMP v3 Coexistence Mode during processing of the PS Configuration File.

7.3.3.1.10 Vendor-specific information

Vendor-specific information for the PS, if present, 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 the VSIF. If the first TLV inside the VSIF is not a Vendor ID, then the PS Configuration File MUST be ignored.

This configuration setting may appear multiple times. The same Vendor ID may appear multiple times. There MUST NOT be more than one Vendor ID Sub-TLV inside a single VSIF.

Type	Length	Value
43	n	Vendor-specific settings

7.3.3.1.10.1 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

7.3.3.1.11 End-of-data marker

This is a special marker for end of data. It has no Length or Value fields.

Type	Length	Value
255	–	–

7.3.3.1.12 PS Message Integrity Check (PS MIC)

Type	Length	Value
53	20	A 160-bit (20 octets) SHA hash

This parameter contains a hash (PS MIC) calculated by a Secure Hash Algorithm (SHA-1) defined in FIPS 180-2. This TLV is only used in the configuration file immediately before the end of data marker.

7.3.3.2 Mode of triggering

Transfer of the Configuration File, from the TFTP server in the Headend network to the PS element, is initiated by an event referred to as a trigger. Requirements for triggering the transfer of a PS Configuration File from the TFTP 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 7.2.3.3) prior to initiating any PS Configuration File download.

PS Configuration File Download Trigger for DHCP Provisioning Mode:

> If the PS receives the TFTP 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 TFTP server address and PS Configuration File name to form a URL-encoded value and write that value into cabhPsDevProvConfigFile. The PS MUST use the following format for the URL-encoded value for the TFTP server address and PS configuration file name:

tftp://IPv4 address of the TFTP 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 server IP address) and name in the DHCP message issued to the PS (CDC) by the DHCP server in the cable network. Refer to 7.2.3.3.

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, the PS MUST issue a TFTP Get request to the server identified in the DHCP message 'siaddr' field to download the file identified in the DHCP message 'file' field.

The PS MUST issue TFTP Get request messages through the PS WAN-Man Interface only.

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.

The PS MUST reject any PS Configuration File that is received through any Interface except the PS WAN-Man Interface.

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 authentication process (refer to clause 11, for details about the SNMP authentication 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 a valid value for cabhPsDevProvConfigHash (PSDev MIB) has not been provisioned by the NMS.

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.

A signalling 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 signalling mechanism.

If a PS (CMP) is not currently requesting, downloading, or processing a configuration file, it MUST set cabhPsDevProvConfigFileStatus = idle(1). When the PS (CMP) has issued a TFTP request for a configuration file specified in cabhPsDevProvConfigFile, it MUST set cabhPsDevProvConfigFileStatus = busy(2). When the PS (CMP) completes the processing of the PS Configuration File, the PS MUST set cabhPsDevProvConfigFileStatus = idle(1).

The PS (CMP) MUST attempt to download and process the configuration file whose name and address are specified in cabhPsDevProvConfigFile when it receives an SNMP set request message for the cabhPsDevProvConfigFile object, if the following conditions are true:

- The PS is operating in SNMP Provisioning Mode;
- The cabhPsDevProvConfigHash object has a valid value; 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.

Post-trigger Operation:

Once triggered, the PS MUST use an RFC 1350 compliant TFTP client to download the PS Configuration Files.

Once triggered to download a PS Configuration File, the PS element MUST continue to attempt to download the specified PS Configuration File from the specified location until the PS Configuration File is successfully downloaded and the hash successfully computed as described in 7.3.3.3. The PS MUST use an adaptive timeout for TFTP based on binary exponential backoff as described below, if the first attempt is not successful, until the PS (CMP) successfully receives the requested file from the TFTP server in the Headend:

- each retry is 2^n second(s) following the previous attempt, where the PS 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 file following the attempt with n = 5, n is to be reset to 0 and the PS is to restart the WAN-Man IP acquisition process via DHCP.

The PS MUST exchange TFTP 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 TFTP download of the PS Configuration File is complete, AND the PS Configuration File is properly authenticated as described in 7.3.3.3, the PS MUST process the TLVs contained within the file as defined below. See 7.3.3.4, for specifics of error handing and event generation while processing the PS Configuration File.

The PS MUST use parameters extracted from the PS 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 PS Configuration File TLV-28 elements into a single SNMP PDU containing (n) MIB OID/instance and value components (SNMP varbinds). In accordance with [RFC 1905], the single PS 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 PS Configuration File or SNMP PDU. The single PS Configuration File-generated SNMP PDU requirement is consistent with SNMP PDU packet behaviours 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 PS Configuration File processing, described in 7.3.3.4.

The size of the PS Configuration File MUST be updated in the MIB object cabhPsDevProvConfigFileSize.

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) MUST be updated in the MIB objects cabhPsDevProvConfigTLVProcessed and cabhPsDevProvConfigTLVRejected, respectively. Configuration parameter Types 255 (End-of-DataMarker), 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 cabhPsDevProvConfigTLVRejected.

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.

7.3.3.3 Means of authenticating the PS Configuration File

This clause defines the procedure for authenticating the PS Configuration File.

The algorithm used to check the PS Configuration File Hash depends upon the provisioning mode of the PS element (see 5.5). There are two types of provisioning modes, DHCP Provisioning Mode and SNMP Provisioning mode. The following clauses describe the security algorithms and requirements needed to check the PS Configuration File Hash based on the provisioning mode of the PS element. The PS element MUST support both security algorithms specified in 7.3.3.3.1 and 7.3.3.3.2.

7.3.3.3.1 PS Configuration File authentication algorithm for DHCP provisioning mode

The procedure for checking of the PS Configuration File hash by the PS element in DHCP 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 contents 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 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.

7.3.3.3.2 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.

Successful download of the PS Configuration File is defined as complete and correct reception by the PS element of the contents of the PS Configuration File within the TFTP timeout period AND computation by the PS of the hash values for the PS Configuration File with no errors resulting from the computation.

7.3.3.4 Means of reporting status

The PS MUST report Configuration File download status and error conditions using the Event Reporting process described in 6.5.

Table 7-12 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 7-12/J.191 – PS Configuration File processing modes

Failure mode	Action
Type field is not valid for IPCable2Home	Disregard the subject TLV and report an event. Continue to process the file.
TFTP failed – Get Request sent, no response received	Report an event (68000500) and retry TFTP.
TFTP failed – configuration file not found	Report an event (68000600) and retry TFTP.
TFTP failed – out of order packets	Report an event (68000700) and retry TFTP.
TFTP download failed – exceeded max retries	Report an event (68000900) and reset.
TFTP download successful	Report an event (68001000) and begin authentication check.
File fails authentication check	Report an event (68000800) and reset. Do not attempt to process the file.
File is too large	Report an event (73040102) and reset. Do not attempt to process the file.
No End Of File marker	Report an event (73040102) and reset. Do not attempt to process the file.

Table 7-12/J.191 – PS Configuration File processing modes

Failure mode	Action
Duplicate TLV-28 OID	Report an event (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 (73040102), reject the configuration file, and reset. Preserve all object values that existed before the attempt to process this bad configuration file.
Unable to set value	Report an event and refuse the configuration file and reset. Set back (to the value before the SNMP Set) any values that were saved in non-volatile memory.
The CMP encounters an unrecognized SNMP OID	Disregard the subject TLV and report an event (73040100). Continue to process the file.

Refer to Annex B for a list of events including those listed in Table 7-12 and information about how events are reported.

Unsuccessful PS Configuration File download attempt – TFTP retries permitted

If the PS Configuration File Retry Counter is less than 5 AND the TFTP Get Request times out, the PS Config File is not found on the TFTP server, OR the TFTP Get failed due to out of order packets, the PS MUST initiate operation of the CDS & CNP, report the appropriate event, and retry the attempt to download the PS Configuration File, in accordance with the retry algorithm described in 7.3.3.2.

The PS MUST report the appropriate event identified in Annex B, indicating unsuccessful PS Configuration File download, each time the PS is unsuccessful in downloading the PS Configuration File.

Unsuccessful PS Configuration File download attempt – TFTP 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 Annex B 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 acquisition process via DHCP.

Successful PS Configuration File download

If the PS successfully downloads the PS Configuration File, the PS MUST reset the PS Configuration File Retry Counter to zero and report the appropriate event identified in Annex B for indicating successful download of the PS Configuration File.

If the PS Configuration File fails the authentication check as specified in 7.3.3.3, the PS MUST stop the provisioning process, reject the PS Configuration File, report the appropriate event, and restart WAN-Man IP acquisition process via DHCP.

If the PS Configuration File contains no EOF TLV or is too large to process, the PS MUST stop the provisioning process, reject the PS Configuration File, report the appropriate event, and restart WAN-Man IP acquisition process via DHCP.

If the PS Configuration File contains duplicate TLV-28 elements (duplicate means SNMP MIB object has an identical OID), the PS MUST stop the provisioning process, reject the

PS Configuration File, report the appropriate event, and restart WAN-Man IP 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 WAN-Man IP 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 Time of Day client architecture

7.4.1 Time of Day client system design guidelines

The following system design guidelines drive the capabilities defined for the PS Time of Day Client:

Table 7-13/J.191 – Time of Day client system design guidelines

Number	Time of Day client system design guidelines
TOD 1	It is necessary to provide a mechanism by which the PS can achieve time synchronization with the Headend network.

7.4.2 Time of Day client 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.

7.4.3 Time of Day client 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) 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 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 ToD 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 stand-alone PS is as follows:

- First priority: time of day acquired from the ToD server;
- Second priority: time initialized to January 1, 1970.

A stand-alone PS MUST use the most recent valid time of day acquired from the ToD server for the system time of day clock.

If a stand-alone 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 DHCP server might offer a PS multiple Time of Day server IP addresses in its DHCP ACK. The PS MUST attempt to acquire time of day from all Time of Day servers included in the DHCP ACK it receives from the DHCP server, until local time is established. The specific timeout for Time of Day Requests is implementation dependent. However, for each server identified in the DHCP ACK, 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, for each server specified, 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 non-zero 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;
- continue to retry communication with the ToD server until local time is established; and
- attempt to download the PS Configuration File as described in 7.3.3.2.

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 non-zero 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);
- attempt to download the PS Configuration File as described in 7.3.3.2.

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.

8 Packet handling and address translation

8.1 Introduction/Overview

8.1.1 Goals

The key goals which drive the IPCable2Home packet handling capabilities include:

- Provide cable friendly address translation functionality, enabling cable operator visibility and manageability of home devices while preserving cable-based source-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 clause describes the key concepts behind the IPCable2Home packet handling and address translation functionality.

8.2.1 System design guidelines

Table 8-1/J.191 – Packet handling and address translation system design guidelines

Number	System design guideline
Pckt Handling 1	Addressing mechanisms will be operator controlled, and will provide operator knowledge of and accessibility to IPCable2Home devices.
Pckt Handling 2	The addressing will do nothing that will compromise current cable network routing architectures (for example source-based routing, MPLS).
Pckt Handling 3	Traffic management mechanisms will insulate the cable network from traffic generated by in house peer-to-peer communications.
Pckt Handling 4	IP Addresses will be conserved when possible (both globally routable addresses and private cable network management addresses).

8.2.2 Packet handling system description

This clause provides an overview of the key packet handling and address translation concepts.

8.2.2.1 Packet handling functional overview

Address translation and packet handling functionality is provided by the functional entity known as the Cable Address Portal (CAP). The CAP encompasses the following address translation and packet forwarding elements:

- Cable Address Translation (CAT);
- Pass-through 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 Pass-through 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 Cable Network Address Translation (C-NAT) Transparent Routing and Cable Network Address and Port Translation (C-NAPT) Transparent Routing. C-NAT Transparent Routing is the Cable compliant version of Basic NAT RFC 3022 section 2.1 and C-NAPT Transparent Routing is the Cable 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 Pass-through function is a 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.

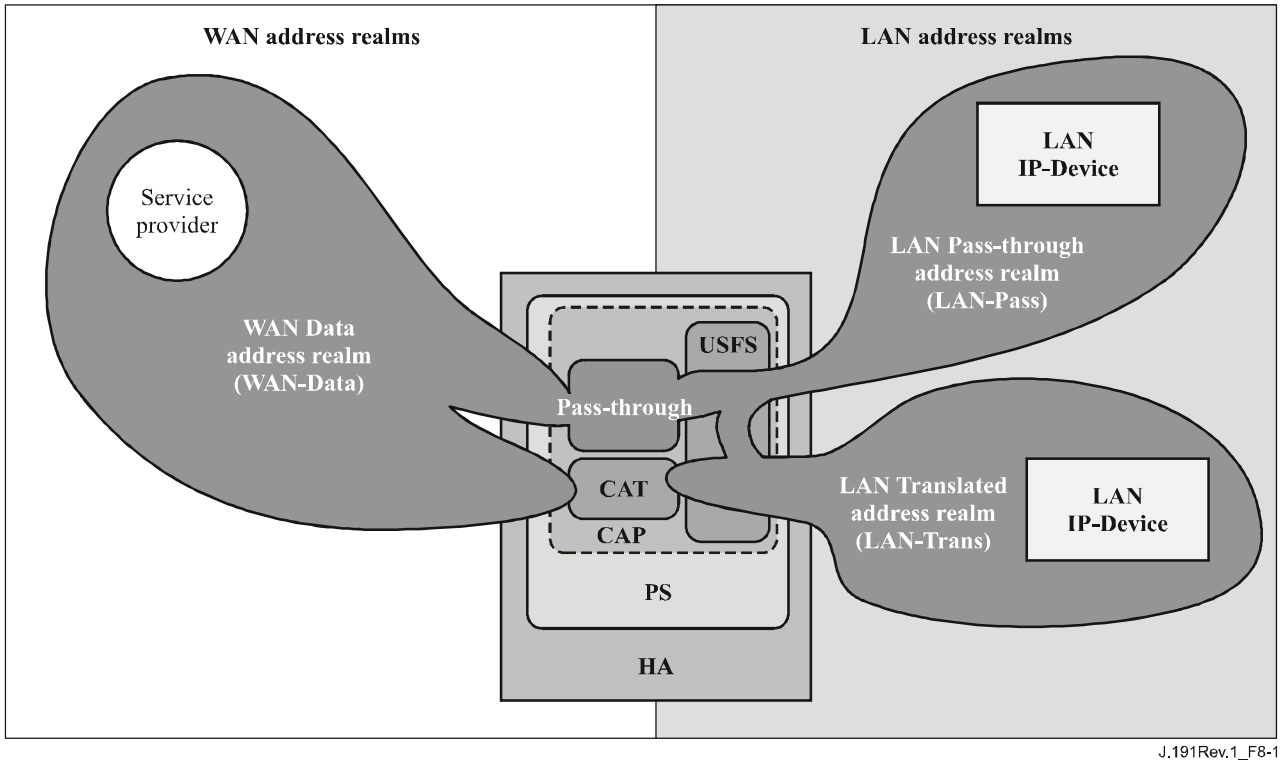

Figure 8-1/J.191 – Cable Address Portal (CAP) functions

Throughout this Recommendation, the terms Address Binding, Address Unbinding, Address Translation, and Session are used as defined in RFC 2663. In addition, the term Mapping is defined 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 as 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.

8.2.2.2 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: Pass-through 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 Pass-through mode, the CAP acts as a transparent bridge [ISO/IEC 15802-3] between the WAN-Data realm and LAN-Pass realm. In Pass-through 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 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. The CAP will enter the value 0 into the WAN and LAN port number entries of the CAP Mapping Table. The 0-value port number entry will serve two purposes:

1) indicate to the CAP that the port numbers are not to be translated; and

2) indicate to anyone reading the CAP Mapping Table that this is a C-NAT mapping, thereby providing a distinction between C-NAT Mappings (port number 0) and C-NAPT Mappings (non-zero port number).

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 cannot 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.

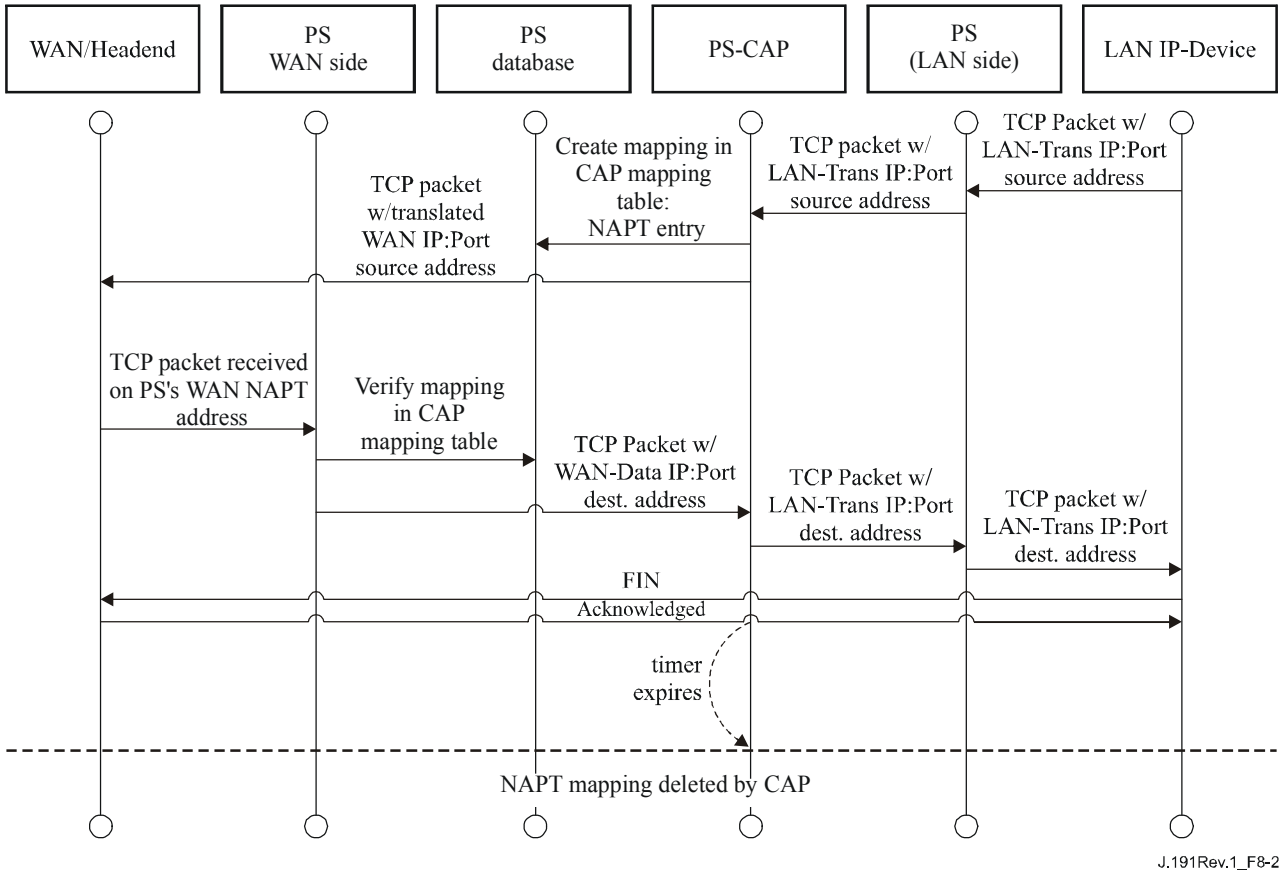

Figure 8-2/J.191 – PS configuration (CAP mapping table – NAPT) sequence diagram

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 Pass-through Table (cabhCapPass-throughTable). 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 cabhCapPass-throughTable, 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 cabhCapPass-throughTable, the frame is transparently bridged to the appropriate LAN interface. If the MAC address does not exist in the cabhCapPass-throughTable, 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), it will process broadcast traffic in accordance with RFCs 919, 922, 1812, and 2644. It is also assumed that when the PS is in Pass-through Mode, the 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 Pass-through 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 (8.2.2.3) 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.

8.2.2.3 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 out the proper PS LAN interface. 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 Pass-through bridging function (depending on the active packet handling mode).

8.2.2.4 Multicast

The CAP supports WAN-to-LAN Multicast traffic by transparently bridging downstream IGMP messaging [RFC 2236] and downstream IP Multicast 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.

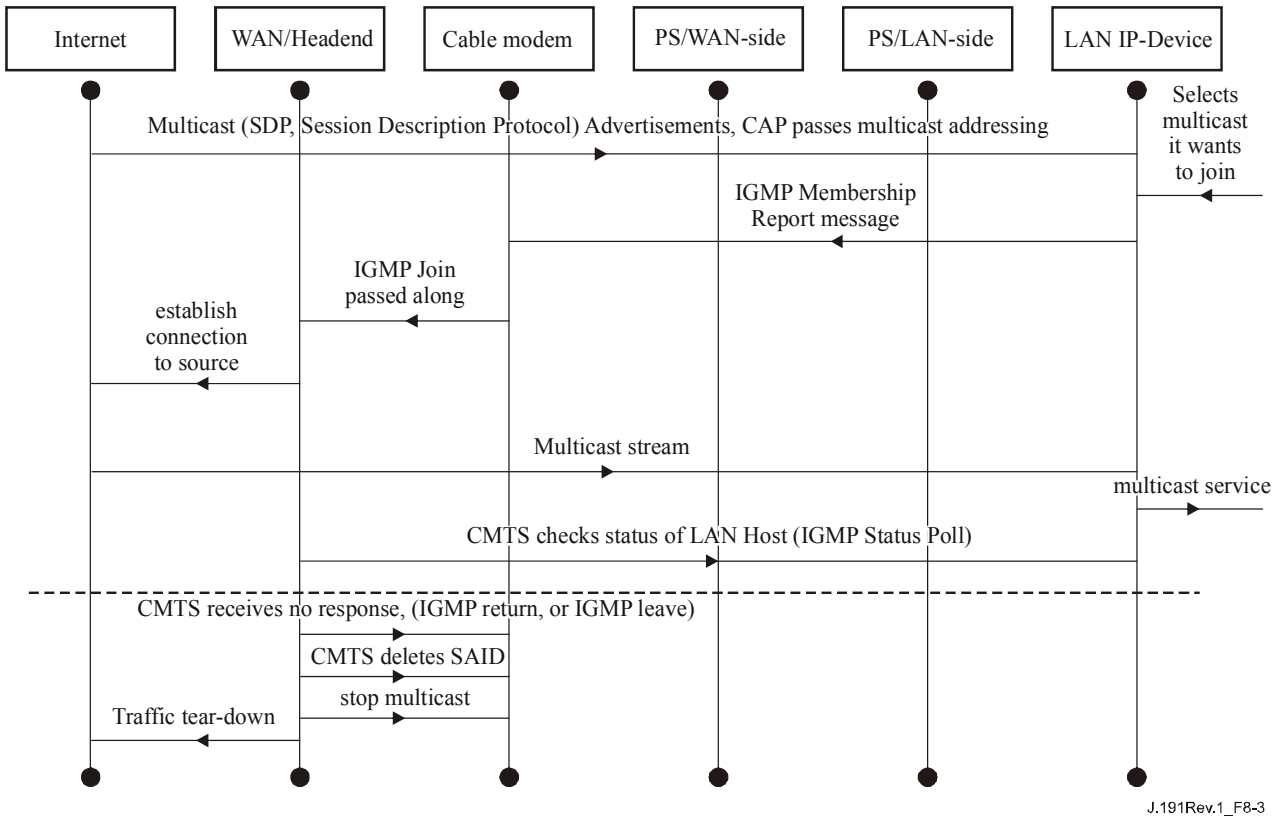

Figure 8-3/J.191 – Multicast via IGMP sequence

8.2.2.5 Packet handling examples

This clause 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. These examples are informative only and do not imply any requirements on implementation.

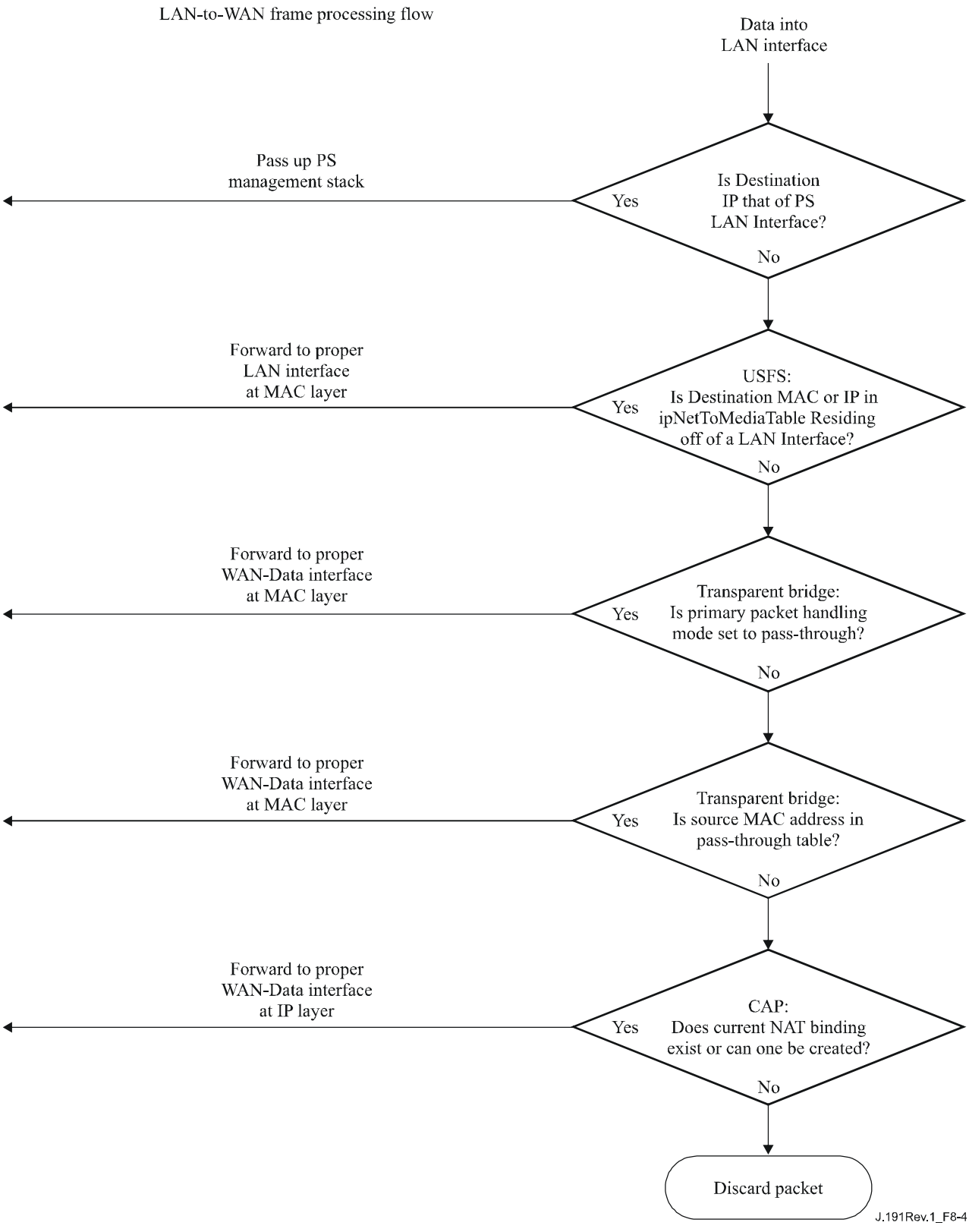

Figure 8-4/J.191 – LAN-to-WAN packet processing example

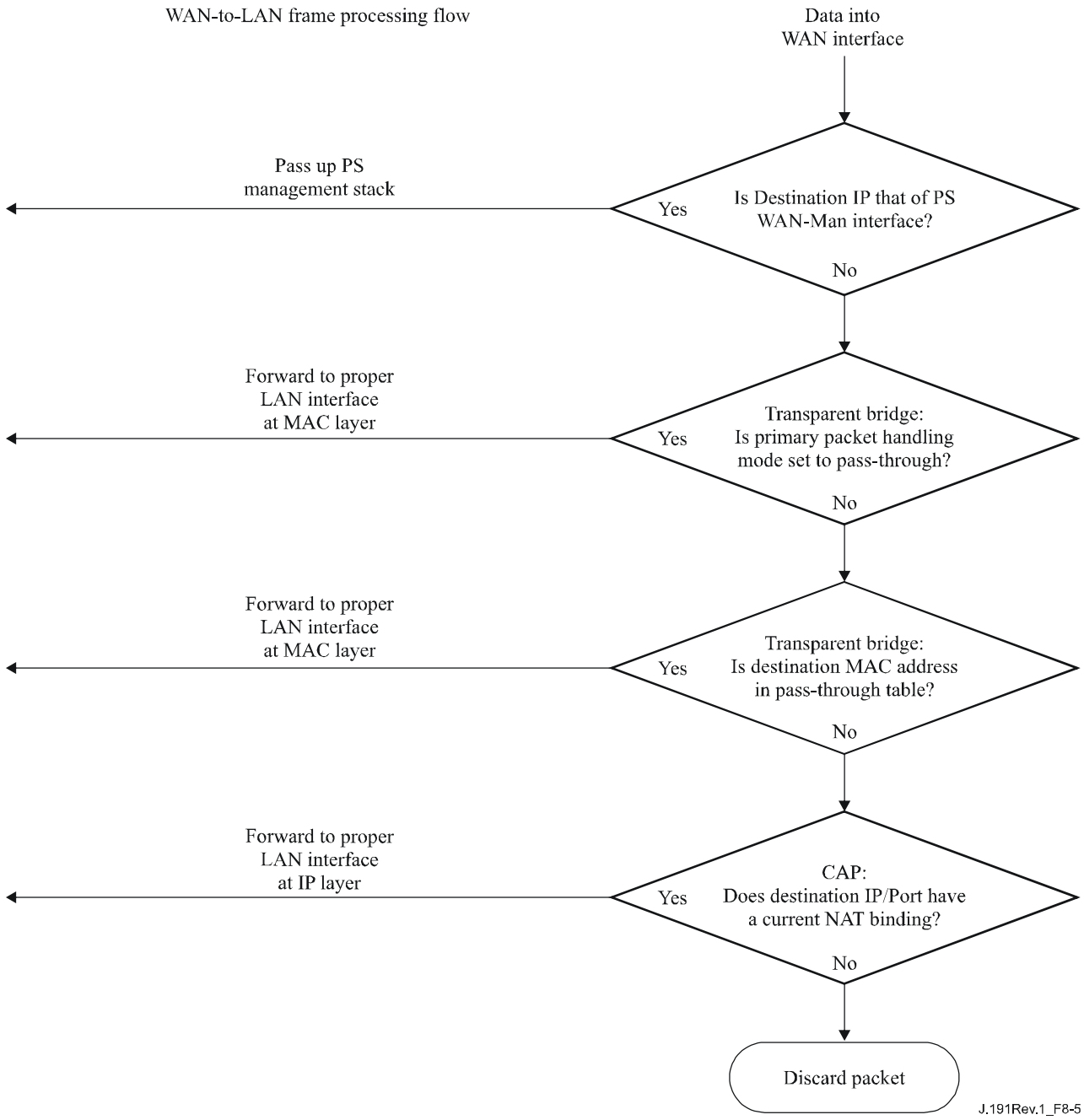

Figure 8-5/J.191 – WAN-to-LAN packet processing example

8.3 CAP requirements

8.3.1 General requirements

All logical IP interfaces on the Portal Services element MUST be compliant with RFC 1122, sections 3 and 4, to enable standard communication with Internet Hosts.

The CAP 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 Pass-through, 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.

8.3.2 Packet handling requirements

The CAP MUST support Pass-through Mode, C-NAT Transparent Routing Mode, and C-NAPT Transparent Routing Mode, and the CAP 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 CAP 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 CAP 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 CAP MUST enter zero for cabhCapMappingLanPort and cabhCapMappingWanPort for that row.

The CAP 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 CAP 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.

8.3.2.1 Pass-through requirements

When the CAP's Primary Packet-handling Mode, cabhCapPrimaryMode, is set to Pass-through mode, the CAP MUST act as a transparent bridge, as defined in ISO/IEC 15802-3, 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 Pass-through, USFS processing MUST take precedence over LAN-to-WAN bridging decisions.

8.3.2.2 C-NAT and C-NAPT transparent routing requirements

When the Primary Packet-handling Mode (cabhCapPrimaryMode) is set to C-NAT the CAP 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 CAP 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 CAP 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 CAP MUST support the creation of Dynamic C-NAT and C-NAPT Mappings, initiated by LAN-to-WAN TCP, UDP, or ICMP traffic. The CAP MUST enable the NMS system to read (via the CMP) Dynamic CAP Mapping (cabhCapMappingTable) entries.

The CAP 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 CAP 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 CAP 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.

8.3.2.3 Mixed Bridging/Routing Mode requirements

The CAP MUST support Mixed Bridging/Routing Mode as described in 8.2.2, 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 cabhCapPass-throughTable, the CAP 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 8.2.2, the USFS function MUST be applied to all LAN originated traffic received.

8.3.3 USFS requirements

Upstream Selective Forwarding Switch (USFS) functionality MUST be applied to packet processing, regardless of the CAP's packet-handling mode (Pass-through, 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 out the proper physical LAN interface.

9 Name resolution

9.1 Introduction/overview

9.1.1 Goals

The goals of the 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/J.191 – 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 Headend DNS server, for DNS clients within LAN IP Devices, for resolution of non-local hostnames.

9.2.2 System description

This clause provides an overview of the name resolution services within the PS element.

9.2.2.1 Name resolution functional overview

The Cable 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.

All 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 Headend DNS server, 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 at this time.

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 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.

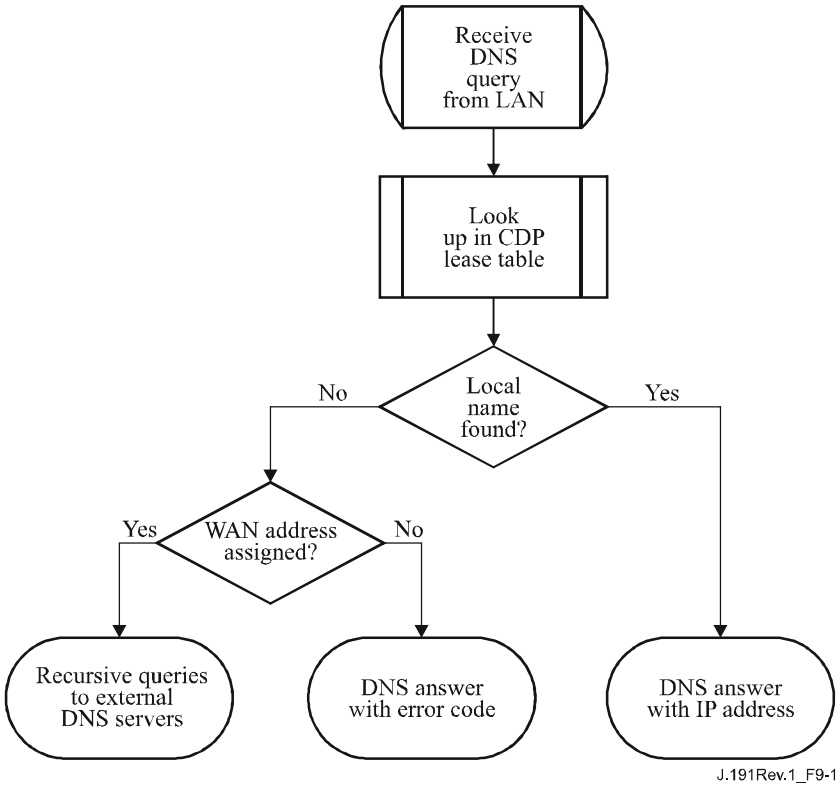

Figure 9-1/J.191 – CNP packet processing

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).

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 (see Table 9-2):

Table 9-2/J.191 – SOA record fields

RFC 1035 RDATA field	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 zone transfers are not specified, the 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 and 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 and 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's 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 hostname cannot be resolved by the CNP from local data AND the CDP's cabhCdpWanDnsServerTable is populated with at least one cabhCdpWanDnsServerIp entry, the CNP MUST attempt to resolve the hostname query via recursive queries to external DNS servers, starting with queries to DNS servers represented by cabhCdpWanDnsServerIp entry in the cabhCdpWanDnsServerTable.

If the hostname cannot be resolved by the CNP from local data AND no cabhCdpWanDnsServerIp entries exist in the cabhCdpWanDnsServerTable, the CNP MUST respond to the hostname 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 clause describes the role of the IPCable2Home environment in enabling home networking applications to utilize IPCablecom and DOCSIS QoS resources. These resources provide a management mechanism that prioritizes data session flows to support real-time application traffic, such as VoIP, A/V streaming, and video gaming, by reducing packet latency and jitter delays. IPCablecom and DOCSIS QoS mechanisms also allow more efficient traffic management over the HFC network.

QoS defines the necessary PS element requirements that enable IPCablecom applications to establish different levels of QoS across the HFC network.

10.1.1 Goals

The goals for QoS include:

- Enable home networking applications to establish prioritized data sessions between the CMTS and HA device using IPCablecom compliant messaging.
- Facilitate design and field-testing leading to the manufacture and interoperability of conforming hardware and software by multiple vendors.

10.1.2 Assumptions

The following assumptions were made for IPCable2Home QoS:

- QoS assumes J.112 and IPCablecom systems exist on the cable network.
- To avoid problems with NAT functions in the CAP element, IPCablecom compliant applications will use LAN-Pass addressing as defined in clauses 7 and 8.

10.2 QoS architecture

The Cable Quality of Service (CQoS) architecture is composed of IPCable2Home functional elements and the HA device class. 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 CQoS-Domain. The basic CQoS elements are presented in 10.2.2.

10.2.1 System design guidelines

The IPCable2Home QoS system design guidelines are listed in Table 10-1.

Table 10-1/J.191 – IPCable2Home QoS system design guidelines

Number	QoS system design guidelines
QoS 1	A standard QoS signalling mechanism will exist that allows residential gateway (HA) products to support the establishment of prioritized service sessions across the DOCSIS network for multimedia applications.
QoS 2	Multimedia applications may be embedded in the residential gateway (HA) device or on an external device connected via a home networking technology.
QoS 4	CQoS 1.0 must support both the Embedded PS and Stand-Alone PS HA configurations.
QoS 5	Multimedia applications may include IPCablecom services (E-MTA/S-MTA).

10.2.2 QoS System description

The CQoS Architecture is composed of the following entities:

- CQoS Domain;
- Portal services function (PS);
- IPCable2Home Quality of Service Portal function (CQP);
- HA device;
- CMTS.

The CQoS-Domain defines the sphere of direct influence of CQoS functionality, which is extended to the HA device from the cable network's Headend. The PS and CQP elements are wholly within the CQoS-Domain and are specified. The CQoS domain exists to provide services to IPCablecom compliant applications.

The reference architecture also describes the HA device. See clause 5.

The cable modem termination system (CMTS) is located at the cable network's Headend and manages the DOCSIS QoS functions.

10.2.2.1 Element – Portal services

The Portal Services (PS) element is a logical element that contains network addressing, management, security, and QoS portal components that provide translation functions between the HFC network and the home network. The PS resides in HA devices only (see clause 5). The QoS component is referred to as the Cable Quality-of-Service Portal (CQP).

10.2.2.1.1 CQP component

The PS element includes a Cable Quality-of-Service Portal (CQP) component. The CQP acts as a CQoS portal for IPCablecom compliant applications. Its primary function is to forward QoS messaging between the CMTS and IPCablecom Applications.

10.2.2.1.2 Stand-alone PS configuration

This Recommendation does not define QoS requirements between a PS and a CM, and thus functions for maintaining data session priorities and avoiding contention due to asynchronous access by multiple devices will not be specified. It is recommended that this interface be a high bandwidth, dedicated PS-to-CM connection (not shared with other devices) to minimize QoS packet jitter due to multi-device contention.

10.2.2.2 CQoS Domain

The CQoS Domain exists on a per-home basis. Individual homes are separate and have independent CQoS Domains. The CQP element bounds the CQoS Domain within a given home.

10.2.2.3 Physical device classes and CQoS functional elements

HA devices contain the PS logical element and the CQP functional element. The CQP acts as a transparent bridge for IPCablecom applications (APP) QoS messaging. An example of the relationship between the CQoS functional elements and the HA device class is presented in Figure 10-1.

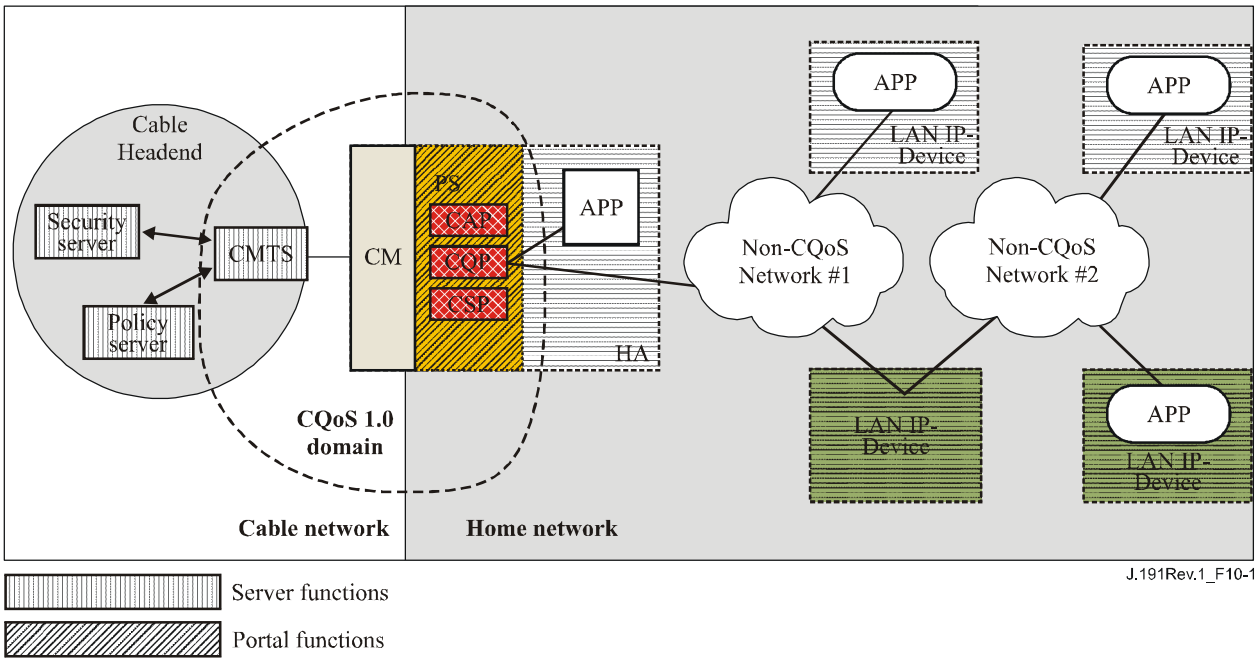

Figure 10-1/J.191 – Example of CQoS functional elements

10.3 Cable QoS messaging requirements

The IPCable2Home QoS (CQoS) architecture consists of the CQP functional element in the CQoS domain. The CQP exists in the PS and 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.

Functional requirements for the CQP and other CQoS elements are defined in the following subclauses.

10.3.1 CQP requirements

The CQP MUST act as a transparent bridge and forward IPCablecom J.161 and J.163 QoS messaging between the CMTS and IPCablecom applications. Application data is associated to a DOCSIS 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 CQP requirement 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 5.5).

10.3.2 QoS policy management and admission control

IPCable2Home QoS messaging is defined by IPCablecom specifications (ITU-T Recs J.161 and J.163). As such, the IPCable2Home QoS policy management and admission control functions are also defined by these IPCablecom Recommendations.

11 Security

11.1 Introduction/Overview

This clause defines the security interfaces, protocols and functional requirements needed to reliably deliver cable-based IP services in a secure environment to the HA.

Supporting the delivery of reliable multimedia IP services to client devices on a home network requires a secure mechanism that protects these services from illegal access, monitoring, and disruption. The purpose of any security technology is to protect value, whether a revenue stream, or a purchasable information asset of some type. Threats to this 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 they perceive extreme value. 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).

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 home connections used to offer high value cable-based services so that it is at least as secure as the Cable Modem and IPCablecom technologies on the hybrid fibre-coax (HFC) network.
- Provide flexible security mechanisms that are compatible with Cable Modem and IPCablecom security mechanisms used on the HFC network.

11.1.2 Assumptions

The assumptions for the IPCable2Home security environment include:

- It is assumed that in the Embedded HA, i.e., a PS/CM enclosed in a single physical device, the CM is a J.112 (or J.122) cable modem.
- Lower security levels may exist on the home network when the services provided are considered to be of low value.

11.2 Security architecture

The Security Architecture is based on the general architecture as defined in clause 5. The architecture defines a Portal Services (PS) element, which includes Management/Provisioning, Security and QoS functions.

The architecture also includes a set of Headend elements. These include the Cable Modem Termination System (CMTS), Dynamic Host Configuration Protocol (DHCP) server, Network Management System, Security server, etc.

The specification focuses on the definition, functionality and interfaces of the security functions and security related 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 specification.

Table 11-1/J.191 – IPCable2Home security system design guidelines

Reference	Security system design guidelines
SEC1	The operator will have the ability to remotely manage compliant firewall products.
SEC2	A firewall event logging/messaging interface that allows the operator to monitor and review firewall activity will be included in the security system design.
SEC3	Firewall management messages between the cable Headend and PS will be authenticated and optionally encrypted to protect against unauthorized monitoring and control.
SEC4	Mutual authentication of elements will be included in the security system design.
SEC5	The home security level will be such that it is not easy for the average subscriber to gain unauthorized access to the HFC network and cable-based services.
SEC6	Once a subscriber's account has been established, authentication of the PS with the operator's provisioning system will be automatic.
SEC7	The operator will have the ability to securely download software images, configuration files and firewall rule sets to the PS element.
SEC8	IPCable2Home security will provide the necessary support for IPCablecom Secured DQoS through the firewall.
SEC9	Network management messages between the cable Headend and PS will be authenticated and optionally encrypted to protect against unauthorized monitoring and control.

This clause limits its scope to these primary system security requirements, but acknowledges that in some cases additional security may be desired. The concerns of individual operators or manufacturers may result in additional security protections. This Recommendation does not restrict the use of further protections, as long as they do not conflict with the intent and guidelines of this Recommendation.

11.2.2 System description

This clause provides an overview of all the elements that are part of the security architecture.

The Security architecture includes the following security elements:

- Security-Domain;
- Portal Services function (PS);
- Cable Security Portal function (CSP);
- Firewall (FW);
- Security Server (KDC, Key Distribution Centre).

The Security-Domain defines the boundary of the sphere of direct influence where security functionality is extended to the PS from the cable network's Headend. The PS, CSP, and FW elements are wholly within the Security Domain. The PS element contains network addressing, management and security portal functions. The CSP acts as the boundary element between the Security-Domain and the non-secure domain. The Security-Domain exists to provide security services to compliant devices.

These elements contain Client, Server or Portal specific functionality and can exist in different types of physical devices. The architecture defines the Home Access (HA) device class. An example of the relationship between the different security elements and HA device classes is presented in Figure 11-1. In Figure 11-1, in-home applications are represented as APP and the OSS server is the NMS server.

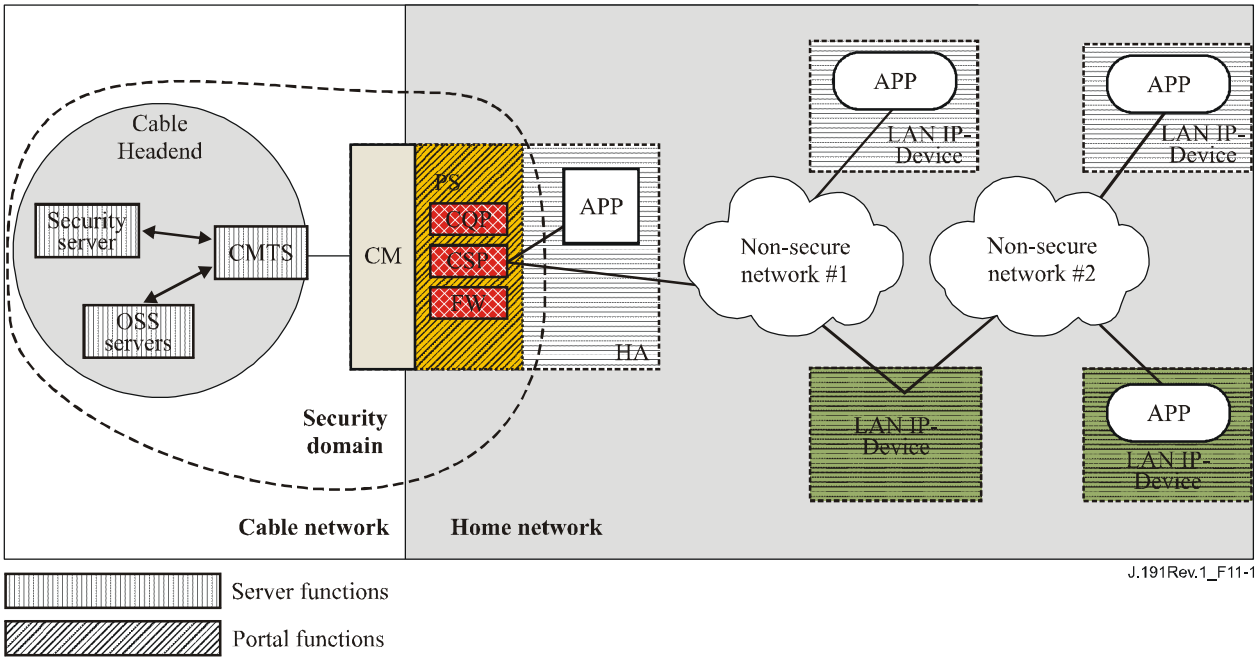

Figure 11-1/J.191 – IPCable2Home security elements

11.2.2.1 Security domain

The Security Domain is defined in Figure 11-1 and encompasses the PS element in the HA and the illustrated Headend servers.

11.2.2.2 PS function – Portal Services

Portal Services (PS) is a logical element that contains network addressing, management and security portal functions. It resides in HA devices only. The PS includes the following elements:

- Cable Security Portal (CSP);
- Firewall (FW).

The CSP acts as a security portal for other PS elements. One of its primary functions is to forward security messaging between Headend OSS servers (including the security server) and IPCablecom applications. The CSP also provides security services, such as authentication and key management, for the PS element.

The PS also includes firewall functionality. The firewall provides protection to the user, as well as the HFC network, from unwanted traffic coming from the WAN or local-area network (LAN) domains. Such traffic may include deliberate attacks on the in-home network as well as traffic limiting for parental control applications.

The security specification will not define a detailed specification for the implementation of a firewall, but will instead define a set of requirements to enable remote management by the operator.

Typically, firewalls are built using a combination of two different components: packet filtering and proxy server. 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-dropping decision is based on static configuration information that mandates inspection of packet header fields including: 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 may have to be configured on a firewall which can be very difficult, requiring a good understanding of the type of services (protocols) to be filtered.

An application-specific proxy (ASP), another typical firewall component, creates a protocol endpoint and relay by implementing the necessary client and server parts of a specific client-server protocol. 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, being protocol specific, an ASP understands the protocol and can be configured to block only subsections of the protocol. 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, say depending on which directions these commands are issued.

The particular combination of packet filters and ASPs on a given firewall product constitutes a trade off between performance and the security level the firewall awards. Typically being a network layer mechanism, packet filters tend to yield better performance than ASPs that are application layer mechanisms. A compromise solution becoming increasingly popular consists in 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.

Static or SPFs and the ASPs in a firewall are ultimately the control knobs the security policy uses to implement 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. It is the rule set derived from the security policy that defines the collection of access control rules (filter and proxy action rules) which then determines which packets the firewall forwards and which it rejects. A big challenge is in deriving the rule set from the statements in the security policy, which is usually expressed in a high-level human language.

Because a firewall only needs the rule set to configure its SPF and ASP components, defining the security policy and deriving a corresponding rule set are considered outside the scope of this Recommendation. An appropriate rule set is to be configured into a firewall via an authenticated firewall configuration file download. The actual format for the file containing the rule set applicable to a particular firewall product and how that file is used in the firewall to configure the SPF and ASP components is implementation specific. This Recommendation only addresses the authentication mechanism used in downloading a firewall rule set to the PS element.

Figure 11-2 illustrates the relationship among the firewall components. In particular, Figure 11-2 suggests that a rule set (RS) is to be used for the internal configuration of all the firewall components. These components consist of the inbound packet filter (IPF), the outbound packet filter (OPF), and the applications specific proxy (ASP) or stateful packet filter (SPF) functions. Figure 11-2 also provides a more detailed view of the PS and its relationship to firewall functions and other components in the HA device. In particular, Figure 11-2 suggests that the firewall Application Specific Proxy/Stateful Packet Filtering (ASP/SPF) function is intimately associated with the CAP Network Address Translation (NAT) function. Because a NAT function breaks some applications, application-specific processing is required as part of the NAT implementation and, therefore, the PS implementation MAY combine the ASP/SPF and NAT functions.

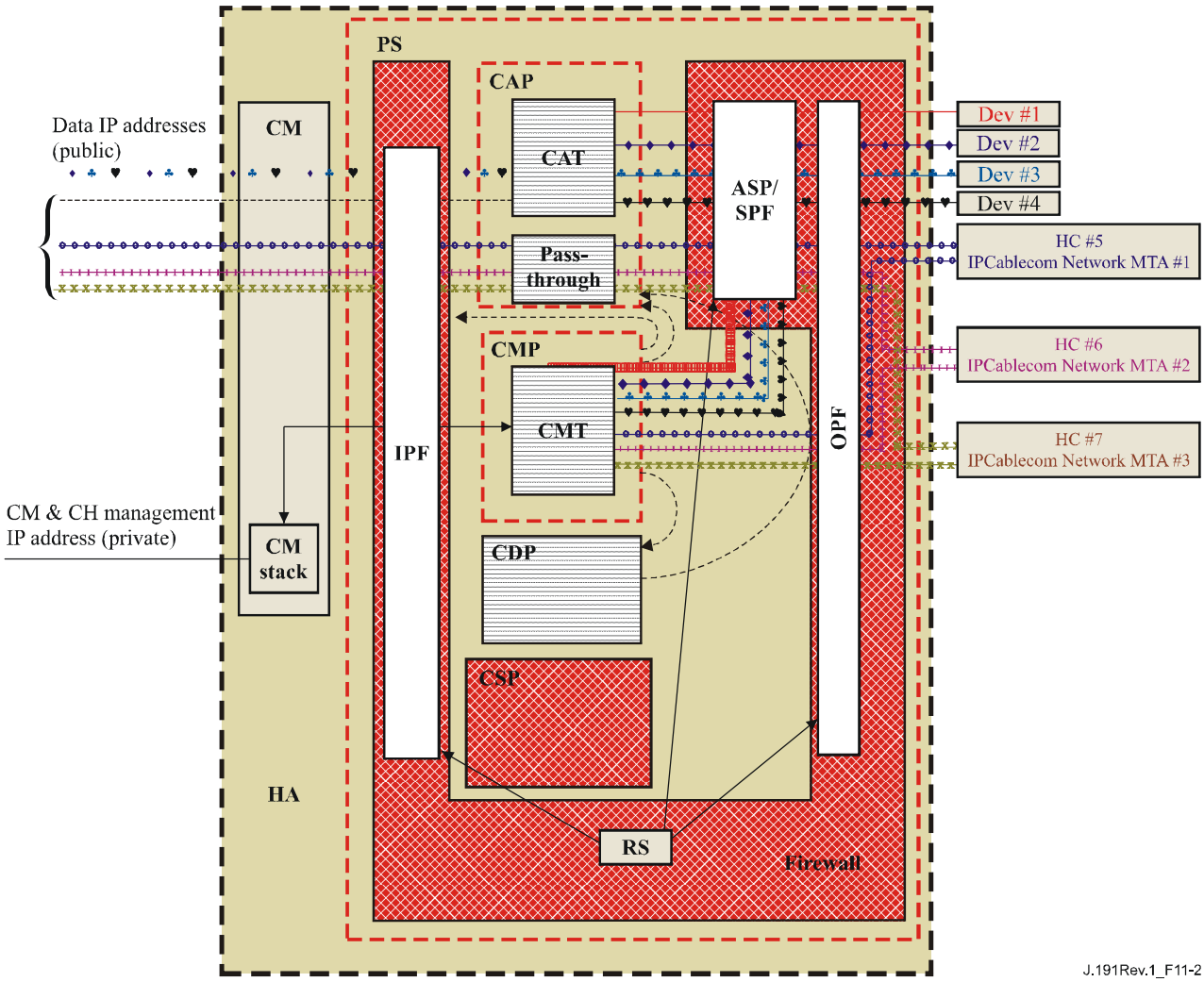

Figure 11-2/J.191 – Example of a PS element in an HA device

11.2.3 Key Distribution Centre (KDC) server

The Security server supported in IPCable2Home is the Key Distribution Centre (KDC) server. If a KDC server that supports IPCable2Home is available in the Headend, it will be used to provide 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.

11.2.4 Other related elements and functions

The following elements are not considered to be security elements, but do use or take part in the management of these security services.

- OSS;
- CMP.

The OSS represents a set of Headend servers that enable management of IPCable2Home elements in the home. The OSS servers communicate with the CMP to manage the security functions and services. The link between the OSS and CMP is secured using the authentication and privacy services defined in this Recommendation.

The CMP is the management function within the PS. The security architecture provides authentication and other security services for its communication with OSS servers at the Headend. The CMP enables management of PS functions including management of security services.

Further detail of these elements and their functions can be found in clauses 12 and 13, and the QoS in clause 10.

11.3 Requirements

For all references to IPCablecom security, please refer to ITU-T Rec. J.170.

11.3.1 Element authentication

For security purposes, it is important to know with whom you are communicating prior to exchanging any meaningful information. Authentication provides a means to securely identify the unknown parties who wish to communicate. There are three parts to authentication, the identity credential, the checking of the identity credential for validity and the common means to communicate the identity information. This Recommendation specifies an industry standard identification credential, the use of X.509 certificates in conjunction with RFC 3280. 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. Additionally, public key certificates provide a secure way to communicate the identity information.

When a KDC that supports this Recommendation is available in the Headend, authentication is supported. If a KDC is available, it is recommended that the cable operator provision the PS Element in SNMP Provisioning Mode (as described in 5.5) to take advantage of the specified mutual authentication protocol with the use of Kerberos using the PKINIT extension. Kerberos provide 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 already been specified by another ITU project, i.e., IPCablecom, this Recommendation references the IPCablecom model when appropriate.

11.3.1.1 Kerberos/PKINIT

When the PS Element is provisioned in SNMP Provisioning Mode, this Recommendation specifies the use of Kerberos with the PKINIT public key extension for authenticating elements and for supporting key management requirements. Elements (clients) authenticate themselves to the KDC with the PKINIT protocol. Once authenticated to the KDC, clients may 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 6.4 and 6.5 of ITU-T Rec. J.170, unless otherwise noted in this Recommendation. 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. The reader should be aware that this SNMP functionality in general should 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 it 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. The certificates used in PKINIT for IPCable2Home are specified in the PKI Section of this Recommendation. Where IPCablecom specifies an MTA device certificate, this Recommendation provides a certificate for the PS Element (PS Element Certificate), and implementations of PS Elements MUST include the PS Element Certificate.

The following clauses for Kerberos functionality from ITU-T Rec. J.170 do not apply to this Recommendation:

- clause 6.4.8.4 Pre-Authenticator for Provisioning Server Location;
- clause 6.4.7 MTA Principal Names;
- clause 6.4.8 Mapping of MTA MAC Address to MTA FQDN;
- clause 6.4.10 Service Key Versioning;
- clause 6.4.11 Kerberos Cross-Realm Operation;
- clause 6.5.4 Rekey Messages;
- clause 6.5.6 Kerberized IPSec;
- clause 6.4.6 Kerberos Server Locations and Naming Conventions.

11.3.1.2 IPCable2Home specific authentication variables

The model IPCablecom specifies some specific variables names for Kerberos in the IPCablecom Network Architecture. In order for this Recommendation 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.

11.3.1.3 IPCable2Home profile for Kerberos Server locations and naming conventions

Kerberos Realm names MAY use the same syntax as a domain name, Kerberos Realms, however; MUST be in all capitals. Kerberos Realm details MUST be followed according to Annex B/J.170.

The KDC conventions listed in 6.4.6.2/J.170 are considered informative for this Recommendation 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>

Where <SNMP entity address> is the service provider's SNMP entity IP address (CDC DHCP Option 177, suboption 3) in dotted notation enclosed in square brackets (e.g., [12.34.56.78]).

11.3.2 Public Key Infrastructure (PKI)

This Recommendation uses public key certificates, which comply with ITU-T Rec. X.509 | ISO/IEC 9594-8 and RFC 3280.

11.3.2.1 Generic structure

11.3.2.1.1 Version

The Version of the certificates MUST be ITU-T Rec. X.509 v3, as is noted as v2 in the actual certificate (because v1 did not have any associated version numbering). All certificates MUST comply with RFC 3280 except where the non-compliance with the RFC is explicitly stated in this clause. Any non-compliance request by this Recommendation for content does not imply non-compliance for format. Any specific non-compliance request for format will be explicitly described.

11.3.2.1.2 Public Key type

RSA Public Keys are used throughout the certificate hierarchies described in 11.3.2.2. The subjectPublicKeyInfo.algorithm OID used MUST be 1.2.840.113549.1.1.1 (rsaEncryption).

The public exponent for all RSA keys MUST be $F_4 – 65537$.

11.3.2.1.3 Extensions

The extensions (subjectKeyIdentifier, authorityKeyIdentifier, KeyUsage, and BasicConstraints) MUST follow RFC 3280. Any other certificate extensions MAY also be included as non-critical. The encoding tags are [c:critical, n:non-critical; m:mandatory, o:optional] and these are identified in the table for each certificate.

11.3.2.1.3.1 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 SHA-1 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).

11.3.2.1.3.2 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.

11.3.2.1.3.3 KeyUsage

The keyUsage extension MUST be used for all Certificate 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.

11.3.2.1.3.4 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 Tables 11-2 through 11-13.

11.3.2.1.4 Signature algorithm

The signature mechanism used MUST be SHA-1 [FIPS 186-2] with RSA Encryption. The specific OID is 1.2.840.113549.1.1.5.

11.3.2.1.5 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 Recommendation.

11.3.2.1.6 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.

11.3.2.2 IPCable2Home certificate hierarchies

There are three distinct certificate hierarchies used. The Manufacturer Chain is used to identify authorized manufacturers; the Code Verification Chain is used to identify compliant software images; the 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 Recommendation can apply to all ITU projects needing certificates. Each project may adopt this hierarchy as there is an opportunity to move to a more generic, shared certificate structure. Also each project may need to 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 may be differences in the end-entity certificates required for each project, but 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 or the same as the IPCablecom MSO KDC (IPCablecom specifies the use of several KDCs).

In Figure 11-3 below, the term Certificate Authority is abbreviated as CA and Code Verification Certificate is abbreviated as CVC.

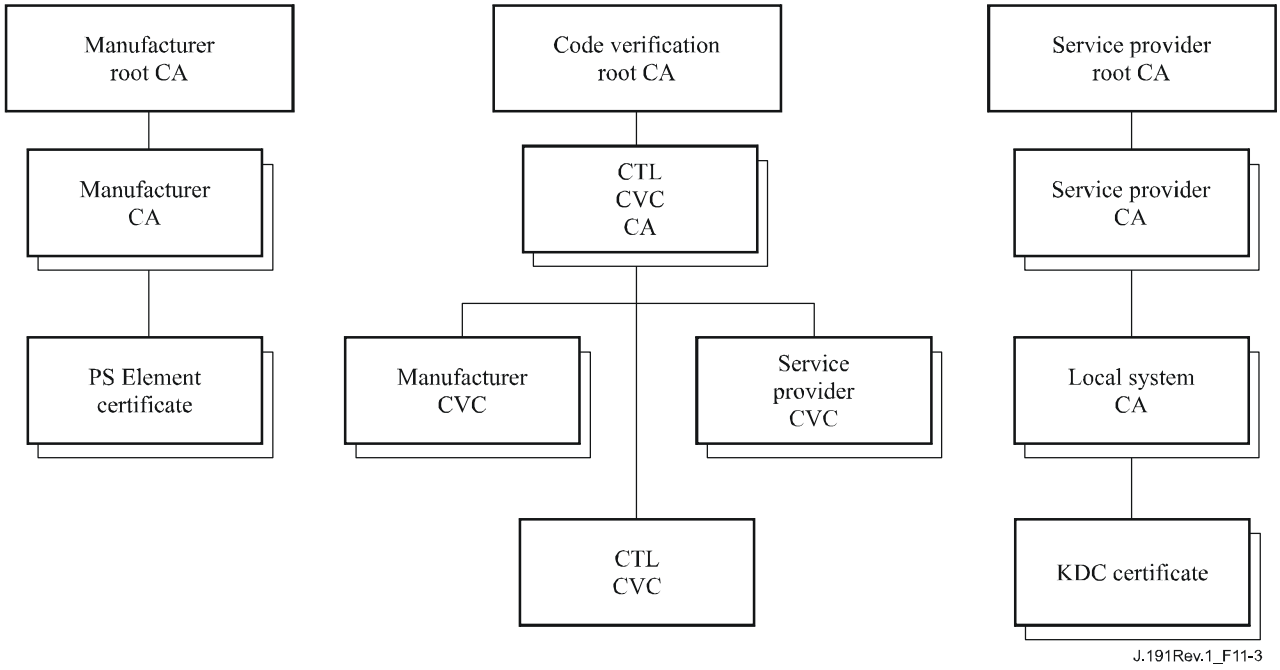

Figure 11-3/J.191 – IPCable2Home certificate hierarchy

11.3.2.2.1 Manufacturer certificate hierarchy

The Manufacturer certificate hierarchy, or Manufacturer chain, is rooted at a Manufacturer Root CA, which is used to issue Manufacturer Certificate 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 Tables 11-2, 11-3 and 11-4. 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, clause 11.3.2.

11.3.2.2.1.1 Manufacturer Root CA Certificate

The Manufacturer Root CA Certificate (see Table 11-2) MUST be verified as part of the certificate chain containing the Manufacturer Root CA Certificate, Manufacturer CA Certificate and the PS Element Certificate.

Table 11-2/J.191 – Manufacturer Root CA Certificate

Subject Name Form	C=<country> O= CN=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).

11.3.2.2.1.2 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 11-3/J.191 – Manufacturer CA Certificate

Subject Name Form	C=<country> O=<CompanyName> [ST=<state/province>] [L=<city>] OU= [OU=<Manufacturer's Facility>] CN=<CompanyName> Mfg CA
Intended Usage	This certificate is issued to each Manufacturer by the 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 MIB. 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	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 (CN) in the Common Name field.

11.3.2.2.1.3 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 11-4/J.191 – PS Element Certificate

Subject Name Form	C=<country> O=<Company Name> [ST=<state/province>] [L=<city>] OU=IPCable2Home [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 MIB. 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), authorityKeyIdentifier [n,m].

11.3.2.2.2 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. The Code Verification CA also issues the CVC. This chain is specifically used to authenticate software downloads. The 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 Tables 11-5, 11-6, 11-7, 11-8 and 11-9 below. 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, clause 11.3.2.

11.3.2.2.2.1 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 11-5/J.191 – Code Verification Root CA Certificate

Subject Name Form	C=<country> O= CN=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).

11.3.2.2.2.2 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 11-6/J.191 – Code Verification CA Certificate

Subject Name Form	C=<country> O= CN=CVC CA
Intended Usage	This certificate is issued to a Certification body by the Code Verification Root CA. This certificate issues Code Verification Certificates.
Signed By	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).

11.3.2.2.2.3 Manufacturer Code Verification Certificate

This certificate MUST be verified as part of the certificate chain containing the Code Verification Root CA Certificate, the Code Verification CA Certificate, and the Code Verification Certificates.

Table 11-7/J.191 – 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].

11.3.2.2.2.4 Code Verification Certificate

The Code Verification Certificate MUST be verified as part of a certificate chain containing the Code Verification Root CA Certificate, the Code Verification CA Certificate, and the Code Verification Certificate.

Table 11-8/J.191 – Code Verification Certificate

Subject Name Form	C=<country> O= CN=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].

11.3.2.2.2.5 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, the Code Verification CA Certificate, and the Service Provider Code Verification Certificate.

Table 11-9/J.191 – 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]

11.3.2.2.3 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 then 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 Tables 11-10 through 11-13 below. 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, clause 11.3.2.

11.3.2.2.3.1 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, the optional Local System CA Certificate and the Ancillary Certificates.

Table 11-10/J.191 – Service Provider Root CA Certificate

Subject Name Form	C=<country> O= CN=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)

11.3.2.2.3.2 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, the optional Local System CA Certificate and the Ancillary Certificates.

Table 11-11/J.191 – 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 Kerberos realm. The 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 this Recommendation, 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 Kerberos client MUST accept any Service Provider OrganizationalName attribute, but it MUST later check that the value added into the MIB 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 (CN) in the Common Name field.

11.3.2.2.3.3 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, the optional Local System CA Certificate and the Ancillary Certificates.

Table 11-12/J.191 – 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 (CN) in the Common Name field.

11.3.2.2.3.4 KDC Certificate

This certificate MUST be verified as part of the certificate chain containing the Service Provider Root CA Certificate, Service Provider CA Certificate, the optional Local System CA Certificate and the Ancillary Certificates (e.g., the KDC Certificates).

The KDC Certificate MUST include the Kerberos PKINIT subjectAltName as specified in 8.2.4.1/J.170.

Table 11-13/J.191 – KDC Certificate

Subject Name Form	C=<country> O=<Company Name> [OU=<Local System Name>] OU=Key Distribution Centre 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 Annex C/J.170)

11.3.2.3 Certificate validation

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 this certificate 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 this 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, 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 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 entities may be before, the same as or after the validity end date for the issuing CA as specified in the Certificate tables.

11.3.2.3.1 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.

11.3.2.3.2 Validation for the Code Verification Chain and Root verification

A back office server may check the validity of the Code Verification Chain prior to beginning the software download process. For details, see 11.3.7.

11.3.2.3.3 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. 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.

11.3.2.4 Certificate revocation

Certificate revocation is for further study.

11.3.3 Secure management messaging

The security algorithm used to initialize SNMP management messaging depends upon the provisioning mode of the PS element (see 5.5). There are two types of provisioning modes, DHCP Provisioning Mode and SNMP Provisioning 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 subclauses 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 11.3.3.1.2 and 11.3.3.2.

11.3.3.1 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.

11.3.3.1.1 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 6.3.6.1.

11.3.3.1.2 Coexistence Mode

If the PS Element is provisioned in DHCP Provisioning Mode with Coexistence Mode and the management messaging protocol is determined to be SNMPv3 (see 6.3.6.1), then the PS Element MUST use SNMPv3 security specified by RFC 3414. SNMPv3 authentication MUST be turned on at all times and SNMPv3 privacy MAY also be utilized.

In order to establish SNMPv3 keys, all SNMP interfaces MUST utilize the SNMPv3 initialization and key changes procedure described below.

To support SNMPv3 initialization and key changes the PS element MUST also be capable of receiving TLVs of type 34, 34.1, and 34.2 as defined in B.C.1.2.8/J.112 and implement the key-change mechanism specified in RFC 2786 which includes the usmDHKickstartTable MIB object.

11.3.3.1.2.1 SNMPv3 initialization

For each of up to 5 different security names, the Ultimate Authorization (PS Administrator) generates a pair of numbers. First, the PS Administrator 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:

$$z = g \text{ ^ } Rm \text{ MOD } p$$

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) = PS Administrator

TLV type 34.2 (SNMPv3 Kickstart Public Number) = z

The PS MUST support the VACM entries defined in 6.3.6.3. 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 = "PS Administrator"
```

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
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).

$$c = g \text{ ^ } xa \text{ MOD } p$$

where g is the 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 = "PS Administrator"
```

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, "PS Administrator" 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)
authentication key <----     PBKDF2 (salt = 0x98dfb5ac,
                     iterationCount = 500,
                     keyLength = 16 (usmHMACMD5AuthProtocol),
                     prf = id-hmacWithSHA1)
```

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.

11.3.3.1.2.2 Diffie-Hellman Key changes

The PS MUST support the key-change mechanism specified in RFC 2786.

11.3.3.2 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.

SNMPv3 authentication MUST be turned on at all times and SNMPv3 privacy MAY also be utilized. In order to establish SNMPv3 keys, all IPCable2Home SNMP interfaces MUST utilize Kerberized SNMPv3 key management as specified in 11.3.3.2.3.

11.3.3.2.1 SNMPv3 encryption algorithms

The encryption Transform Identifiers for Kerberized SNMPv3 key management MUST be followed as defined in 6.3.1/J.170.

11.3.3.2.2 SNMPv3 authentication algorithms

The authentication algorithms for Kerberized SNMPv3 key management MUST be followed as defined in 6.3.2/J.170.

11.3.3.2.3 Kerberized SNMPv3

The Kerberized key management profile specific for SNMPv3 MUST be followed as defined in 6.5.7/J.170.

11.3.3.2.4 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 [ITU-T Rec. J.170], the following defines the rule to be used in generating SNMPv3 Engine IDs for use in this application:

- The SNMPv3 Engine ID follows the format defined in RFC 2576, i.e., the first bit is set to 1 (one) and the appropriate value is used for the first four bytes [RFC 2576];
- 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 8-bit 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 2576. The above selection, though, is intended to reduce implementation complexity that would be required if all of the approaches in RFC 2576 were allowed.

11.3.3.2.5 Populating the usmUserTable

SNMPv3 security settings for the cable operator "CHAdministrator" user are defined in 6.3.6.3. The CHAdministrator is the ultimate authority for management of the Portal Services element. Other users can also be defined. In this clause, 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 (see Table 13-1, step CHPSWMD-11; Table 13-2, step CHPSWMS-11 and step CHPSWMS-13; and 13.3.3).

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 an element to process these messages, the necessary information requires to be entered in the *usmUserTable* [RFC 3414] for the element engine. The *usmUserTable* MUST be

populated in the PS Element right after the AP Reply message receipt with the following information:

- usmUserEngineID: the local SNMP Engine ID as defined in 11.3.3.2.4;
- usmUserName: PS Administrator-XXXXXX;
- usmUserSecurityName: PS Administrator-XXXXXX;
- 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 by 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: "".

The value XXXXXX MUST be the PS Element WAN-Man MAC address for that PS element.

New SNMPv3 users MAY be created by with standard SNMPv3 cloning as defined in RFC 2475. For additional information refer to 7.1.1.3.1/J.170.

11.3.4 Secure CQoS

CQoS provides QoS to IPCablecom applications that require a pass-through address. The IPCablecom DQoS messages between the MTA and the CMTS, CMS or CM are secured by the IPCablecom Security Specification. For IPCable2Home Security it is necessary to ensure these IPCablecom messages, already secured by IPCablecom, can pass through the firewall in the Portal Services Element (PS). It is not within the scope of this Recommendation to add security for IPCablecom messages. Because the PS element CQoS security requirement for this Recommendation is to just forward IPCablecom security messaging, there is no dependency on the NMS to support this function. Therefore, the CQoS security function remains the same for both DHCP Provisioning Mode and SNMP Provisioning Mode (see 5.5).

The requirement for securing CQoS is to provide security that is not unduly burdensome on the system. The key point to securing QoS is to ensure that theft of service and network disruption is reduced to an insignificant loss. It is also critical to understand that CQoS is the QoS gateway into the home and therefore will likely either control or support all the applications and appliances in the home requiring QoS on the cable network, to and through the PS. Therefore, it is especially critical to ensure this one entry point, not be the weak link in the QoS system.

11.3.4.1 CQoS architecture

The CQoS architecture consists of the CQP functional element that facilitates the establishment of QoS flows across the HFC for IP applications. The CQP element exists in the HA. See CQoS, clause 10. The CQP element acts as a transparent bridge for CQoS messaging between IPCablecom compliant applications and the CMTS. The IPCable2Home firewall will need to be capable of passing IPCablecom compliant security and QoS messaging.

See clause 10 for more complete details on CQoS.

11.3.4.2 IPCablecom secured DQoS architecture

Table 11-14/J.191 – Secure DQoS architecture

E-MTA		
Link to the MTA in the Home	**Protocol**	**Security protocol**
E-MTA/CM – CMS	NCS	IPSec
E-MTA/CM – CMTS	DOCSIS	BPI+

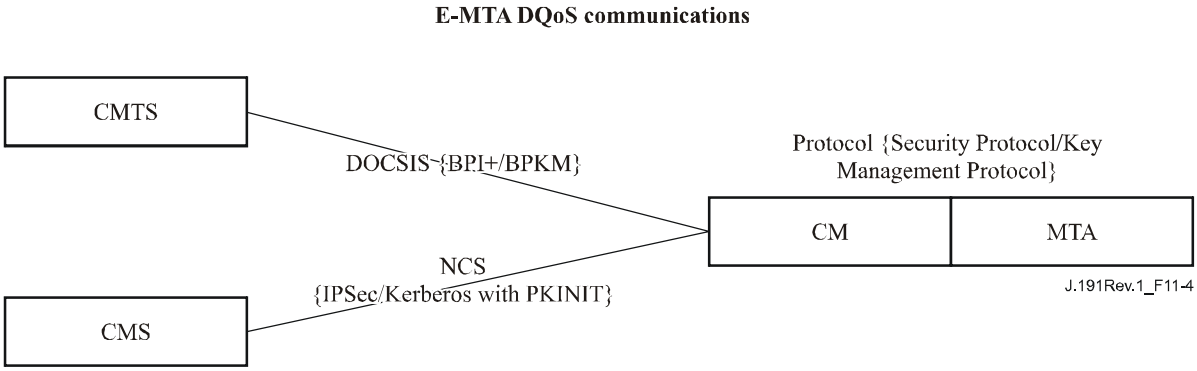

Figure 11-4/J.191 – Secure DQoS architecture to the MTA

11.3.4.3 CQos security architecture

CQoS requires IPCablecom DQoS messaging [ITU-T Rec. J.163] be passed to the E-MTA. All DQoS messaging MUST be secured as described in the IPCablecom Security Specification. The diagram below shows the protocols needed to support the E-MTA for DQoS. The only difference in the CQoS Secured Architecture and the IPCablecom DQoS Secured Architecture is that the PS is logically between the CM and the MTA. However, since the PS acts as a transparent bridge there are no changes in protocols or communication links.

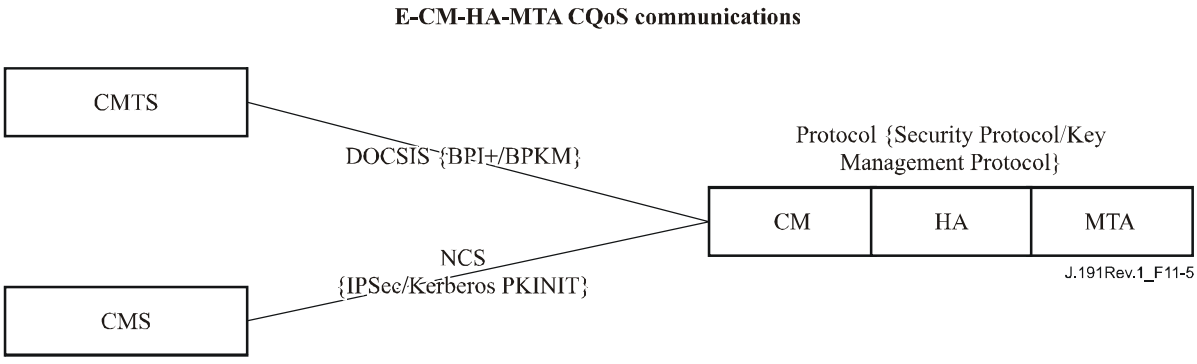

Figure 11-5/J.191 – Secure CQoS architecture to the MTA

11.3.4.4 The role of the CSP in CQoS

The Cable Security Portal (CSP) is the single point of security control within the Portal Service (PS) function in the IPCable2Home Architecture; therefore the CSP provides security in the CQoS Architecture. The CQP acts as a transparent bridge for the DQoS messages it supports; therefore the CSP does not provide any services for CQoS.

11.3.5 Firewall management

While security issues have long been a major concern for networked corporations, the increasing ubiquity of always on Internet connectivity through a Cable Modem (CM) brings security concerns to the home. Because the average subscriber lacks the technical knowledge, understanding of the security issues and the time to keep their home computers in top-notch secure operation, a firewall becomes a necessary first line of defense in protecting the insecure computers in the home.

There are many definitions for firewall including:

> "A firewall is an approach to security; it helps implement a larger security policy that defines the services and access to be permitted" [ICSA].
>
> "A firewall is an agent which screens network traffic in some way, blocking traffic it believes to be inappropriate, dangerous, or both" [RFC 2979].

Hence, a firewall implements a security policy by using some mechanism to block traffic that the security policy stipulates to be undesirable.

Firewall traffic handling requirements include:

- IPCablecom (see Table 11-15) and IPCable2Home protocols defined in this Recommendation MUST not be broken by the firewall. For instance, a firewall should have appropriate application-specific proxy or stateful packet filtering support to open UDP ports that are defined as a result of IPCablecom signalling.

Table 11-15/J.191 – Relevant IPCablecom Specifications for IPCable2Home firewall

Description	Specification
Audio/Video Codecs Specification	J.161
Dynamic Quality of Service Specification	J.163
Network-Based Call Signalling Protocol Specification	J.162
MTA Device Provisioning Specification	J.167
Security Specification	J.170
Management Event Mechanism Specification	J.172
Audio Server Protocol Specification	J.175
Call Management Server Signalling Specification	J.178

IPCablecom-defined protocols include the following:

- Provisioning SNMPv3, DHCP, DNS, TFTP, SYSLOG
- Media Stream RTP, RTCP
- QoS RSVP
- Network Call Signalling MGCP, SDP
- Security Kerberos Messaging, IPSec

IPCable2Home-defined protocols include the following:

- Provisioning SNMPv3, DHCP, DNS, TFTP, SYSLOG
- Management ICMP
- Security Kerberos

The firewall SHOULD protect against port or network scanning launched from inside and outside of the home network. It SHOULD also protect against the following list of denial-of-service attacks: "Ping of Death", "Teardrop", "Bonk", "Nestea", "SYN Flood", "LAND Attack", "IP Spoofing", "Smurf Attack" and "WinNuke".

The firewall MUST be capable of allowing the access of the same popular Internet application protocols as defined in Annex D. For the purpose of this Recommendation, a simple NAT or packet filter is not sufficient. In order to provide a flexible and secure solution, the firewall MUST implement either an Application-Specific Proxy (ASP) or a Stateful Packet Filtering (SPF) firewall.

11.3.5.1 Remote download of firewall rule set

Features in the PS element will be enabled that allow the operator to remotely manage firewall functions. The bulk of this management is accomplished via a configuration file download. The Firewall Configuration File contains the rule set for a particular security policy. Firewall management is achieved by accessing management objects of the Security MIB.

The security policy defines the desired level of security/functionality for a subscriber's firewall. More than one may exist to choose from. The files containing the corresponding rule set for these security policies are maintained on an operator file server. The PS MUST use an RFC 1350 compliant TFTP client to download the firewall rule set configuration file.

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.

The procedure for checking the integrity of the Firewall Configuration File by the PS element follows:

1) The Firewall Config 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 value of the cabhSecFwPolicyFileHash MIB object via a type 28 TLV in the PS Configuration File;
   b) sends an SNMP SET to update the cabhSecFwPolicyFileHash 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 cabhSecFwPolicyFileURL MIB object via a type 28 TLV in the PS Configuration File;
   b) sends an SNMP SET to update the cabhSecFwPolicyFileURL MIB object.
4) If the cabhSecFwPolicyFileOperStatus is not inProgress(1) and the value used to SET the cabhSecFwPolicyFileURL MIB object is different than the value of the cabhSecFwPolicySuccessfulFileURL MIB, then the PS element MUST immediately download the named file from the configured TFTP server.

5) The PS element MUST compute a SHA-1 [FIPS 186-2] hash over the entire contents of the Firewall Configuration File and compare the computed hash to the hash represented by the value of the cabhSecFwPolicyFileHash MIB object. If the computed hash and the value of the cabhSecFwPolicyFileHash MIB object are the same, the integrity of the Firewall Configuration File is verified and the Firewall Configuration File MUST be used, otherwise the file MUST be rejected.

Successful download of the Firewall Configuration File is defined as complete and correct reception of the file by the PS element within the TFTP timeout period and error-free file validation as defined by the integrity check procedure above. After a successful download of the Firewall Configuration File, the PS MUST update the cabhSecFwPolicySuccessfulFileURL MIB with the same value as the cabhSecFwPolicyFileURL MIB.

If the download of the Firewall Configuration File is not successful, the PS MUST NOT update the cabhSecFwPolicySuccessfulFileURL MIB with the same value as the cabhSecFwPolicyFileURL MIB. In any case, the cabhSecFwPolicyFileURL 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 cabhSecFwPolicyFileURL MIB object MUST be populated with its default value.

The Firewall Configuration File policy settings MUST be persistent across reboots of the PS element.

Activation and deactivation of the PS firewall is controlled by the cabhSecFwPolicyEnable MIB object. If the value of cabhSecFwPolicyEnable is enable(1) the PS firewall MUST be activated after, and not before, the cabhPsDevProvState MIB has a value of 'pass'(1) indicating that the provisioning process is complete. This will provide the ability to change the firewall policy via a power cycle of the PS when the WAN management access has been accidentally restricted. The PS firewall MUST NOT be enabled if the value of cabhSecFwPolicyEnable is disable(2).

The cabhSecFwPolicyCurrentVersion MIB MUST always reflect the version of the policy installed on the PS regardless of whether it is currently enabled or disabled in the cabhSecFwPolicyEnable MIB.

11.3.5.2 Firewall rule set management parameters

The following management parameters MUST be implemented in the PS as defined by the Security MIB to support the firewall rule set file:

- **cabhSecFwPolicyFileURL** – Contains the name of the policy rule set file and the IP address of the TFTP server containing the policy rule set file, in a TFTP 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 cabhSecFwPolicySuccessfulFileURL MIB.
- **cabhSecFwPolicySuccessfulFileURL** – 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 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.
- **cabhSecFwPolicyFileHash** – Defines the SHA-1 digest for the corresponding rule set file.
- **cabhSecFwPolicyFileOperStatus** – This object indicates the status of the Firewall Configuration File download and is defined as: InProgress(1) indicates a firewall configuration file download is under way. Complete(2) indicates the firewall configuration file downloaded and processed successfully. Failed(4) indicates that the last attempted firewall configuration file download failed.
- **cabhSecFwPolicyFileCurrentVersion** – The rule set file version currently operating in the PS element. This object should be in the syntax used by the individual vendor to identify rule set file versions. The 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.

- **cabhSecFwPolicyFileEnable** – Allows for activation and deactivation of the firewall security policy.

11.3.5.3 Firewall event log

The firewall MUST be capable of logging the following types of events:

TYPE 1: Attempts from both private and public clients to traverse the Firewall that violate the Security Policy.

TYPE 2: Identified Denial-of-Service attack attempts.

TYPE 3: Changes made to any of the following firewall management parameters:

- cabhSecFwPolicyFileURL;
- cabhSecFwPolicyFileCurrentVersion;
- cabhSecFwPolicyFileEnable.

The choice of which types of firewall events actually get logged is configured through the Security MIB interface as described in 11.3.5.4.

The firewall MUST log events associated with the download via TFTP of the firewall policy file as appropriate. Refer to Annex B, Table B.1 (CSP Process, Firewall TFTP sub-process).

Operators can monitor firewall events using the event messaging mechanism defined in 6.5. Event logging management parameters are accessed via the Security MIB and are defined in 6.5.

The firewall event message log allows an operator to assess the level of hacker activity across the operator network and monitor changes to the firewall's security policy. When event message types have been enabled via the Security MIB management parameters, these firewall events MUST be logged with an event message entry using the event logging mechanism defined in 6.5.

A firewall event message entry will contain the following information:

- Event Priority;
- Date and Time – when the event occurred;
- Protocol – indicated by the IP header field (TCP, UDP, ICMP);
- Source IP Address;
- Destination IP Address;
- Destination Port (TCP and UDP) or Message Type (ICMP);
- Relevant Policy Rule;
- Event description (optional).

Clause 6.5.2.1 defines an Event Priority field that describes different levels of priority for logged events. If the field is not applicable, it must be left blank. The PS element MUST format firewall event messages as defined in Annex B.

To assist in monitoring hacker activity on a subscriber's firewall hacker alert management objects have been defined in the Security MIB. This feature alerts the operator when the number of TYPE 1 and 2 firewall events exceeds an alert threshold for a given alert period (in hours). The alert threshold and alert period are configurable by the operator. The PS element accumulates the number of TYPE 1 and 2 firewall events that have occurred over the past number hours defined by the alert period. If this number exceeds the alert threshold, a hacker alert event message is logged to inform the operator.

11.3.5.4 Management parameters for event logging

The following management parameters MUST be implemented in the PS as defined by the Security MIB to monitor/configure firewall event logging:

- **cabhSecFwEventType1Enable** – Enables or disables logging of type 1 firewall event messages. Default = disable (2).
- **cabhSecFwEventType2Enable** – Enables or disables logging of type 2 firewall event messages. Default = disable (2).
- **cabhSecFwEventType3Enable** – Enables or disables logging of type 3 firewall event messages. Default = disable (2).
- **cabhSecFwEventAttackAlertThreshold** – If the number of type 1 or 2 hacker attacks exceeds this threshold in the period defined by the cabhSecFwEventAttackAlertPeriod object, a firewall message event MUST be logged. The default is set to the highest allowed integer value. This MIB MUST be ignored if the cabhSecFwEventAttackAlertPeriod is set to 0 and an event message MUST NOT be sent. Default = 65535.
- **cabhSecFwEventAttackAlertPeriod** – Indicates the period to be used in past hours for the cabhSecFwEventAttackAlertThreshold object. Default = 0.

11.3.6 MIBs

The stand-alone PS MUST support the following software download support MIBs defined in RFC 2669:

- **docsDevSwAdminStatus** – If set to upgradeFromMgt(1), the device will initiate a TFTP software image download using docsDevSwFilename.
- **docsDevSwFilename** – The file name of the software image to be loaded into the device.
- **docsDevSwCurrentVers** – The software version currently operating in the device.
- **docsDevSwServer** – The address of the TFTP server used for software upgrades.
- **docsDevSwOperStatus** – Status of software download.

The stand-alone PS MUST support the following software download support MIBs defined in [draft-ietf-ipcdn-bpiplus-mib-12]:

- **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.

- **docsBpi2CmPublicKey** – A DER-encoded RSAPublicKey ASN.1 type string, as defined in the RSA Encryption Standard [RFC 2437].
- **docsBpi2CmDeviceCmCert** – The X.509 DER-encoded device certificate.
- **docsBpi2CmDeviceManufCert** – The X.509 DER-encoded manufacturer CA ertificate that signed the device certificate.

The stand-alone PS MUST support the following configuration download support MIB:

- **cabhPsDevProvConfigHash** – HA-1 [FIPS 186-2] hash of the contents of the configuration file, taken as a byte string. See 7.3.3.

11.3.7 Secure software download

A stand-alone PS Element MUST be capable of remotely downloading a software image over the network. As described in 6.3.7, secure software download to an Embedded PS is controlled by the cable modem. The new software image would allow the operator to improve performance, accommodate new functions and features, correct design deficiencies, and to allow a migration path of IPCable2Home devices as this Recommendation 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 stand-alone 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 MAY 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 originator of the software download.
- 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 an operator with mechanisms to upgrade or downgrade the code version of the IPCable2Home elements.
- The software download process MUST provide options for an 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 Recommendation 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 (operator or Certification body) MAY countersign the code file in addition to the manufacturer's signature.
- The PS element MUST be able to process a PKCS#7 digital signature and an IPCable2Home X.509 certificate as defined in 11.3.7.2.1.1 and 11.3.7.3, 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.

11.3.7.1 Software download into embedded or stand-alone PS elements

As shown in Figure 11-6 below, a complete Home Access (HA) device may implement the cable modem and the PS Element as separate entities or embedded as defined in 5.1.3.1.

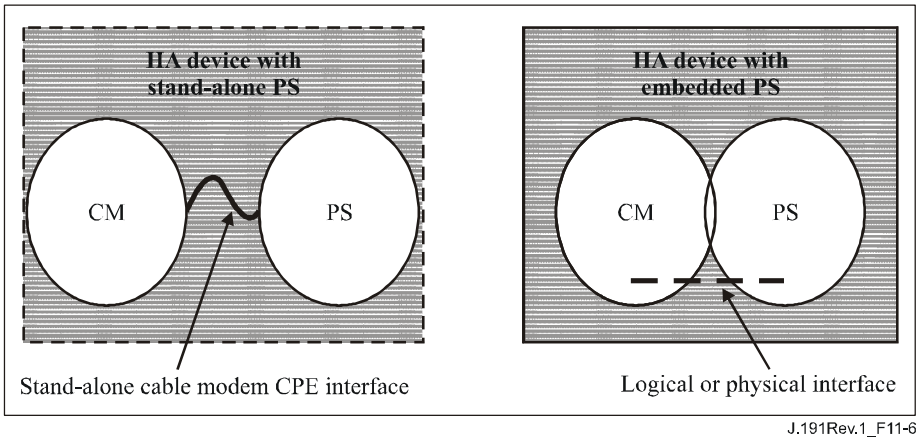

Figure 11-6/J.191 – HA device

For IPCable2Home:

- 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 stand-alone entities, then the software download for the IPCable2Home elements MUST be performed by the PS Element as described below.

11.3.7.2 Code file requirements

11.3.7.2.1 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 11-16.

When certificates are downloaded as a part of the Code File, the certificates MAY be contained in the fields as specified in Table 11-16, and separated from the actual code image contained in the CodeImage field.

Table 11-16/J.191 – Code file structure

Code file	Description
PKCS#7 Digital Signature {	
ContentInfo	
ContentType	SignedData
SignedData ()	EXPLICIT signed-data content value: includes CVS and X.509 compliant CVCs.
} *end PKCS#7 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*	

11.3.7.2.1.1 Signed data

The code download file will contain the information in a [RFC 2315] Signed Data content type as shown below in Table 11-17. 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 11-17/J.191 – 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 Certificate (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	

Table 11-17/J.191 – PKCS#7 signed data

PKCS#7 field	Description
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*	

11.3.7.2.1.2 Signed content

The signed content field of the code file contains the code image and the download parameters field, which possibly contains additional optional items Service Provider Root CA Certificate, Certification Testing Laboratory (CTL) CVC Root CA Certificate, CTL CVC CA Certificate, and/or the Manufacturer CA Certificate.

The final code image is in a format compatible with the destination PS element. In support of the PKCS#7 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, then the PS element MUST replace its currently stored certificate with the new certificate received in the signed content field. This new certificate will then be used for subsequent verification.

11.3.7.2.1.3 Code signing keys

The [RFC 2315] digital signature uses the RSA Encryption Algorithm [RFC 2437] with SHA-1 [FIPS 186-2]. The PS element MUST be able to verify code file signatures. The public exponent is F_4 (65537 decimal).

11.3.7.2.1.4 Manufacturer CA-Certificate

This Attribute is a string attribute containing an X.509 CA Certificate, as defined in ITU-T Rec. X.509 | ISO/IEC 9594-8.

Type	Length	Value
17	Variable	X.509 CA Certificate (DER-encoded ASN.1)

11.3.7.2.1.5 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 Rec. X.509 | ISO/IEC 9594-8. 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)

11.3.7.2.1.6 CVC Root CA-Certificate

This Attribute is a string attribute containing an X.509 CVC Root CA Certificate, as defined in ITU-T Rec. X.509 | ISO/IEC 9594-8. This certificate must be used by the stand-alone PS Element in the secure software downloading process.

Type	Length	Value
51	Variable	X.509 CA Certificate (DER-encoded ASN.1)

11.3.7.2.1.7 CVC CA-Certificate

This Attribute is a string attribute containing an X.509 CVC CA Certificate, as defined in ITU-T Rec. X.509 | ISO/IEC 9594-8. This certificate must be used by the stand-alone PS Element in the secure software downloading process.

Type	Length	Value
52	Variable	X.509 CA Certificate (DER-encoded ASN.1)

11.3.7.3 Code Verification Certificate (CVC) format

11.3.7.3.1 CVC format for secure software download

For secure software download, the format used for the CVC is 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 11-18 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 11-18. The DER encoding MUST meet the requirements of 11.3.2.

Table 11-18/J.191 – X.509 compliant code verification certificate

X.509 certificate	Description
Certificate {	
version	2 (i.e., ITU-T Rec. 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

Table 11-18/J.191 – X.509 compliant code verification certificate

X.509 certificate	Description
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*	

11.3.7.3.2 Certificate revocation

This Recommendation does not require or define the use of certificate revocation lists (CRLs). The PS element is not required to support CRLs. Operators may want to define and use CRLs outside of the HFC network 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.

11.3.7.4 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.

11.3.7.4.1 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. These include:

- 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 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.

11.3.7.4.2 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. These values are:

- 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 Recommendation.

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.

11.3.7.5 Code upgrade initialization

11.3.7.5.1 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:
  a) codeAccessStart initialization value;
  b) 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 stand-alone 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.

11.3.7.5.2 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; but 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 Recommendation. 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, then 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 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 IPCable2Home 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:

1) co-signing agent's organizationName;

2) co-signer's time-varying control values:
   a) cvcAccessStart;
   b) 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.

11.3.7.6 CVC processing

To expedite the delivery of an updated CVC without requiring the HA to process a code upgrade, the CVC MAY be delivered in either the configuration file or an SNMP MIB. The format of the CVC is the same whether it is in a code file, configuration file, or SNMP MIB.

11.3.7.6.1 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 element configuration file does not include a CVC that validates properly, the PS element MUST NOT download upgrade code files whether triggered by the PS element configuration file or via an SNMP MIB. In addition, if the PS element configuration files does not include a CVC that validates properly, the PS element is not required to process CVC's subsequently delivered via an SNMP MIB, and MUST NOT accept information from a CVC subsequently delivered via an SNMP MIB.

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 11.3.2.2.2.
2) Check the CVC subject organization name.
   a) If the CVC is a Manufacturer's CVC (Type 32) then:
      i) 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.
      ii) 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:
      i) 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.
      ii) 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 CTL CVC 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.

11.3.7.6.2 Processing the SNMP CVC

The PS element MUST process SNMP delivered CVCs when enabled to upgrade code files; otherwise, all CVCs delivered via SNMP MUST be rejected. When validating the CVC delivered via SNMP, 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, 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 11.3.2.2.2.

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.

11.3.7.7 Code signing requirements

11.3.7.7.1 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 11.3.7.3. The CTL CVC CA MUST not sign any CVC unless it is identical to the format specified in 11.3.7.3. Before signing a CVC, the CTL CVC CA MUST verify that the certificate request is authentic.

The CTL CVC CA will be responsible for registering names of authorized CVC subscribers. CVC Subscribers include PS Element manufacturers and 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 organization that issued the root certificate.
- 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 non-zero integer, with an integer value of 0 representing the absence of a code-signer.

11.3.7.7.1.1 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 operator for remote upgrade of a device MUST be signed according to the requirements defined in this Recommendation. When signing a code file, a manufacturer MAY choose not to update the PKCS#7 signingTime value in the manufacturer's signing information. This Recommendation requires that the PKCS#7 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 an operator to use the code file to either 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.

11.3.7.7.1.2 Operator requirements

When an operator receives software upgrade code files from a manufacturer, the operator should validate the code image using the CTL CVC CA Public Key. This will allow the operator to verify that the code image is as built by the trusted manufacturer. The operator can re-verify the code file at any time by repeating the process.

If an operator wants to exercise the option of co-signing the code image destined for a device on their network, the operator MUST obtain a valid CVC from the CTL CVC CA.

When signing a code file, the operator MUST co-sign the file content according to the PKCS#7 signature standard, and include their operator CVC as defined in 11.3.7.2.1.1. This Recommendation does not require an operator to co-sign code files; but when the operator follows all the rules defined in this Recommendation for preparing a code file, the PS element MUST accept it.

11.3.7.8 Triggering process

Code downloads, regardless of the provisioning mode, may 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.

11.3.7.8.1 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 overwritten 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 where 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 IPCable2Home 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 last known working image and 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 and after the device is operational, it MUST adhere to the following requirements:

- set its docsDevSwAdminStatus to ignoreProvisioningUpgrade{3};
- set its docsDevSwOperStatus to completeFromMgt{3};
- reboot.

The PS element MUST properly use ignoreProvisioningUpgrade status to ignore software upgrade value that may be included in the PS element configuration file and become operational with the correct software image and after the device is operational, 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 IPCable2Home device the:

- DocsDevSwAdminStatus MUST be allowProvisioingUpgrade{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 retry has not been reached, the PS element MUST fall back to the last known working image and proceed to an operational state, generate appropriate event notification as specified in 11.3.7.10, 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 retry 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.

11.3.7.8.2 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 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 last known working image and 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 IPCable2Home device;
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the IPCable2Home device;
- docsDevSwOperStatus MUST be completeFromProvisioning{2};
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the device.

11.3.7.9 Code verification

For secure software download, the PS element MUST perform the verification checks presented in this clause. 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 verification checks can be made in any order, as long as all of the applicable checks presented in this clause are made.

1) The PS element MUST validate the manufacturer's signature information by verifying that the PKCS#7 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 11.3.2.2.2.
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 does not 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:
      i) Co-signer's signature information is included in the code file.
      ii) PKCS#7 signingTime value is equal-to or greater-than the corresponding codeAccessStart value currently held in the PS element.

iii) PKCS#7 signingTime value is equal-to or greater-than the corresponding CVC validity start time.

iv) PKCS#7 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:

i) CVC subject organizationName is identical to the co-signer's organization name currently stored in the PS element's memory.

ii) CVC validity start time is equal-to or greater-than the cvcAccessStart value currently held in the PS element for the corresponding subject organizationName.

iii) Extended key usage extension is in the CVC as defined in 11.3.2.2.2.

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 does not 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 PKCS#7 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 PKCS#7 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 PKCS#7 signingTime value.

b) Update the current value cvcAccessStart with the CVC validity start value.

9) When the code installation is successful, 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 PKCS#7 signingTime value.

b) Update the current value of cvcAccessStart with the CVC validity start value.

11.3.7.10 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 PKCS#7 signingTime value is less-than the codeAccessStart value currently held in the PS element.

d) The manufacturer's PKCS#7 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 PKCS#7 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 PKCS#7 signingTime value is less-than the codeAccessStart value currently held in the PS element.

i) The co-signer's PKCS#7 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 PKCS#7 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.

11.3.7.11 Software Downgrade

The Software Downgrade defines the process of removing the upgraded version of the software image download, thus reverting the 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 operator, Certification body) may choose not to update the signing-time. In this manner, multiple code files with the same code signing-time allow 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 operator, but these advantages should 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.

11.3.8 Physical security

This Recommendation requires the PS to maintain, in its memory, keys and other cryptovariables related to network security. All elements and devices MUST deter unauthorized physical access to this cryptographic material.

The level of physical protection of keying material that is required for network elements and devices is specified in terms of the security levels defined in the FIPS PUBS 140-2, Security Requirements for Cryptographic Modules. In particular, IPCable2Home elements 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.

11.3.9 Cryptographic algorithms

11.3.9.1 SHA-1

The IPCable2Home implementation of SHA-1 MUST use the SHA-1 hash algorithm as defined in FIPS 180-2.

12 Management processes

12.1 Introduction/Overview

This clause provides examples of processes associated with the use of the tools described in clause 6 (Management Tools) and additional processes that facilitate other required management functions defined in this Recommendation. PS Database access and other PS operations of the Cable Management Portal (CMP) are described in clause 6. Typical MIB access rules are provided in 6.3.6.

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 clause is primarily composed of informative text, intended to aid in reader 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 clause 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 clause 6.

12.2.1 CTP operation

The Cable Test Portal (CTP) provides Connection Speed Tool and Ping Tool capabilities, described in 6.4.3.1 and 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.

- The Network Management System (NMS) starts the test by initializing the test parameters and setting the Begin Test flag, via SNMP SET Request.
- The CMP SNMP Agent updates the PS Database with the test parameters and notifies the CTP to begin the test.
- The CTP queries the PS database for the test parameters.
- 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.
- The target LAN IP Device simply echoes the UDP packet payload back to the CTP.
- 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.
- The NMS verifies that the command is complete by checking Status = complete.
- The NMS requests the test results via SNMP GET Request.
- 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.

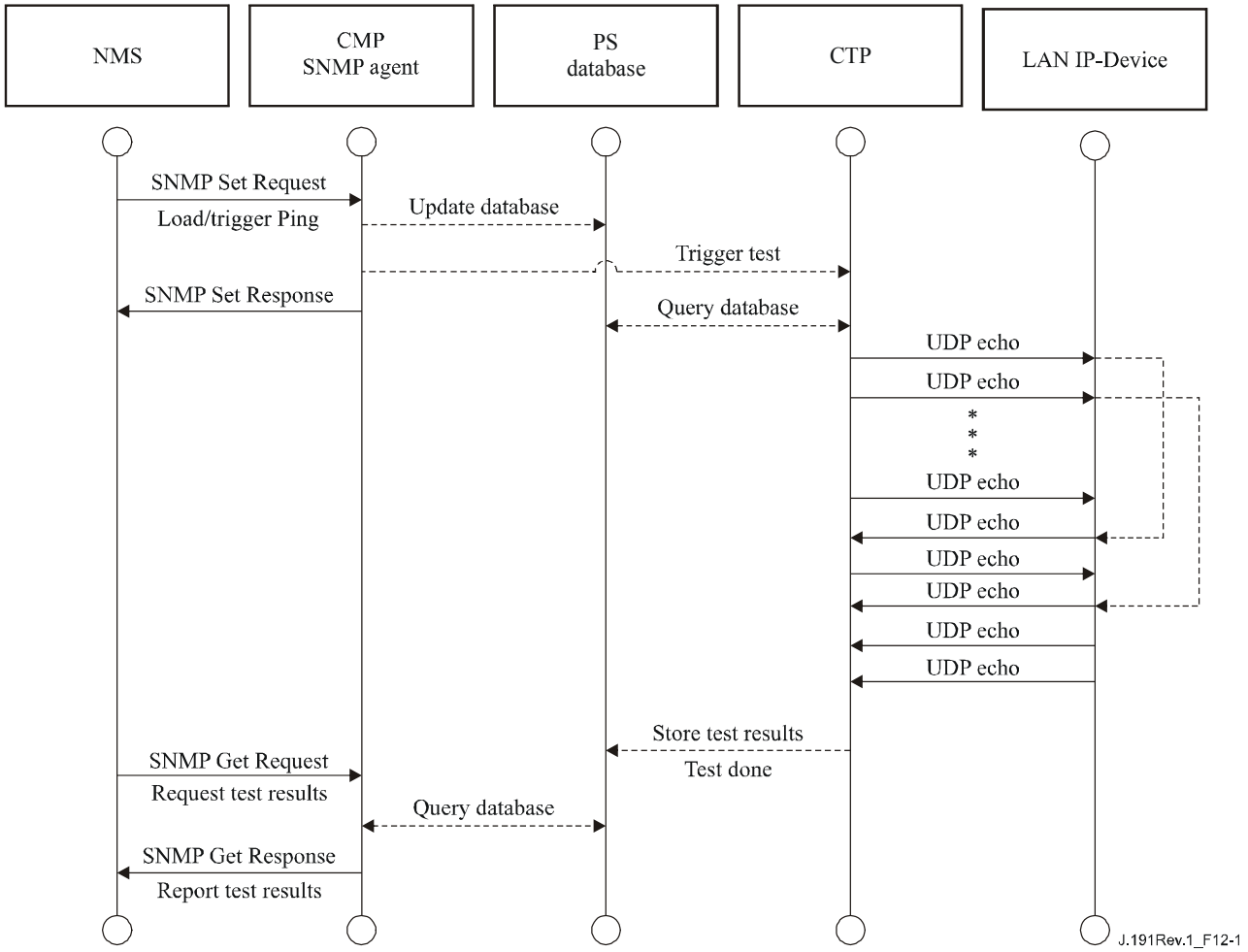

Figure 12-1/J.191 – Connection speed tool process sequence diagram

12.2.1.2 Ping Tool process

The Ping Tool can be useful in validating connectivity state, performance levels, and identifying possible configuration errors.

- The NMS starts the test by initializing the test parameters and setting the Begin Test flag, via SNMP SET Request.
- The CMP SNMP Agent updates the PS Database with the test parameters and notifies the CTP to begin the test.
- The CTP queries the PS database for the test parameters.
- The CTP issues an ICMP Echo Request packet to the specified LAN IP Device.
- The target LAN IP Device responds with an ICMP Echo Response.
- The CTP updates the PS Database with the results of the test and sets the Test Complete flag.
- The NMS verifies that the command is complete by checking Status = complete.
- The NMS requests the test results via SNMP GET Request.
- 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.

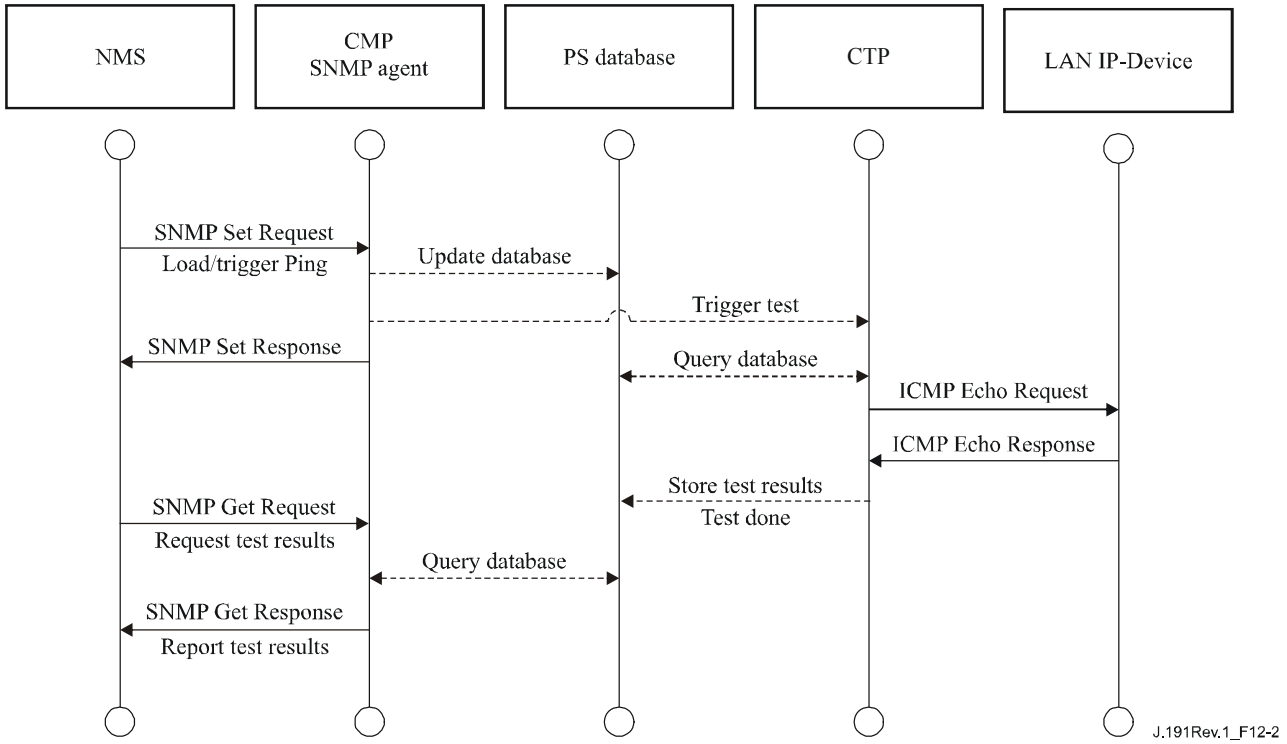

Figure 12-2/J.191 – Ping Tool process sequence diagram

12.3 PS operation

The Cable Management Portal (CMP) provides access to the PS Database via the PS WAN-Man interface, as described in clause 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 message sequences assume a secure SNMPv3 link has been established.

- 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.
- The CMP SNMP Agent queries the PS Database for the specified parameters.
- The CMP SNMP reports the data to the NMS with the SNMP GET Response.

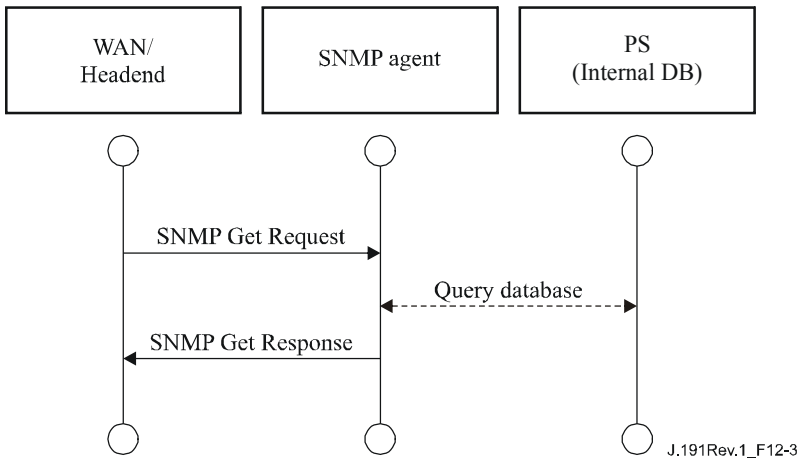

Figure 12-3/J.191 – PS database access from the PS WAN-Man interface sequence diagram

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. This process is triggered by the NMS. The PS is told 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 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 to the NMS what happened.

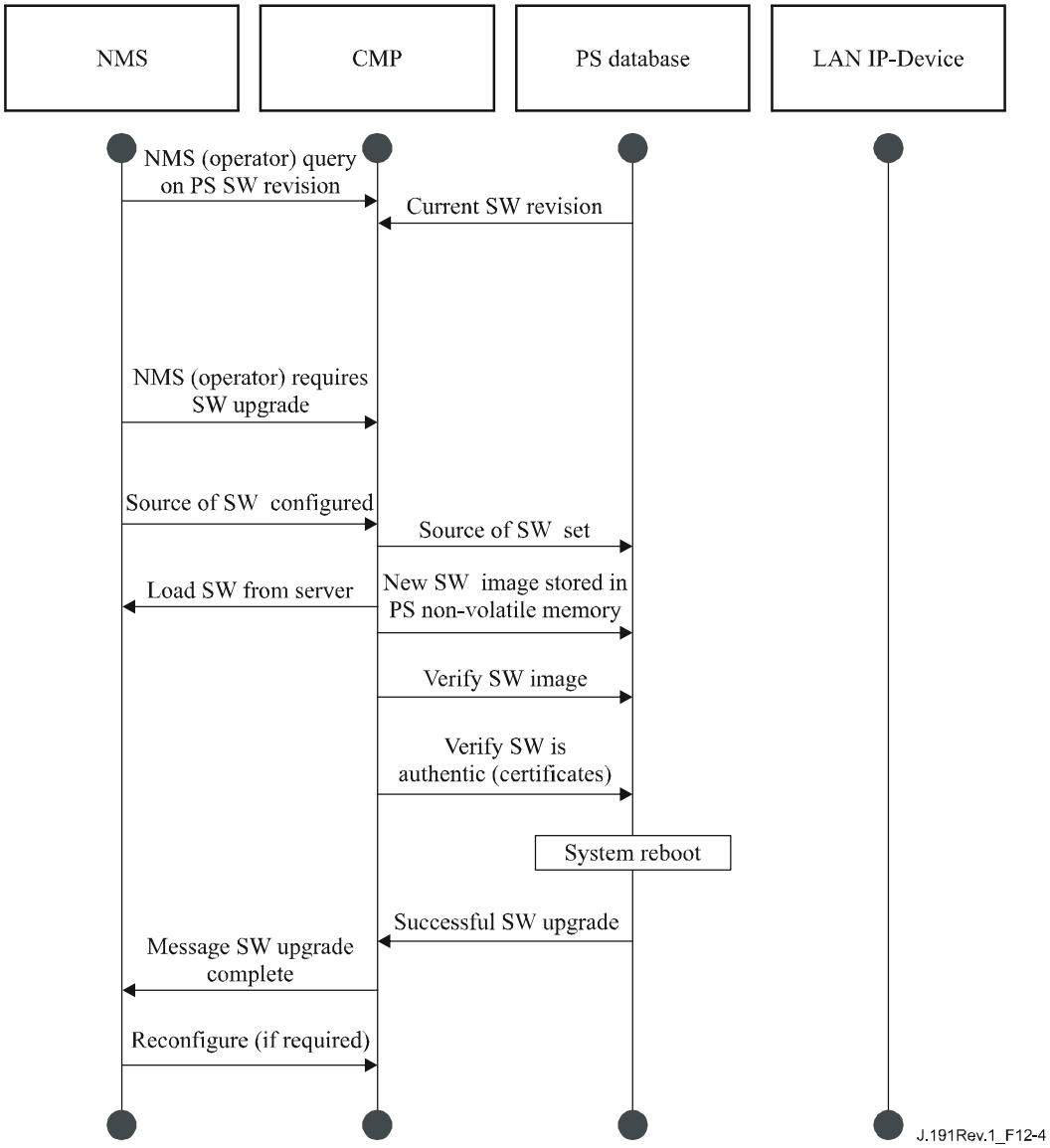

Figure 12-4/J.191 – PS software download sequence diagram

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. This process 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.

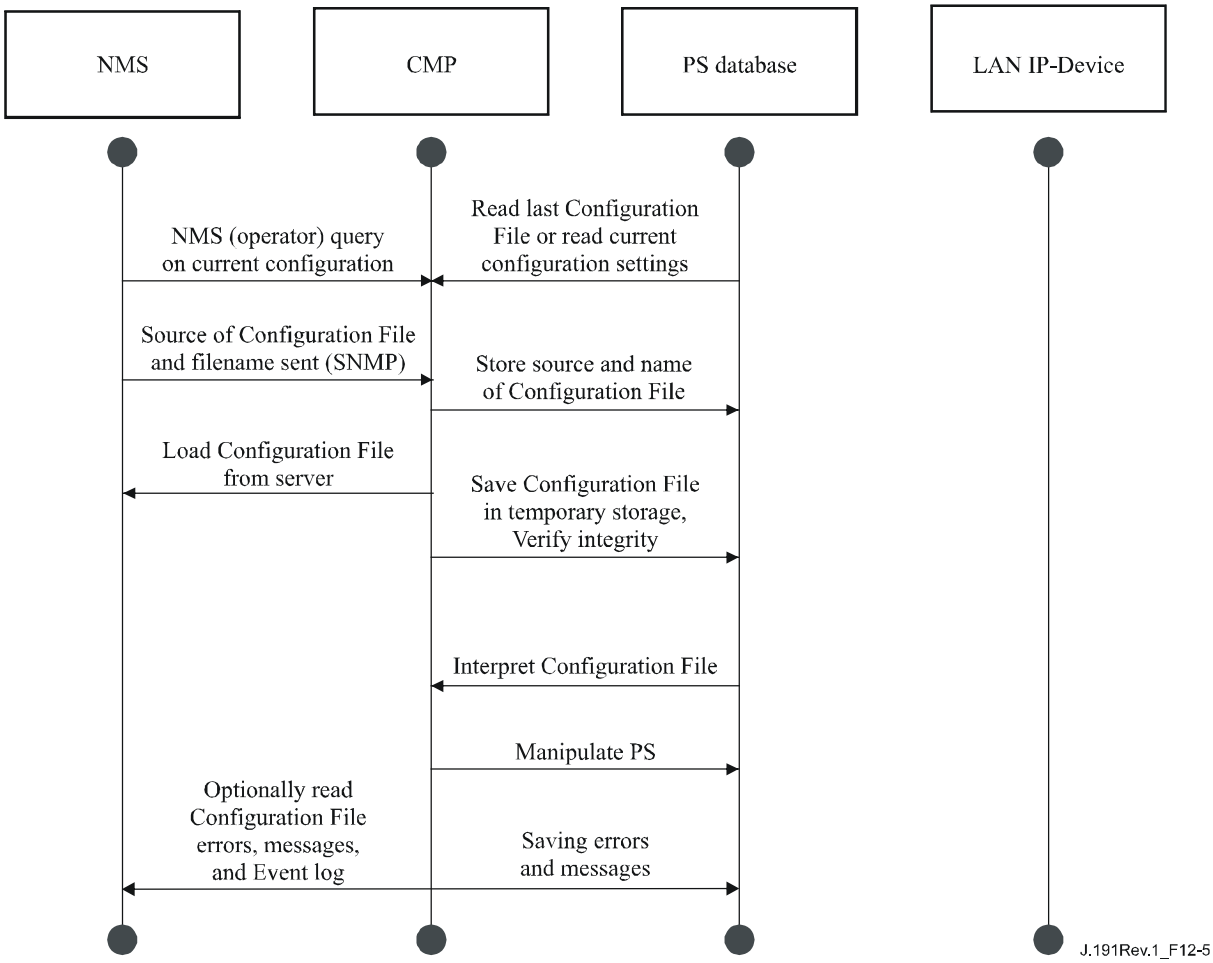

Figure 12-5/J.191 – PS reconfiguration (Configuration File Download) sequence diagram

12.4 MIB access

12.4.1 VACM configuration

The cable operator has control of the management domain. An example of the configuration of VACM parameters is shown in Figure 12-6.

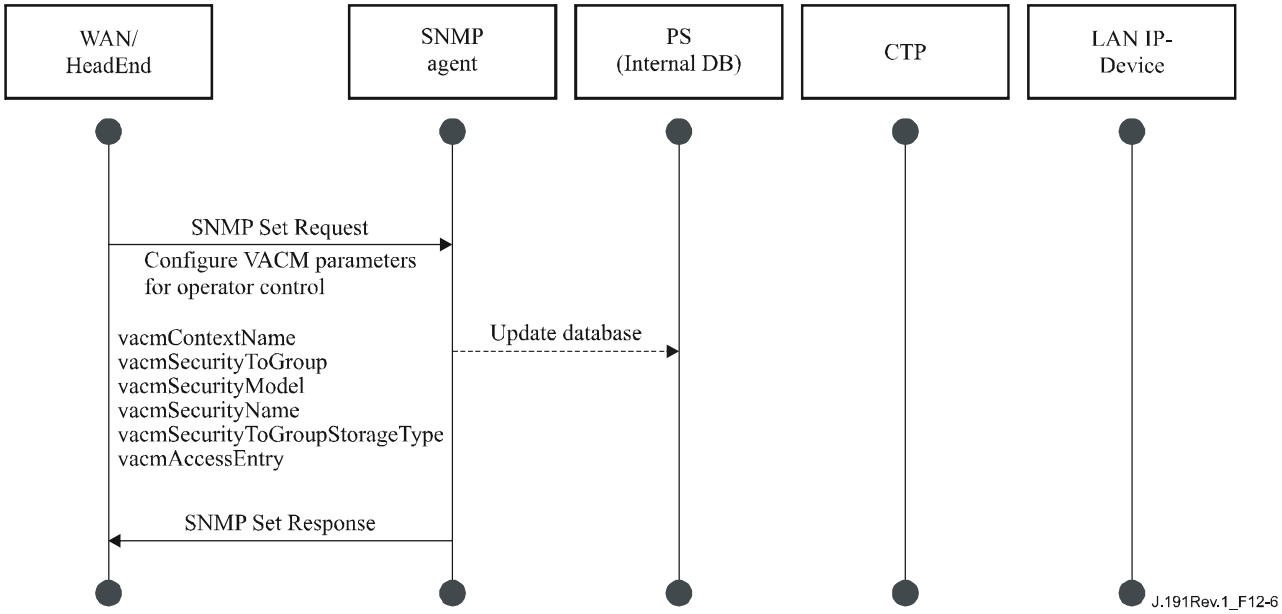

Figure 12-6/J.191 – PS configuration (VACM Parameters) sequence

12.4.2 Management event messaging configuration

12.4.2.1 CMP event notification operation

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. Non-volatile events persist across reboots.

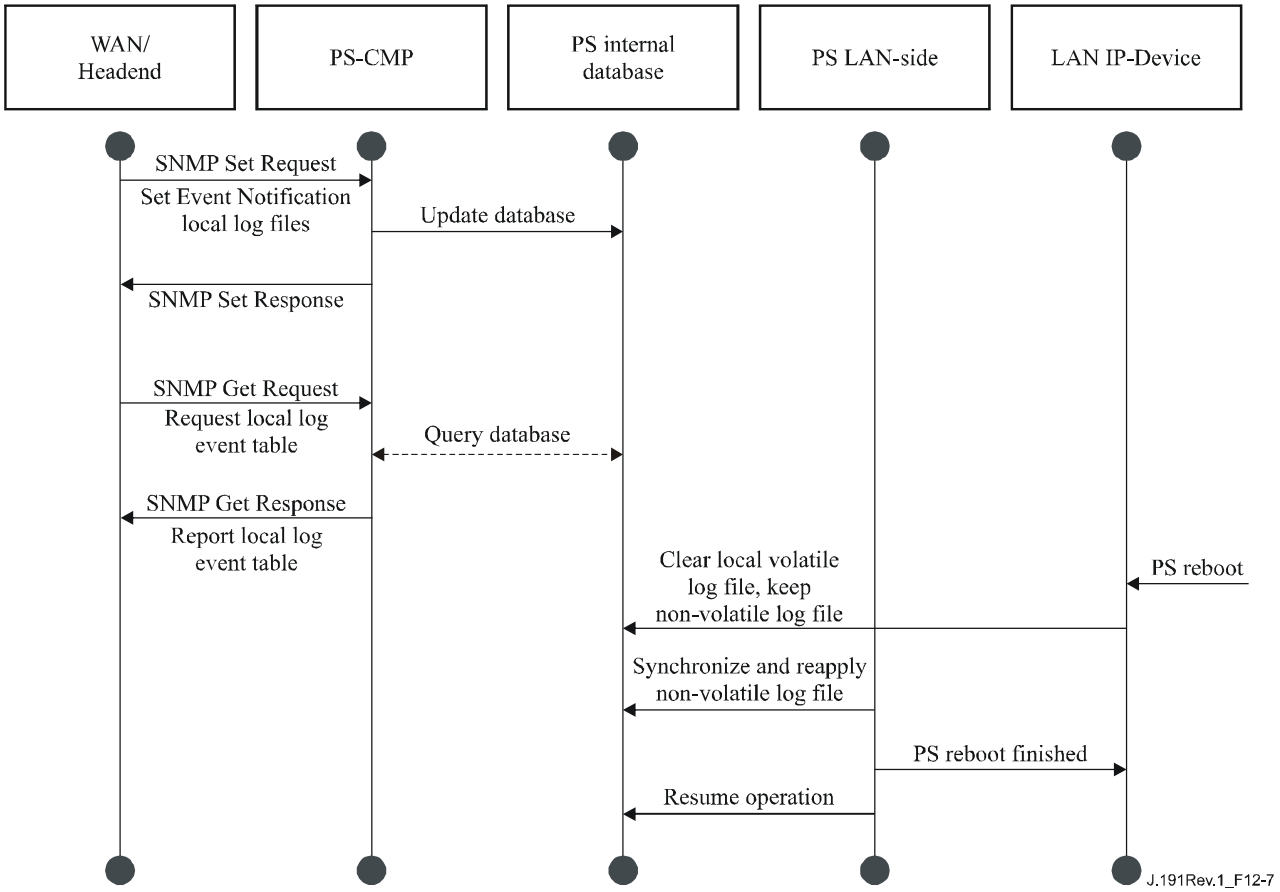

Figure 12-7/J.191 – PS Configuration (Event Control) sequence

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.

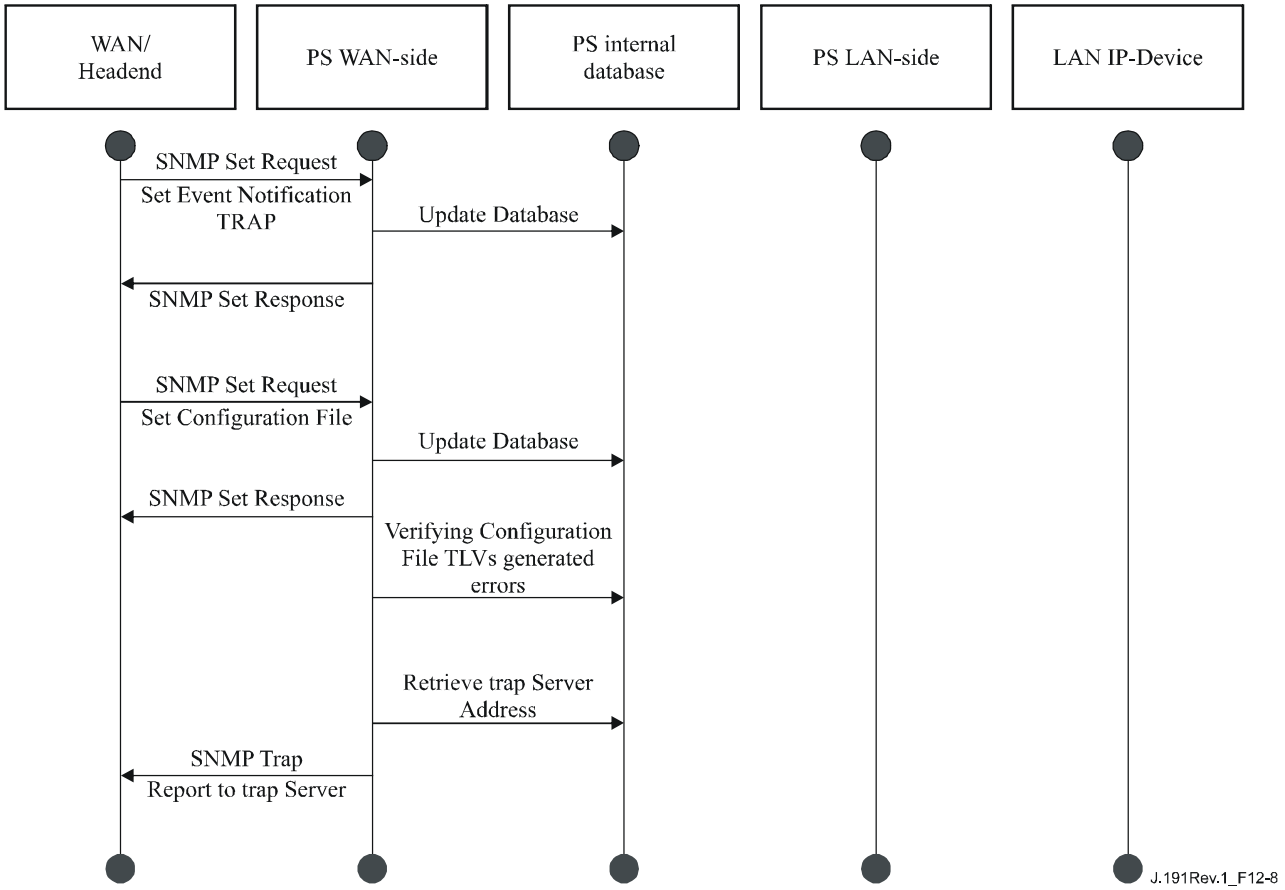

Figure 12-8/J.191 – PS Configuration File Download (with Invalid TLVs) sequence

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 it generates an event to SYSLOG.

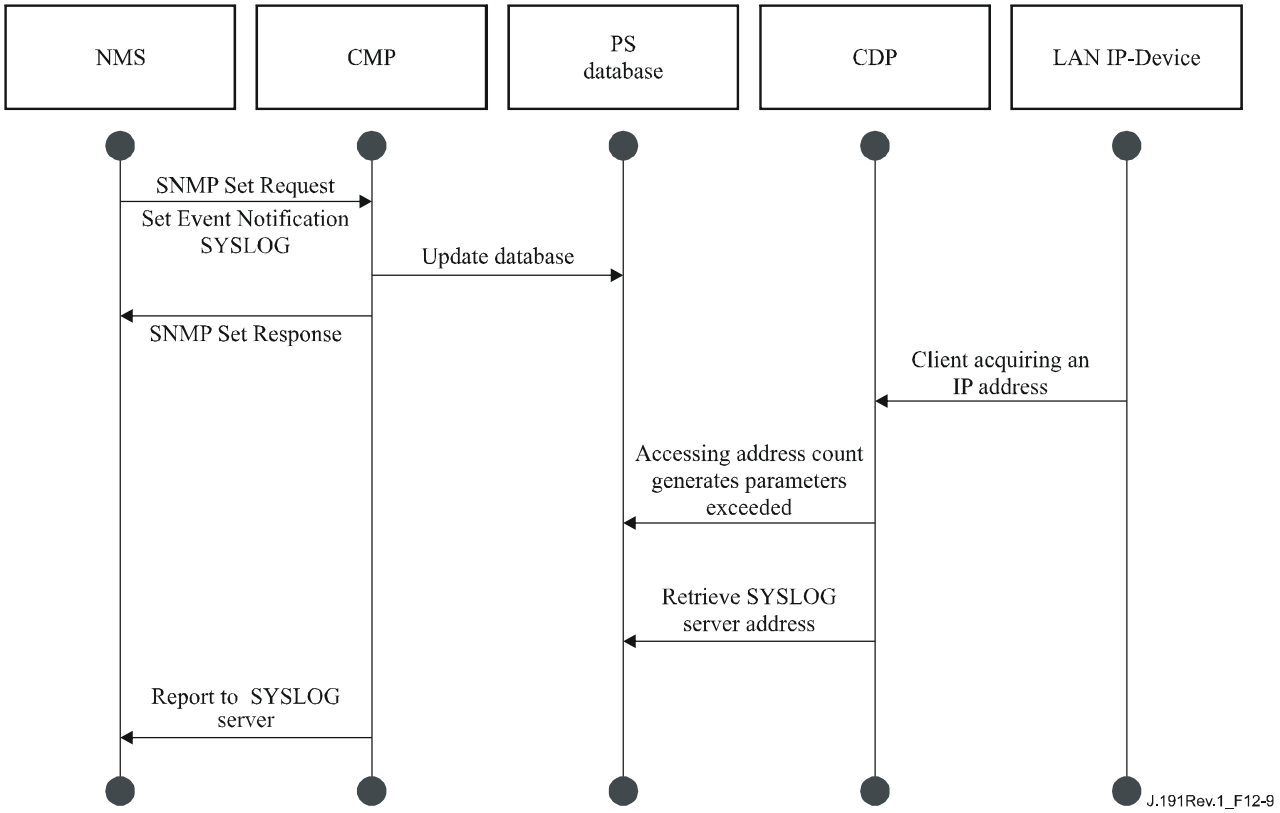

Figure 12-9/J.191 – LAN IP Device address acquisition (request exceeds provisioned count) sequence

12.4.2.2 Example CMP event throttling and limiting operation

An event throttling mechanism is provided 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 6.5.3 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.

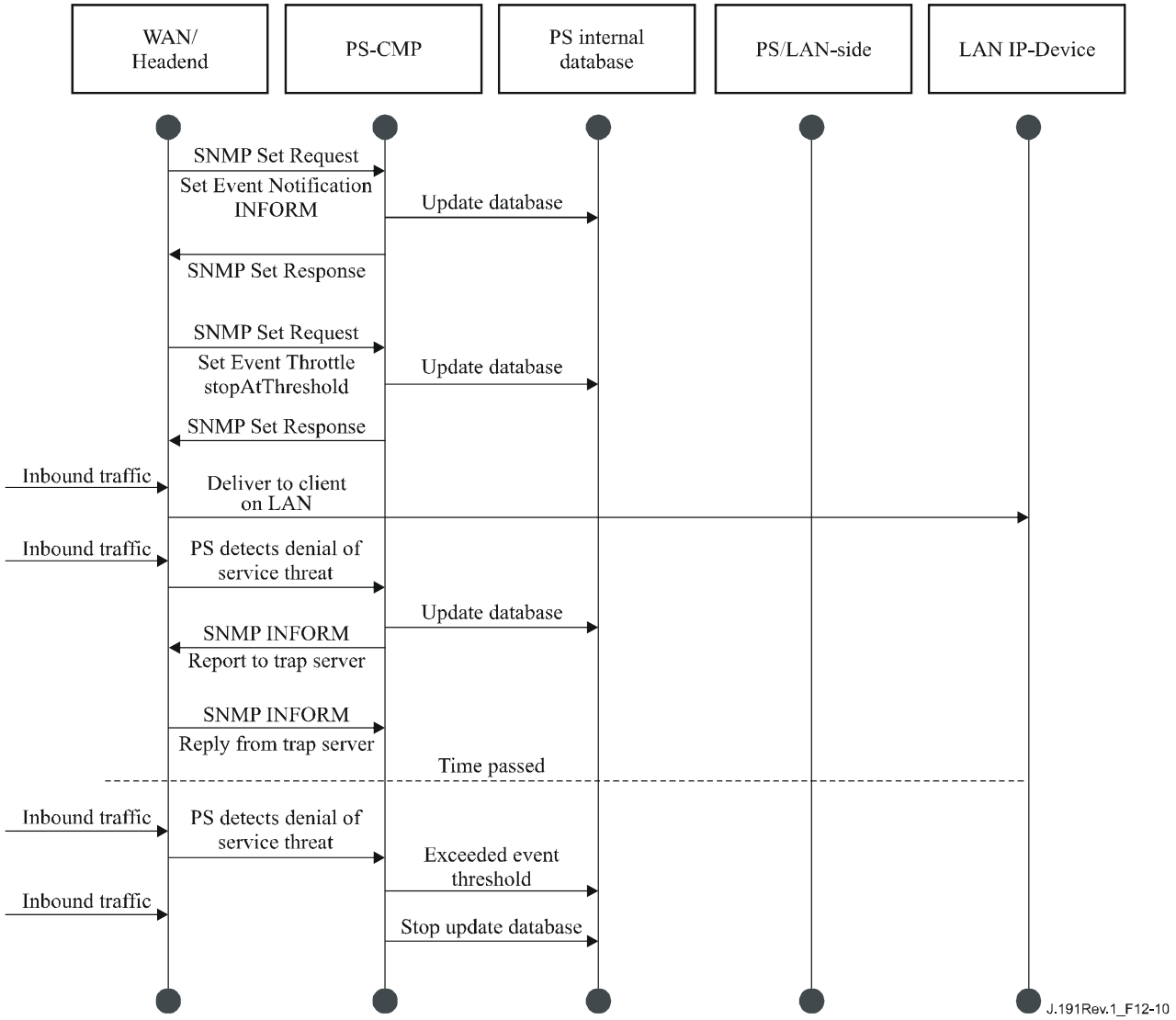

Figure 12-10/J.191 – CMP event throttling and limiting operation

13 Provisioning processes

This clause describes the processes involved when using the Provisioning Tools, described in clause 7, for initial provisioning of LAN IP Device and the PS element. This 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 clause 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 PS provisioning, and is out of scope for this Recommendation. The reader is referred to cable modem Recommendations 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 Centre (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.

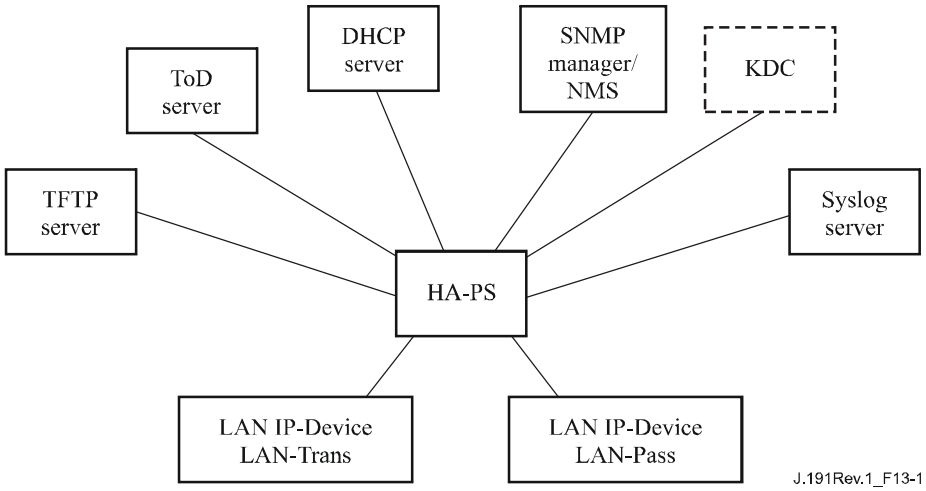

Figure 13-1/J.191 – Provisioning functional elements

The Trivial File Transfer Protocol (TFTP) 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) Simple Network Management Protocol (SNMP) Manager complies with RFC 1157 and possibly with more current versions of the SNMP, e.g., [RFC 2576], [RFC 3412], [RFC 3414], and [RFC 3415]. The Key Distribution Centre (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 Headend servers, and uses these client functions during the provisioning processes described in this clause to accomplish the tasks listed at the beginning of this clause.

13.1 Provisioning modes

Clauses 5.5 and 7.1.1 introduce two provisioning modes supported by the Portal Services element: DHCP Provisioning Mode and SNMP Provisioning Mode. In this clause, each of the two modes is presented in more detail. Figure 13-2 illustrates a possible event flow for the two provisioning modes. The key point of Figure 13-2 is the switch used by the PS to determine the provisioning 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 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 suboption 51 to the PS CDC, during the DHCPOFFER phase of the initialization process. DHCP Provisioning Mode is intended to enable the PS to operate on a J.112 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 suboption 51. SNMP Provisioning Mode is intended to enable the PS to take advantage of advanced features of a IPCablecom infrastructure.

The PS defaults to Dormant CableHome Mode if it receives none of the fields or suboptions defined as triggers for DHCP Provisioning Mode and for SNMP Provisioning Mode, or if it receives an invalid combination of the fields and suboptions.

Not all error conditions are shown in Figure 13-2. Refer to 7.2.3.3 for a description of PS behaviour in the event of incorrect Provisioning Mode decision criteria.

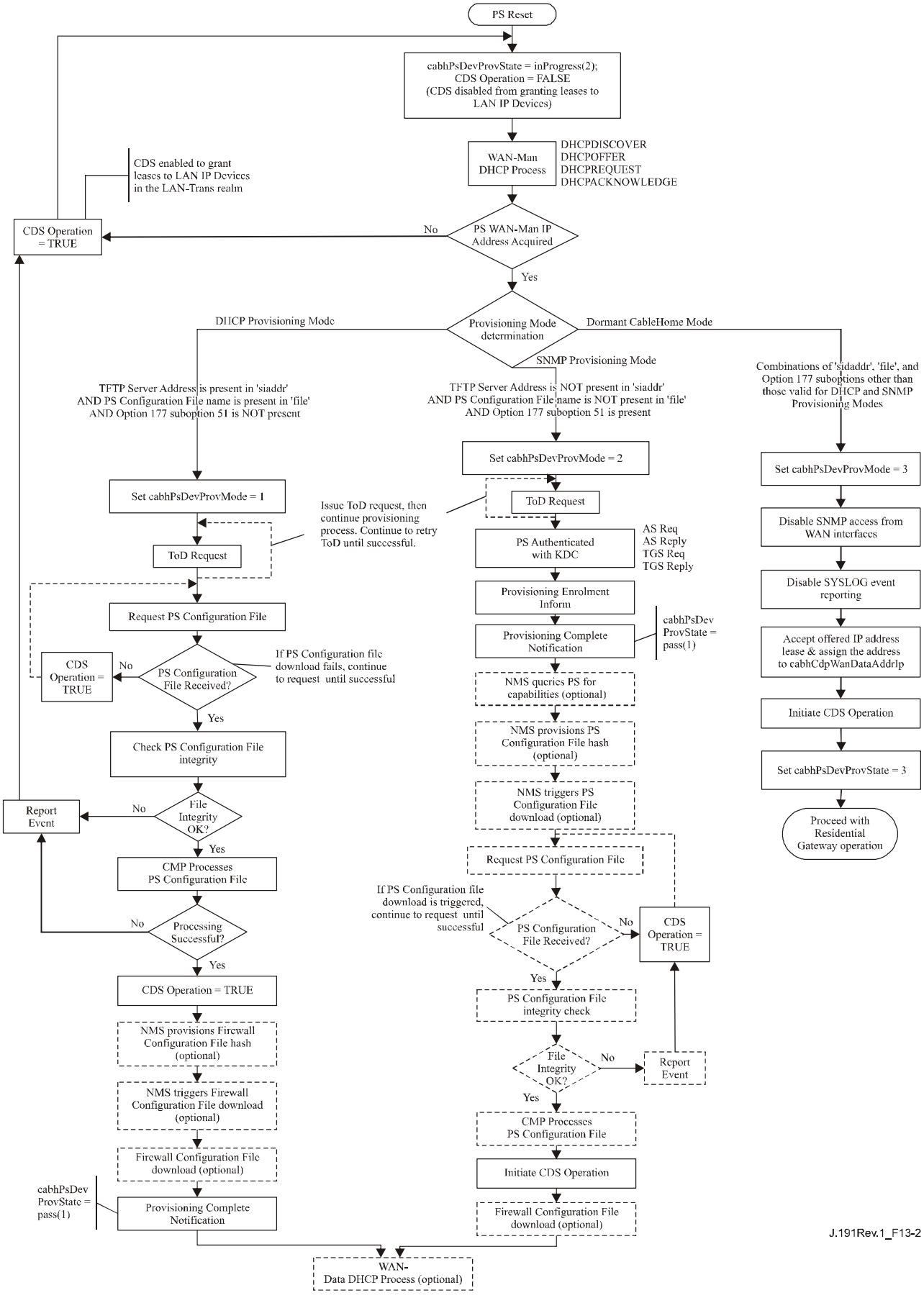

Figure 13-2/J.191 – Provisioning modes

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 (see 7.2.3.3). Clause 7.2.2.2.2 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 7.2. 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 6.3.3.

Figure 13-3 and Table 13-1 describe the sequence of messages needed to initialize a PS operating in DHCP Provisioning Mode. The process for provisioning a PS operating in DHCP Provisioning Mode is the same for the PS embedded with a 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.

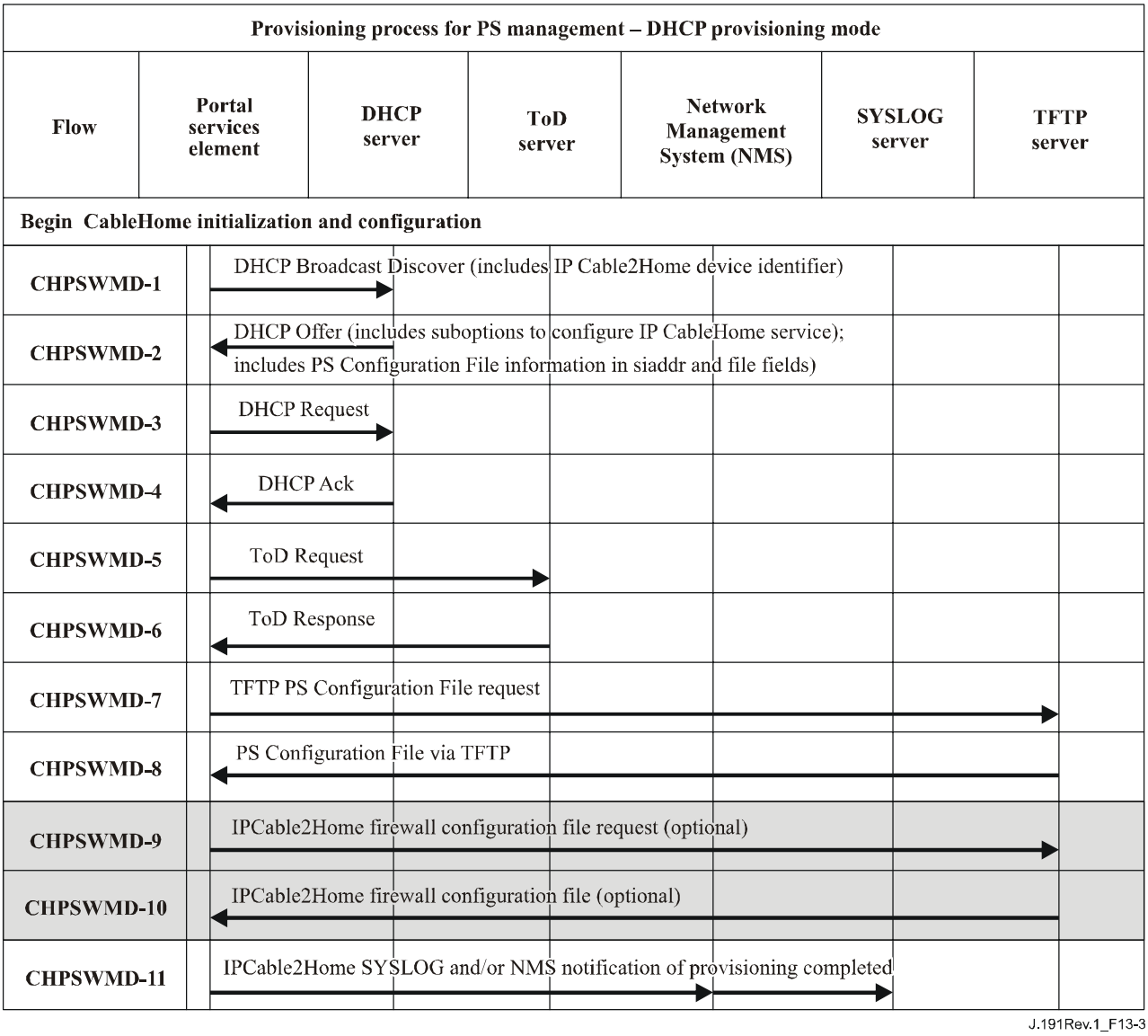

Figure 13-3/J.191 – Provisioning process for PS management – DHCP provisioning mode

Table 13-1 describes the individual messages CHPSWMD-1 – CHPSWMD-11 shown in Figure 13-3.

Table 13-1/J.191 – 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 7.2.3.3. The DHCP DISCOVER broadcast by the CDP (CDC) includes mandatory options listed in Table 7-7. The PS sets cabhPsDevProvState to status 'inProgress' (2) when the CDC sends a broadcast DHCP DISCOVER. The PS MUST start the Provisioning Timer using the starting value accessible via cabhPsDevProvTimer AND set 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 7.2.3.3.
CHPSWMD-2	*DHCP OFFER* The DHCP OFFER issued by the DHCP server in the cable network is expected to include no CableHome option code 177 with suboptions 3, 6, and 51 AND is expected to include PS configuration file information in the siaddr and file fields of the DHCP message. (See 7.2.3.3.)	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 (see 7.2.3.3). The PS stores the Time of Day server address in cabhPsDevTimeServerAddr.	CHPSWMD-4 MUST occur after CHPSWMD-3 completion.	If failure per DHCP protocol return to CHPSWMD-1 and report an error. If the expected configuration file information is not received in the DHCP ACK after 5 attempts, the PS operates in "Dormant CableHome mode" as described in 5.5 and 7.2.3.3.

Table 13-1/J.191 – 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-5	*Time of Day (ToD) Request per RFC 868* The PS issues a ToD Request to the ToD server identified in the DHCP OFFER.	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 7.3.3.	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 7.3.3.3. The PS Configuration File is then processed. Refer to 7.3.3 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.	CHPSWMD-8 MUST occur after CHPSWMD-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.

Table 13-1/J.191 – 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-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 11.3.5.1). 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 CHPSWMD-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 11.3.5.	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.	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.

Table 13-1/J.191 – 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
	If the PS successfully completes all required provisioning steps from CHPSWMD-1 through CHPSWMD-10 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-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 if the PS received valid parameters for 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'.		

13.3 Process for 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 7.3.3.3 that it is to operate in SNMP Provisioning Mode, the PS will secure its management messages using SNMPv3, following the authentication procedure described in 11.3.3.

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 6.5.

Figure 13-4 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 stand-alone 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 13-4 and described in detail in Table 13-2. Optional steps are shown with a shaded background in Figure 13-4. These optional steps may be done immediately following step CHPSWMS-13, at a later time, or not at all.

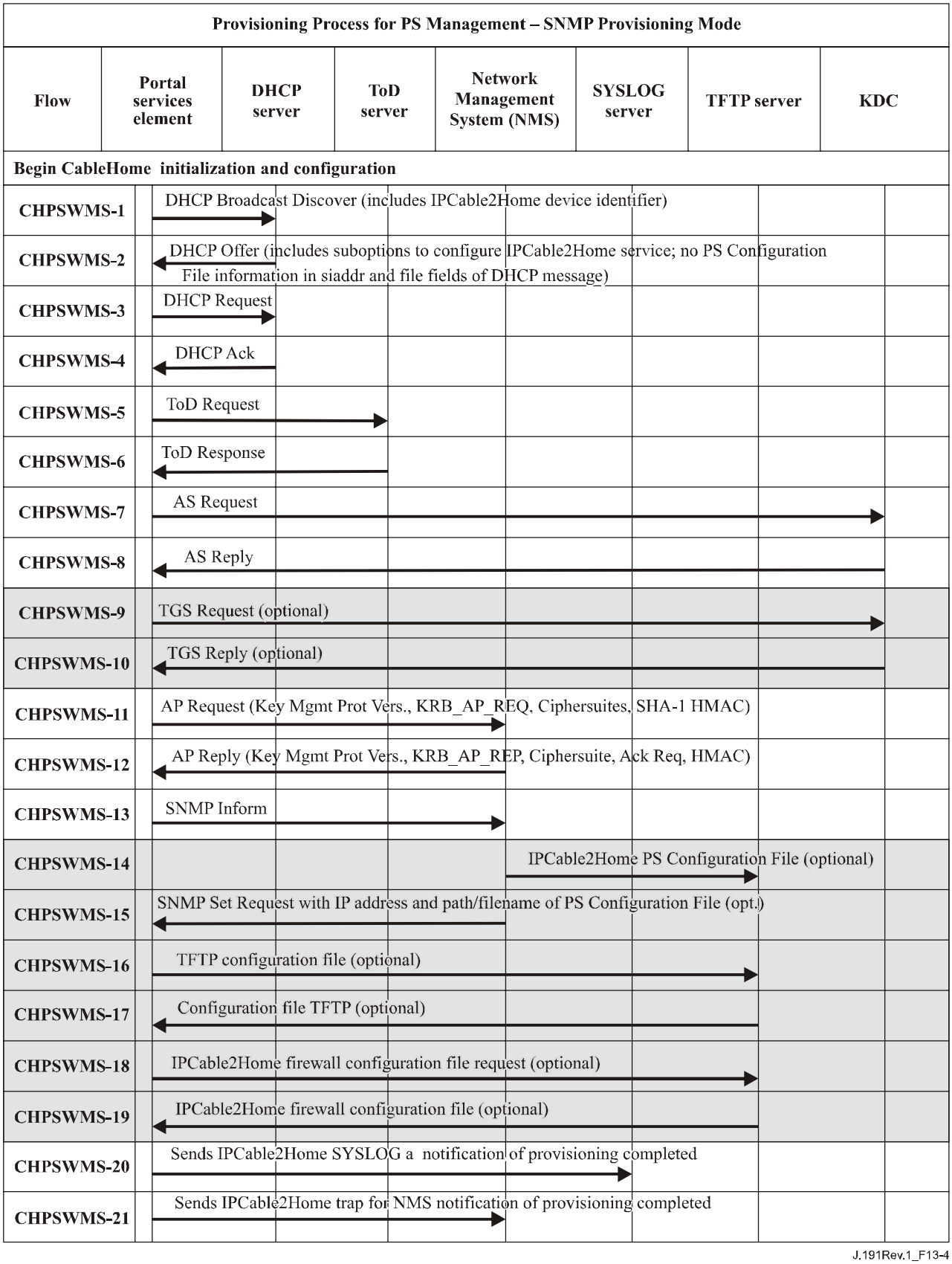

Figure 13-4/J.191 – Provisioning process for PS management – SNMP provisioning mode

Table 13-2 describes the individual steps of the provisioning process depicted in Figure 13-4.

Table 13-2/J.191 – 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 DHCP DISCOVER message to acquire the WAN-Man IP address as described in 7.2.3. The DHCP DISCOVER broadcast by the CDP (CDC) includes mandatory options listed in Table 7-7. 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 Headend DHCP server fails, initiate operation of the CDS as specified in 7.2.3.3.
CHPSWMS-2	*DHCP OFFER* The DHCP OFFER issued by the DHCP server in the cable network is expected to include the option code 177 with suboptions 3, 6, & 51 AND no PS configuration file information in the siaddr and file fields of the DHCP message.	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 CDC 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 CableHome option code 177 with suboptions 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 (see 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.

Table 13-2/J.191 – 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-5	*Time of Day (ToD) Request per [RFC 868]* The PS issues a ToD Request to the ToD server identified in the DHCP ACK.	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.	Continue with CHPSWMS-7, report an error, and return to CHPSWMS-5 (continue to retry ToD until successful).
CHPSWMS-7	*AS Request (Note)* The PS sends the AS Request message to the operator IPCable2Home KDC 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 operator 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	*(Optional) TGS Request* If the PS obtained a Ticket Granting Ticket (TGT) during step CHPSWMS-8, the PS sends the TGS Request message to the operator KDC server whose address was passed to the PS (CDC) in DHCP Option 177 suboption 51.	CHPSWMS-9 MUST occur after CHPSWMS-8 completion.	Return to CHPSWMS-1. PS initiates operation of CDS.
CHPSWMS-10	*(Optional) TGS Reply* The TGS Reply message containing the ticket is received from the operator 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 11.3.3.2.	CHPSWMS-11 MUST occur after CHPSWMS-10 completion.	Return to CHPSWMS-1. PS initiates operation of CDS.

Table 13-2/J.191 – 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-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 (see 11.3 for additional detail.)	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 enrolment 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	*(Optional) Configuration File Create* The provisioning system uses information from previous PS provisioning steps 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-14 occurs, CHPSWMS-14 MUST occur after CHPSWMS-13 completes.	N/A
CHPSWMS-15	*(Optional) 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 7.3.3.2 (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 be loaded, and this method is selected to specify it.	If CHPSWMS-15 occurs, CHPSWMS-15 MUST occur after CHPSWMS-14 completes.	Return to CHPSWMS-1 if the set was received, but there was a processing error.

Table 13-2/J.191 – 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-16	*(Optional) TFTP Request* If the NMS triggers the PS to download a PS Configuration File as described in 7.3.3.2, the PS sends the TFTP Server a TFTP Get Request to request the specified PS Configuration File.	If CHPSWMS-16 occurs, CHPSWMS-16 MUST occur after CHPSWMS-15 completes.	Continue with CHPSWMS-17.
CHPSWMS-17	*(Optional) 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-15. The PS then processes the PS Configuration File. Refer to 7.3.3 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.	If CHPSWMS-17 occurs, CHPSWMS-17 MUST occur after CHPSWMS-16 completes.	If the TFTP download fails, report an error, proceed to CHPSWMS-18, and continue to retry CHPSWMS-16 (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-18	*(Optional) TFTP Request – Firewall Configuration File* The PS sends the Firewall Configuration TFTP Server a TFTP Get Request to request the specified firewall configuration data file.	If CHPSWMS-18 occurs, it MUST occur after CHPSWMS-17 completes.	Continue with CHPSWMS-19.
CHPSWMS-19	*(Optional) 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-15 or CHPSWMS-17. The file is then processed. Refer to 11.3 for additional detail.	If CHPSWMS-19 occurs, CHPSWMS-19 MUST occur after CHPSWMS-18 completes.	If the TFTP download fails, continue with PS operation but report an error and continue to retry CHPSWMS-18. If processing of the firewall configuration file produces an error, continue and report the error as an event.

Table 13-2/J.191 – 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-20	*SYSLOG notification* If the PS received a SYSLOG server address in the DHCP ACK, the PS MUST send the SYSLOG server a "provisioning complete" notification. The general format of this notification is as defined in 6.5.1.	CHPSWMS-20 MUST occur after CHPSWMS-19 completion.	N/A
CHPSWMS-21	*SNMP Trap* The PS MUST send the NMS an SNMP TRAP (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-13 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-21 MUST occur after CHPSWMS-20 completion.	N/A
NOTE – Steps CHPSWMS-5-CHPSWMS-8 are optional in some cases. Refer to clause 11 for details.			

13.3.1 PS WAN-Man configuration file download

The PS operating in SNMP Provisioning Mode MAY 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 PS Configuration File MAY be downloaded 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.

13.3.2 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.

13.3.3 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.

13.3.4 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.

13.3.5 Provisioning state and error reporting

As indicated in Tables 13-1 and 13-2, failure of the steps in the provisioning process generally results in the process restarting at the first step, CHPSWMD-1 or CHPSWMS-1.

13.4 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 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 Figure 13-5 and described in detail in Table 13-3.

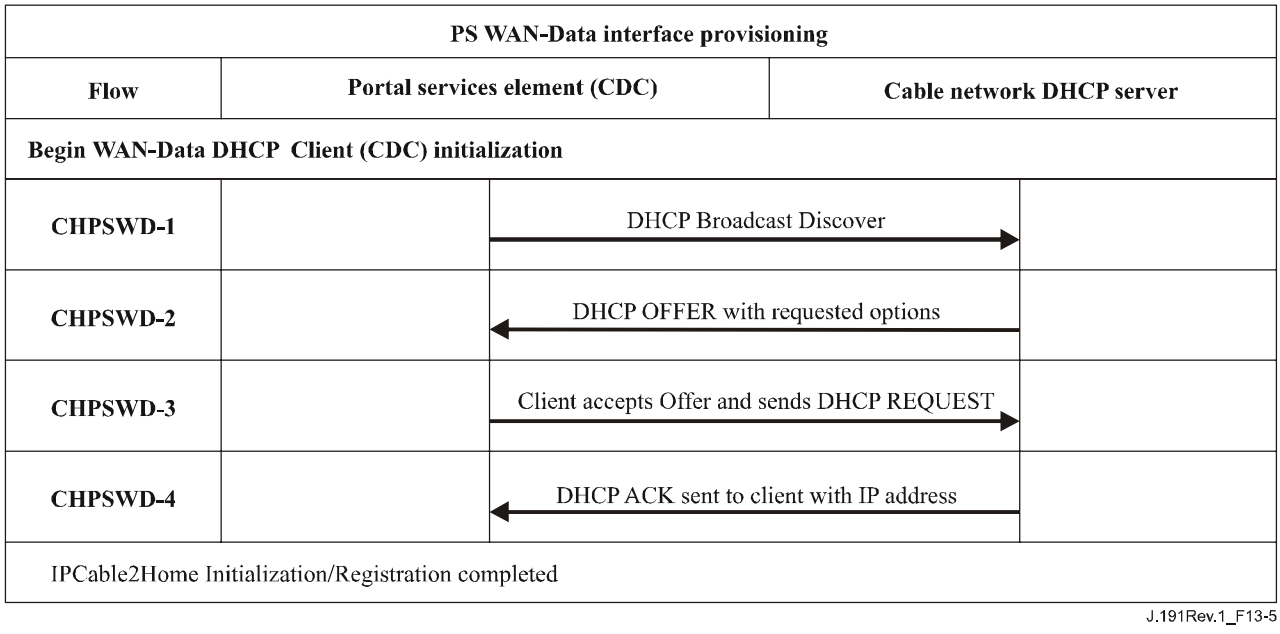

Figure 13-5/J.191 – PS WAN-Data provisioning process

Table 13-3/J.191 – 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 DHCP DISCOVER message including the mandatory options listed in Table 7-7.	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 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.	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.

13.5 Provisioning process: DHCP client in the LAN-Trans realm

LAN IP Devices request IP addresses via DHCP processes. The PS element handles these messages according to the provisioning parameters assigned by the cable network NMS (see 7.2.3.2).

This clause 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 clause 8). There is no difference in LAN-Trans realm IP Device provisioning process between the DHCP and SNMP Provisioning Modes.

Provisioning process message flows for a LAN IP Device in the LAN-Trans address realm are described in Figure 13-6. Additional detail about the process is provided in Table 13-4.

The provisioning process for the LAN IP Device in the LAN-Trans realm MUST occur via the sequence depicted in Figure 13-6 and described in detail in Table 13-4.

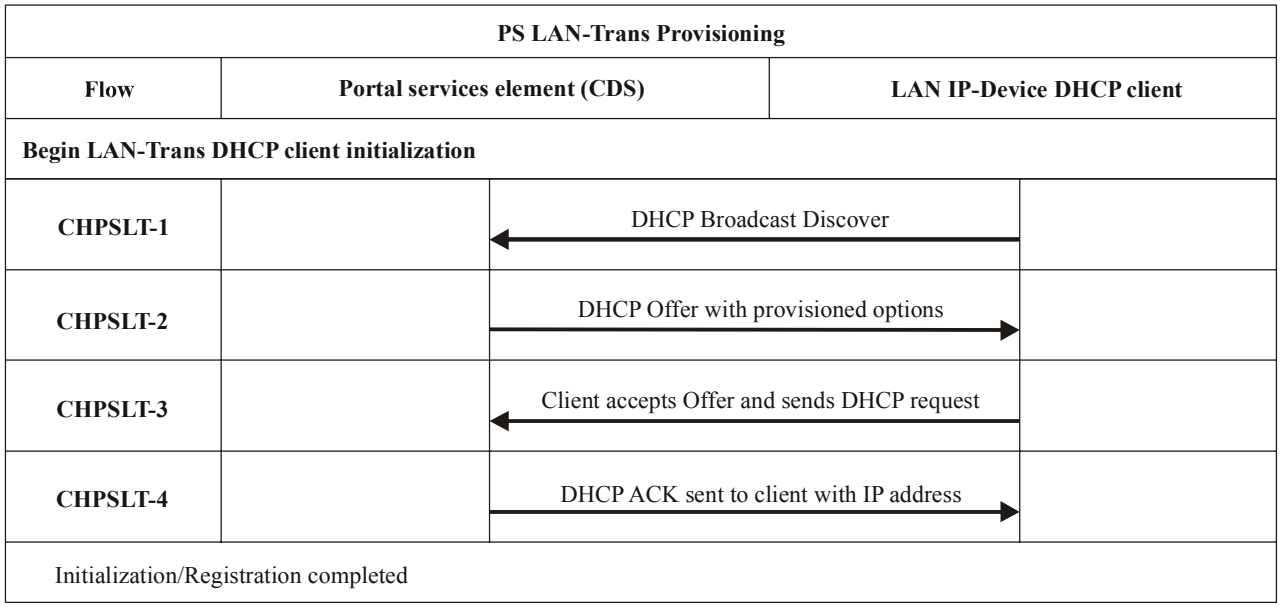

Figure 13-6/J.191 – Provisioning process for LAN IP Device in LAN-Trans realm

Table 13-4/J.191 – Flow descriptions for PS LAN-Trans provisioning process

Flow step	Client LAN-Trans address provisioning	Normal sequence	Failure sequence
CHPSLT-1	*DHCP Broadcast Discover* The Client (Note 1) sends a broadcast DHCP DISCOVER message on its local LAN (Note 2).	Proceed to CHPSLT-2.	If failure per DHCP protocol repeat CHPSLT-1.
CHPSLT-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).	Proceed to CHPSLT-3.	If failure, the client will time out per DHCP protocol and CHPSLT-1 will be repeated.
CHPSLT-3	*DHCP REQUEST* The LAN IP Device's DHCP client receives the DHCP OFFER 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 (Note 3).	Proceed to CHPSLT-4.	If failure, the client will time out per DHCP protocol and CHPSLT-1 will be repeated.

Table 13-4/J.191 – Flow descriptions for PS LAN-Trans provisioning process

Flow step	Client LAN-Trans address provisioning	Normal sequence	Failure sequence
CHPSLT-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).	Provisioning Complete.	If failure, the client will time out per DHCP protocol and CHPSLT-1 will be repeated.
NOTE 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. NOTE 2 – If the client is located on a non-broadcast network, it is expected to unicast the message to the DHCP Server. NOTE 3 – If the client is located on a non-broadcast network, it is expected that it will unicast the message to the PS.			

13.5.1 LAN-Trans address selection and DHCP Options

The PS MUST select the Lan-Trans address that it offers from the range indicated by MIB variables cabhCdpLanPoolStart and cabhCdpLanPoolEnd.

The PS CDS MUST include in the DHCP OFFER the mandatory options listed in Table 7-3.

13.6 Provisioning process: DHCP client 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 Pass-through (transparent bridging) mode. As described in 8.2.2.2, bridging occurs when the cable network NMS sets the Primary Packet-handling mode (cabhCapPrimaryMode) to Pass-through, or by writing individual LAN IP Device MAC addresses into the Pass-through Table (cabhCapPass-throughTable). Figure 13-7 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.

The provisioning process for the LAN IP Device in the LAN-Pass realm MUST occur via the sequence depicted in Figure 13-7 and described in detail in Table 13-5.

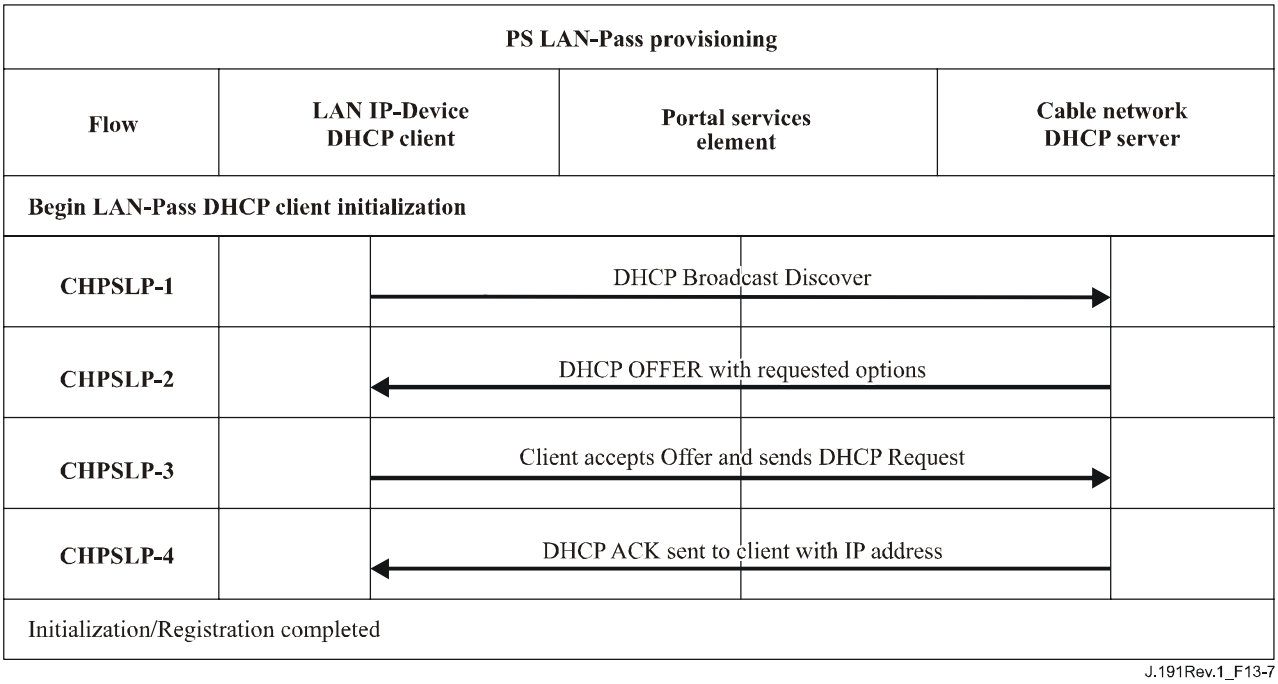

Figure 13-7/J.191 – Provisioning process for LAN IP device in the LAN-Pass realm

Table 13-5/J.191 – Flow descriptions for LAN-Pass provisioning process

Flow step	Client pass thru address provisioning	Normal sequence	Failure sequence
CHPSLP-1	*DHCP Broadcast Discover* The LAN IP Device broadcasts a DHCP DISCOVER message on its local LAN (Note). The PS receives the broadcast DHCP DISCOVER packet on its LAN interface and MUST transparently bridge the packet to the WAN interface without changing the content of the packet.	Proceed to CHPSLP-2.	If failure per DHCP protocol, repeat CHPSLP-1.
CHPSLP-2	*DHCP OFFER* The DHCP Server at the Headend 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 MUST transparently bridge the DHCP OFFER from its WAN interface to its LAN interface without changing the content of the IP packet.	Proceed to CHPSLP-3.	If failure, the LAN IP Device will time out per DHCP protocol and CHPSLP-1 will be repeated.

Table 13-5/J.191 – Flow descriptions for LAN-Pass provisioning process

Flow step	Client pass thru address provisioning	Normal sequence	Failure sequence
CHPSLP-3	*DHCP REQUEST* The LAN IP Device receives the DHCP OFFER and issues a DHCP REQUEST message. The PS MUST transparently bridge the DHCP REQUEST from its LAN interface to its WAN interface without changing the content of the IP packet.	Proceed to CHPSLP-4.	If failure per DHCP protocol, repeat CHPSLP-1.
CHPSLP-4	*DHCP ACK* The Headend DHCP server receives the DHCP REQUEST and sends the DHCP ACK to the LAN IP Device with the LAN IP Device's IPv4 address. The PS MUST transparently bridge the DHCP ACK from its WAN interface to its LAN interface without changing the content of the IP packet.	Provisioning complete.	If failure, the LAN IP Device will time out per DHCP protocol and CHPSLP-1 will be repeated.
NOTE – If the client is located on a non-broadcast network, it must unicast the message to the DHCP Server or DHCP Relay Agent in the cable network.			

Annex A

MIB objects

This annex lists all required MIB objects, as indicated in 6.3.7.

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
mib-2			
system			
sysDescr	read-only	N/A	N/A
sysObjectID	read-only	N/A	N/A
sysUpTime	read-only	N/A	N/A
sysContact	read-write	Yes	1
sysName	read-write	Yes	1
sysLocation	read-write	Yes	1
sysServices	read-only	N/A	N/A
interfaces [RFC 2863]			
ifNumber	read-only	N/A	N/A
ifTable/ifEntry			
ifIndex	read-only	N/A	N/A
ifDescr	read-only	N/A	N/A

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
ifType	read-only	N/A	N/A
ifMtu	read-only	N/A	N/A
ifSpeed	read-only	N/A	N/A
ifPhysAddress	read-only	N/A	N/A
ifAdminStatus	read-write	N/A	N/A
ifOperStatus	read-only	N/A	N/A
ifLastChange	read-only	N/A	N/A
ifInOctets	read-only	N/A	N/A
ifInUcastPkts	read-only	N/A	N/A
ifInDiscards	read-only	N/A	N/A
ifInErrors	read-only	N/A	N/A
ifInUnknownProtos	read-only	N/A	N/A
ifOutOctets	read-only	N/A	N/A
ifOutUcastPkts	read-only	N/A	N/A
ifOutDiscards	read-only	N/A	N/A
ifOutErrors	read-only	N/A	N/A
ip [RFC 2011]			
ipForwarding	read-write	No	N/A
ipDefaultTTL	read-write	No	N/A
ipInReceives	read-only	N/A	N/A
ipInHdrErrors	read-only	N/A	N/A
ipInAddrErrors	read-only	N/A	N/A
ipForwDatagrams	read-only	N/A	N/A
ipInUnknownProtos	read-only	N/A	N/A
ipInDiscards	read-only	N/A	N/A
ipInDelivers	read-only	N/A	N/A
ipOutRequests	read-only	N/A	N/A
ipOutDiscards	read-only	N/A	N/A
ipOutNoRoutes	read-only	N/A	N/A
ipReasmTimeout	read-only	N/A	N/A
ipReasmReqds	read-only	N/A	N/A
ipReasmOKs	read-only	N/A	N/A
ipReasmFails	read-only	N/A	N/A
ipFragOKs	read-only	N/A	N/A
ipFragFails	read-only	N/A	N/A
ipFragCreates	read-only	N/A	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

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
icmp			
icmpInMsgs	read-only	N/A	N/A
icmpInErrors	read-only	N/A	N/A
icmpInDestUnreachs	read-only	N/A	N/A
icmpInTimeExcds	read-only	N/A	N/A
icmpInParmProbs	read-only	N/A	N/A
icmpInSrcQuenchs	read-only	N/A	N/A
icmpInRedirects	read-only	N/A	N/A
icmpInEchos	read-only	N/A	N/A
icmpInEchosReps	read-only	N/A	N/A
icmpInTimestamps	read-only	N/A	N/A
icmpInTimestampsReps	read-only	N/A	N/A
icmpInAddrMasks	read-only	N/A	N/A
icmpInAddrMaskReps	read-only	N/A	N/A
icmpOutMsgs	read-only	N/A	N/A
icmpOutErrors	read-only	N/A	N/A
icmpOutDestUnreachs	read-only	N/A	N/A
icmpOutTimeExcds	read-only	N/A	N/A
icmpOutParmProbs	read-only	N/A	N/A
icmpOutSrcQuenchs	read-only	N/A	N/A
icmpOutRedirects	read-only	N/A	N/A
icmpOutEchos	read-only	N/A	N/A
icmpOutEchosReps	read-only	N/A	N/A
icmpOutTimestamps	read-only	N/A	N/A
icmpOutTimestampReps	read-only	N/A	N/A
icmpOutAddrMasks	read-only	N/A	N/A
icmpOutAddrMaskReps	read-only	N/A	N/A
udp [RFC 2013]			
udpInDatagrams	read-only	N/A	N/A
udpNoPorts	read-only	N/A	N/A
udpInErrors	read-only	N/A	N/A
udpOutDatagrams	read-only	N/A	N/A
udpTable/udpEntry			
udpLocalAddress	read-only	N/A	N/A
udpLocalPort	read-only	N/A	N/A

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
transmission [draft-ietf-ipcdn-bpiplus-mib-12]			
docsIfMib			
docsBpi2MIB			
docsBpi2MIBObjects			
docsBpi2CmObjects			
docsBpi2CmCertObjects			
docsBpi2CmDeviceCertTable/docsBpi2CmDeviceCertEntry			
docsBpi2CmDeviceCmCert	read-write	Yes	5
docsBpi2CmDeviceManufCert	read-only	N/A	N/A
docsBpi2CodeDownloadGroup			
docsBpi2CodeDownloadStatusCode	read-only	N/A	N/A
docsBpi2CodeDownloadStatusString	read-only	N/A	N/A
docsBpi2CodeMfgOrgName	read-only	N/A	N/A
docsBpi2CodeMfgCodeAccessStart	read-only	N/A	N/A
docsBpi2CodeMfgCvcAccessStart	read-only	N/A	N/A
docsBpi2CodeCoSignerOrgName	read-only	N/A	N/A
docsBpi2CodeCoSignerCodeAccessStart	read-only	N/A	N/A
docsBpi2CodeCoSignerCvcAccessStart	read-only	N/A	N/A
docsBpi2CodeCvcUpdate	read-write	Yes	1
snmp [RFC 3416]			
snmpInPkts	read-only	N/A	N/A
snmpInBadVersions	read-only	N/A	N/A
snmpInBadCommunityNames	read-only	N/A	N/A
snmpInBadCommunityUses	read-only	N/A	N/A
snmpInASNParseErrs	read-only	N/A	N/A
snmpEnableAuthenTraps	read-write	No	N/A
snmpSilentDrops	read-only	N/A	N/A
ifMIB [RFC 2863]			
ifMIBOjects			
ifXTable/ifXEntry			
ifName	read-only	N/A	N/A
ifInMulticastPkts	read-only	N/A	N/A
ifInBroadcastPkts	read-only	N/A	N/A
ifOutMulticastPkts	read-only	N/A	N/A
ifOutBroadcastPkts	read-only	N/A	N/A
ifLinkUpDownTrapEnable	read-write	No	N/A
ifHighSpeed	read-only	N/A	N/A
ifPromiscuousMode	read-write	N/A	N/A
ifConnectorPresent	read-only	N/A	N/A
ifAlias	read-write	No	N/A
ifCounterDiscontinuityTime	read-only	N/A	N/A

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
ifStackTable/ifStackEntry			
ifStackHigherLayer	read-only	N/A	N/A
IfStackLowerLayer	read-only	N/A	N/A
ifStackStatus	read-only	N/A	N/A
docsDev [RFC 2669]			
docsDevMIBObjects			
docsDevNmAccessTable/docsDevNmAccessEntry			
docsDevNmAccessIndex	not-accessible	N/A	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	No	1
docsDevSwOperStatus	read-only	N/A	N/A
docsDevSwCurrentVers	read-only	N/A	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	N/A
docsDevEvThrottleThreshold	read-write	No	N/A
docsDevEvThrottleInterval	read-write	No	N/A
docsDevEvControlTable/docsDevEvControlEntry			
docsDevEvPriority	not-accessible	N/A	N/A
docsDevEvReporting	read-write	No	N/A
docsDevEventTable/docsDevEventEntry			
docsDevEvIndex	not-accessible	N/A	N/A
docsDevEvFirstTime	read-only	Yes	1
docsDevEvLastTime	read-only	Yes	1
docsDevEvCounts	read-only	Yes	1
docsDevEvLevel	read-only	Yes	1
docsDevEvId	read-only	Yes	1
docsDevEvText	read-only	Yes	1

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
private			
enterprises			
cableLabs			
clabProject			
clabProjCableHome			
cabhPsDevMib			
cabhPsDevBase			
cabhPsDevDateTime	read-write	No	N/A
cabhPsDevResetNow	read-write	No	N/A
cabhPsDevSerialNumber	read-only	N/A	N/A
cabhPsDevHardwareVersion	read-only	N/A	N/A
cabhPsDevWanManMacAddress	read-only	N/A	N/A
cabhPsDevWanDataMacAddress	read-only	N/A	N/A
cabhPsDevTypeIdentifier	read-only	N/A	N/A
cabhPsDevSetToFactory	read-write	No	N/A
cabhPsDevTodSyncStatus	read-only	N/A	N/A
cabhPsDevProvMode	read-only	N/A	N/A
cabhPsDevLastSetToFactory	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	N/A
cabhPsDevProvConfigFileStatus	read-only	N/A	N/A
cabhPsDevProvConfigTLVProcessed	read-only	N/A	N/A
cabhPsDevProvConfigTLVRejected	read-only	N/A	N/A
cabhPsDevProvSolicitedKeyTimeout	read-write	Yes	1
cabhPsDevProvState	read-only	N/A	N/A
cabhPsDevProvAuthState	read-only	N/A	N/A
cabhPsDevTimeServerAddrType	read-only	N/A	N/A
cabhPsDevTimeServerAddr	read-only	N/A	N/A
cabhSecMib			
cabhSecFwObjects			
cabhSecFwBase			
cabhSecFwPolicyFileEnable	read-write	No	N/A
cabhSecFwPolicyFileURL	read-write	no	N/A
cabhSecFwPolicyFileHash	read-write	No	N/A
cabhSecFwPolicyFileOperStatus	read-only	N/A	N/A
cabhSecFwPolicyFileCurrentVersion	read-only	N/A	N/A
cabhSecFwPolicySuccessfulFileURL, Max-Access	read-only	yes	1

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
cabhSecFwLogCtl			
cabhSecFwEventType1Enable	read-write	No	N/A
cabhSecFwEventType2Enable	read-write	No	N/A
cabhSecFwEventType3Enable	read-write	No	N/A
cabhSecFwEventAttackAlertThreshold	read-write	No	N/A
cabhSecFwEventAttackAlertPeriod	read-write	No	N/A
cabhSecCertObjects			
cabhSecCertPsCert	read-only	Yes	1
cabhCapMib			
cabhCapObjects			
cabhCapBase			
cabhCapTcpTimeWait	read-write	Yes	1
cabhCapUdpTimeWait	read-write	Yes	1
cabhCapIcmpTimeWait	read-write	Yes	1
cabhCapPrimaryMode	read-write	No	N/A
cabhCapSetToFactory	read-write	No	N/A
cabhCapLastSetToFactory	read-only	–	N/A
cabhCapMap			
cabhCapMappingTable/cabhCapMappingEntry			
cabhCapMappingIndex	not-accessible	Yes (Note)	16
cabhCapMappingWanAddrType	read-create	Yes (Note)	16
cabhCapMappingWanAddr	read-create	Yes (Note)	16
cabhCapMappingWanPort	read-create	Yes (Note)	16
cabhCapMappingLanAddrType	read-create	Yes (Note)	16
cabhCapMappingLanAddr	read-create	Yes (Note)	16
cabhCapMappingLanPort	read-create	Yes (Note)	16
cabhCapMappingMethod	read-only	N/A	16
cabhCapMappingProtocol	read-create	Yes (Note)	16
cabhCapMappingRowStatus	read-create	Yes	16
cabhCapPass-throughTable/cabhCapPass-throughEntry			
cabhCapPass-throughIndex	not-accessible	Yes	16
cabhCapPass-throughMACAddr	read-create	Yes	16
cabhCapPass-throughRowStatus	read-create	Yes	16
NOTE – cabhCapMappingEntry objects are persistent if provisioned by the NMS and non-persistent if created dynamically based on outbound traffic. Refer to 8.3.2.2.			
cabhCdpMib			
cabhCdpObjects			
cabhCdpBase			
cabhCdpSetToFactory	read-write	No	N/A
cabhCdpLanTransCurCount	read-only	N/A	N/A
cabhCdpLanTransThreshold	read-write	No	N/A

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
cabhCdpLanTransAction	read-write	No	N/A
cabhCdpWanDataIpAddrCount	read-write	No	N/A
cabhCdpLastSetToFactory	read-only	–	N/A
cabhCdpAddr			
cabhCdpLanAddrTable/cabhCdpLanAddrEntry			
cabhCdpLanAddrIpType	not-accessible	Yes	16
cabhCdpLanAddrIp	not-accessible	Yes	16
cabhCdpLanAddrClientID	read-create	Yes	16
cabhCdpLanAddrLeaseCreateTime	read-only	Yes	16
cabhCdpLanAddrLeaseExpireTime	read-only	Yes	16
cabhCdpLanAddrMethod	read-only	Yes	16
cabhCdpLanAddrHostName	read-only	Yes	16
cabhCdpLanAddrRowStatus	read-create	Yes	16
cabhCdpWanDataAddrTable/cabhCdpWanDataAddrEntry			
cabhCdpWanDataAddrIndex	not-accessible	N/A	N/A
cabhCdpWanDataAddrClientId	read-create	No	N/A
cabhCdpWanDataAddrIpType	read-only	N/A	N/A
cabhCdpWanDataAddrIp	read-only	N/A	N/A
cabhCdpWanDataAddrRenewalTime	read-only	N/A	N/A
cabhCdpWanDataAddrRowStatus	read-create	No	N/A
cabhCdpWanDataAddrServerTable/cabhCdpWanDataAddrServerEntry			
cabhCdpWanDataAddrDnsIpType	not-accessible	N/A	N/A
cabhCdpWanDataAddrDnsIp	not-accessible	N/A	N/A
cabhCdpWanDataAddrDnsRowStatus	read-create	No	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

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
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
cabhCdpServerControl	read-write	No	N/A
cabhCdpServerCommitStatus	read-only	–	N/A
cabhCtpMib			
cabhCtpObjects			
cabhCtpBase			
cabhCtpSetToFactory	read-write	No	N/A
cabhCtpLastSetToFactory	read-only	–	N/A
cabhCtpConnSpeed			
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	N/A
cabhCtpConnPktsSent	read-only	N/A	N/A
cabhCtpConnPktsRecv	read-only	N/A	N/A
cabhCtpConnRTT	read-only	N/A	N/A
cabhCtpConnThroughput	read-only	N/A	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	N/A
cabhCtpPingNumSent	read-only	N/A	N/A
cabhCtpPingNumRecv	read-only	N/A	N/A

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
cabhCtpPingAvgRTT	read-only	N/A	N/A
cabhCtpPingMinRTT	read-only	N/A	N/A
cabhCtpPingMaxRTT	read-only	N/A	N/A
cabhCtpPingNumIcmpError	read-only	N/A	N/A
cabhCtpPingIcmpError	read-only	N/A	N/A
experimental			
snmpUSMDHObjectsMIB [RFC 2786]			
usmDHKeyObjects			
usmDHPublicObjects			
usmDHPParamaters	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	No	N/A
usmDHKickstartMyPublic	read-only	N/A	N/A
usmDHKickstartMgrPublic	read-only	N/A	N/A
usmDHKickstartSecurityName	read-only	N/A	N/A
snmpV2			
snmpModules			
snmpMIB			
snmpMIBObjects			
snmpSet			
snmpSetSerialNo	read-write	No	N/A
snmpFrameworkMIB [RFC 2576]			
snmpEngine			
snmpEngineID	read-only	N/A	N/A
snmpEngineBoots	read-only	Yes	1
snmpEngineTime	read-only	N/A	N/A
snmpEngineMaxMessageSize	read-only	N/A	N/A
snmpMPDMIB [RFC 3412]			
snmpMPDObjects			
snmpMPDStats			
snmpUnknownSecurityModels	read-only	N/A	N/A
snmpInvalidMsgs	read-only	N/A	N/A
snmpUnknownPDUHandlers	read-only	N/A	N/A

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
snmpTargetMIB [RFC 3413]			
snmpTargetObjects			
snmpTargetSpinLock	read-write	No	N/A
snmpTargetAddrTable/snmpTargetAddrEntry			
snmpTargetAddrName	not-accessible	No	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	No	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	N/A
snmpUnknownContexts	read-only	N/A	N/A
snmpNotificationMIB [RFC 3413]			
snmpNotifyObjects			
snmpNotifyTable/snmpNotifyEntry			
snmpNotifyName	not-accessible	No	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	No	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

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
snmpUsmMIB [RFC 3414]			
usmStats			
usmStatsUnsupportedSecLevels	read-only	N/A	N/A
usmStatsNotInTimeWindows	read-only	N/A	N/A
usmStatsUnknownUserNames	read-only	N/A	N/A
usmStatsUnknownEngineIDs	read-only	N/A	N/A
usmStatsWrongDigests	read-only	N/A	N/A
usmStatsDecryptionErrors	read-only	N/A	N/A
usmUser			
usmUserSpinLock	read-write	No	N/A
usmUserTable/usmUserEntry			
usmUserEngineID	not-accessible	N/A	N/A
usmUserName	not-accessible	N/A	N/A
usmUserSecurityName	read-only	N/A	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	No	N/A
vacmSecurityToGroupTable/vacmSecurityToGroupEntry			
vacmSecurityModel	not-accessible	No	N/A
vacmSecurityName	not-accessible	No	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	No	N/A
vacmAccessSecurityModel	not-accessible	No	N/A
vacmAccessSecurityLevel	not-accessible	No	N/A
vacmAccessContextMatch	read-create	No	N/A
vacmAccessReadViewName	read-create	No	N/A

MIB NAME/Parameter	Max-Access	Persistent	# of Persistent Entries
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	No	N/A
vacmViewTreeFamilySubtree	not-accessible	No	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	No	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	N/A
clabCVCRootCACert	read-only	N/A	N/A
clabCVCCACert	read-only	N/A	N/A
clabMfgCVCCert	read-only	N/A	N/A

Annex B

Format and content for event, SYSLOG and SNMP trap

Table B.1 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 6.5.1.3 and SNMP trap format is defined in this annex, following the table.

The first and second columns 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 4th 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 6.5.1.3. 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/J.191 – 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	DHCP	Critical	DHCP FAILED – Discover sent, no offer received		D01.0	68000100	
Init	DHCP	Critical	DHCP FAILED – Request sent, No response		D02.0	68000200	
Init	DHCP	Critical	DHCP FAILED – Requested Info not supported.		D03.0	68000300	

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
Init	DHCP	Critical	DHCP ERROR – Response does not contain ALL the valid fields or the PS is unable to determine provisioning mode		D03.1	68000301	
ToD Errors before provisioning complete							
Init	ToD	Warning	ToD Request sent – no response received		D04.1	68000401	
Init	ToD	Warning	ToD Response received – invalid data format		D04.2	68000402	
TFTP Errors before provisioning complete							
Init	TFTP	Critical	TFTP failed – Request sent – No Response		D05.0	68000500	
Init	TFTP	Critical	TFTP failed – configuration file NOT FOUND	For SYSLOG only: append: File name = <P1> P1 = requested file name	D06.0	68000600	
Init	TFTP	Critical	TFTP Failed – OUT OF ORDER packets		D07.0	68000700	
Init	TFTP	Critical	TFTP file complete – but failed SHA-1 hash check	For SYSLOG only: append: File name = <P1> P1 = filename of TFTP file	D08.0	68000800	
Init	TFTP	Critical	TFTP Failed Exceeded maximum number of retries	For Syslog only: append: Retry limit = <P1> P1 = maximum number of retries	D09.0	68000900	

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
TFTP Success							
Init	TFTP	Notice	TFTP success		D10.0	68001000	
TLV Parsing							
Init	TLV parsing	Notice	TLV-28 – unrecognized OID		I401.0	73040100	cabhPsDevInitTLVUnknownTrap
Init	TLV parsing	Notice	Unknown TLV <P1>	For SYSLOG only, <P1> = the complete TLV in hexadecimal	I401.1	73040101	cabhPsDevInitTLVUnknownTrap
Init	TLV parsing	Notice	Invalid TLV Format/contents <P1>	For SYSLOG only, <P1> = the complete TLV in hexadecimal	I401.2	73040102	
Provisioning							
Init	SNMP Inform	Notice	SNMP Inform sent signalling provisioning complete (pass/fail)	For SYSLOG only, append MAC Addr: <P1>. P1 = PS MAC address	I11.0	73001100	cabhPsDevInitTrap
Init	SNMP Inform retransmission	Critical	SNMP Inform sent signalling provisioning complete (pass/fail), no response. SNMP Inform resent	For SYSLOG only, append: MAC Addr: <P1>. P1 = PS MAC address	I11.1	73001101	cabhPsDevInitRetryTrap
SW upgrade init (Note)							
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

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
SW Upgrade	SW upgrade init	Notice	SW Download INIT – Via Config file <P1>	P1 = CM config file nameFor SYSLOG only, append: SW file: <P2> – SW server: <P3>. P2 = SW file name and P3 = TFTP server IP address	E102.0	69010200	cabhPsDev SwUpgrade InitTrap
SW upgrade general failure (Note)							
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade FailTrap

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade FailTrap

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
SW Upgrade	SW upgrade general failure	Error	Disruption during SW download – RF removed	For SYSLOG only, append: SW file: <P1> – SW server: <P2>. P1 = SW file name and P2 = TFTP server IP address	E110.0	69011000	cabhPsDev SwUpgrade FailTrap
SW upgrade success (Note)							
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	cabhPsDev SwUpgrade SuccessTrap
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	cabhPsDev SwUpgrade SuccessTrap
DHCP failure after provisioning complete							
DHCP		Error	DHCP RENEW sent – No response		D101.0	68010100	cabhPsDev DHCPFailTrap
DHCP		Error	DHCP REBIND sent – No response		D102.0	68010200	cabhPsDev DHCPFailTrap
DHCP		Error	DHCP RENEW sent – Invalid DHCP option		D103.0	68010300	cabhPsDev DHCPFailTrap
DHCP		Error	DHCP REBIND sent – Invalid DHCP option		D104.0	68010400	cabhPsDev DHCPFailTrap

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
ToD failure after provisioning complete							
ToD	ToD	Warning	ToD Request sent – no response received		D04.3	68000403	cabhPsDev ToDFail Trap
ToD	ToD	Warning	ToD Response received – invalid data format		D04.4	68000404	cabhPsDev ToDFail Trap
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade FailTrap

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade FailTrap
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	cabhPsDev SwUpgrade CVCFail Trap
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	cabhPsDev SwUpgrade CVCFail Trap
SW Upgrade	Verification of CVC	Error	Improper SNMP CVC Format – SNMP manager: <P1>	P1 = IP Address of SNMP Manager	E208.0	69020800	cabhPsDevS wUpgradeC VCFailTrap
SW Upgrade	Verification of CVC	Error	SNMP CVC Validation Failure – SNMP manager: <P1>	P1 = IP Addr of SNMP manager	E209.0	69020900	cabhPsDevS wUpgradeC VCFailTrap

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
CDP Events							
CDP	CDS	Notice	Attempt to allocate more LAN TRANS IP addresses than allowed		P01.0	80000100	cabhPsDevCDP Threshold Trap
CDP	CDS	Notice	Unable to obtain all WAN-Data IP addresses the PS was configured to obtain		P02.0	80000200	cabhPsDevCdpWanDataIpTrap
CDP	CDS	Notice	Unable to provision DHCP LAN client-IP address pool exhausted		P03.0	80000300	cabhPsDevCdpLanIp PoolTrap
CSP Events							
CSP	Firewall	Notice	Firewall Type 1 and Type 2 hacker threshold exceeded		P101.0	80010100	cabhPsDevCSPTrap
CSP	Firewall	Notice	Firewall Type 1 event detected	P1= IP address of source, P2 = IP address of destination, P3 = type of protocol, P4 = active rule set file name, P5 = event description	P102.0	80010200	cabhPsDev CSPTrap
CSP	Firewall	Notice	Firewall Type 2 event detected	P1= IP address of source, P2 = IP address of destination, P3 = type of protocol, P4 = active rule set file name, P5 = event description	P103.0	80010300	cabhPsDev CSPTrap

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
CSP	Firewall	Notice	Firewall configuration has changed	P1 = description of change in firewall configuration parameters	P120.0	80012000	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 hash 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

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
CAP Events							
CAP	C-NAT	Notice	CAP unable to make C-NAT mapping. No WAN-data IP address available		P201.0	80020100	cabhPsDevCAPTrap
CAP	C-NAPT	Notice	CAP unable to make C-NAPT mapping. No WAN IP address available		P250.0	80025000	cabhPsDevCAPTrap
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 (ms)	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 (ms)	P303.0	80030300	cabhPsDevCtpTrap

Table B.1/J.191 – Defined events for IPCable2Home

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	EventID	Trap name
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
NOTE – Software upgrade (secure software download) events apply to stand-alone Portal Services only. Software upgrade is controlled by the 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 11.3.7.1.							

B.1 Trap descriptions

All traps specified by IPCable2Home are defined in the PS DEV MIB specification (see E.1).

Annex C

Security threats and preventative measures

C.1 Security threats

When developing a security technology, it is important to understand what the primary threats are 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 network operators have been identified:

C.1.1 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 neighbour) 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 neighbour) 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.

C.1.2 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.

C.1.3 service confidentiality: The service confidentiality threat involves a 3rd party (neighbours, 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.

C.2 Preventive measures

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:

C.2.1 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.

C.2.2 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.

C.2.3 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.

C.2.4 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.

C.2.5 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

The existence of NAT and Firewall functionality are known to disrupt a number of protocols and applications. The following list of protocols and applications MUST work through CAT and IPCable2Home Firewall implementations. This list is NOT prioritized.

1) FTP;
2) Peer-to-peer application (i.e., Gnutella, LimeWire, BearShare, Morpheus, etc.);
3) IPSec;
4) IGMP and IP Multicast;
5) H.323 (Used in Windows for various applications);
6) Instant Messaging applications (i.e., AOL, Microsoft, Yahoo, etc.);
7) E-mail (SMTP and POP);
8) Streaming Media applications (i.e., Real, MediaPlayer, etc.).

In addition, vendors SHOULD make every attempt to support online gaming applications through CAT and Firewall implementations.

RFC 3235 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 an IPCable2Home environment adhere to these guidelines.

Annex E

MIBs

E.1 Portal Service (PS) MIB

The 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,
   InetAddressIPv4,
   InetAddressIPv6                    FROM INET-ADDRESS-MIB

   docsDevSwCurrentVers,
   docsDevEvLevel,
   docsDevEvId,
   docsDevEvText,
   docsDevSwFilename,
   docsDevSwServer                    FROM DOCS-CABLE-DEVICE-MIB -- RFC 2669

   cabhCdpServerDhcpAddress,
   cabhCdpWanDataAddrClientId,
   cabhCdpLanTransThreshold           FROM CABH-CDP-MIB

   clabProjCableHome                  FROM CLAB-DEF-MIB;

--============================================================================
--
--   History:
--
--
--============================================================================


cabhPsDevMib MODULE-IDENTITY
    LAST-UPDATED    "0209200000Z"-- September 20, 2002
    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 describes
             general PS Device attributes and behaviour characteristics.
             Most of the PS Device MIB is needed for configuration download."


    ::=  { 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. 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      current
     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)
     }
     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, 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)."
::={ 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 config file, calculated and
       sent to the PS prior to sending the config file. For the
       SHA-1 authentication algorithm the hash length is 160 bits."
     ::= { 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      current
    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, docsDevEvId, 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 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  cabhCdpServerDhcpAddress is 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 occured."
    ::= { 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 Cable Security Portal."
    ::= { cabhPsNotification 12 }
```

```
cabhPsDevCapTrap  NOTIFICATION-TYPE
   OBJECTS {
       docsDevEvLevel,
       docsDevEvId,
       docsDevEvText,
       cabhPsDevWanManMacAddress
   }
   STATUS      current
   DESCRIPTION
           "To report an event with the Cable 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 cabhCdpLanPoolStart 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 PS MIB."
    ::= { cabhPsGroups 1 }

cabhPsNotificationGroup     NOTIFICATION-GROUP
     NOTIFICATIONS { cabhPsDevInitTLVUnknownTrap, cabhPsDevInitTrap,
cabhPsDevInitRetryTrap,
                     cabhPsDevDHCPFailTrap, cabhPsDevSwUpgradeInitTrap,
cabhPsDevSwUpgradeFailTrap,
                     cabhPsDevSwUpgradeSuccessTrap, cabhPsDevSwUpgradeCVCFailTrap,
cabhPsDevTODFailTrap,
                     cabhPsDevCdpWanDataIpTrap, cabhPsDevCdpThresholdTrap,
cabhPsDevCspTrap,
                     cabhPsDevCapTrap, cabhPsDevCtpTrap, cabhPsDevProvEnrollTrap }
     STATUS   current
     DESCRIPTION
       "These notifications deal with change in status of
       PS Device."
     ::= { cabhPsGroups 2 }

END
```

E.2 Cable Test Portal MIB

The 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
--
--============================================================================

cabhCtpMib MODULE-IDENTITY
    LAST-UPDATED  "0209200000Z" -- September 20, 2002
    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 Cable Test Portal (CTP)."

    ::=  { 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
--   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
       " 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 }

cabhCtpConnRTTOBJECT-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
             " 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  { 5000 } -- 5 seconds
::={ 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 CTP MIB."
    ::= { cabhCtpGroups 1 }

END
```

E.3 Security MIB

The 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 -- RFC 2571
     X509Certificate         FROM DOCS-BPI2-MIB
     clabProjCableHome  FROM CLAB-DEF-MIB;

--=========================================================================
--
--   History:
--
--   Date     Modified by         Reason
--
--
--=========================================================================

cabhSecMib MODULE-IDENTITY
    LAST-UPDATED    "0209200000Z" --September 20, 2002
    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."


    ::=  { 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),
               completeFromProvisioning(2),
               completeFromMgt(3),
               failed(4)
     }
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
        "InProgress(1) indicates that a TFTP download is under way,
         either as a result of a version mismatch at provisioning
         or as a result of a upgradeFromMgt request.
         CompleteFromProvisioning(2) indicates that the last
         software upgrade was a result of version mismatch at
         provisioning. CompleteFromMgt(3) indicates that the last
         software upgrade was a result of setting
         docsDevSwAdminStatus to upgradeFromMgt.
         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 defined 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
" Specification
 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 Cable 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 Cable Firewall MIB"
    ::= { cabhSecGroups 1 }

END
```

E.4 Definition

The Definition MIB MUST be implemented as defined below.

```
CLAB-DEF-MIB DEFINITIONS ::= BEGIN
IMPORTS
     MODULE-IDENTITY,
     X509Certificate         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 }
clabProjOpenCable  OBJECT IDENTIFIER ::= { clabProject 3}
clabProjCableHome   OBJECT IDENTIFIER ::= { clabProject 4 }
clabSecurity  OBJECT IDENTIFIER ::= { cableLabs 3}

clabSecCertObject OBJECT IDENTIFIER ::= { clabSecurity 1 }

clabSrvcPrvdrRootCACert     OBJECT-TYPE
    SYNTAX    X509Certificate
    MAX-ACCESS   read-only
    STATUS    current
    DESCRIPTION
            "The X509 DER-encoded Service Provider Root CA Certificate."
    REFERENCE
    " Specification Section 11"
    ::= { clabSecCertObject 1 }

clabCVCRootCACert      OBJECT-TYPE
    SYNTAX    X509Certificate
    MAX-ACCESS   read-only
    STATUS    current
    DESCRIPTION
            "The X509 DER-encoded CVC Root CA Certificate."
    REFERENCE
    " Specification Section 11 for Standalone PS Elements only"
    ::= { clabSecCertObject 2 }

clabCVCCACert   OBJECT-TYPE
    SYNTAX    X509Certificate
    MAX-ACCESS   read-only
    STATUS    current
    DESCRIPTION
            "The X509 DER-encoded CableLabs CVC CA Certificate."
    REFERENCE
    " Specification Section 11 for Standalone PS Elements only"
    ::= { clabSecCertObject 3 }

clabMfgCVCCert     OBJECT-TYPE
    SYNTAX    X509Certificate
    MAX-ACCESS   read-only
    STATUS    current
```

```
    DESCRIPTION
            "The X509 DER-encoded Manufacturer CVC Certificate."
    REFERENCE
    " Specification Section 11 for Standalone PS Elements only"
    ::= { clabSecCertObject 4 }

END
```

E.5 Cable DHCP Portal (CDP) MIB

The CDP MIB MUST be implemented as defined below.

```
CABH-CDP-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
     Integer32,
     Unsigned32
                    FROM SNMPv2-SMI
     TruthValue,
     TimeStamp,
     RowStatus,
    TEXTUAL-CONVENTION
                    FROM SNMPv2-TC
    OBJECT-GROUP,
    MODULE-COMPLIANCE
                    FROM SNMPv2-CONF
    InetAddressType,
    InetAddress,
    InetAddressIPv4,
    InetAddressIPv6
                       FROM INET-ADDRESS-MIB
     SnmpAdminString         FROM SNMP-FRAMEWORK-MIB -- RFC 2571
    clabProjCableHome
                    FROM CLAB-DEF-MIB;

--============================================================================
--
--   History:
--
--   Date Modified by       Reason
--
--============================================================================

cabhCdpMib MODULE-IDENTITY
    LAST-UPDATED    "0209200000Z" -- September 20, 2002
    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 Cable DHCP Portal (CDP) portion of the PS database."



    ::=  { clabProjCableHome 4 }
```

```
-- Textual conventions
CabhCdpLanTransDhcpClientId ::= TEXTUAL-CONVENTION
     STATUS   current
     DESCRIPTION
       "LAN-Trans DHCP option61 information."
     SYNTAX OCTET STRING (SIZE (1..80))


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
--   DHCP Portal. The rest of this group deals with addresses defined on
--   the LAN side.
--

cabhCdpSetToFactory     OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Setting this object to true(1) causes the DHCP default options to
be returned back to factory defaults. Reading this object always returns
false(2). When cabhCdpSetToFactory is set to true, the following actions occur:
1. Clear all cabhCdpLanAddrEntries in the CDP LAN Address Table.
2. Reset all default CDS DHCP options to the factory defaults.
3. The CDS will offer the factory default DHCP options at the next lease renewal
time.
            The objects set to factory defaults are:
       cabhCdpLanTransThreshold,
       cabhCdpLanTransAction,
       cabhCdpWanDataIpAddrCount,
       cabhCdpLanStartType,
       cabhCdpLanPoolStart,
       cabhCdpLanPoolEndType,
       cabhCdpLanPoolEnd,
       cabhCdpNetworkNumber,
       cabhCdpServerSubnetMaskType,
       cabhCdpServerSubnetMask,
       cabhCdpServerTimeOffset,
       cabhCdpServerRouterType,
       cabhCdpServerRouter,
       cabhCdpServerDnsAddressType,
       cabhCdpServerDnsAddress,
       cabhCdpServerSyslogAddressType,
       cabhCdpServerSyslogAddress,
       cabhCdpServerDomainName,
       cabhCdpServerTTL,
       cabhCdpServerInterfaceMTU,
       cabhCdpServerVendorSpecific,
       cabhCdpServerLeaseTime,
       cabhCdpServerDhcpAddressType,
       cabhCdpServerDhcpAddress"
REFERENCE
""
::= { cabhCdpBase 1 }
```

```
cabhCdpLanTransCurCount OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The current number of LAN-Trans IP addresses for
            Translated addresses (NAT and NAPT Interconnects).
            This is a count of LAN side addresses."
     REFERENCE
       ""
    ::= { 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 and treat the
               interconnection between the LAN and WAN as
               would normally occur if the threshold was not
               exceeded.


          noAssignment - do not assign a LAN-Trans IP address and do
                not create an interconnection"
     REFERENCE
       ""
     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 that the CDC needs to
acquire via DHCP."

     REFERENCE
       ""
     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
            list has one entry for each allocated LAN-Trans IP
            address."
    ::= {  cabhCdpAddr 1 }

cabhCdpLanAddrEntry OBJECT-TYPE
    SYNTAX      CabhCdpLanAddrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of general parameter for CDP mappings."
    INDEX { cabhCdpLanAddrIpType, cabhCdpLanAddrIp }
    ::= { cabhCdpLanAddrTable 1 }

CabhCdpLanAddrEntry ::= SEQUENCE {
cabhCdpLanAddrIpType              InetAddressType,
     cabhCdpLanAddrIp             InetAddress,
     cabhCdpLanAddrClientID            CabhCdpLanTransDhcpClientId,
     cabhCdpLanAddrLeaseCreateTime         TimeStamp,
     cabhCdpLanAddrLeaseExpireTime         TimeStamp,
     cabhCdpLanAddrMethod              INTEGER,
     cabhCdpLanAddrHostName       SnmpAdminString,
     cabhCdpLanAddrRowStatus      RowStatus
    }

cabhCdpLanAddrIpType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The address type assigned on the LAN side for the CDP Address
Table."
    DEFVAL  { ipv4 }
    ::= { cabhCdpLanAddrEntry 1 }



cabhCdpLanAddrIp OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
"The address assigned on the LAN side for the CDP Address Table. 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 created by the NMS through the CMP, when the NMS creates a new DHCP
address reservation by accessing the cabhCdpLanAddrRowStatus object with an
index comprised of a new cabhCdpLanAddrIp and its Type."

    ::= { cabhCdpLanAddrEntry 2 }

cabhCdpLanAddrClientID OBJECT-TYPE
    SYNTAX      CabhCdpLanTransDhcpClientId
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
     "The client ID as indicated in Option 61 of the DHCP Discover. There is a
     one-to-one relationship between the Client ID and the assigned LAN address.
     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 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 new
     cabhCdpLanAddrIp and a new cabhCdpLanAddrClientID."

    ::= { cabhCdpLanAddrEntry 3 }

cabhCdpLanAddrLeaseCreateTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The time the LAN side of the CDP LAN Table was created.
       This entry is only set when the cabhCdpLanAddrTable
       entry is created and the entry does not already exist. In
       other words, this value is not overwritten at lease renewal
       time."
    ::= { cabhCdpLanAddrEntry 4 }

cabhCdpLanAddrLeaseExpireTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "This is the time that the LAN side lease expires.  When
       the lease expires this entry will be deleted from the table."
    ::= { cabhCdpLanAddrEntry 5 }

cabhCdpLanAddrMethod OBJECT-TYPE
    SYNTAX      INTEGER {
     cmp (1),
     cdp (2)
     }
     MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The method that created this Address Entry. cmp
       indicates that configuration through the CMP established this
       row (entry). cdp indicates that a DHCP discover established
       this row (entry)."
    ::= { 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."
    ::= { 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 to 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."
    ::= { cabhCdpWanDataAddrEntry 6 }

--------------------------------------------------------------------------
--
--   cabhCdpWanDataAddrServerTable (CDP WAN-Data DNS Server Table)
--
--   The cabhCdpWanDataAddrServerTable contains a table of referral DNS Servers.

--
--------------------------------------------------------------------------

cabhCdpWanDataAddrServerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CabhCdpWanDataAddrServerEntry
     MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains the IP addresses used for the WAN-Data DNS
            hosts obtained via the DHCP option 6 during the WAN-Data process."
    ::= {  cabhCdpAddr 3 }

cabhCdpWanDataAddrServerEntry OBJECT-TYPE
    SYNTAX      CabhCdpWanDataAddrServerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of WAN-Data DNS Hosts."
    INDEX { cabhCdpWanDataAddrDnsIpType, cabhCdpWanDataAddrDnsIp }
```

```
    ::= { cabhCdpWanDataAddrServerTable 1 }

CabhCdpWanDataAddrServerEntry ::= SEQUENCE {
    cabhCdpWanDataAddrDnsIpType  InetAddressType,
    cabhCdpWanDataAddrDnsIp      InetAddress,
    cabhCdpWanDataAddrDnsRowStatus    RowStatus
    }


cabhCdpWanDataAddrDnsIpType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This parameter indicates the IP address type of a DNS server."
     DEFVAL  { ipv4 }
     ::= { cabhCdpWanDataAddrServerEntry 1 }


cabhCdpWanDataAddrDnsIp OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This parameter indicates the IP address of a DNS server."
     ::= { cabhCdpWanDataAddrServerEntry 2 }

cabhCdpWanDataAddrDnsRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The RowStatus interlock for creation and deletion."
     ::= { cabhCdpWanDataAddrServerEntry 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      INTEGER (68..4096)
     MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Option value 26 - LAN-Trans Interface MTU."
     ::= { 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 }


--
-- 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,
cabhCdpWanDataAddrIp,
cabhCdpWanDataAddrRenewalTime,
cabhCdpWanDataAddrRowStatus,
```

```
cabhCdpWanDataAddrDnsRowStatus,

cabhCdpLanPoolStartType,
cabhCdpLanPoolStart,
cabhCdpLanPoolEndType,
cabhCdpLanPoolEnd,
cabhCdpServerNetworkNumberType,
cabhCdpServerNetworkNumber,
cabhCdpServerSubnetMaskType,
cabhCdpServerSubnetMask,
cabhCdpServerTimeOffset,


cabhCdpServerRouterType,
cabhCdpServerRouter,
cabhCdpServerDnsAddressType,
cabhCdpServerDnsAddress,
cabhCdpServerSyslogAddressType,
cabhCdpServerSyslogAddress,
cabhCdpServerDomainName,
cabhCdpServerTTL,
cabhCdpServerInterfaceMTU,
cabhCdpServerVendorSpecific,
cabhCdpServerLeaseTime,
cabhCdpServerDhcpAddressType,
cabhCdpServerDhcpAddress
      }
    STATUS    current
    DESCRIPTION
        "Group of objects for Cable CDP MIB."
    ::= { cabhCdpGroups 1 }



END
```

E.6 Cable Address Portal

The CAP MIB MUST be implemented as defined below

```
CABH-CAP-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
      Unsigned32
          FROM SNMPv2-SMI
      TimeStamp,
      TruthValue,
      RowStatus,
      PhysAddress
          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
--
--
--============================================================================

cabhCapMib MODULE-IDENTITY
    LAST-UPDATED    "0209200000Z" --September 20, 2002
    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 Cable
           Address Portal (CAP) portion of the PS database."



    ::=  { 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 [RFC 793]."
    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 inactivty 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
            "Setting this object to true(1) causes all the tables in the CAP
            to be cleared, and all CAP objects with defaults to be reset back to
            their default values.

         The objects to set to factory default values when this object is set to
'true' are listed below:
         cabhCapTcpTimeWait,
         cabhCapUdpTimeWait,
         cabhCapIcmpTimeWait,
         cabhCapPrimaryMode,
         cabhCapMappingWanAddrType,
         cabhCapMappingWanPort,
         cabhCapMappingLanAddrType,
         cabhCapMappingLanPort"
    ::= { cabhCapBase 5 }
```

```
--============================================================================
--
--   cabhCapMappingTable (CAP Mapping Table)
--
--   The cabhCapMappingTable contains the mappings for all CAP mappings.
--
--============================================================================

cabhCapMappingTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CabhCapMappingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains IP address mapping for all CAP mappings."
    ::= {  cabhCapMap 1 }

cabhCapMappingEntry OBJECT-TYPE
    SYNTAX      CabhCapMappingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of CAP IP mappings."
    INDEX { cabhCapMappingIndex }
    ::= { cabhCapMappingTable 1 }

     CabhCapMappingEntry ::= SEQUENCE {
     cabhCapMappingWanAddrType        InetAddressType,
     cabhCapMappingIndex              INTEGER,
     cabhCapMappingWanAddr            InetAddress,
     cabhCapMappingWanPort            INTEGER,
     cabhCapMappingLanAddrType        InetAddressType,
     cabhCapMappingLanAddr            InetAddress,
     cabhCapMappingLanPort            INTEGER,
     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. IP version 4 is
       typically used."
    DEFVAL { ipv4 }
    ::= { cabhCapMappingEntry 2 }

cabhCapMappingWanAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The IP address assigned on the WAN side.  IP version 4
       is typically used."
```

```
    ::= { cabhCapMappingEntry 3 }

cabhCapMappingWanPort OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The TCP/UDP port number on the WAN side."
        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.  IP version
         4 is typically used."
    DEFVAL { ipv4 }
    ::= { cabhCapMappingEntry 5 }

cabhCapMappingLanAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The IP address assigned on the LAN side.  IP version 4
       is typically used."
    ::= { cabhCapMappingEntry 6 }

cabhCapMappingLanPort OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The TCP/UDP port number on the LAN side."
    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), -- not specified
       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 CMP will prevent
modification of this table's columns when the cabhCapMappingRowStatus object is
in the active state."
    ::={ 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-ACCESSnot-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-ACCESSnot-accessible
    STATUS   current
    DESCRIPTION
            "The index into the CAP Passthrough Table."
    ::= { cabhCapPassthroughEntry 1 }

cabhCapPassthroughMacAddr        OBJECT-TYPE
    SYNTAX   PhysAddress
    MAX-ACCESSread-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-ACCESSread-create
    STATUS    current
    DESCRIPTION
            "The RowStatus interlock for the creation and deletion of a
             cabhCapPassthroughTable entry.
             There are no restrictions on setting the read-create column of this
             table (i.e., cabhCapPassthroughMacAddr) when the status of
             cabhCapPassthroughRowStatus is active."
    ::= { 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,
       cabhCapMappingWanAddrType,
       cabhCapMappingWanAddr,
       cabhCapMappingWanPort,
       cabhCapMappingLanAddrType,
       cabhCapMappingLanAddr,
       cabhCapMappingLanPort,
       cabhCapMappingMethod,
       cabhCapMappingProtocol,
       cabhCapMappingRowStatus,
       cabhCapPassthroughMacAddr,
```

```
       cabhCapPassthroughRowStatus
       }
    STATUS    current
    DESCRIPTION
        "Group of objects for CAP MIB."
    ::= { cabhCapGroups 1 }


END
```

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Cable networks and transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure, Internet protocol aspects and Next Generation Networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2005