INTERNATIONAL TELECOMMUNICATION UNION

ITU-T J.191

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(07/2002)

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

Miscellaneous

IP feature package to enhance cable modems

ITU-T Recommendation J.191

ITU-T J-SERIES RECOMMENDATIONS

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

General Recommendations	J.1–J.9
General specifications for analogue sound-programme transmission	J.10–J.19
Performance characteristics of analogue sound-programme circuits	J.20–J.29
Equipment and lines used for analogue sound-programme circuits	J.30–J.39
Digital encoders for analogue sound-programme signals	J.40–J.49
Digital transmission of sound-programme signals	J.50–J.59
Circuits for analogue television transmission	J.60–J.69
Analogue television transmission over metallic lines and interconnection with radio-relay links	J.70–J.79
Digital transmission of television signals	J.80–J.89
Ancillary digital services for television transmission	J.90–J.99
Operational requirements and methods for television transmission	J.100–J.109
Interactive systems for digital television distribution	J.110–J.129
Transport of MPEG-2 signals on packetised networks	J.130–J.139
Measurement of the quality of service	J.140–J.149
Digital television distribution through local subscriber networks	J.150–J.159
IPCablecom	J.160–J.179
Miscellaneous	**J.180–J.199**
Application for Interactive Digital Television	J.200–J.209

For further details, please refer to the list of ITU-T Recommendations.

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 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 prepared by ITU-T Study Group 9 (2001-2004) and approved under the WTSA Resolution 1 procedure on 29 July 2002.

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.

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 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.

2 References

2.1 Normative 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.

[ITU-T J.112] ITU-T Recommendation J.112 Annex B (2001), *Data-over-cable service interface specifications*: *Radio-frequency interface specification*.

[ITU-T J.161] 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 J.163] ITU-T Recommendation J.163 (2001), *Dynamic quality of service for the provision of real-time services over cable televison networks using cable modems*.

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

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

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

[FIPS 140-2] FIPS PVB 140-2 (2001), *Security Requirements for Cryptographic Modules*, *Department of commerce, NIST*.

[ISO/IEC 10038] ISO/IEC 10038 (ANSI/IEEE Std 802.1D):1993, *Information technology – Telecommunications and information exchange between systems – Local area networks – Media access control (MAC) bridges*.

[RFC 768] IETF RFC 768 (1980), *User Datagram Protocol*.

[RFC 792] IETF RFC 792 (1981), *Internet Control Message Protocol*.

[RFC 868] IETF RFC 868 (1983), *Time Protocol*.

[RFC 1034] IETF RFC 1034 (1987), *Domain Names – Concepts and Facilities*.

[RFC 1035] IETF RFC 1035 (1987), *Domain Names – Implementation and Specification*.

[RFC 1122] IETF RFC 1122 (1989), *Requirements for Internet Hosts – Communication Layers*.

[RFC 1123] IETF RFC 1123 (1989), *Requirements for Internet Hosts – Application and Support*.

[RFC 1157] IETF RFC 1157 (1990), *A Simple Network Management Protocol (SNMP)*.

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

[RFC 1901] IETF RFC 1901 (1996), *Introduction to Community-based SNMPv2*.

[RFC 1905] IETF RFC 1905 (1996), *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*.

[RFC 1907] IETF RFC 1907 (1996), *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*.

[RFC 2011] IETF RFC 2011 (1996), *SNMPv2 Management Information Base for the Internet Protocol using SMIv2*.

[RFC 2013] IETF RFC 2013 (1996), *SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2*.

[RFC 2131] IETF RFC 2131 (1997), *Dynamic Host Configuration Protocol*.

[RFC 2132] IETF RFC 2132 (1997), *DHCP Options and BOOTP Vendor Extensions*.

[RFC 2236] IETF RFC 2236 (1997), *Internet Group Management Protocol, Version 2*.

[RFC 2349] IETF RFC 2349 (1998), *TFTP Time-out Interval and Transfer Size Options*.

[RFC 2570] IETF RFC 2570 (1999), *Introduction to Version 3 of the Internet-standard Network Management Framework.*

[RFC 2571] IETF RFC 2571 (1999), *An Architecture for Describing SNMP Management Frameworks*.

[RFC 2572] IETF RFC 2572 (1999), *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*.

[RFC 2573] IETF RFC 2573 (1999), *SNMP Applications*.

[RFC 2574] IETF RFC 2574 (1999), *User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*.

[RFC 2575] IETF RFC 2575 (1999), *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*.

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

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

[RFC 2579] IETF RFC 2579 (1999), *Textual Conventions for SMIv2*.

[RFC 2580] IETF RFC 2580 (1999), *Conformance Statements for SMIv2*.

[RFC 2669] IETF RFC 2669 (1999), *DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems*.

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

[RFC 2786] IETF RFC 2786 (2000), *USM Key Management Information Base and Textual Convention*.

[RFC 2863] IETF RFC 2863 (2000), *The Interfaces Group MIB*.

[RFC 3022] IETF RFC 3022 (2001), *Traditional IP Network Address Translator (Traditional NAT)*.

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

2.2 Informative references

[FIPS 186-2] FIPS PUB 186-2 (2000), *Digital Signature Standard*, *Department of commerce, NIST*.

[RFC 347] IETF RFC 347 (1972), *Echo Process*.

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

[DOCSIS2] *Management Information Base for DOCSIS Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus*, draft-ietf-ipcdn-bpiplus-mib-01.txt (work in progress).

draft-ietf-ipcdn-bpiplus-mib-06 INTERNET DRAFT – DOCSIS Baseline Privacy Plus MIB – *Management Information Base for DOCSIS Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus*, November 2001.

[ID-IGMP] FENNER (W.) et al., *IGMP-based Multicast Forwarding ("IGMP Proxying")*, IETF Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-magma-igmp-proxy-00.txt.

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 the 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 and acronyms:

ASP	Application-Specific Proxy
CA	Certificate Authority
CAP	Cable Address Portal
CAT	Cable Address Translation
CDC	Cable DHCP Client
CDP	Cable DHCP Portal
CDS	Cable DHCP Server
CM	Cable Modem
CMP	Cable Management Portal
CMS	Call Management Server
CMTS	Cable Modem Termination System
C-NAPT	Cable Network Address and Port Translation
C-NAT	Cable Network Address Translation
CNP	Cable Naming Portal
CQoS	Cable Quality of Service
CQP	Cable Quality-of-Service Portal
CRL	Certificate Revocation List
CSP	Cable Security Portal
CTP	CableHome 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
HA	Home Access
HEX	Hexadecimal

HFC	Hybrid Fiber Coax
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
IP	Internet Protocol
KDC	Key Distribution Center
LAN-Pass	Pass-through LAN address
LAN-Trans	Translated LAN address
MAC	Media Access Control
MGCP	Media Gateway Control Protocol
MIB	Management Information Base
MPLS	Multiprotocol Label Switching
MSO	Multiple Service Operator
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 System 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-Data	Portal Service element WAN data interface
PS WAN-Man	Portal Service element WAN management 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 keywords "MUST" and "SHALL" as well as "REQUIRED" are to be interpreted as indicating a mandatory aspect of this Recommendation. The keywords indicating a certain level of significance of a particular requirement that are used throughout this Recommendation are summarized below.

"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 behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

5 IP feature package requirements, architecture and overview

This Recommendation provides a set of IP-based features that may be added to a Cable Modem 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 Cable Modem that contains these enhanced features is referred to as an IP-enhanced Cable Modem (IPCM), which is an implementation of a J.190 HA device class. As described in ITU-T Rec. J.190, the HA device class includes both Cable Modem functionality as well as Portal Services functionality.

Major areas and features are as follows:

– *Management and provisioning*
  - Remote management and configuration of the PS;
  - Simple management proxy for IP-based home devices (e.g., a PC);
  - Hands off provisioning for the PS.

– *Addressing and packet handling*
  - One-to-one address translation for home devices;
  - One-to-many address translation for home devices;
  - Non-translated addressing for home devices;
  - Simple DNS server in the PS.

– *Quality of service*
  - Transparent bridging functionality for IPCablecom QoS messaging to/from IPCablecom-compliant applications.

– *Security*
  - PS device authentication;
  - Secure management messages;
  - Secure download of configuration and software files;
  - Secure QoS on the HFC link;
  - Remote PS firewall management.

5.1 Architecture

See Figure 1.

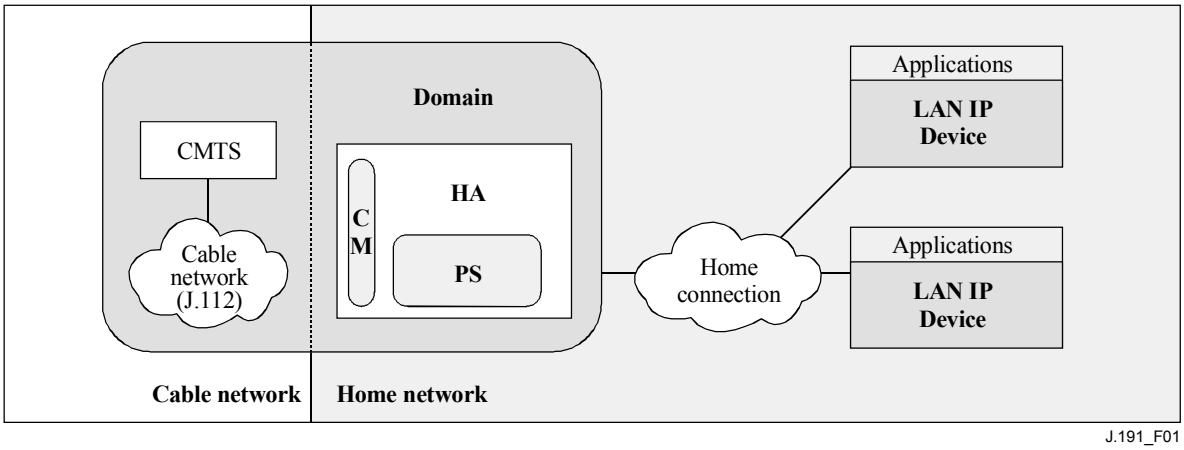


Figure 1/J.191 – Key concepts

5.1.1 Portal service

The Portal Service 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.

5.1.2 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 2).

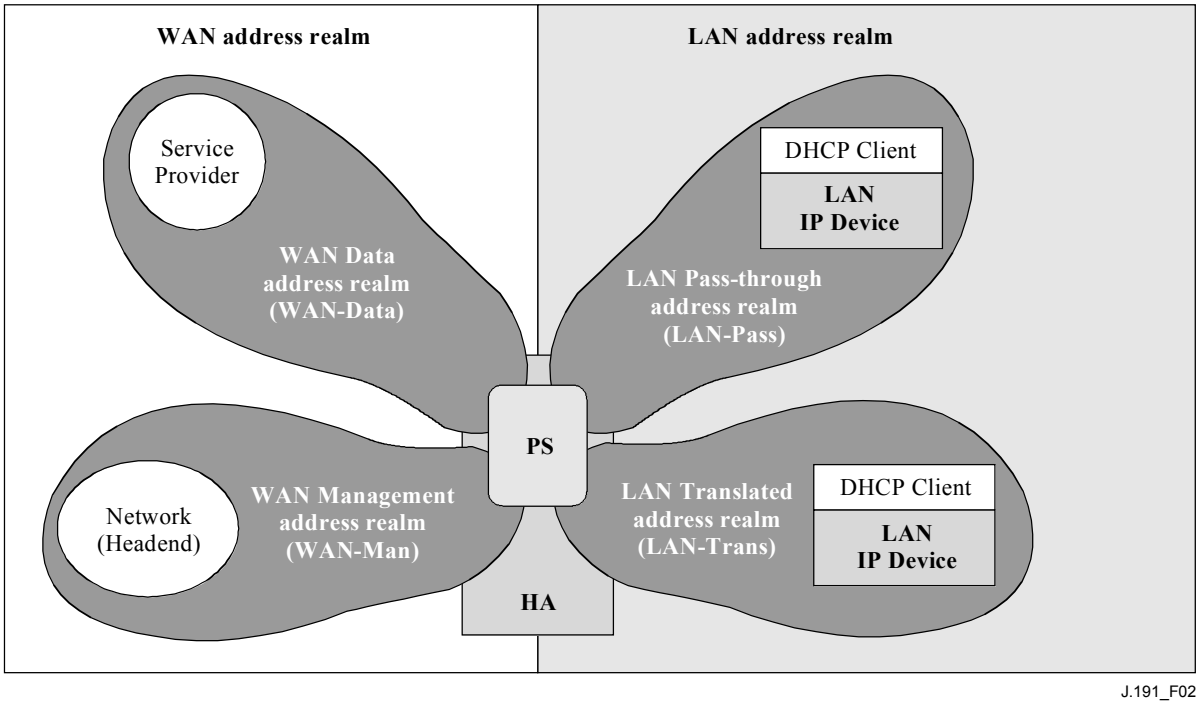


Figure 2/J.191 – Address realms

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. 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 link, 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 the 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).

5.2 Management functions

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

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

Several of the Management Server Functions reside within the headend (HE). Management Client Functions are typically found within LAN IP Devices. Management Service Portal Functions are located within the PS logical element of the Cable Modem 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 introduced in Tables 1, 2, and 3 are illustrated in Figure 3.

Table 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 DNS server	The headend DNS server is a back-office component used to map between ASCII domain names and IP addresses.
Headend Management Messaging server	The headend management messaging, download, event notification servers including protocols such as SNMP, SYSLOG, and TFTP.

Table 2/J.191 – Management and provisioning PS 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 interfaces between the headend 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.
CableHome Testing Portal (CTP)	The CTP provides a remote means to initiate pings and loopbacks within the LAN.

Table 3/J.191 – Management client function description

Management Client Functions	Description
LAN IP Device DHCP Client	The Cable DHCP client function is a in-home component used during the LAN IP Device provisioning process to dynamically request IP addresses and other logical element configuration information.
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.

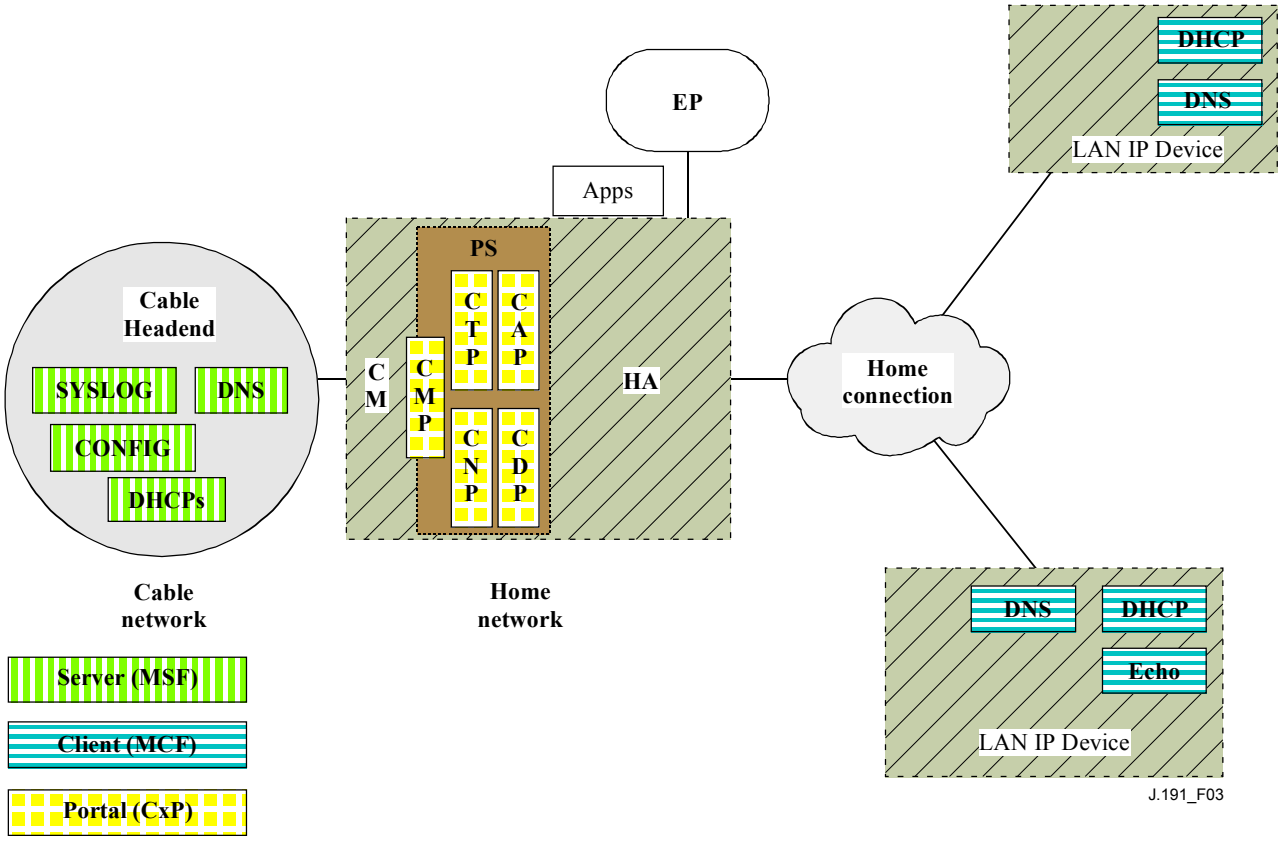


Figure 3/J.191 – Management client-server relationships

5.3 Security functions

Security functions are categorized as Security Portal Functions or Security Server Functions. The relationship between the different security elements and their classification as Server and Portal functions is presented in Figure 4 and described in Tables 4 and 5.

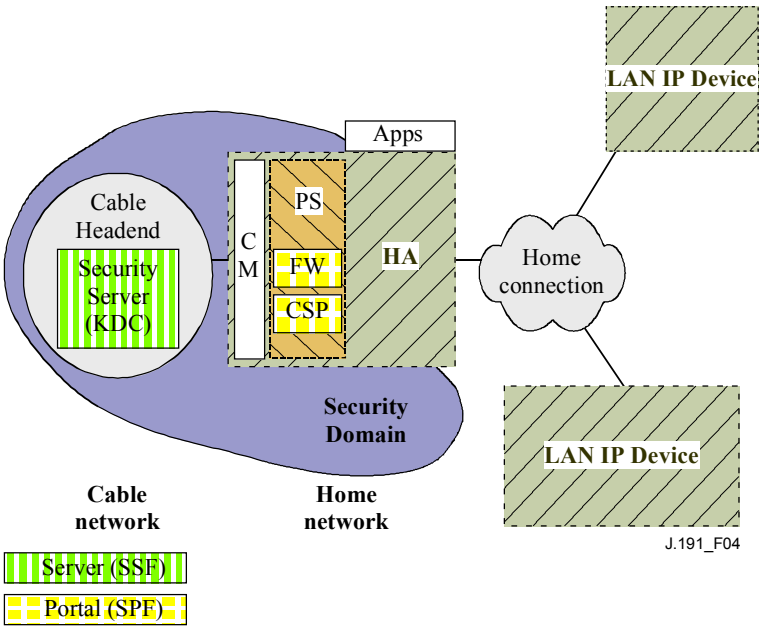


Figure 4/J.191 – Security elements

Table 4/J.191 – Security portal function description

Security Portal Functions	Description
Cable Security Portal (CSP)	The CSP acts as a portal for security material for all of the other security functions within the PS. The CSP communicates on the WAN side with a Security Server (Key Distribution Center, KDC).
Firewall (FW)	The firewall provides protection of the home IP environment from malicious attack.

Table 5/J.191 – Security server function description

Security Server Functions	Description
KDC	The KDC servers in the headend provides for authentication services and key distribution for the home. They communicate with the CSP function to establish these services.

5.4 QoS functions

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

5.5 Messaging interface model

The communication between the functions in the 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. Messaging interfaces are illustrated in Figure 5.

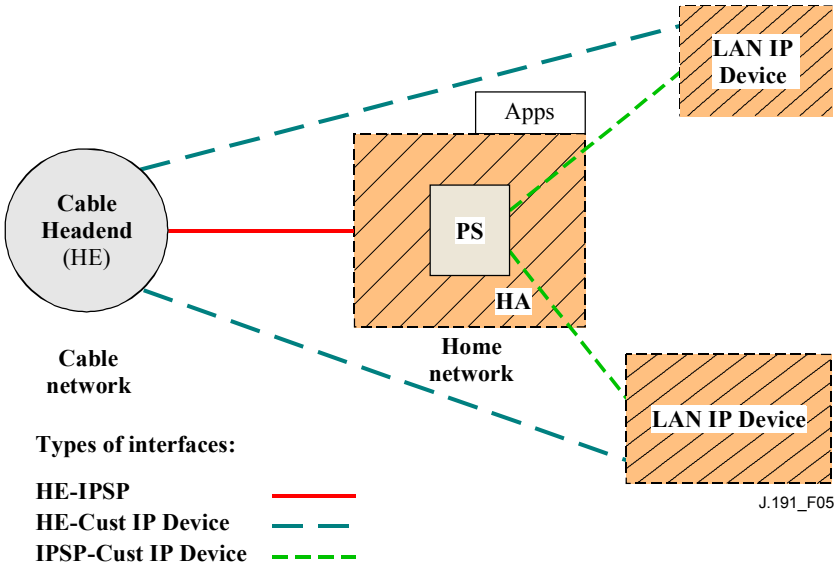


Figure 5/J.191 – Reference interfaces

The Messaging interfaces are summarized in Table 6.

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

Functionality	Protocol	Interface		
		HE-PS	HE-LAN IP Dev	PS-LAN IP Dev
Name service	DNS	Unspecified	Unspecified	Unspecified
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

5.6 Information reference model

The operation of the management model is based upon a store of information maintained in the Portal by the various Portal functions (CAP, CDP, CMP, etc.). These functions must have a means of interacting via information exchange, and the Portal Database is a conceptual entity that represents a store for this information. The Portal 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 6 shows the relationship between the database and the Portal functions, Table 7 describes the typical information associated with each of these functions. Figure 7 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.

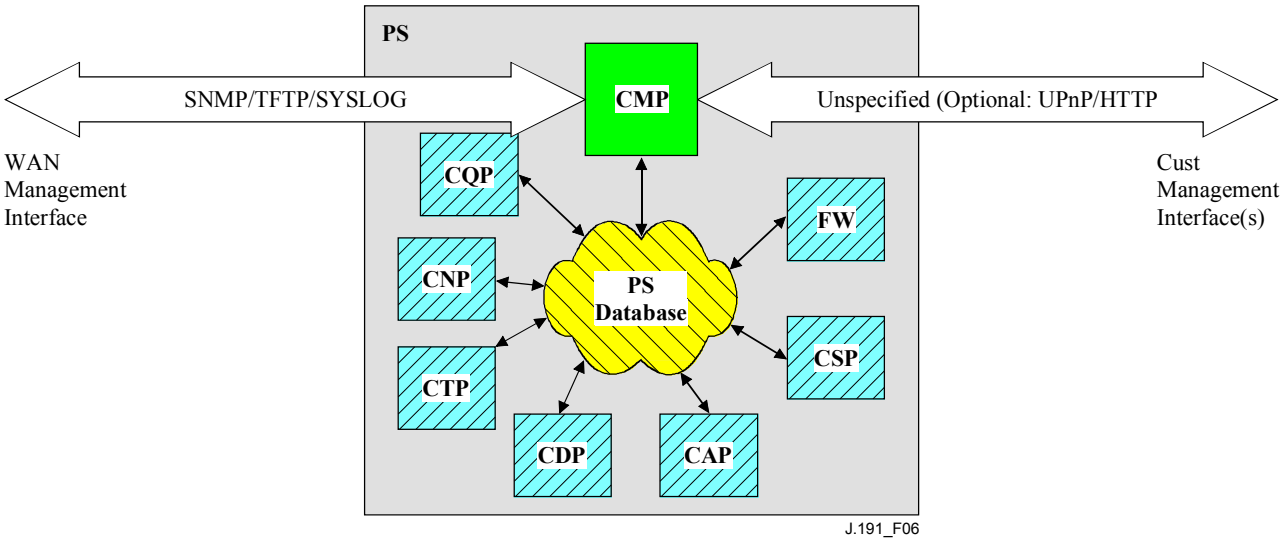


Figure 6/J.191 – Portal function and database relationship

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

Table 7/J.191 – Typical portal database information examples

Name	Usage (in general)
CDP Information	Information associated with addresses acquired and allocated via DHCP
CAP information	Information associated with 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 behavior of the firewall (rule set) and firewall logging
Event information	Information associated with the local log for all general events, traps, etc.

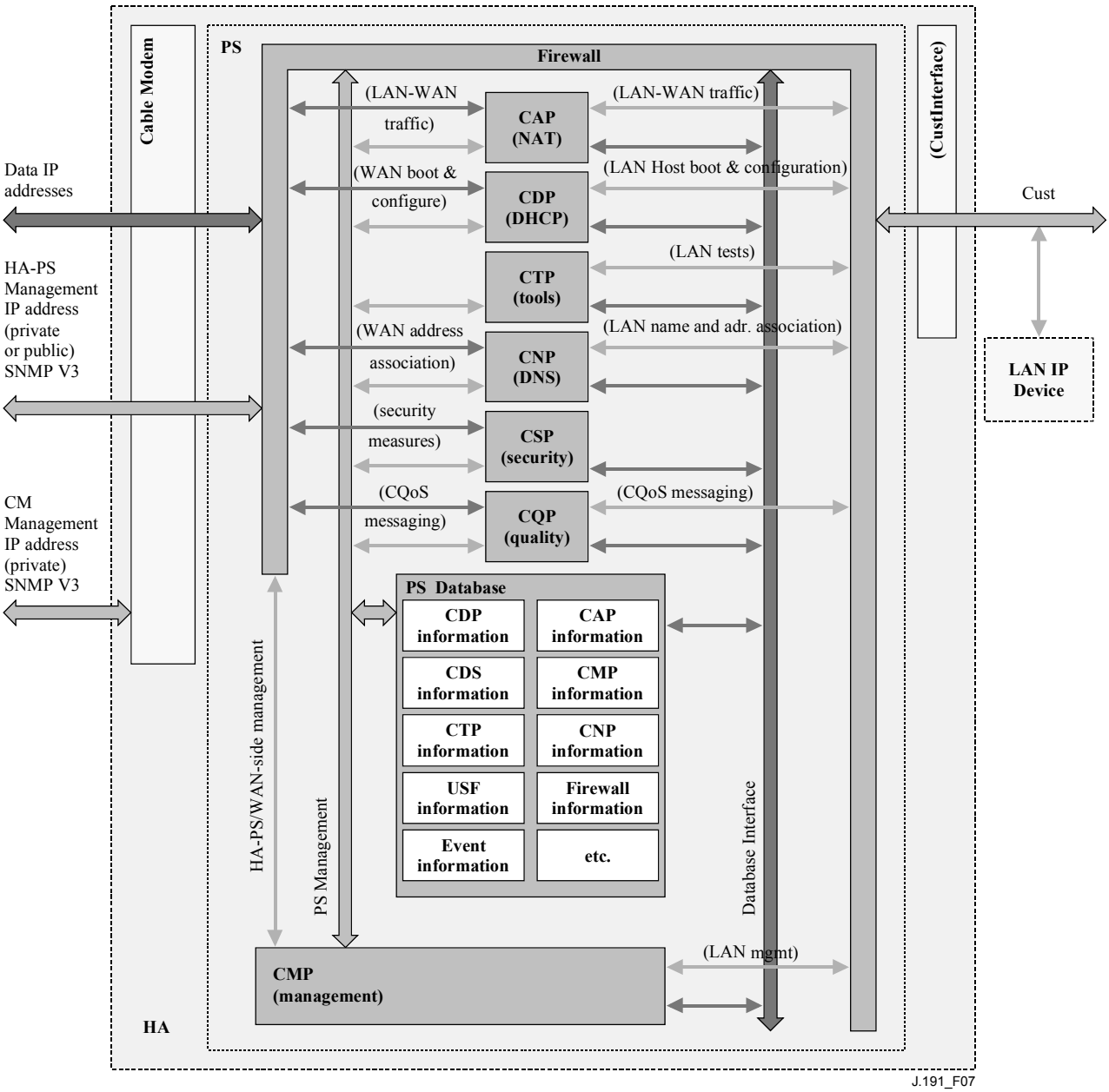


Figure 7/J.191 – Portal database detailed example implementation

The Portal is managed from the WAN via the CMP, and to a large degree this involves access to the information in the Portal Database. Management is used for initialization and provisioning of the WAN side network elements, and diagnostics or status of the LAN side. 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 CNP only responds to DNS queries on the LAN-Trans Realm.

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 PORTAL authentication capabilities as well as key exchange activities.

The CQP is part of a system that enables IPCablecom Quality of Service (QoS) through the portal. The CQP, acting as a transparent bridge, forwards IPCablecom-compliant QoS messaging between IPCablecom applications and the IPCablecom QoS infrastructure.

The firewall is implementation-specific, and this Recommendation does not specify the details of firewall implementation.

5.7 Operational models

This enhanced infrastructure builds upon a cable modem infrastructure 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 Portal may operate within one of two provisioning modes:

- the DHCP Provisioning Mode;
- the SNMP Provisioning 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 8 illustrates the various PS operational modes along with the associated triggers for each.

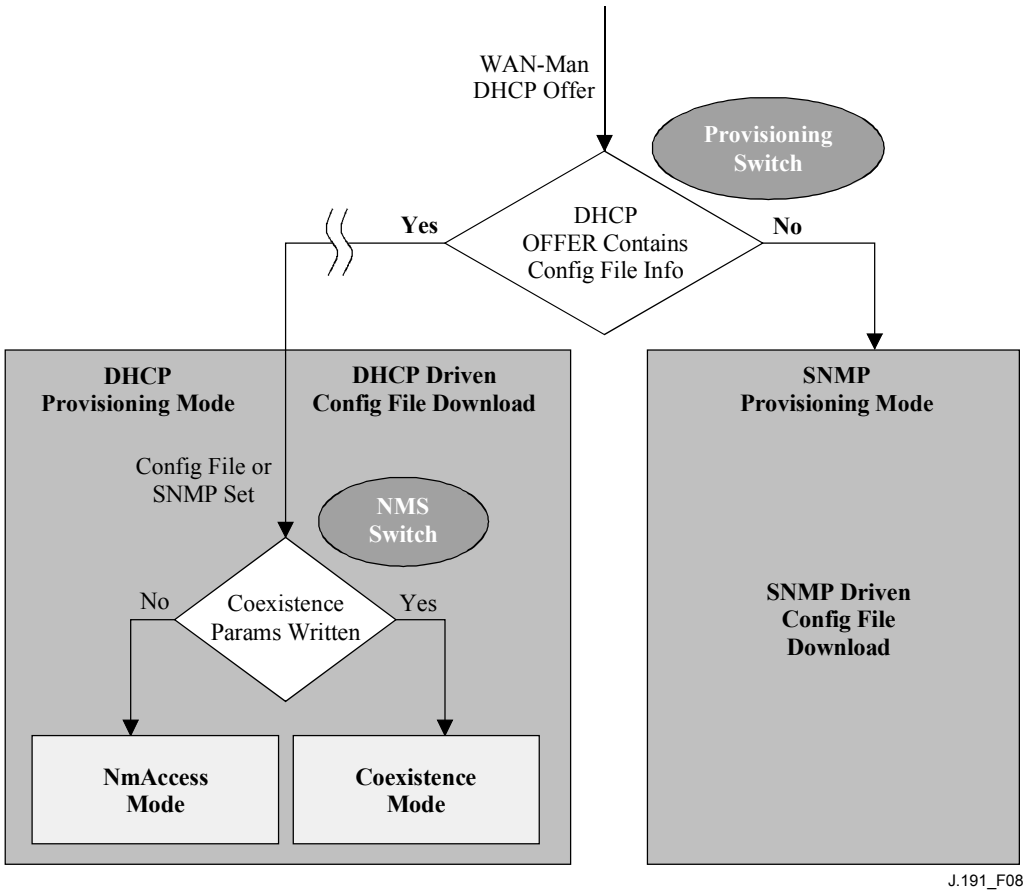


Figure 8/J.191 – Portal operational modes

If Portal Configuration File information (server location and file name) is provided to the Portal in the DHCP OFFER issued by the cable network DHCP server, the Portal will operate in DHCP Provisioning Mode. When in DHCP Provisioning Mode, the Portal may operate in one of two Network Management Modes (NmAccess and Coexistence). Within DHCP Provisioning Mode, the Portal will operate in NmAccess Network Management Mode by default, but can be configured by the NMS to operate in Coexistence Mode.

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

Table 8 describes the capabilities impacted by each operational mode described above.

Table 8/J.191 – Portal infrastructures

Mode	Capability directly effected
SNMP Provisioning Mode	Configuration file download
DHCP Provisioning Mode	Configuration file download
DHCP Provisioning Mode: NmAccess Mode	SNMP version used between NMS and PS
DHCP Provisioning Mode: Coexistence Mode	SNMP version used between NMS and PS

These various operational modes are meant to accommodate a variety of infrastructures from a back-office server perspective, including various SNMP versions, and various types of security servers. More details can be found in 13.1 to 13.3.

6 Management Tools

6.1 Introduction/overview

Management Tools provide the cable operator with functionality to monitor and configure the IPService Portal, 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:

- to provide cable operators with visibility to LAN IP Devices;
- to provide cable operators with a minimum set of remote diagnostic tools that will allow the cable operator to verify connectivity between the PS element and any LAN IP Device in the LAN-Trans address realm;
- to 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 re-configure specified capabilities as necessary;
- to 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:

- Compliant devices implement the Internet Protocol (IP) suite of protocols.
- SNMP is used for the exchange of management messages between the cable network NMS and the IPService Portal in the Cable Modem. SNMP provides visibility for the NMS to interfaces on the Portal, via access to internal Portal data, through required MIBs.
- Any of SNMPv1/v2c/v3 can be used as a management protocol between the NMS and the Portal Service.
- 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 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 Management Tools system design guidelines are listed in Table 9. This list provided guidance for the development of the management tools specifications.

Table 9/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 PS 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 9, the Management Tools architecture consist of the following components:

1) the Cable Management Portal (CMP);
2) the CableHome Testing Portal (CTP);
3) an Event Reporting mechanism within the CMP; and
4) an SNMP Network Management Server (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.

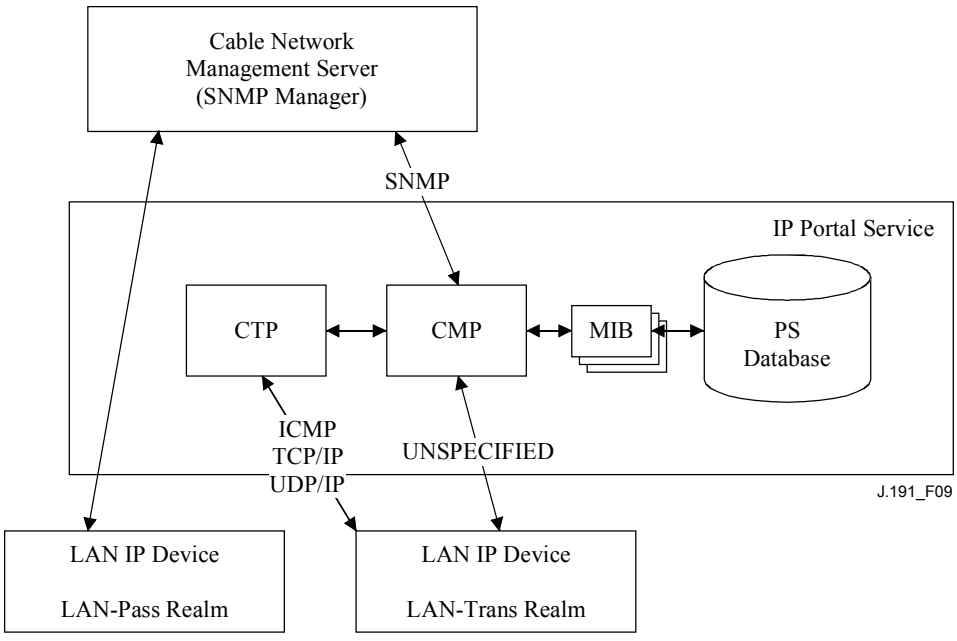


Figure 9/J.191 – Management architecture

The CMP and CTP functional elements reside within the PS.

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 a PS. 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. This does not preclude the use of a single SNMP agent.

The PS element supports SNMPv1, SNMPv2c, and SNMPv3 protocols. Clause 5.7 introduced the two provisioning modes supported by a PS 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:

- to enable viewing and updating of Cable Address Portal (CAP) configuration information;
- to enable viewing and updating of firewall configuration information;
- to enable Remote Ping for LAN IP Devices in the LAN-Trans realm, via the CableHome Testing Portal (CTP);
- to enable viewing of LAN IP Device information obtained via the Cable DHCP Portal (CDP);
- to enable viewing of the results of LAN IP Device performance monitoring done by the CableHome Testing Portal (CTP);
- to enable access to other PS configuration parameters;
- to process bulk SNMP commands passed from the cable network NMS in a PS Configuration File;
- to facilitate security by providing access to security parameters, and through the use of SNMPv1/v2c/v3 in the appropriate network management mode;
- to provide the capability to disable LAN segments.

6.3.2 CMP design guidelines

The CMP design guidelines are listed in Table 10. This list provide guidance for the specification of CMP functionality.

Table 10/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 into the home.
CMP 2	Loss of connection between broadband service provider(s) and the IP enhanced device will not disable or degrade the operation of other internal home functions
CMP 3	The PS will recover gracefully from a power outage, and devices connected to the PS must return to the operational state they were in prior to the outage.
CMP 4	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 5.7, when in SNMP provisioning mode, the PS:

1) operates using SNMPv3 protocol;

2) supports USM and VACM; and

3) uses Kerberos to distribute keying material.

As described in 5.7, when in DHCP provisioning mode, the PS can operate in either of the other two network management modes, NmAccessTable mode and Coexistence mode. In NmAccessTable mode, management access is controlled by the NmAccessTable of [RFC 2669] and the SNMPv1/v2c protocols are supported. In Coexistence mode, management access is controlled as described in [RFC 2576], the SNMPv1/v2c/v3 protocols are supported, USM and VACM are possible, and SNMPv3 keying material is distributed using [RFC 2786] and TLVs in the PS Configuration File.

Table 11 contains definitions for terms that are specific to the CMP.

Table 11/J.191 – Definition of terms

Management-control	Read or write access to a set of parameters that control or monitor the behavior of the PS.
PS database	A set of parameters that controls or monitors the behavior of the PS element readable by the WAN management system. It can be thought of as a repository of information describing the current state of the PS.
User	As defined in SNMP [RFC 2574, section 2.1], 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 [RFC 2575, section 2.4].

Table 11/J.191 – Definition of terms

Ultimate Authorization	The single authority that establishes, modifies, or deletes User IDs, authentication keys, encryption keys, and access rights to the PS database. Ultimate Authorization MAY be switched between a User in the cable network NMS and a User in the home but SHOULD NOT be both. This User is entrusted with all security management operations.
Maintenance User	A User that typically performs only read-only operations on the PS database. This is typically used for performance monitoring and accounting.
Administrator User	A User that typically performs both read and write operations on the PS database. These operations are used for Configuration and Fault Management.

Examples of the types of information manipulated via Cable 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 10 indicates three possible 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);
- NMS – LAN IP Device: management message exchange between the cable network NMS and LAN IP Devices in the LAN-Pass realm (not specified);
- 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.

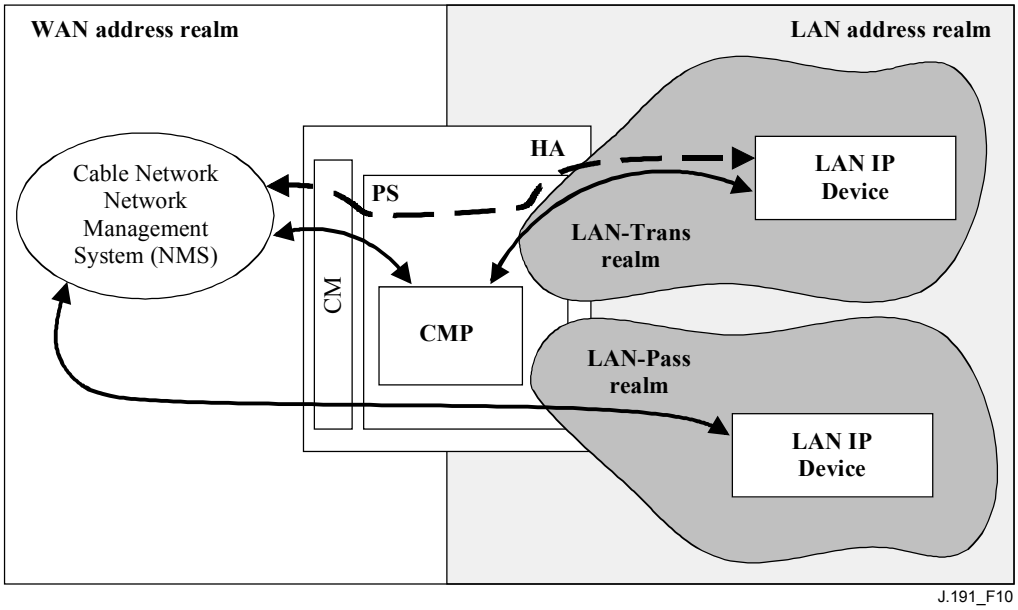


Figure 10/J.191 – Management message interfaces

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 11 to convey concepts for CMP functionality.

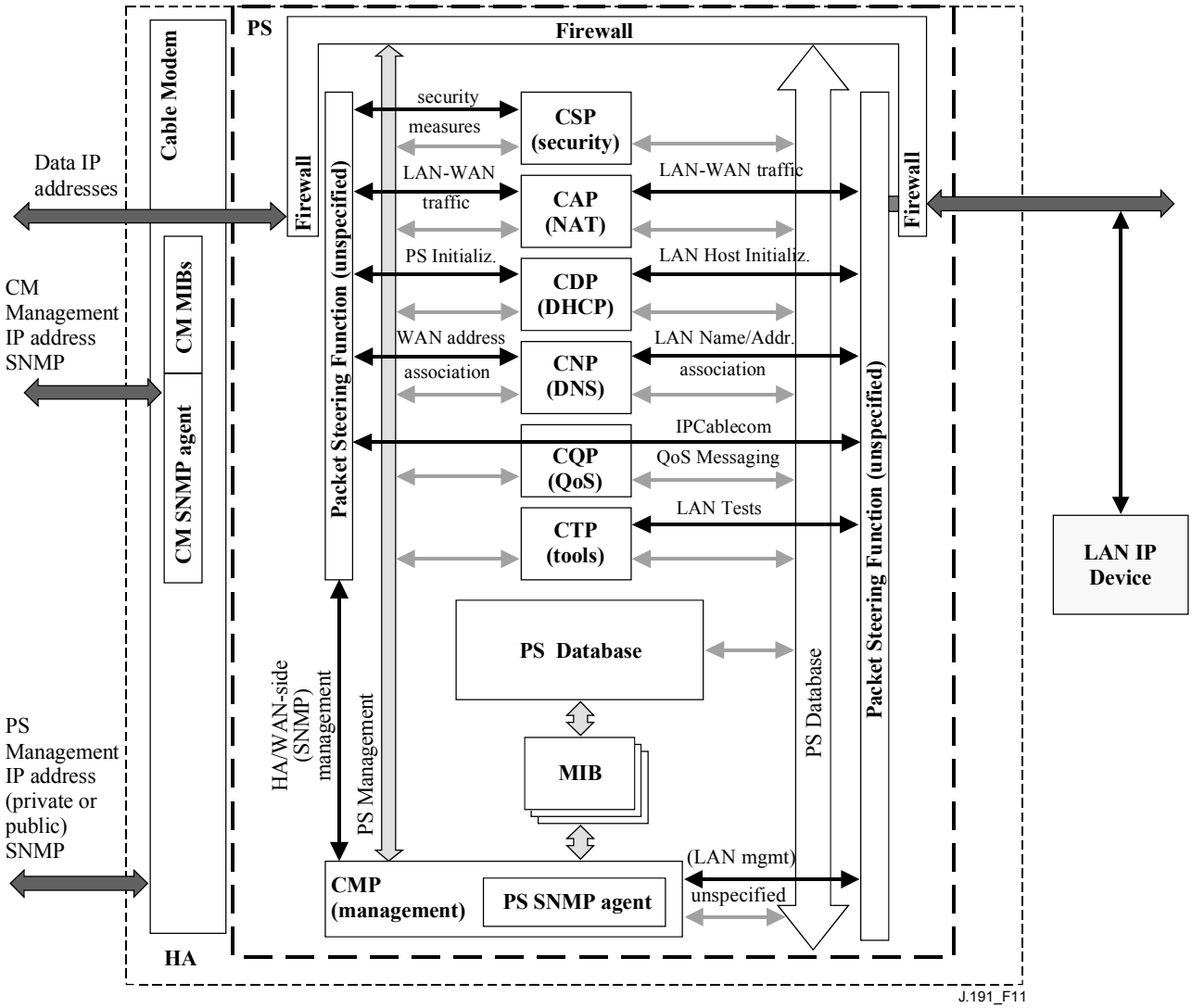


Figure 11/J.191 – PS block diagram

The NMS management tools use SNMP to access and manage objects in the PS. 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 PS SNMP agent 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 CMP MUST provide Management-control to the WAN through SNMP v3 [RFC 2571, RFC 2572].

The CMP MUST implement ICMP [RFC 792] and reply to ICMP Echo Requests from the NMS.

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.

The CMP MUST be able to grant Ultimate Authorization to either the LAN Administrator or the Cable WAN Administrator (PS Administrator).

The default Ultimate Authorization setting MUST be WAN Administrator. The Ultimate Authorization setting MAY be overwritten via SNMP access or a configuration file.

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

The sysDescr object of the MIB-2 System group (MIB-2 1) [RFC 1907] MUST be implemented and MUST persist across device resets and power cycles.

The sysDescr MUST contain five fields in the specific order as follows: HW_REV: hardware_version; VENDOR: vendor_name; BOOTR: Boot_ROM_version; SW_REV: Software_version; Model: Model_number.

The sysDesr is composed of a list of five Type/Value pairs. 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. The required five pairs of the SysDescr MUST be enclosed in double angle brackets. For example, a sysDescr for PS of vendor XYZ, hardware version 5.2, Boot ROM version 1.4, software (SW) version 2.2, and model number ABC MUST appear as follows:

> any text «HW_REV: 5.2; VENDOR: XYZ; BOOTR: 1.4; SW_REV: 2.2; MODEL: ABC» any text

The PS needs to report, through sysDescr fields, all of the information necessary to determine what SW the PS is capable of being upgraded to. If any of the required sysDescr fields are not applicable, the SysDescr MUST report "NONE" as the value. For example, a PS with no BOOTR will report BOOTR: NONE.

The sysObjectID object of the MIB-2 System group [RFC 1907] MUST be implemented and MUST be persistent across device reset and power cycles.

The sysUpTime object of the MIB-2 System group [RFC 1907] 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 1907] 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 1907] MUST be implemented and MUST be persistent across device reset and power cycles.

The sysServices object of the MIB-2 System group [RFC 1907] 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 1907] 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.

The MIB-2 SNMP group [RFC 1907] MUST be implemented.

The snmpSetSerialNo object of the snmpSet group [RFC 1907] 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.

6.3.5 SNMP protocol requirements

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

- A Simple Network Management Protocol [RFC 1157];
- Introduction to Community-based SNMPv2 [RFC 1901];
- Protocol Operations for SNMPv2 [RFC 1905];
- Transport Mappings for SNMPv2 [RFC 1906];
- Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) [RFC 1907];
- Introduction to SNMPv3 [RFC 2570];
- SNMP FrameWork MIB [RFC 2571];
- Message Processing and Dispatching for SNMP [RFC 2572];
- SNMP Applications MIB [RFC 2573];
- SnmpUSM MIB Group [RFC 2574];
- SnmpVACM MIB Group [RFC 2575];
- SNMP Community MIB [RFC 2576];
- SNMPv2-CONF.

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

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

6.3.6 Network management mode requirements

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 NmAccessTable mode and Coexistence mode for a PS operating in DHCP provisioning mode

The PS MUST support SNMPv1, SNMPv2c, and SNMPv3 and SNMP Coexistence as described by [RFC 2571] through [RFC 2576]. The PS MUST also support NmAccessTable mode as defined by [RFC 2669]. Support for the network management modes for a PS operating in DHCP Provisioning Mode is subject to the following guidelines:

6.3.6.1.1 Basic operation for a PS operating in DHCP provisioning mode

a) Following receipt of DHCP ACK, the PS operating in DHCP Provisioning Mode (indicated by a cabhPsDevProvMode value of '1' (DHCPmode)) MUST operate as follows:

- SNMPv1/v2c read-only Access to all MIB variables, which are required to be in view during SNMPv1/v2c operation, is allowed from the LAN. No access is allowed from the WAN, to prevent unauthorized management access before the PS is configured via the PS Configuration File.
- SNMPv1/v2c packets are accepted which contain any community string.
- All SNMPv3 packets are dropped.
- Access SHOULD be prohibited to any MIB variable that would allow determination of the PS WAN-Man IP address, like the MIB-2 IpAddrTable.
- None of the SNMPv3 MIBs (Community MIB, TARGET-MIB, VACM-MIB, USM-MIB, NOTIFICATION-MIB) are accessible, except that they may be set from the PS Configuration File.
- None of the elements in the SNMP-USM-DH-OBJECTS-MIB is accessible except that they may be set from the PS Configuration File.
- Successful processing of all MIB elements in the PS Configuration File MUST be completed before beginning the calculation of the public values in the USMDHKickstart Table.

b) If a PS is operating in DHCP Provisioning Mode, the content of the PS Configuration File determines the network management mode, as described below:

- The PS is in SNMPv1/v2c docsDevNmAccess mode if the PS Configuration File contains ONLY docsDevNmAccess Table setting for SNMP access control.
- If the PS Configuration File does not contain SNMP access control items (docsDevNmAccessTable or snmpCommunityTable or TLV 34.1/34.2 or TLV38), then the PS is in NmAccess mode.
- If the PS Configuration File contains snmpCommunityTable setting and/or TLV type 34.1 and 34.2 and/or TLV type 38, then the PS is in SNMP Coexistence Mode. In this case, any entries made to the docsDevNmAccessTable are ignored.

c) 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.

NmAccess Mode (using docsDevNmAccess Table) using SNMPv1/v2c:

- Only SNMPv1/v2c packets are processed.
- SNMPv3 packets are dropped.
- docsDevNmAccessTable controls access and trap destinations as described in [RFC 2669].
- None of the SNMPv3 MIBs (Community MIB, TARGET-MIB, VACM-MIB, USM-MIB, NOTIFICATION-MIB) is accessible.

Coexistence Mode using SNMPv1/v2c/v3.

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 2571] and [RFC 2576].
- docsDevNmAccessTable is not accessible.
- Access control and trap destinations are determined by the snmpCommunityTable, NOTIFICATION-MIB, TARGET-MIB, VACM-MIB, and USM-MIB.
- Community MIB controls the translation of SNMPv1/v2c packet community string into security name which select entries in the USM-MIB. Access control is provided by the VACM-MIB.
- USM-MIB and VACM-MIB controls SNMPv3 packets.
- Trap destinations are specified in the TARGET-MIB and NOTIFICATION-MIB.

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

6.3.6.1.2 Coexistence Mode SNMPv3 initialization and key changes

When in Coexistence Mode, the PS MUST support the "SNMPv3 Initialization" and "DH Key Changes" requirements specified in the following clauses.

6.3.6.1.2.1 SNMPv3 initialization

For each of up to 5 different security names, the PS Administrator generates a pair of numbers. First, the PS Administrator generates a random number Rm.

Then, the IPCable2Home Administrator uses the DH equation to translate Rm to a public number z. The equation is as follows:

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

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 6.3.6.4. Only VACM entries specified by the corresponding security name in the PS Configuration File will (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).

$$c = (g \wedge xa) \text{ MOD } p$$

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 = "docsisManager".

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.

6.3.6.1.2.2 Diffie-Hellman Key Changes

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

6.3.6.2 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 the network management mode using SNMPv3, USM and VACM, and Kerberos for exchanging key material (as described in 6.3.3) following rules described in this clause.

6.3.6.2.1 Management Views

The management controls are in the PS element. Settings, based on management mode, define the access rights that are granted to a command generator for access to the PS database, through specified MIBs, via SNMP from the cable network NMS. A single command generator is defined by the specification.

Figure 12 illustrates some example Management Views using SNMPv3. A WAN Administrator View (PS Administrator view) and a WAN Administrator User (PS Administrator user) are defined. 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 2574] and [RFC 2575].

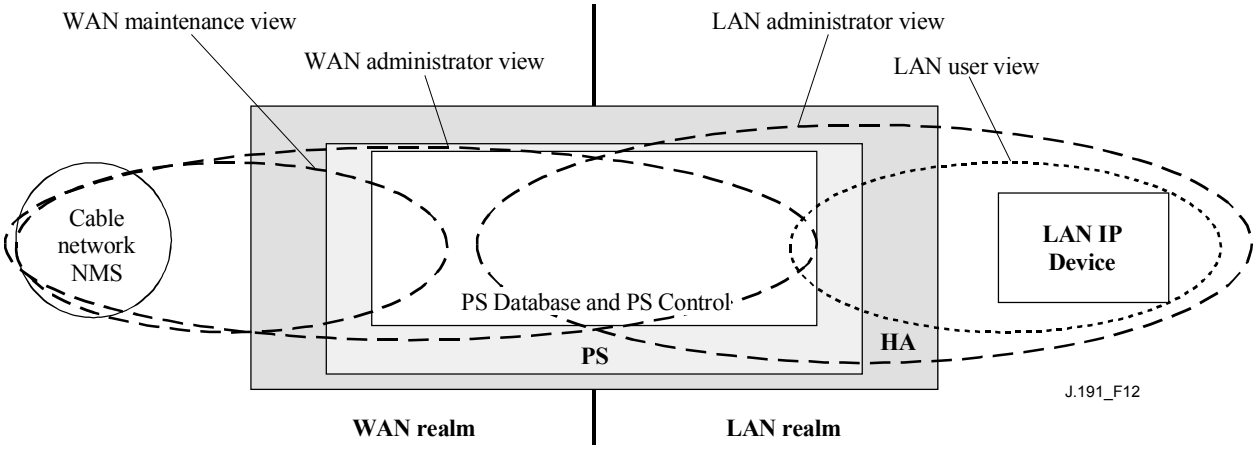


Figure 12/J.191 – Management Views

Managed parameters are stored in the PS database. As shown in Figure 12, 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 the following responsibilities:

- for setting up all access Views on both the LAN and WAN management interface.
- for own user ID and Keys.
- for creating and managing all User profiles including user IDs, Keys, and PS database access privileges.
- 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 6.3.6.4 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 6.3.6.4.

6.3.6.2.2 WAN-Access Control

SNMP Access Control, per [RFC 2575], will be used for the WAN side Views. The View-based Access Control Model (VACM) [RFC 2575] 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 2575] section 2.4, a "MIB View" is a specific set of managed object types that can be defined, and this concept is used to support WAN Management of the PS. The PS Administrator User access and View are specified in 11.3.3.2.2 and in 6.3.6.4. 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 SNMP v3 to counter threats identified in Annex C.

To protect against replay attacks, a real-time clock is utilized to provide timestamps for messaging. Management messaging security requirements are specified in 11.3.3.

6.3.6.3 Security requirements

Management messaging security requirements are specified in 11.3.3.

6.3.6.4 View-based Access Control Model (VACM) requirements

To provide controlled access to management information and the creation of distinct management realms View-based Access Control Model (VACM) MUST be employed as defined by [RFC 2575].

The WAN Administrator View MUST be implemented in the PS element. Default Views other than the WAN Administrator View MUST NOT be available on the PS. Other Views MAY be created by 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).

6.3.7 MIB requirements

MIB objects listed in Annex A MUST be implemented in a PS element. Required MIB objects are from the following MIB documents:

Interfaces Group MIB [RFC 2863]

1) DOCSIS Cable Device MIB [RFC 2669];
2) Cable CLAB DEF MIB [Annex E.4];
3) Cable PSDev MIB [Annex E.1];
4) Cable CAP MIB [Annex E.6];
5) Cable CDP MIB [Annex E.5];
6) Cable CTP MIB [Annex E.2];
7) Cable Security MIB [Annex E.3];
8) draft-ietf-ipcdn-bpiplus-mib-06.txt;
9) IP MIB (SNMPv2) [RFC 2011];
10) UDP MIB (SNMPv2) [RFC 2013];
11) Diffie-Hellman USM Key [RFC 2786];
12) INET Address MIB [RFC 2851];
13) DOCS IF MIB [RFC 2670];
14) IANA ifType MIB.

With the exception of the SNMP group of MIB 2, USM MIB, and VACM MIB, which are directly accessed by the SNMP agent in the PS (CMP), AND the DOCSIS Cable Device MIB for the case of software download to an PS, the PS MUST maintain separate instances of PS-specified MIBs from the cable modem. Information accessed from the PS database through the PS WAN-Man address MUST be separate and distinct from information accessed via the CM management address.

The general MIB hierarchy is illustrated in Figure 13. Specific OIDs required for individual MIBs are listed in Annex A.

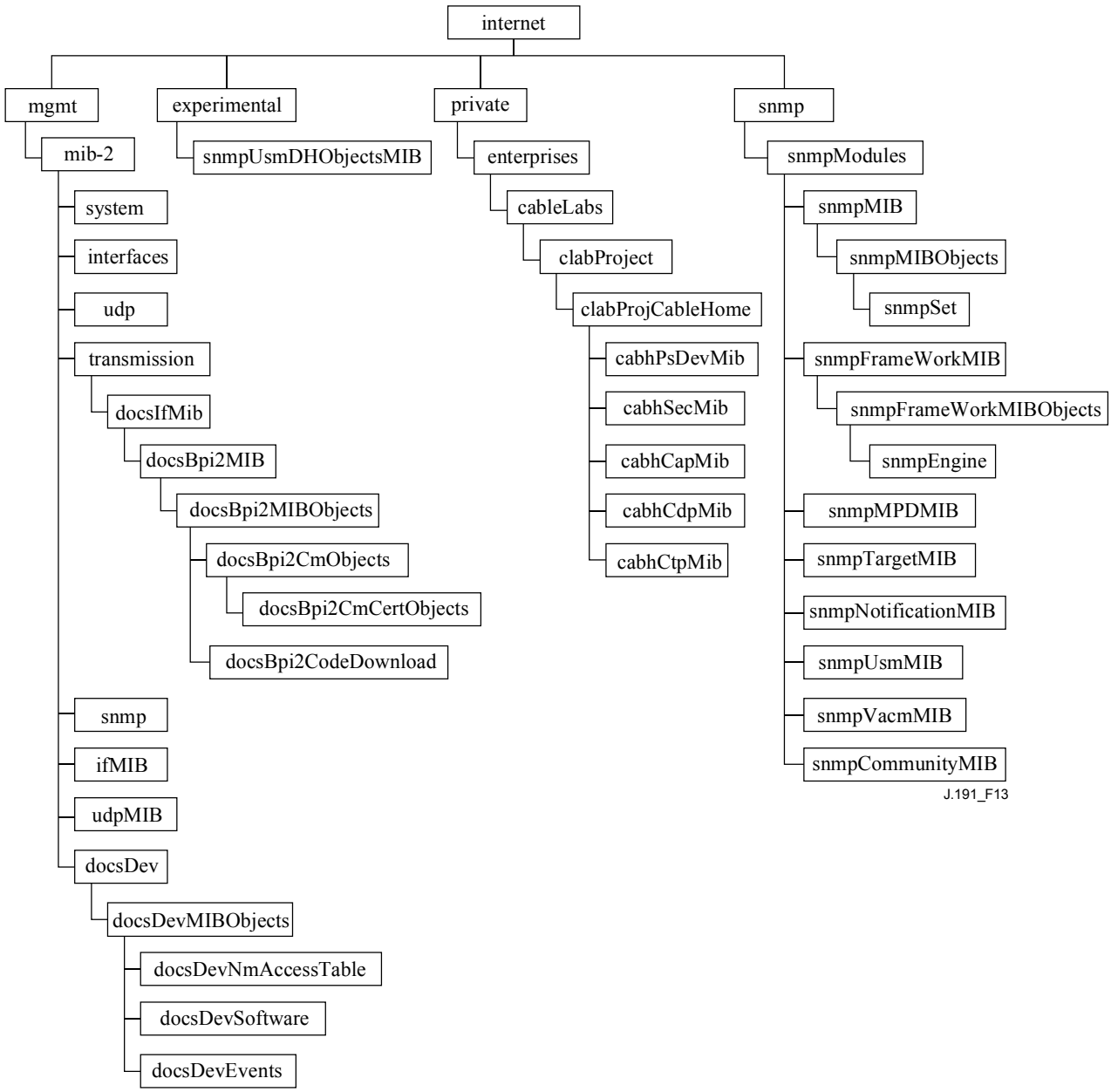


Figure 13/J.191 – MIB hierarchy

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 interface of the PS element, even if that WAN interface is internal – as exists in the case of an Embedded PS utilizing an integrated chip design.

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

The interfaces MUST be numbered as shown in Table 12.

Table 12/J.191 – Numbering interfaces in the ifTable

Interface	Description
1	WAN interface
1 + n	Each LAN interface

If a given interface's ifAdminStatus = down, that interface MUST NOT accept or forward any traffic.

6.3.9 CMP Configuration File processing requirements

The CMP is the functional entity in the PS responsible for processing parameters passed in PS Configuration Files. PS Configuration Files are used for reconfiguration of the PS by providing values for manageable parameters in the PS database.

The received PS Configuration File is first checked for integrity and authenticated, as described in 11.3.7. Then, the TLV tuples in the PS Configuration Files are analyzed, and the SNMP object identifiers and their parameters are extracted. The CMP 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 CMP MUST unconditionally update the objects corresponding to recognized OIDs.

Configuration settings MUST be processed in the same order that they appear in the PS Configuration File. The CMP MUST be capable of accepting a series of TLV parameters contained in a PS Configuration File. There is no preconceived state of the PS when a PS Configuration File is received. The process of loading and executing a PS Configuration File may interrupt data processing in the PS. The CMP MUST disregard any configuration setting for which no valid database parameter exists.

For SNMP sets in the PS Configuration File, the PS MUST treat all SNMP variable bindings (Varibinds) in the PS Configuration File as if they were received in a single SNMP PDU. If duplicate Varibinds are received in the PS Configuration File, then the PS MUST stop the provisioning process.

The objects defined by TLVs that are passed in the PS Configuration File and are not supported or cannot be written in the particular PS implementation, MUST be ignored. The CMP MUST disregard any unknown TLV.

The size of the PS Configuration File, the number of TLVs processed and the number of TLVs ignored MUST be updated in the MIB objects: cabhPsDevProvConfigFileSize, cabhPsDevProvConfigTLVProcessed and cabhPsDevProvConfigTLVRejected, respectively.

PS Configuration File requirements are specified in 7.3.

6.4 The CableHome Testing Portal (CTP)

6.4.1 CTP goals

The goals for the CableHome Testing 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.

6.4.2 CTP design guidelines

The Management Tools system design guidelines are listed in Table 13. A number of these guidelines are common with the CMP design guidelines. This list provided guidance for the specification of CTP functionality.

Table 13/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 into the home.
CTP 2	Local and remote monitoring capabilities are needed that can monitor home 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 in the home.
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 (CableHome Testing Portal) contains the "remote tools" with which the NMS management 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: connectivity and throughput diagnostics.

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

6.4.3.1 CTP connection speed tool

This function is used to get a rough measure of the 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 delay 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 source IP address is always that of the PS LAN default gateway (cabhCdpServerRouter). This test feature only works on LAN IP Devices in the LAN Trans address realm.

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.

Clause 6.4.4 lists the parameters and responses for the Ping Tool. Note that the time for the request reply is not stored, since typical frame propagation time in the home may be faster than standard time units (ms) can accurately measure. For performance metering, the Connection Speed Tool should be used.

Clause 12.2.1.2 details the operation of the Ping Tool.

6.4.4 CTP requirements

The CTP MUST implement the Connection Speed Tool with parameters listed below, where the angle brackets indicate the CTP MIB object. Numbers in square brackets are the options or the lower and upper bounds of the parameter range, and the number in parenthesis is the default value:

– <cabhCtpConnSrcIp> (equal to the value of cabhCdpServerRouter) – the LAN IP address used as the source of the Connection Speed Tool;

– <cabhCtpConnDestIp> – the LAN IP address used as the destination of the Connection Speed Tool;

NOTE 1 – May be set to any valid IPv4 address, to find LAN IP Devices in the LAN-Trans address realm.

– <cabhCtpConnProto> [UDP (1), TCP (2)] (UDP) – the protocol used for the Connection Speed Tool;

– <cabhCtpConnPort> [1 to 65535] (7) – the port used for the Connection Speed Tool.

NOTE 2 – IANA reserves port 7 for this use. Other ports may be useful.

– <cabhCtpConnNumPkts> [1 to 255] (1) – the number of packets to send for the Connection Speed Tool;

– <cabhCtpConnPktSize> [–64 to 1518] (64) – the size of the test frames for the Connection Speed Tool in bytes;

– <cabhCtpConnTimeOut> [0 to 600000] (600000) – the time-out value, in milliseconds, for the response to the Connection Speed Tool.

NOTE 3 – A value of zero indicates no time-out and can be used for TCP only.

– <cabhCtpConnControl> [notRun (1), start (2), abort (3)] – control for the Connection Speed test;

– <cabhCtpConnStatus> [running (1), complete (2), aborted (3)] – status of the Connection Speed test;

– <cabhCtpConnPktsSent> [1 to 255] – the number of packets sent during the Connection Speed test;

– <cabhCtpConnPktsRecv> [0 to 255] – the number of packets received during the Connection Speed test.

NOTE 4 – This value allows the operator to determine whether the time-out was achieved (PktsSent > PktsRecv) due to packet loss, assuming the time-out was properly calculated. This pair of parameters was included to support detection of UDP packet loss. Under normal operation, PktsRecv is equal to PktsSent.

– <cabhCtpConnAvgRTT> [0 to 600000] – the resulting average of round-trip time for acknowledged packets in milliseconds;
– <cabhCtpConnMaxRTT> [0 to 600000] – the resulting maximum of round-trip times for acknowledged packets in milliseconds;
– <cabhCtpConnMinRTT> [0 to 600000] – the resulting minimum of round-trip times for acknowledged packets in milliseconds;
– <cabhCtpConnNumIcmpError> [0 to 255] – the number of ICMP errors.

NOTE 5 – The value may include net or host "prohibited" or "unreachable". This parameter is null by default, or when no errors occur.

– <cabhCtpConnIcmpError> [0 to 255] – the last ICMP error.

The CTP MUST implement the CTP Ping Tool with the parameters listed below, where the angle brackets indicate the CTP MIB object, numbers in brackets are the lower and upper bounds of the parameter range, and the number in parenthesis is the default value:

– <cabhCtpPingSrcIp> (equal to the value of cabhCdpServerRouter) – the LAN IP address used as the source of the Remote Ping Tool;
– <cabhCtpPingDestIp> – the LAN IP address used as the destination of the Remote Ping Tool;
– <cabhCtpPingProto> [icmp (1)] (icmp) – the protocol used for the Remote Ping Tool;
– <cabhCtpPingNumPkts> [1 to 4] (1) – the number of packets to send to each host for the Remote Ping test;
– <cabhCtpPingPktSize> [–64 to 1518] (64) – the size of the test frames for the Remote Ping test in bytes;
– <cabhCtpPingTimeBetween> [0 to 600000] (1000) – the time between sending one packet and the next during the Remote Ping test in milliseconds;
– <cabhCtpPingTimeOut> [0 to 600000] (5000) – the time-out for response of sending a single ping during the Remote Ping test in milliseconds;
– <cabhCtpPingControl> [notRun(1), start (2), abort (3)] – control for the Remote Ping test;
– <cabhCtpPingStatus> [running (1), complete (2), aborted (3)] – status of the Remote Ping test;
– <cabhCtpPingNumSent> [0 to 254] – the number of pings sent during the Remote Ping test;
– <cabhCtpPingNumRecv> [0 to 254] – the number of pings received during the Remote Ping test.

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 in 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 a pre-defined SYSLOG server.

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-nonvolatile(0);
- 2: traps(1);
- 3: syslog(2);
- 4: local-volatile(3);
- 5: inform(4).

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

Event descriptions MUST 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-nonvolatile necessitates a method for synchronizing docsDevEvIndex values between the two types of events after a PS reboot. After a PS reboot, to synchronize the docsDevEvIndex values for volatile and non-volatile events, the following procedure MUST be used:

- The values of docsDevEvIndex for local log events tagged as local-nonvolatile MUST be renumbered beginning with 1;
- The local log MUST then be initialized with the events tagged as local-nonvolatile in the same order as they had been immediately prior to the reboot;

- Subsequent events recorded in the local log, whether tagged as local-volatile or local-nonvolatile, MUST use incrementing values of docsDevEvIndex.

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

6.5.1.2 SNMP TRAP and INFORM

The PS MUST support the SNMP Trap PDU as described in [RFC 2571]. The PS MUST support the SNMP INFORM PDU as described in [RFC 2571]. 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 "CABLE" for the standard Cable 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 Cable events, this number is converted from the error code using the following rules:

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

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

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

text – for the standard Cable 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>PS Element[CABLE]: <68000402> Time of the day received in invalid format.

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

6.5.2 Format of Events

The 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. Standard management events are defined in Annex B. Each management event described in the annex is assigned an Event ID;
- Event Text – describes the event in human readable form;
- MAC Address – describes the MAC address of the device.

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 subclause.

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 document utilize these priority levels.

1) 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.

2) 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.

3) 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.

4) 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.

5) Warning event (priority 5)

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

6) 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'.

7) 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.

8) Debug event (priority 8)

Reserved for vendor-specific non-critical events.

The priority associated with the standard events MUST NOT be changed.

Table 14 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 nonvolatile entries.

Table 14/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	No	Yes	Yes	Yes	Standard
5) Warning	No	No	No	Yes	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

Table 15 shows the minimum level of support required for notification types for the various event priorities. For example, the PS has to minimally support nonvolatile entries in the local log for event priorities of emergency, alert, and critical. The PS MUST support the minimum requirements for implementing event priorities for each type of event reporting. The PS MAY choose to report an event priority with more notification types than required in Table 15.

Table 15/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

Table 15/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
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 1907] and [RFC 2863]:

- coldStart [RFC 1907];
- linkUp [RFC 2863];
- linkDown [RFC 2863];
- SNMP authentication-Failure [RFC 1907].

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.

7 Provisioning tools

7.1 Introduction/overview

The PS 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. 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 PS element;
- Bulk PS Configuration (BPSC) tool;
- Time of Day Client in the PS 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 16.

Table 16/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 behavior of the Provisioning Tools is dependent upon the Provisioning Mode in which the PS operates.

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

7.1.2 Provisioning architecture

The provisioning architecture is illustrated in Figure 14. PS 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.

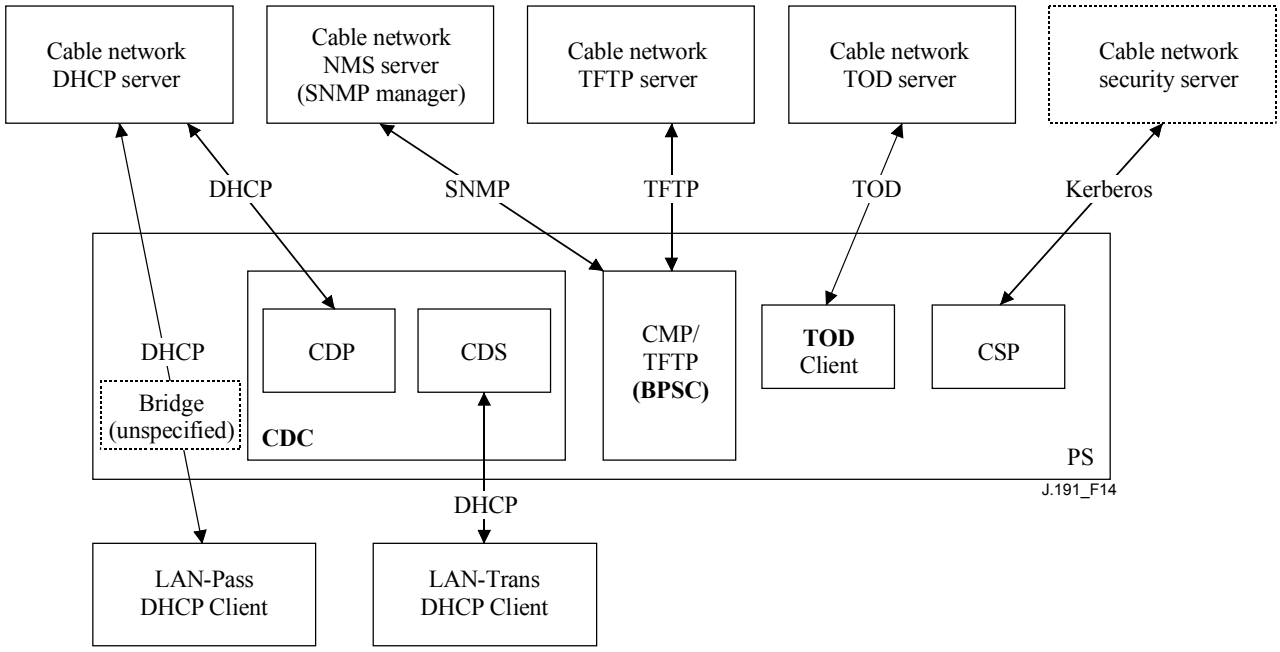


Figure 14/J.191 – Provisioning architecture

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 PS 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 Cable 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 in Table 17 drive the capabilities defined for the CDP:

Table 17/J.191 – CDP system design guidelines

Number	CDP system design guidelines
CDP 1	Addressing mechanisms will be operator controlled, and will provide the cable operator knowledge of and accessibility to the **Portal Service** 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.

Table 17/J.191 – CDP system design guidelines

Number	CDP system design guidelines
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 addressing activities. The CDP address request and address allocation responsibilities 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 PS element;
- providing information to the Cable Naming Portal (CNP) in support of LAN IP Device host name services.

The PS element requires an IP Address for its role as a router of traffic in the home (see clause 8, Packet Handling and Address Translation), DHCP server (CDS), and DNS server (see clause 9, Name Resolution). For each of these three Portal Service Element server and router functions, a LAN IP address is saved in the PS database. Each can be accessed via a different MIB object, which are listed below and in Table 17.

– Router (default gateway) Address cabhCdpServerRouter

– Domain Name System (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 15, 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 15 also illustrates 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.

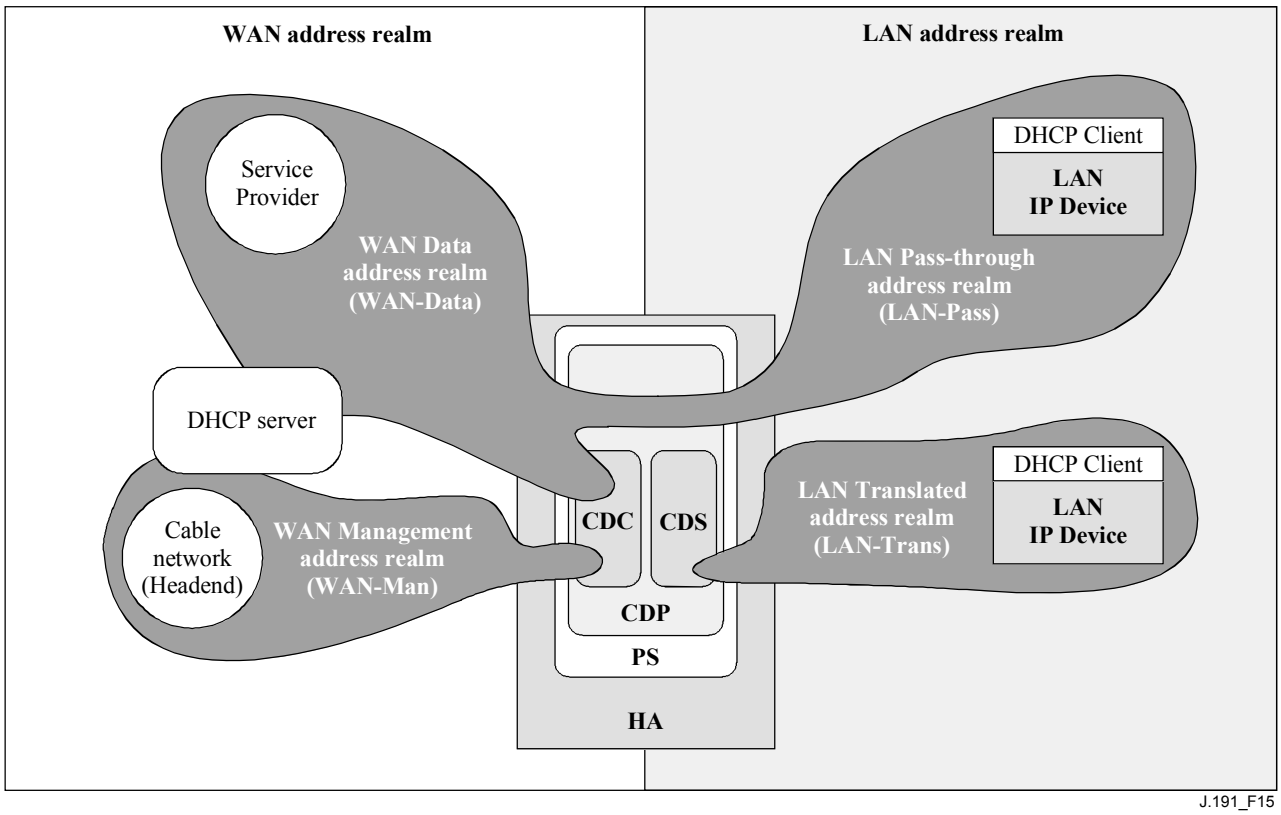


Figure 15/J.191 – CDP functions

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 behavior of the CDS when a cable operator settable limit is exceeded is also configurable via the NMS. Possible CDS actions when the limit is exceeded include:
  1) assign a LAN-Trans IP address and treat the WAN to LAN CAT interconnection as would normally occur if the limit had not been exceeded; and
  2) do not assign an address to requesting LAN IP devices.
- In the absence of time of day information from the Time of Day (TOD) server, the CDS uses the PS default starting time of 0 (January 1, 1900), 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.
- Upon PS re-boot or reset, the CDS remains inactive until activated by the PS after successful download of the PS Configuration File or after 5 unsuccessful attempts by the PS to download the PS Configuration File, whichever occurs first. The CDS is thereby prevented from granting DHCP leases in the LAN-Trans realm until there has been a

reasonable opportunity for the cable operator to update LAN-Trans lease parameters such as cabhCdpServerLeaseTime, cabhCdpLanPoolStart, and cabhCdpLanPoolEnd.

- If the PS Primary Packet-handling mode (cabhCapPrimaryMode) has been set to Pass-through, then the CDS is disabled.

LAN IP Devices may receive addresses that reside in the LAN-Pass realm. As shown in Figure 15, 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 "Automatic Allocation", "Dynamic Allocation", and "Manual Allocation" are used as defined in [RFC 2131]. Automatic Allocation of IP addresses within a LAN-Trans address realm will be permanent, and the CDS may reuse Automatic addresses if all available addresses have been allocated. The **CDS Provisioned DHCP Options**, cabhCdpServer objects in the CDP MIB, are used by the CDS to indicate the DHCP options offered 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 17) 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 contains 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 Head-end 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 automatically (via the CDP) (cabhCdpLanAddrConfig).

The CDS uses the MAC address to identify LAN IP Devices.

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], as described in 7.2.2.2.1. Cable DHCP Client options 60 and 43.

The Vendor Class Identifier Option (DHCP option 60) defines a specific device class. The Vendor Class Identifier Option will contain a specific string to identify a 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, the components that are contained in the device (CM, MTA, PS, etc.), the device serial number, and also allows device specific parameters.

Details of the requirements for supporting DHCP options 60 and 43 are in Tables 19 and 20.

7.2.2.2.1 Cable DHCP client option 61

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

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

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 1: The PS element makes use of a single WAN IP Address. The PS element has one WAN-Man and one WAN-Data IP Interface, which share a common MAC Address. These two Interfaces share a single, common IP address. This is the factory default configuration of the PS element. The cable operator's Headend DHCP server typically needs no software modifications to support this provisioning mode.

WAN Address Mode 2: The PS element makes use of two or more different WAN IP Addresses. The PS element could have one WAN-Man and one or more WAN-Data IP Interface(s), which share a common MAC Address. These two or more Interfaces would each have their own, unshared IP address. The cable operator's Headend DHCP server may need software modification to support assigning multiple IP Addresses to a single MAC Address. In this mode, Headend DHCP server will need to support IP assignment based on Client ID (option 61), as well as MAC Address.

WAN Address Mode 2 is triggered by writing a unique Client ID string into the cabhCdpWanDataAddrClientId entry of the CDP MIBs cabhCdpWanDataAddrTable, for each WAN-Data interface to be used. To support this provisioning mode, the cable operator will need to provide (via NMS, config file, or manual customer entry via a proprietary interface) the PS element with a unique Client ID string for each WAN-Data IP Interface.

7.2.3 Cable DHCP portal requirements

7.2.3.1 CDP requirements

CDP Manual address allocation MUST be supported using CDP Table (cabhCdpLanAddrTable) entries created via the NMS system or config file.

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, independent of the WAN connectivity state.

7.2.3.2 CDS requirements

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

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

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, following successful download of the PS Configuration File or following 5 successive unsuccessful PS Configuration File download attempts, whichever occurs first.

The CDS MUST assign addresses and deliver DHCP configuration parameters only to LAN IP Devices receiving an address in the LAN-Trans address realm.

Automatic addresses allocation within a LAN-Trans address realm MUST be permanent and the CDS MAY reuse Automatic addresses if all available addresses have been allocated.

The CDS MUST use the hardware (MAC) address of LAN IP Devices as their client identifier value.

The CDS MUST support the CDP MIB 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 18.

Table 18/J.191 – CDS DHCP options

Option number	Option function	CDS protocol support (M)andatory or (O)ptional	CDS Mgmt Support (M)andatory or (O)ptional	CDS factory defaults	CDS cable outage retention (M)andatory	CDS power outage persistent (M)andatory	MIB object name
0	Pad	M	–	N/A	N/A	N/A	N/A
255	End	M	M	N/A	N/A	N/A	N/A
1	Subnet Mask	M	M	255.255.255.0	M	M	cabhCdpServer SubnetMask
2	Time Offset	M	O	O	N/A	N/A	cabhCdpServer TimeOffset
3	Router Option	M	M	192.168.0.1	M	M	cabhCdpServer Router
6	Domain Name Server	M	M	192.168.0.1	M	M	cabhCdpServer DnsAddress
7	Log Server	M	M	0.0.0.0	M	M	cabhCdpServer SyslogAddress
12	Host Name	M	O	N/A	N/A	N/A	N/A
15	Domain Name	M	M	Null String	M	M	cabhCdpServer DomainName
23	Default Time-to-live	M	M	255	M	M	cabhCdpServer TTL
26	Interface MTU	M	M	1520	M	M	cabhCdpServer InterfaceMTU
43	Vendor-Specific Information	M	M	Vendor Selected	M	M	cabhCdpServer VendorSpecific
50	Requested IP Address	M	N/A	N/A	N/A	N/A	N/A
51	IP Address Lease Time	M	M	60	M	M	cabhCdpServer LeaseTime
54	Server Identifier	M	M	192.168.0.1	M	M	cabhCdpServer DhcpAddress
55	Parameter Request List	M	N/A	N/A	N/A	N/A	N/A
60	Vendor Class Identifier	M	N/A	N/A	N/A	N/A	N/A
61	Client-identifier	M	N/A	N/A	N/A	N/A	N/A

The CDS MUST support NMS provisioning of the options indicated as Mandatory in the CDS Mgmt Support column of Table 18.

The CDS DHCP options indicated as Mandatory in the CDS Cable Outage Retention column of Table 18 MUST be retained during a cable service outage.

The CDS DHCP options indicated as Mandatory in the CDS Power Outage Persistent column of Table 18 MUST Persist after a CDP power cycle.

The CDS MUST support offering the default values indicated in the CDS Factory Defaults column of Table 18, if the DHCP option has not been provisioned.

If the PS Primary Packet-handling mode (cabhCapPrimaryMode) has been set to Pass-through, then the CDS MUST be disabled.

In support of Automatic address allocation, the CDS MUST be capable of creating, modifying and deleting CDP Table entries for devices allocated a LAN-Trans address.

The CDS MUST maintain the Address Count parameter (cabhCdpLanTransCurCount) indicating the number of LAN-Trans address assigned 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. 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 addresses as normal or do not assign an address.

The specific action taken by the CDS MUST be as indicated by the Threshold Exceeded Action (cabhCdpLanTransAction) provisioned parameter.

7.2.3.3 CDC requirements

The CDC behavior MUST be in accordance with the Client requirements of [RFC 2131].

The PS element MUST have a WAN interface hardware address that is distinct from the cable modem.

If the CDC receives, in the DHCP response [RFC 2131] from the DHCP server in the cable network, a valid IP address in the 'siaddr' field AND a valid file name in the 'file' field AND does not receive DHCP option 177 sub-option 51, the PS MUST set cabhPsDevProvMode to '1' (DHCP Mode).

If the CDC receives, from the DHCP server in the cable network, a valid IP address for DHCP option 177 sub-option 51 AND does not receive a valid IP address in the 'siaddr' field AND does not receive a valid file name in the 'file' field, the PS MUST set cabhPsDevProvMode to '2' (SNMP Mode).

If the CDC receives, in the DHCP message [RFC 2131] from the DHCP server in the cable network, DHCP option 177 sub-option 51 AND a valid IP address in the 'siaddr' field, OR if the CDC receives DHCP option 177 sub-option 51 AND a valid file name in the 'file' field, the PS MUST log an error in the local log and re-broadcast a DHCP DISCOVER message (i.e., restart the provisioning sequence in the event of this invalid condition).

If the CDC does not receive DHCP option 177 sub-option 51 AND does not receive a valid IP address in the 'siaddr' field AND does not receive a valid file name in the 'file' field, the PS MUST log an error in the local log and re-broadcast a DHCP DISCOVER message (i.e., restart the provisioning sequence in the event of this invalid condition).

The DHCP option 43, sub-option 11 is a device-specific parameter. It indicates whether an address is being requested in the PS WAN Management or PS WAN Data realm. Table 19 indicates the how the values for DHCP option 43, sub-option 11 MUST be set for these interfaces.

Table 19/J.191 – DHCP option 43, sub-option 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

The CDC MUST implement the Vendor Class Identifier Option (DHCP option 60) as specified in Table 20.

The cable modem and PS element each send separate DHCP requests. Table 20 describes how the CDC MUST set the contents of options 60 and 43 for the PS when separate PS WAN Management and PS WAN Data addresses are requested.

Table 20/J.191 – DHCP options for embedded PS WAN-Man and WAN-Data Address Requests

DHCP request options	Value	Description
PS DHCP request for WAN Management address		
CPE option 60	"PS"	
CPE option 43 sub-option 1	request sub-option vector	List of sub-options (within option 43) to be returned by server. None defined
CPE option 43 sub-option 2	"EPS"	Embedded PS
CPE option 43 sub-option 3	"ECM:EPS"	List of embedded devices (Embedded CM and embedded PS)
CPE option 43 sub-option 4	e.g., "123456"	Device serial number
CPE option 43 sub-option 11	PS WAN-Man (0x01)	Defines that an address is being requested in the PS WAN Management realm
PS DHCP request for WAN-Data address		
CPE option 60	"PS"	
CPE option 43 sub-option 1	request sub-option vector	List of sub-options (within option 43) to be returned by server. None defined
CPE option 43 sub-option 2	"EPS"	Embedded PS
CPE option 43 sub-option 3	"ECM:EPS"	List of embedded devices (Embedded CM and embedded PS)
CPE option 43 sub-option 4	e.g., "123456"	Device serial number
CPE option 43 sub-option 11	PS WAN-Data (0x02)	Defines that an address is being requested in the PS WAN-Data realm

The CDC MUST support the DHCP options indicated as mandatory in the CDC Protocol Support column in Table 21.

Table 21/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	Sub-option 3 – Service Provider's SNMP Entity Address	M
177	Sub-option 51 – Kerberos Server IP address	M

Table 21 represents the DHCP options that are mandatory and optional for the CDC to support. DHCP options listed as mandatory in Table 21 MUST be included in DHCP DISCOVER and DHCP REQUEST messages sent by the CDC to the cable network DHCP server.

The PS MUST support a Service Provider's SNMP Entity Address (DHCP option 177 sub-option 3) configured as an IPv4 address.

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:

(This IP address is used for the WAN mapping for the Dynamic NAPT tuple. This address cannot be used for NAT mapping because WAN side of NAT mapping is persistent. It also cannot be used for Pass-through addresses, which are assigned from the service provider's IP address pool.)

Management IP address: 192.168.100.5

Netmask: 255.255.255.0

Default Gateway: 192.168.100.1

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.

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.

The CDC MUST continue to broadcast its DHCP DISCOVER message (in accordance with [RFC 2131]) until it receives an address and DHCP ACK. The specific time-out for DHCP server access is implementation-dependent. However, the CDC MUST NOT broadcast DHCP DISCOVER more than 3 times in any 30-second period. At minimum, the CDC MUST broadcast DHCP DISCOVER at least once per 30-second interval, until it successfully acquires an address.

If the CDC does not receive a DHCP OFFER after 5 attempts to broadcast a DHCP DISCOVER message, the PS MUST initiate operation of the CDS, so that LAN IP Devices in the LAN-Trans realm can be served with IP addresses.

7.3 Bulk PS configuration architecture

7.3.1 Bulk PS configuration system design guidelines

The following system design guidelines in Table 22 drive the capabilities defined for the Bulk PS Configuration tool:

Table 22/J.191 – Bulk PS 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 PS configuration system description

Bulk PS 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:

The format of the Configuration File:

1) modes of triggering the download process;
2) means of authenticating the file;
3) 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 does not need a Configuration File loaded before it can operate. It is expected that a PS will initialize itself to a known state and a PS could run for a lifetime without having a 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 PS Configuration parameter download. Refer to 11.3.5.2 for a description of the firewall Configuration File Download procedure.

7.3.3 Bulk PS configuration requirements

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 23.

Table 23/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 file length MUST be 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 configuration settings which MAY be present;
- vendor-specific configuration settings.

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

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

Each PS element MUST support and a PS Configuration File MAY include configuration parameter Types 0, 4, 9, 17, 21, 28, 32, 33, and 255, which are described in this clause.

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

The Length value for each Type described in 7.3.3.1.1 to 7.3.3.1.10 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 RSA-Public-Key

This Attribute is a string attribute containing a DER-encoded RSAPublicKey ASN.1 type, as defined in the RSA Encryption Standard PKCS #1 v2.0 [RSA1].

PKCS #1 v2.0 specifies that an RSA public key consists of both an RSA public modulus and an RSA public exponent; the RSAPublicKey type includes both of these as DER-encoded INTEGER types.

PKCS #1 v2.0 states that the RSA public exponent may be standardized in specific applications, and the document suggests values of 3 or 65537 (F4). This Recommendation requires F4 for a public exponent and employs a 2048-bit modulus.

Type	Length	Value
4	106, 140, or 270[a)]	DER-encoded RSAPublicKey ASN.1 Type
[a)] Length of DER-encoding, using F4 as the public exponent, and a 2048-bit public modulus, respectively.		

7.3.3.1.3 Software upgrade filename

The filename of the software upgrade file for the PS. 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 [a)]	filename
[a)] Length MUST NOT cause resulting MAC management message to exceed the maximum allowed size.		

7.3.3.1.4 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 Identifier (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 encoding of the OID prefix plus one byte for the control flag.

The control flag may take the following 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.

7.3.3.1.5 CA-Certificate

This Attribute is a string attribute containing an X.509 CA Certificate, as defined in ITU-T Rec. X.509.

Type	Length	Value
17	Variable [a)]	X.509 CA Certificate (DER-encoded ASN.1)
[a)] Length MUST NOT cause resulting MAC management message to exceed the maximum allowed size.		

7.3.3.1.6 Software upgrade TFTP server

The IP address of the TFTP server, on which the software upgrade file for the PS resides.

Type	Length	Value
21	4	ip1, ip2, ip3, ip4

7.3.3.1.7 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 [a)]	variable binding
[a)] Length MUST NOT cause resulting MAC management message to exceed the maximum allowed size.		

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.

7.3.3.1.8 Manufacturer Code Verification Certificate

The Manufacturer's Code Verification Certificate (M-CVC) for Secure Software Downloading. The PS Configuration File MUST contain a M-CVC and/or C-CVC in order to allow the device to download the code file from TFTP server.

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

If the length of the M-CVC exceeds 65 535 bytes, the M-CVC MUST be fragmented into two or more successive Type 32 elements. Each fragment, except the last, MUST be 65 535 bytes in length. The PS reconstructs the M-CVC by concatenating the contents (Value of the TLV) of successive Type 32 elements in the order in which they appear in the Configuration File. For example, the first byte following the length field of the second Type 32 element is treated as if it immediately follows the last byte of the first Type 32 element.

7.3.3.1.9 Co-signer Code Verification Certificate

The Co-signer's Code Verification Certificate (C-CVC) for Secure Software Downloading. The PS Configuration File MUST contain a C-CVC and/or M-CVC in order to allow the device to download the code file from TFTP server.

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

If the length of the C-CVC exceeds 65 535 bytes, the C-CVC MUST be fragmented into two or more successive Type 33 elements. Each fragment, except the last, MUST be 65 535 bytes in length. The PS reconstructs the C-CVC by concatenating the contents (Value of the TLV) of successive Type 33 elements in the order in which they appear in the Configuration File. For example, the first byte following the length field of the second Type 33 element is treated as if it immediately follows the last byte of the first Type 33 element.

7.3.3.1.10 SNMPv3 kickstart value

Compliant PS 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.

7.3.3.1.10.1 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.

7.3.3.1.10.2 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.

7.3.3.1.11 Configuration File Element – docsisV3Notification Receiver

Type	Length	Value
38	Variable	(See below.)

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. Up to 10 of these elements may be included in the PS Configuration File.

Here is the format of this element:

Definition of fields of docsisV3NotificationReceiver Element;

All multi-byte fields have the most significant bytes first in the field.

This TLV (38) consists of several Sub-TLVs inside of the TLV Configuration File element:

Sub-TLV 38.1 – IP Address of trap receiver, in binary

IP Address 4 bytes IP Address of the trap receiver, in binary.

Sub-TLV 38.2 – UDP Port number of the trap receiver, in binary

Port 2 bytes UDP Port number of the trap receiver, in binary.

(If not present, the default value 162 is used)

Sub-TLV 38.3 – Type of trap sent by the PS (Note 2)

Trap type 2 bytes

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.

Sub-TLV 38.4 – Time-out, in milliseconds, used for sending inform

Time-out 2 bytes 0-65535.

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

Retries 2 bytes 0-65535.

Sub-TLV 38.6 – Notification Filtering Parameters

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. <z> is the length, in bytes, of the ASN.1 encoding. This field starts with the ASN.1 Universal type 6 (Object Identifier) byte, then the ASN.1 length field, then the ASN.1 encoded object identifier components.

Sub-TLV 38.7 – Security Name to use when sending SNMP V3 Notification

This Sub-TLV is not required for Trap type = 1, 2, or 3 above. 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 "@config" (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 Configuration File TLV Type 34 as part of the DH Kickstart procedure. The notifications will 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 SHALL 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 SHALL be "public". The Security Name in traps and informs in SNMP V3 packets where no security name has been specified SHALL be "@config and in that case the security level SHALL 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 SHALL be sent to the trap receiver.

NOTE 4 – Config file TLV number: The type field of this TLV SHALL be (38).

NOTE 5 – 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 SHALL NOT use index columns that start with the characters "@config".

NOTE 6 – This TLV element SHALL be processed only if the PS has entered SNMP V3 Coexistence Mode during processing of the PS Configuration File.

7.3.3.1.12 End-of-Data marker

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

Type	Length	Value
255	–	–

7.3.3.2 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 OFFER, 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 Configuration hash appended to the PS Configuration File name MUST NOT be included in the URL-encoded value.

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 DHCPACK 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.

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 Security 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.

If the PS is operating in SNMP Provisioning Mode (as indicated by the value of cabhPsDevProvMode) AND the cabhPsDevProvConfigHash object from the PSDev MIB has a valid value, the PS Configuration File download MUST be triggered when an SNMP Set-Request message, addressed to the PS WAN-Man interface, contains a valid value for the cabhPsDevProvConfigFile PSDev MIB object. The format of cabhPsDevProvConfigFile MUST be a URL-encoded TFTP server IP address and configuration file name.

Post-trigger Operation:

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

If the PS Configuration File is properly authenticated, when the TFTP download of the PS Configuration File is complete, the PS MUST process the TLVs contained within the file. Refer to 6.3.9 for a description of how the CMP processes the Configuration File.

7.3.3.3 Means of authenticating the PS Configuration File

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

A hash calculation is used to authenticate the PS Configuration File. The NMS calculates the hash of the PS Configuration File then sends the resulting hash value to the PS element. The identity of the NMS that generated the PS Configuration File is authenticated by comparing the hash of the PS Configuration File that was generated by the NMS and transported to the PS element against the hash (calculated by the PS) on the PS Configuration File downloaded from the TFTP server. The identity of the PS element requesting the file is not required.

The security algorithm used to authenticate the PS Configuration File depends upon the provisioning mode of the PS element (see 5.7). There are two types of provisioning modes: DHCP Provisioning Mode and SNMP Provisioning Mode. The following subclauses describe the security algorithms and requirements needed to authenticate the PS Configuration File 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 authentication of the PS Configuration File by the PS element in DHCP Provisioning Mode follows:

1) When the NMS creates a new PS Configuration File or modifies an existing file, the NMS will create a SHA-1 hash of the entire content of the PS Configuration File, taken as a byte string.
2) The NMS appends the hash value to the PS Configuration File name that is sent to the PS element in the DHCP Offer (see 7.2.3.3 and 13.2). The delimiter used between the PS Configuration File name and hash value is the '@' character (e.g., "configfile1.txt@23423487987345"). The PS element updates the cabhPsDevProvConfigHash MIB object with the received hash value.
3) The PS element downloads the named file from the configured TFTP server.
4) The PS element MUST compute a SHA-1 hash over the entire content of the PS Configuration File and compare the computed hash to the hash in cabhPsDevProvConfigHash MIB object. If the computed and configured hash values are the same, the PS Configuration File is authenticated; otherwise, the file MUST be rejected.
5) When authentication is successful, the PS element MUST use the PS Configuration File contents for its configuration.

7.3.3.3.2 Configuration File authentication algorithm for SNMP Provisioning Mode

The procedure for authentication of the PS Configuration File by the PS element in SNMP Provisioning Mode follows:

1) When the NMS creates a new PS Configuration File or modifies an existing file, the NMS will create a SHA-1 hash of the entire content of the PS Configuration File, taken as a byte string.
2) The NMS sends the hash value calculated in step 1 to the PS element via SNMP SET and updates the cabhPsDevProvConfigHash MIB object.
3) The NMS sends the Name and location of the PS Configuration File via SNMP SET and updates the cabhPsDevProvConfigFile MIB object (this triggers the TFTP download, see 7.3.3.2).
4) The PS element downloads the named file from the configured TFTP server.
5) The PS element MUST compute a SHA-1 hash over the entire content of the PS Configuration File and compare the computed hash to the hash in

cabhPsDevProvConfigHash MIB object. If the computed and configured hash values are the same, the PS Configuration File is authenticated; otherwise, the file MUST be rejected.

6) When authentication is successful, the PS element MUST use the PS Configuration File contents for its configuration.

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 time-out 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 24 identifies the processing modes that MUST be handled and the action that MUST be taken when these processing modes are detected.

Table 24/J.191 – PS Configuration File processing modes

Failure Mode	Action
Type field is not valid	Disregard the subject TLV and report an event. Continue to process the file.
File fails integrity check (file integrity still needs to be defined)	Report an event. Do not attempt to process the file.
File is too large	Report an event. Do not attempt to process the file.
Configuration File not found	Report an event. Do not attempt to process the file.
File is not properly padded	Report an event. Do not attempt to process the file.
No End Of File marker	Report an event. Do not attempt to process the file.
Unable to set value	Report and 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.
Encounters a value where the SNMP OID is unrecognized	Disregard the subject TLV and report an event. Continue to process the file.

Refer to Annex B for a list of events including those listed in Table 24 and information about how events are reported.

If any configuration settings are processed, then an event MUST be generated when the end of the file is detected, and this event MUST include the number of TLVs successfully processed and the number of TLVs skipped.

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 specific time-out for TFTP server access is implementation-dependent. However, the PS MUST NOT attempt to access the TFTP server more than 3 times in any 5-minute period. At minimum, the PS MUST attempt at least once per 5-minute interval to download the PS Configuration File, until the PS Configuration File is successfully downloaded.

The PS MUST generate 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.

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

If the PS is operating in DHCP Mode (as indicated by the value of cabhPsDevProvMode) AND aborts the PS Configuration File download process, the PS MUST generate 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 re-issue a DHCP DISCOVER in accordance with [RFC 2131], i.e., the PS must re-start the initialization process.

The PS MUST use an adaptive time-out for TFTP based on binary exponential backoff as described in [RFC 1123] and [RFC 2349].

7.4 Time of Day Client architecture

7.4.1 Time of Day Client system design guidelines

The following system design guidelines in Table 25 drive the capabilities defined for the PS Time of Day Client:

Table 25/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 PS 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 DHCP 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 messaging and security functions.

Time of Day client requirements

The PS element MUST implement a Time of Day client.

The PS 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 PS element MUST initialize its time to 0 (0:0.0 January 1, 1900) in accordance with [RFC 868].

The PS element MUST attempt Time of Day time synchronization with the TOD server indicated by the DHCP option 4, that is received in the DHCP Offer made to the WAN-Man interface.

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 PS 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 event messaging and security functions and need only be accurate to the nearest second.

The PS element MUST continue to attempt to communicate with the Time of Day server, until local time is established. The specific time-out for Time of Day requests is implementation-dependent. However, the PS Time of Day client MUST NOT exceed more than 3 TOD requests in any 5-minute period. At minimum, the PS Time of Day client MUST issue at least 1 TOD request per 5-minute period, until local time is established.

If the TOD server does not respond with a valid response, the PS MUST do the following, not necessarily in the order listed:

- Set the value of cabhPsDevTodSyncStatus to '2' (TOD access failed).
- If there are active leases in the LAN-Trans realm as indicated by a nonzero value for cabhCdpLanTransCurCount, set cabhCdpLanAddrCreateTime to the current time and set cabhCdpLanAddrExpireTime to the value of cabhCdpLanAddrCreateTime plus the value of cabhCdpServerLeaseTime for each active lease (Expire Time = CreateTime + LeaseTime);
  - log the failure and generate a standard event defined in Annex B; and
  - continue to retry communication with the TOD server until local time is established.

If the PS successfully synchronizes its time reference with the TOD server in the cable network, the PS MUST set the value of cabhPsDevTodSyncStatus to '1' (TOD synchronization successful).

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.

8 Packet handling and address translation

8.1 Introduction/overview

8.1.1 Goals

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

- providing cable friendly address translation functionality, enabling cable operator visibility and manageability of home devices while preserving cable network source based routing architectures;
- preventing unnecessary traffic on the cable and home network;
- conserving globally routable public IP addresses as well as cable network private management addresses;
- facilitating 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.
- The PS packet-handling function may forward broadcast traffic to all LAN and WAN-Data interfaces transparently. Throttling of broadcast traffic is not required. It is assumed that the DOCSIS cable modem has the capability to filter broadcast IP traffic.

8.2 Architecture

This clause describes the key concepts behind the packet-handling and address translation functionality.

8.2.1 System design guidelines

See Table 26.

Table 26/J.191 – Packet handling and address translation system design guidelines

Number	System design guidelines
Pckt Handling 1	Addressing mechanisms will be operator controlled, and will provide the operator knowledge of and accessibility to the PS.
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, if any.
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

The 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 16, 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.

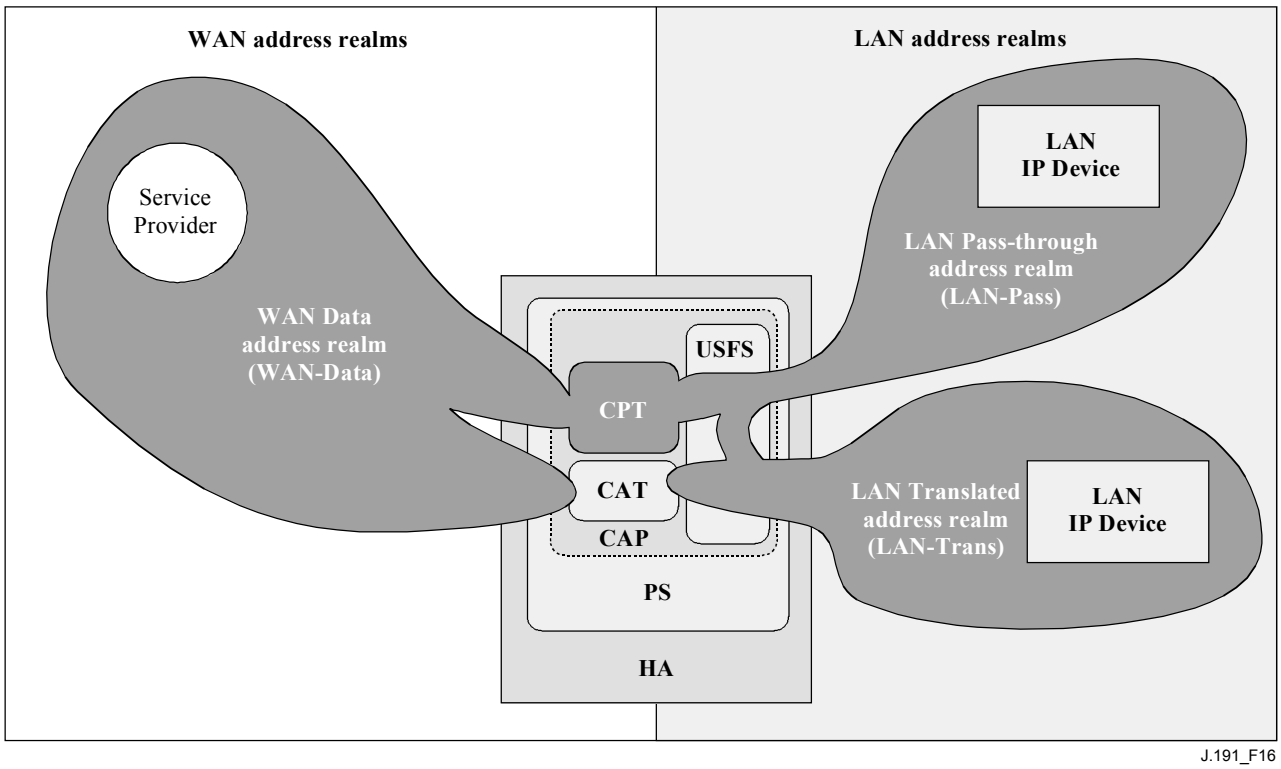


Figure 16/J.191 – Cable Address Portal (CAP) functions

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 CableHome Pass-through (CPT) 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 in home traffic to the home, even when in home 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.

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 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 PS 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:

1) Pass-through Mode;

2) C-NAT Transparent Routing Mode; and

3) 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 DIS 10038 MAC Bridges] 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).

Dynamic C-NAT Mappings for UDP traffic are destroyed when an inactivity time-out period, cabhCapUdpTimeWait, expires. Dynamic C-NAT Mappings for TCP traffic are destroyed when an inactivity time-out period, cabhCapTcpTimeWait, expires or a TCP session terminates. Dynamic C-NAT Mappings for ICMP traffic are destroyed when an inactivity time-out period, cabhCapIcmpTimeWait, expires. In addition, Static C-NAT Mappings may be created or destroyed when the NMS system writes to or deletes from the cabhCapMappingTable MIB table.

In C-NAPT Mode (the factory default mode for the system) the PS element (CDC) acquires one IP address, used for WAN-Data traffic. After acquisition, via DHCP, this IP address is used as the WAN-Data IP address portion of Dynamically created C-NAPT Mapping tuples. If the WAN-Data IP address has been acquired, Dynamic C-NAPT Mappings are created when the CAP first sees LAN-to-WAN IP traffic that does not have an existing Mapping. If the WAN-Data IP address has not been acquired (i.e., does not have an active DHCP lease), the Dynamic C-NAPT Mapping can not be created, and this traffic is dropped, and a standard event is generated (see Annex B).

Dynamic C-NAPT Mappings for UDP traffic are destroyed when an inactivity time-out period, cabhCapUdpTimeWait, expires. Dynamic C-NAPT Mappings for TCP traffic are destroyed when an inactivity time-out period, cabhCapTcpTimeWait, expires or a TCP session terminates. Dynamic C-NAPT Mappings for ICMP traffic are destroyed when an inactivity time-out 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 17 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 address in the LAN-Trans realm.

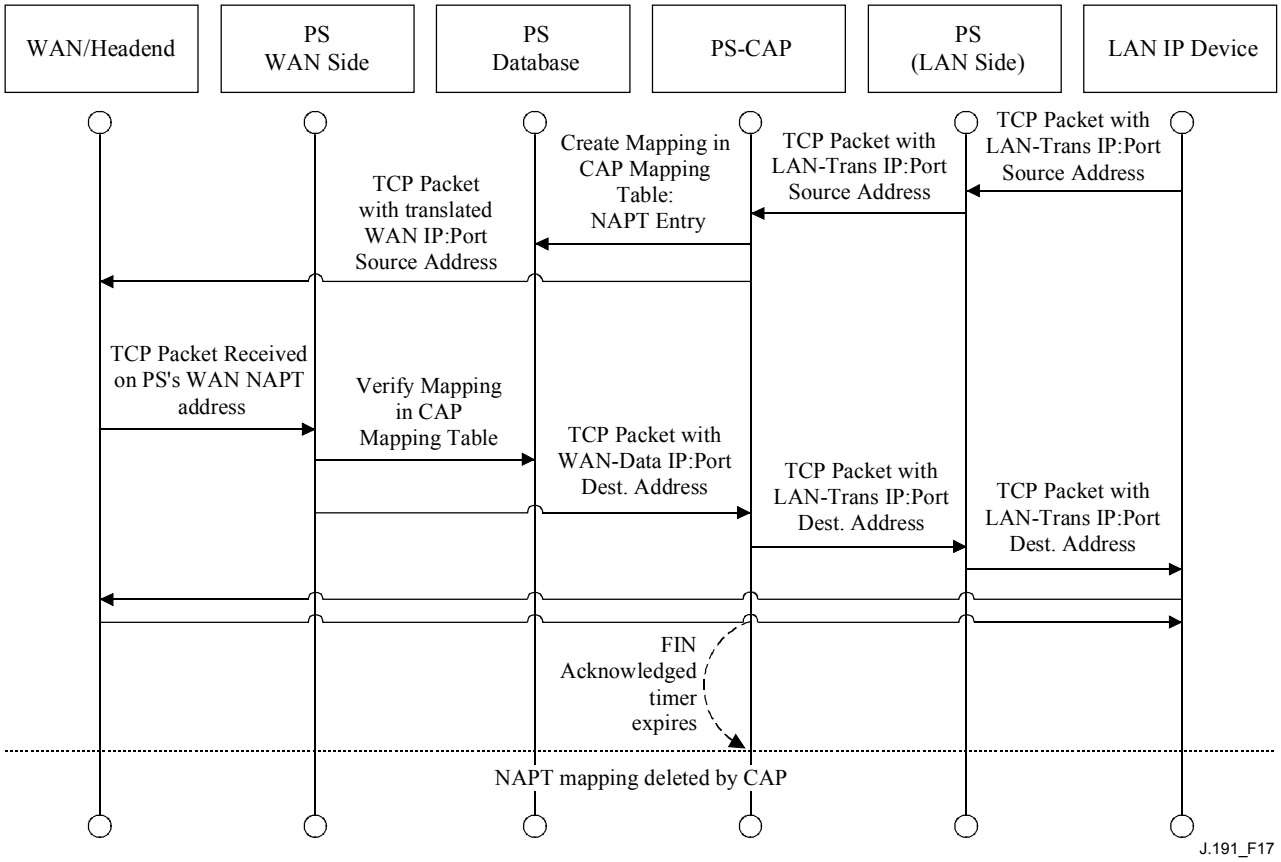


Figure 17/J.191 – PS configuration (CAP Mapping Table – NAPT) sequence diagram

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 (cabhCapPassthroughTable). In this Mixed Mode, the PS examines MAC addresses of received frames to determine whether to transparently bridge the frame or to perform any C-NAT or C-NAPT Transparent Routing functions at the IP layer. In the case of LAN-to-WAN traffic, the PS examines the source MAC address, and if that MAC address exists in the cabhCapPassthroughTable, the frame is transparently bridged to the WAN-Data interface. In the case of WAN-to-LAN traffic, the PS examines the destination MAC address, and if that MAC address exists in the cabhCapPassthroughTable, the frame is transparently bridged to the appropriate LAN interface. If the MAC address does not exist in the cabhCapPassthroughTable, the packet is processed by higher layer functions, including the C-NAT/C-NAPT Transparent Routing function.

It should be noted that the USFS functionality (see 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 and is destined to the home – 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 time-out 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 Multicast traffic by transparently bridging IGMP messaging [RFC 2236] and IP Multicast packets. 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 18 provides a detailed example of IGMP and Multicast processes passing through a PS.

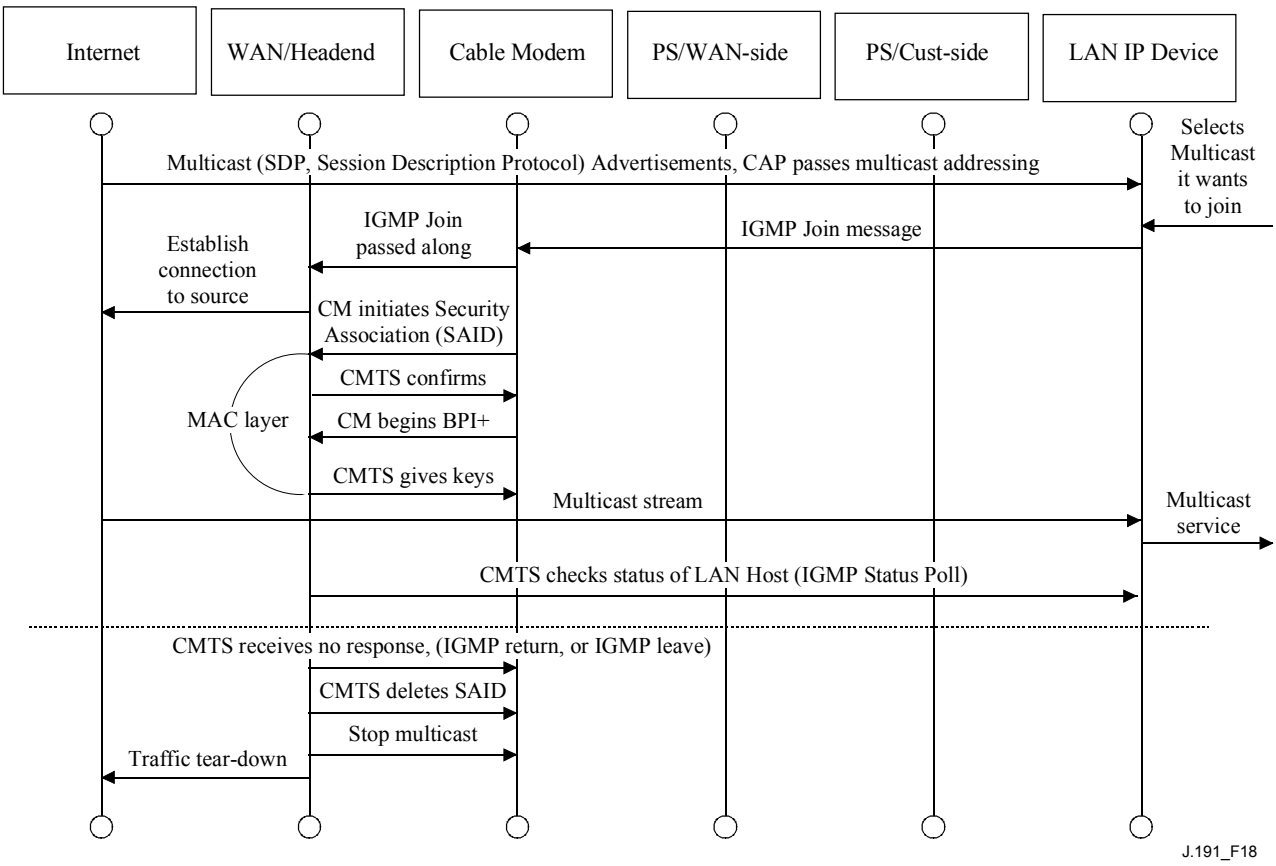


Figure 18/J.191 – Multicast via IGMP sequence

8.2.2.5 Packet-handling examples

This clause provides an informative look at processing involved for packet-handling. Figure 19 shows an example of possible packet processing steps for LAN-to-WAN unicast traffic, and Figure 20 shows an example of possible packet processing steps for WAN-to-LAN unicast traffic. These examples are informative only and do not imply any requirements on implementation.

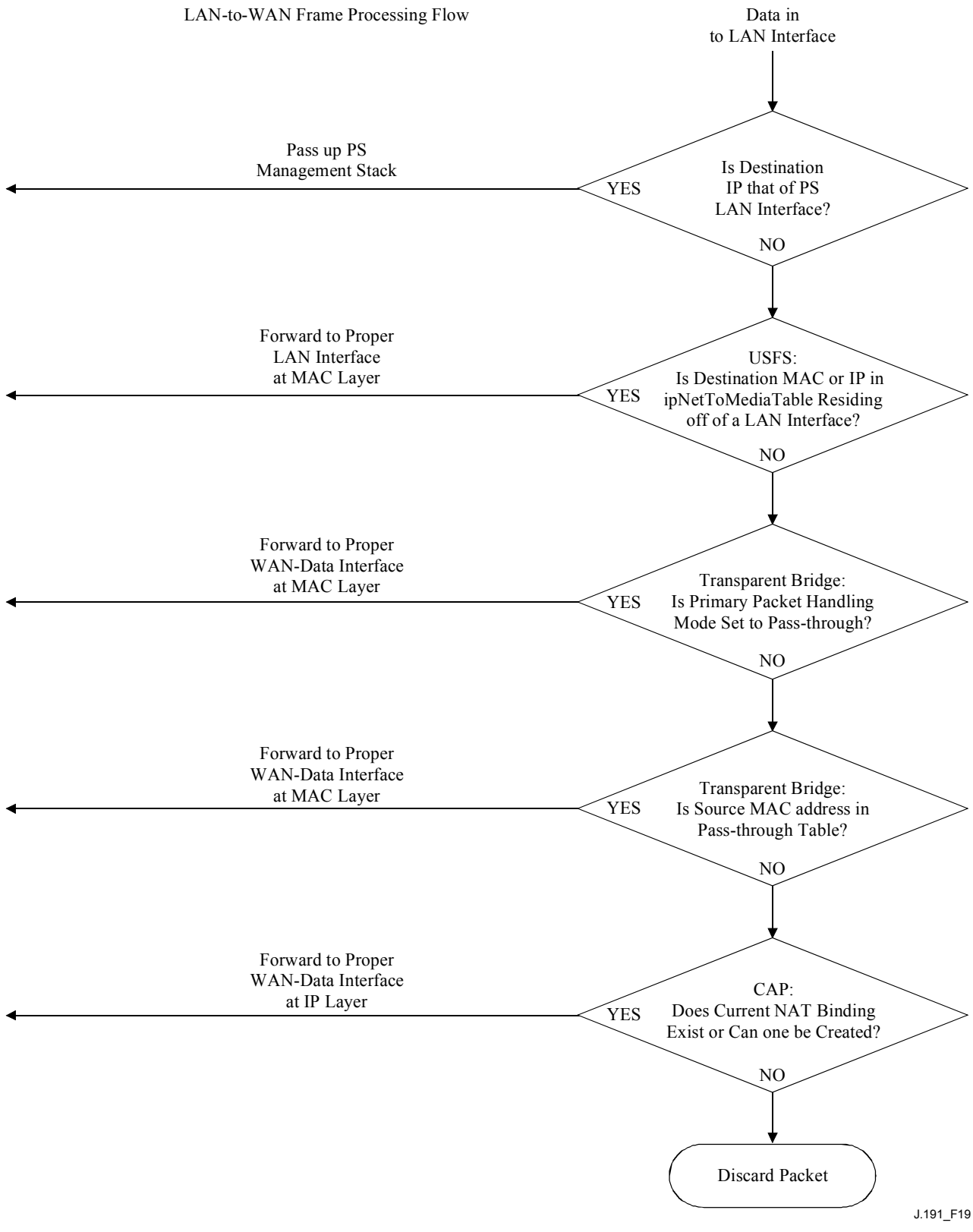


Figure 19/J.191 – LAN-to-WAN packet processing example

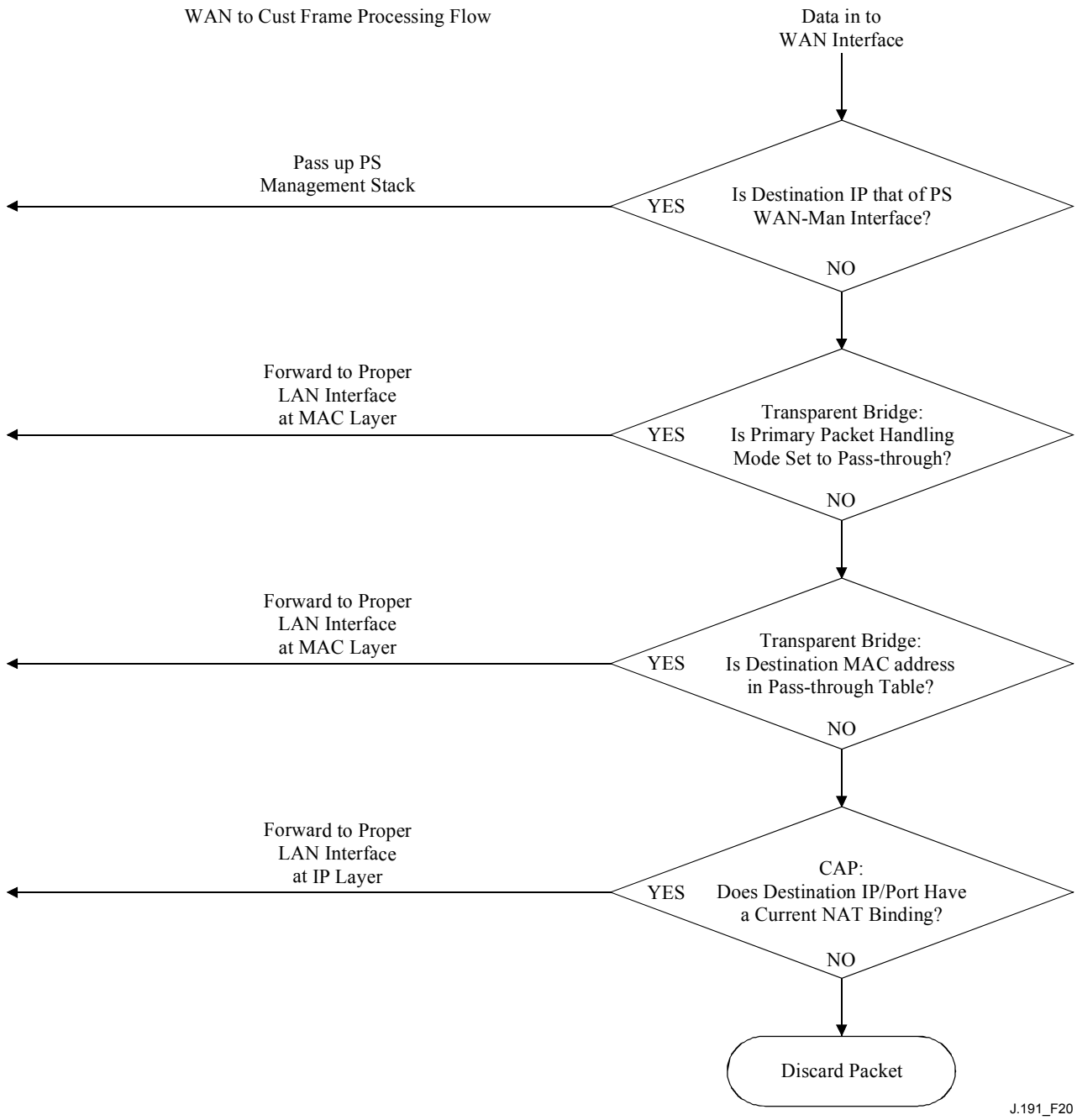


Figure 20/J.191 – WAN-to-LAN packet processing example

8.3 CAP requirements

8.3.1 General requirements

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

The CAP MUST support Multicast traffic, by transparently bridging IGMP messaging and IP Multicast packets as defined in [RFC 2236].

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).

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 traffic MUST never be routed or bridged out a WAN interface.

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 10038], 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 time-out, 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 time-out, 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 time-out, 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 cabhCapPassthroughTable, 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 time-out 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 name resolution include:

- Provide Domain Name System (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 then by IP address.
- Refer LAN DNS clients to Headend DNS servers, 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

See Table 27.

Table 27/J.191 – Name Resolution system design guidelines

Reference	System design guidelines
Name Rsln 1	Provide Domain Name Service 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 Referral to Headend DNS servers, 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.
- Refers LAN IP Devices to external DNS servers 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 customer premises 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 21. The CNP responds to initial standard DNS queries [RFC 1035], directed to cabhCdpServerDnsAddress, for all name lookups. If the CNP responds with a referral to external DNS servers, it is assumed to be the responsibility of the LAN IP Device to send a query directly to the referred server.

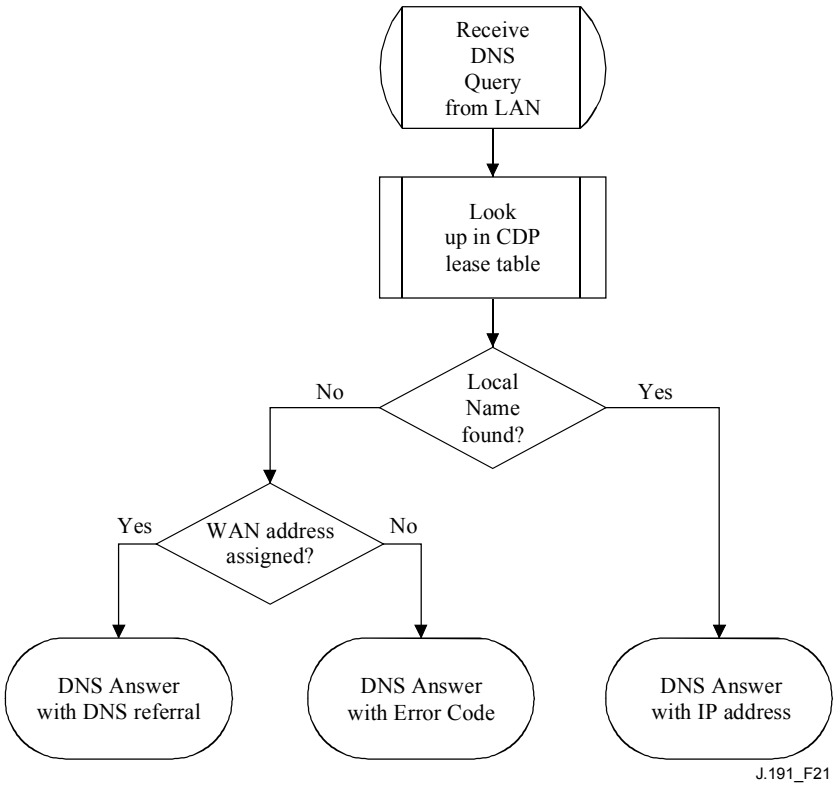


Figure 21/J.191 – CNP packet processing

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 returns a DNS referral which points to an external DNS server (which is learned by the CDC via DHCP options). The IP address of the external DNS server is the last cabhCdpWanDataAddrDnsIp entry in the CDP's cabhCdpWanDataAddrServerTable.

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. Table 28 is an example of the SOA record fields (see section 3.3.13 of [RFC 1035]).

Table 28/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, 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 operate at least in non-recursive mode, as defined in [RFC 1034].

The CNP answers name queries, using only local information within the PS, and its response messages MUST contain an error, an answer, or a referral to an external DNS server.

The CNP MUST 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 state of the cabhCdpWanDataAddrDnsIp entry in the CDP's cabhCdpWanDataAddrServerTable, 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.

When functioning as a non-recursive DNS server: if the hostname can not be resolved by the CNP from local data AND the last cabhCdpWanDataAddrDnsIP entry in the CDP's cabhCdpWanDataAddrServerTable is populated, the CNP MUST respond to the hostname resolution query with a referral to an external DNS server, represented by the IP address contained in the cabhCdpWanDataAddrDnsIp object.

If the hostname can not be resolved by the CNP from local data AND the cabhCdpWanDataAddrDnsIp object is not populated, the CNP MUST respond to the hostname resolution query with the appropriate error specified by [RFC 1035].

When the last remaining WAN-Data DHCP lease expires, the CDC MUST clear all cabhCdpWanDataAddrDnsIp entries from the cabhCdpWanDataAddrServerTable.

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 IP enhanced Cable Modem in enabling home 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 applications to establish prioritized data sessions between the CMTS and PS 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 QoS:

- QoS assumes 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 quality-of-service (CQoS) architecture is composed of functional elements and the HA device class. Developers of 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 QoS system design guidelines are listed in Table 29.

Table 29/J.191 – QoS system design guidelines

Number	QoS system design guidelines
QoS 1	A standard QoS signalling mechanism will exist that allows IP-enhanced Cable Modems to support the establishment of prioritized service sessions across the DOCSIS network for multimedia applications.
QoS 2	Multimedia applications may be embedded in the HA device or on an external device connected to the HA device.
QoS 4	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);
- Cable Quality-of-Service Portal (CQP) function;
- 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. The PS resides in HA devices only (see clause 5). The QoS component is referred to as the Cable Quality-of-Service Portal (CQP). 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.

10.2.2.2 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 22.

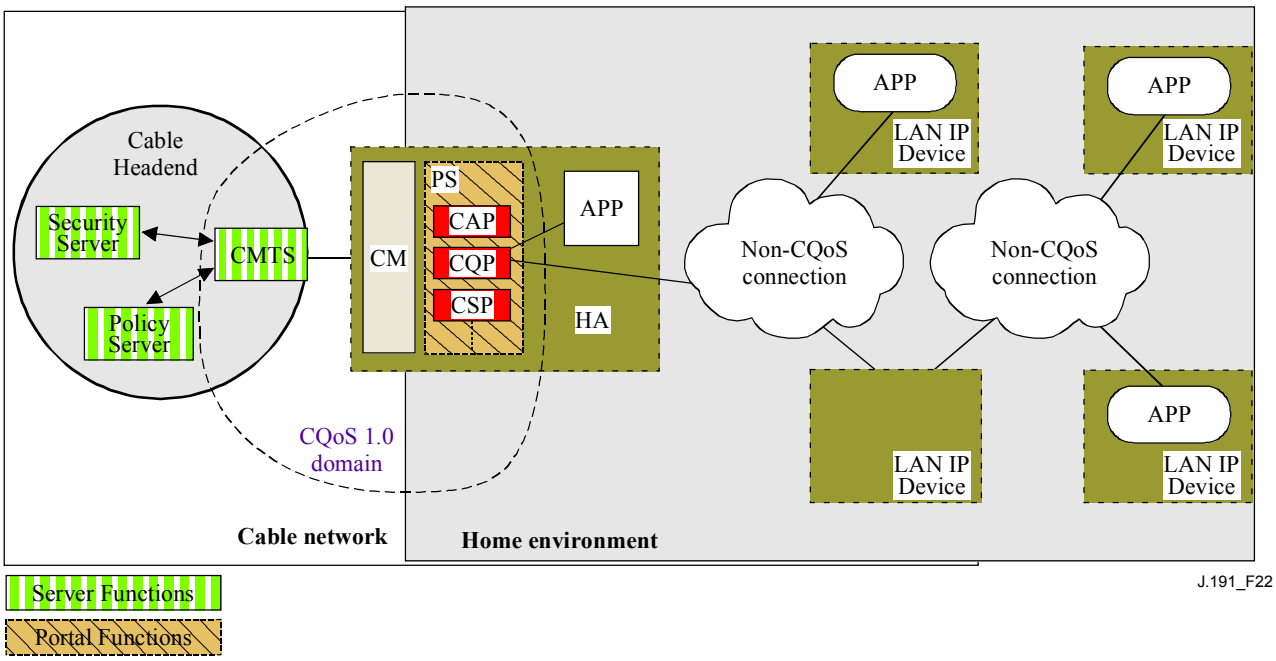


Figure 22/J.191 – Example of CQoS functional elements

10.3 Cable QOS messaging requirements

The 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 (ITU-T Recs 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 5.7).

10.3.2 QoS policy management and admission control

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

Supporting the delivery of reliable multimedia IP services to client devices in a home 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 DOCSIS and IPCablecom technologies on the hybrid fiber-coax (HFC) network.
- Provide flexible security mechanisms to be compatible with DOCSIS and IPCablecom security mechanisms used on the HFC network.

11.1.2 Assumptions

The assumptions for the security environment include:

- The PS and CM functionality reside in a single physical device.
- Lower security levels may exist in the home 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 IPService Portal (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 server, Security server, etc.

The security 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 in Table 30. This list provided guidance for the development of the security specification.

Table 30/J.191 – 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.

Table 30/J.191 – Security system design guidelines

Reference	Security system design guidelines
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	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

The following subclauses provide an overview of all the elements that are part of the security architecture.

The security architecture includes the following security elements:

- Security Domain;
- IPService Portal function (PS);
- Cable Security Portal (CSP) function;
- Firewall (FW);
- Security Server.

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 HA device class. An example of the relationship between the different security elements and HA device classes is presented in Figure 23. In the figure, in-home applications are represented as APP, and the OSS server is the NMS server.

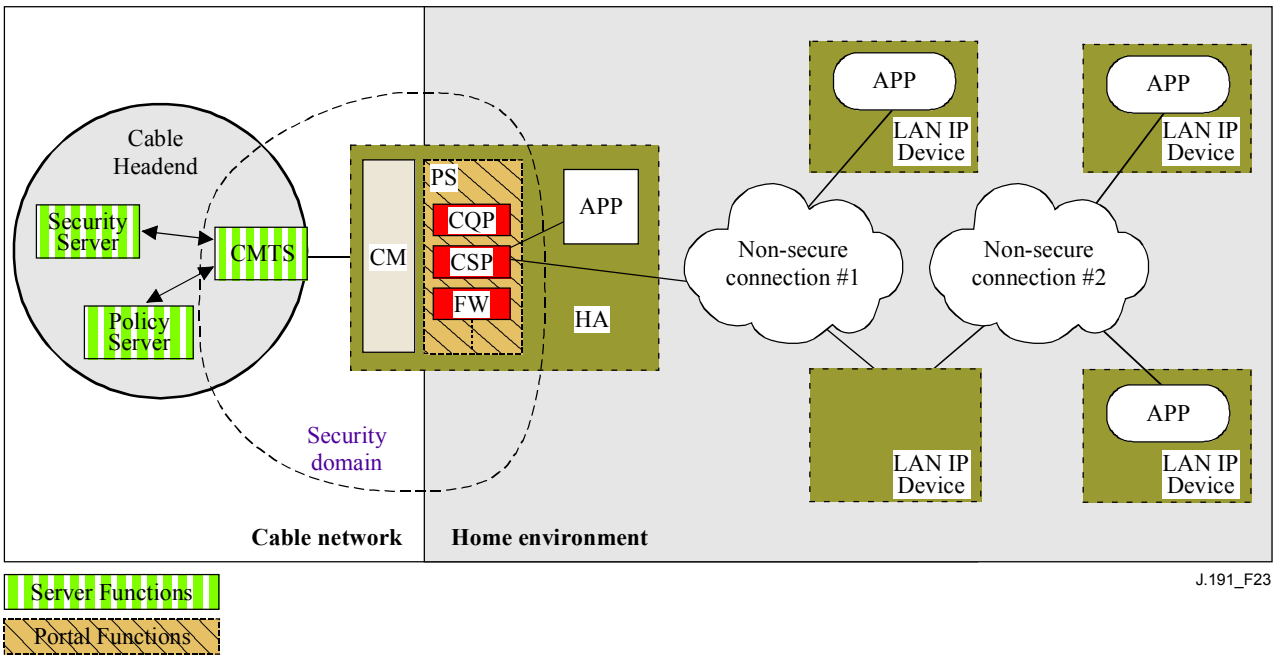


Figure 23/J.191 – Security elements

11.2.2.1 Security Domain

The Security Domain is defined in Figure 23 and encompasses the PS element in the HA and the illustrated headend servers.

11.2.2.2 PS function

Portal Service (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 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 filers 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 firewall scope. 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 24 illustrates the relationship among the firewall components. In particular, the figure 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 24 also provides a more detailed view of the PS and it's relationship to firewall functions and other components in the HA device. In particular, the figure 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.

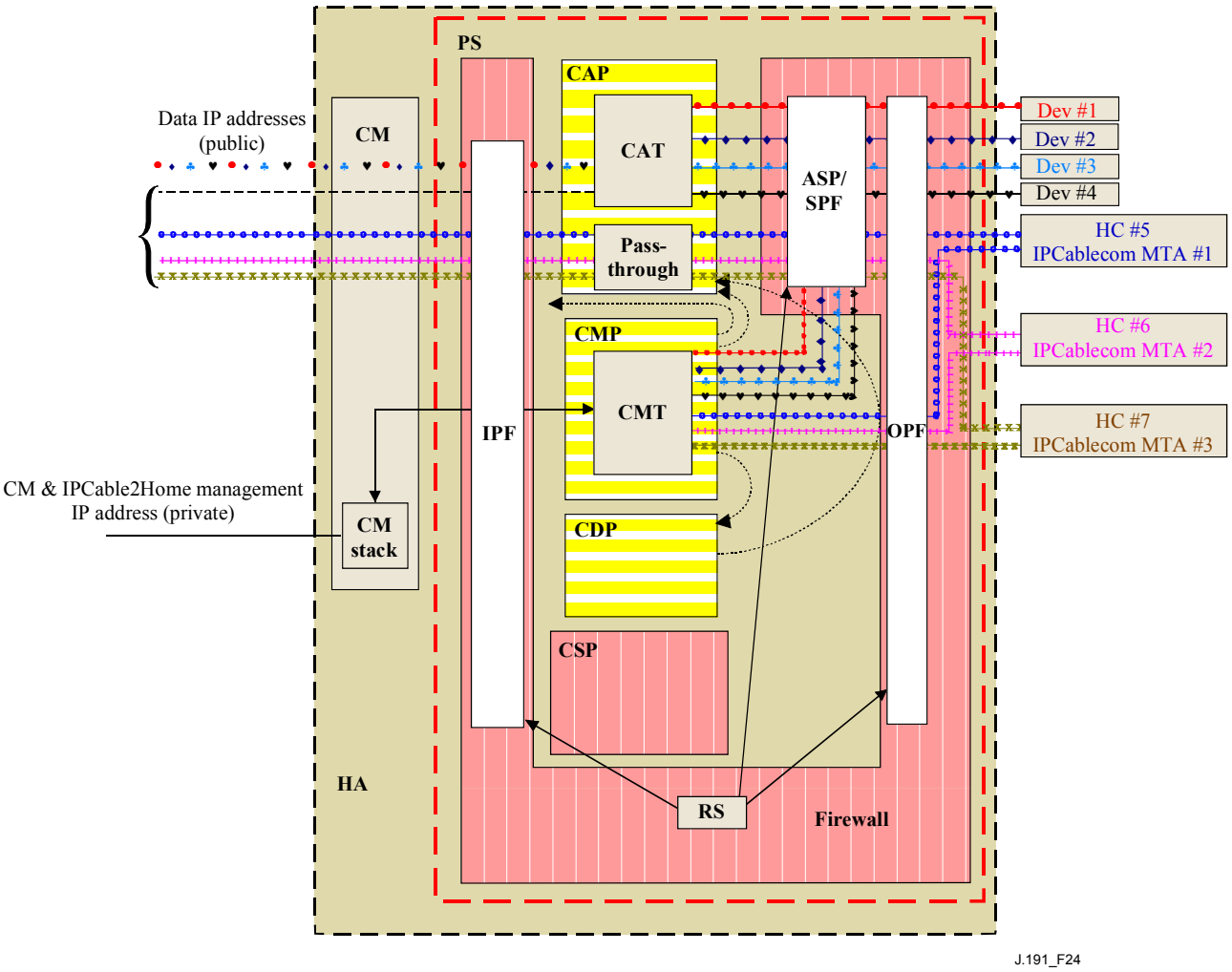


Figure 24/J.191 – Example of a PS element in an HA device

11.2.3 Key Distribution Center (KDC) server

The Security server supported is the Key Distribution Center (KDC) server. If a KDC server that supports 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 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 in clause 10 (QoS).

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 2459]. The PS Element Certificate provides the identity of the associated PS element by cryptographically binding the PS Element MAC address to a public key certificate issued for that PS element. Additionally, public key certificates provide a secure way to communicate the identity information.

When a KDC that supports this application 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 5.1) to take advantage of the specified mutual authentication protocol with the use of Kerberos using the PKINIT extension. Kerberos provides a protocol to secure mutual authentication in order to provide keying material and communication establishment only between authenticated parties. Because this authentication model has already been specified by 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.

The KDC enabled authentication MUST follow the specification for Kerberos/PKINIT as defined in ITU-T Rec. J.170. The KDC is equivalent to or the same as the IPCablecom operator KDC (IPCablecom specifies the use of several KDCs). The PS element MUST act as the client to the KDC. In the IPCablecom security Recommendation, the MTA is the client. It is expected that 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 are specified in 11.3.2. 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.

IPCablecom security for the following Kerberos functionality does not apply to this Recommendation:

1) Service Key Versioning (see 6.4.10/J.170);
2) Kerberos Cross-Realm Operation (see 6.4.11/J.170);
3) Rekey Messages (see 6.5.4/J.170);
4) Kerberized IPsec (see 6.5.6/J.170);
5) Kerberos Server Locations and Naming Conventions (see 6.4.6.3, CMS).

11.3.1.2 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 need to be changed:

- Replace pktcKdcToMtaMaxClockSkew as defined in the IPCablecom Security Specification with KdcToClientMaxClockSkew.
- Replace pktcSrvrToMtaMaxClockSkew as defined in the IPCablecom Security Specification with SrvrToClientMaxClockSkew.
- Replace MTAProvSrvr as defined in the IPCablecom Security Specification with ProvSrvr.
- Replace MTA-FQDN-Map as defined in the IPCablecom Security Specification with FQDN-Map.

Kerberos implementations MUST ignore the Object Identifier (OID) field portion, which reads clabProjPacketCable (2) within the AppSpecificTypedData within the KRB-ERROR messages.

11.3.2 Public Key Infrastructure (PKI)

This Recommendation uses public key certificates, which comply with the ITU-T Rec. X.509 specification and the IETF [RFC 3280].

11.3.2.1 Generic structure

11.3.2.1.1 Version

The version of the certificates MUST be 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 F4 – 65537.

11.3.2.1.3 Extensions

The extensions (subjectKeyIdentifier, authorityKeyIdentifier, KeyUsage, BasicConstraints, Signature Algorithm, SubjectName and IssuerName) 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 ASN.1 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]).

11.3.2.1.3.3 KeyUsage

The keyUsage extension MUST be used for all Certification Authority (CA) certificates and Code Verification Certificates (CVCs). For CA certificates the keyUsage extension MUST be marked as critical with a value of keyCertSign and cRLSign. For CVC certificates the keyUsage extension MUST be marked as critical with a value of digitalSignature and keyEncipherment. The end-entity certificates may use the keyUsage extension as listed in [RFC 3280].

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 31 to 42).

11.3.2.1.4 Signature algorithm

The signature mechanism used MUST be SHA-1 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.

11.3.2.2 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 are generic in nature and applicable to all applications needing certificates. This means the basic infrastructure can be reused for every application (DOCSIS, IPCablecom, PS). 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 to support the overlap. For example, IPCablecom requires a KDC for the service provider and this Recommendation can take advantage of a KDC that supports IPCablecom to provide mutual authentication. 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 this application. In this case, this application's KDC is equivalent to or the same as the IPCablecom operator KDC (IPCablecom specifies the use of several KDCs).

In Figure 25, the term "Certificate Authority" is abbreviated as CA and "Code Verification Certificate" is abbreviated as CVC.

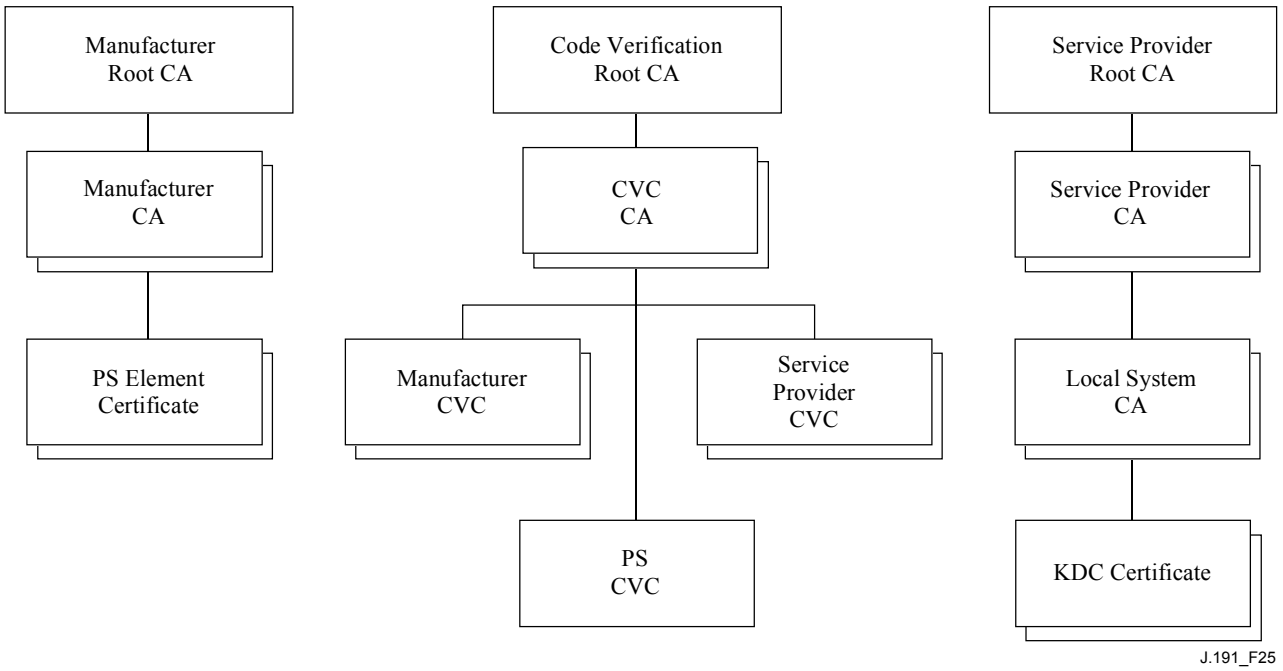


Figure 25/J.191 – Certificate hierarchy

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 31 through 33. 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 11.3.2 (PKI).

11.3.2.2.1.1 Manufacturer Root CA Certificate

The Manufacturer Root CA Certificate (see Table 31) MUST be verified as part of the certificate chain containing the Manufacturer Root CA Certificate, Manufacturer CA Certificate and the PS Element Certificate.

Table 31/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. It is intended that the validity period is long enough so that this certificate is never re-issued.
Modulus length	2048
Extensions	keyusage [c, m] (keyCertSign, cRL Sign), subjectkeyidentifier, basicConstraints [c, m](cA = true)

11.3.2.2.1.2 Manufacturer CA Certificate

The Manufacturer CA Certificate (see Table 32) MUST be verified as part of a certificate chain containing the Manufacturer Root CA Certificate, Manufacturer CA Certificate and the PS Element Certificate.

Table 32/J.191 – Manufacturer CA certificate

Subject name form	C = <country>, O = <CompanyName>, [S = <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.
Signed by	Manufacturer Root CA
Validity period	20 years
Modulus length	2048
Extensions	keyUsage[c, m](keyCertSign, cRLSign), subjectKeyIdentifier, authorityKeyIdentifier basicConstraints[c, m](cA = true, pathLenConstraint = 0)

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. If present, the authorityKeyIdentifier of the PS Element Certificate MUST be matched to the subjectKeyIdentifier of the manufacturer certificate as described in [RFC 3280].

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 and manufacturer's facility are optional attributes.

The PS element 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 MUST have a validity period greater than the operational lifetime of the specific device (see Table 33).

Table 33/J.191 – Device certificate

Subject name form	C = <country>, O = <Company Name>, [S = <state/province>,] [L = <city>], OU = [OU = <Product Name>,] [OU = <Manufacturer's Facility>,] CN = <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	2048
Extensions	keyUsage[n, o](digitalSignature, keyEncipherment), authorityKeyIdentifier, The keyUsage extension is optional. When the keyUsage extension is used it SHOULD be marked as non-critical.

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 34 through 38. 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 11.3.2 (PKI).

11.3.2.2.2.1 Code Verification Root CA Certificate

This certificate (see Table 34) 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 34/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. It is intended that the validity period is long enough so that this certificate is never re-issued.
Modulus length	2048
Extensions	keyUsage [c, m] (keyCertSign, cRL Sign), subjectkeyidentifier, basicConstraints [c, m](cA = true, pathLenConstraint = 0)

11.3.2.2.2.2 Code Verification CA Certificate

The Code Verification CA Certificate (see Table 35) 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. There MAY be more than one Code Verification CA.

Table 35/J.191 – Code Verification CA Certificate

Subject name form	C = <country>, O =, CN = CVC CA
Intended usage	This certificate is issued to 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, authorityKeyIdentifier, basicConstraints [c, m](cA = true, pathLenConstraint = 0)

11.3.2.2.2.3 Manufacturer Code Verification Certificate

This certificate (see Table 36) 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 36/J.191 – Manufacturer Code Verification Certificate

Subject name form	C = <country>, O = <CompanyName>, [S = <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.
Signed by	Code Verification CA
Validity period	2 years
Modulus length	2048
Extensions	keyUsage[c, m](digitalSignature, keyEncipherment), authorityKeyIdentifier

11.3.2.2.2.4 Code Verification Certificate

The Code Verification Certificate (see Table 37) 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 37/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 Root CA
Validity period	2 years
Modulus length	2048
Extensions	keyUsage[c, m](digitalSignature, keyEncipherment), authorityKeyIdentifier

11.3.2.2.2.5 Service Provider Code Verification Certificate

The Service Provider Code Verification Certificate (see Table 38) 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 38/J.191 – Service Provider Code Verification Certificate

Subject name form	C = <country>, O = <CompanyName>, [S = <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.
Signed by	Code Verification Root CA
Validity period	2 years
Modulus length	2048
Extensions	keyUsage[c, m](digitalSignature, keyEncipherment), authorityKeyIdentifier

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 39 through 42. 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 11.3.2 (PKI).

11.3.2.2.3.1 Service Provider Root CA Certificate

This certificate (see Table 39) 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 39/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. It is intended that the validity period is long enough so that this certificate is never re-issued.
Modulus length	2048
Extensions	keyUsage [c, m] (keyCertSign, cRL Sign), subjectkeyidentifier, basicConstraints [c, m](cA = true)

11.3.2.2.3.2 Service Provider CA certificate

The Service Provider CA certificate (see Table 40) 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 40/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 application, 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, authorityKeyIdentifier, basicConstraints[c, m](cA = true, pathLenConstraint = 1)

11.3.2.2.3.3 Local System CA Certificate

This certificate (see Table 41) 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 41/J.191 – Local System CA Certificate

Subject name form	C = <country>, O = <CompanyName>, 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	2048
Extensions	keyUsage[c, m](keyCertSign, cRLSign), subjectKeyIdentifier, authorityKeyIdentifier, basicConstraints[c, m](cA = true, pathLenConstraint = 0)

11.3.2.2.3.4 KDC certificate

This certificate (see Table 42) 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 the IPCablecom security specification, under "Key Distribution Center Certificate".

Table 42/J.191 – KDC Certificate

Subject name form	C = <country>, O = <Company Name>, [OU = <Local System Name>], OU = <KDC>, CN = <IP Address of the KDC server>
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	2048
Extensions	keyUsage[n, o](digitalSignature, keyEncipherment), authorityKeyIdentifier. The keyUsage extension is optional. When it is used it SHOULD be marked as non-critical. subjectAltName [n, m] (see IPCablecom Security specification).

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. This 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 then 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, Secure software download.

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. The Service Provider Root CA Certificate MUST NOT contain any changes to the certificate with the possible exception of the certificate serial number, validity period and the value of the signature. If changes other than the certificate serial number, validity period and the value of the signature, exist in the Service Provider Root CA Certificate that was passed over the wire in comparison to the known Service Provider Root CA Certificate, the PS element making the comparison MUST fail the certificate verification.

11.3.2.4 Certificate revocation

Certificate revocation is out of the scope of this Recommendation.

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 5.7). 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 2574]. SNMPv3 authentication MUST be turned on at all times and SNMPv3 privacy MAY also be utilized.

In order to establish SNMPv3 keys, PS compliant CM MUST support the "SNMPv3 Initialization" described below.

NOTE – The cable modem is designated as having "very-secure" security posture in the context of RFC-2574 Appendix A and RFC-2575 Appendix A. This means that default usmUser and vacmAccess entries defined in RFC-2574 Appendix A and RFC-2575 Appendix A MUST NOT be present.

1) For each of up to 5 different security names, the Manager generates a pair of numbers:
   a) Manager generates a random number Rm;
   b) Manager uses DH equation to translate Rm to a public number z:

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

      where g is from the set of Diffie-Hellman parameters, p is the prime from those parameters.
2) CM configuration file is created to include (security name, public number) pair and CM MUST support a minimum of 5 pairs. For example:

   TLV type 34.1 (SnmpV3 Kickstart Security Name) = docsisManager

   TLV type 34.2 (SnmpV3 Kickstart Public Number) = z

During the CM boot-up process, the above values (security name, public number) will (MUST) be populated in the usmDHKickstartTable.

At this point:

```
usmDHKickstartMgrpublic.1 = "z"  (octet string)
usmDHKickstartSecurityName.1 = "docsisManager"
```

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
usmUserCloneForm: 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 (CM) dhKickstart entries in usmUserTable, Permanent means it MUST be written to but not deleted and is not saved across reboots.

After the CM has registered with the AN:

– CM generates a random number xa for each row populated in the usmDHKickstartTable which has a non-zero length usmDHKickstartSecurityName and usmDHKickstartMgrPublic.

– CM 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, p is the prime from those parameters.

At this point:

```
usmDHKickstartMyPublic.1 = "c" (octet string)
usmDHKickstartMgrPublic.1 = "z" (octet string)
usmDHKickstartSecurityName.1 = "docsisManager"
```

3) CM calculate shared secret sk where sk = z ^xa mod p;

4) CM 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, "docsisManager" in this case, is derived from sk by applying the key derivation function PBKDF2 defined in PKCS#5v2.0.

```
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 CM 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.

   Compliant CM 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 CM'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.

   Compliant CM MUST provide preinstalled 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.

   SNMP manager gets the value of CM's usmDHKickstartMypublic number associated with the security name that manager wants to derive authentication and privacy keys for. With the manager's knowledge of 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 security name that the manager is going to use to communicate with the CM.

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, the DOCSIS Radio Frequency Interface specification, and implement the key-change mechanism specified in [RFC 2786] which includes the usmDHKickstartTable MIB object.

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 2574]. 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 Kerberized SNMPv3 key management as specified in 11.3.3.2.3.

11.3.3.2.1 SNMPv3 encryption algorithms

The encryption Transform Identifiers to be used by the Kerberized key management to negotiate an encryption algorithm for use by SNMPv3 are the same ones defined in 6.3.1/J.170.

11.3.3.2.2 SNMPv3 authentication algorithms

The authentication Transform Identifiers to be used by the Kerberized key management to negotiate a message authentication algorithm for use by SNMPv3 in are the same ones defined in 6.3.2/J.170.

11.3.3.2.3 Kerberized SNMPv3

The Kerberized key management profile specific for SNMPv3 is the same profile 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 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 2571], i.e., the first bit is set to 1 (one) and the appropriate value is used for the first four bytes [RFC 2571];
- 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 2571]. The above selection, though, is intended to reduce implementation complexity that would be required if all of the approaches in [RFC 2571] were allowed.

11.3.3.2.5 Populating the usmUserTable

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 user information MUST be entered in the usmUserTable [RFC 2574] 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.

The value XXXXXX will be the Element 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 Security it is necessary to ensure these IPCablecom messages, already secured by IPCablecom, can pass through the firewall in the PS. It is not within the scope of this Recommendation to add security for IPCablecom messages. Because the PS element CQoS security requirement 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 5.7).

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 PS. See clause 10. The CQP element acts as a transparent bridge for CQoS messaging between IPCablecom compliant applications and the CMTS. The firewall will need to be capable of passing IPCablecom compliant security and QoS messaging.

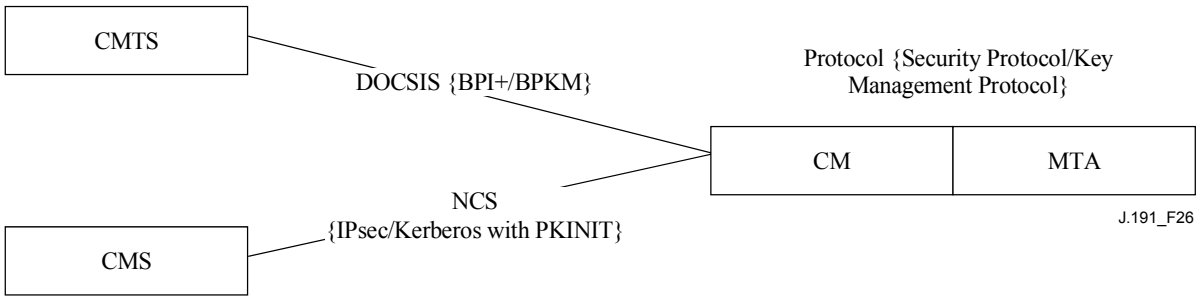
See clause 10 for more complete details on CQoS.

11.3.4.2 IPCablecom secured DQoS architecture

This clause describes the IPCablecom secured DQoS architecture in order to discuss how these messages interact with the firewall in the PS. Within DQoS, the Multimedia Terminal Adaptor (MTA) communicates with the CMTS and Call Management Server (CMS) to establish the necessary QoS for its IPCablecom services. The MTA is embedded with the DOCSIS CM. Below are a table (Table 43) and diagram (Figure 26) of the devices, the communication protocol and the security protocol for DQoS.

Table 43/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+

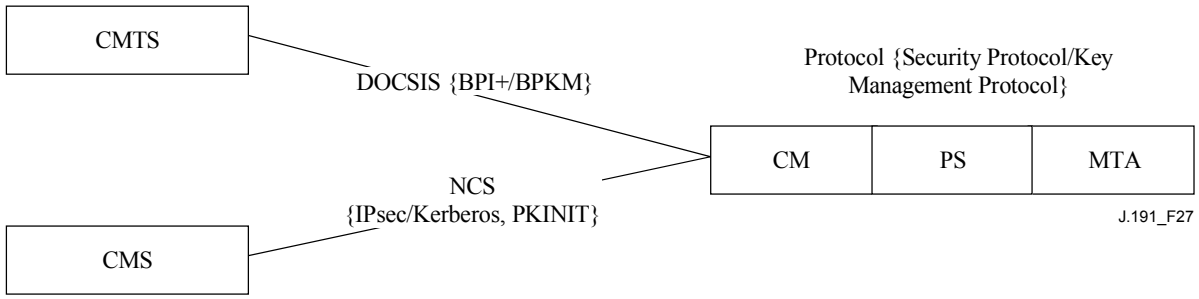


E-MTA DQoS communications

Figure 26/J.191 – Secure DQoS architecture to the MTA

11.3.4.3 CQoS Security architecture

CQoS requires IPCablecom DQoS messaging [ITU-T J.163]. All CQoS messaging MUST be secured as described in the IPCablecom security specification. Figure 27 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.



E-CM-IPSP-MTA CQoS communications

Figure 27/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 PS function in the 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".

– "A firewall is an agent which screens network traffic in some way, blocking traffic it believes to be inappropriate, dangerous, or both".

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 44) and 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 44/J.191 – Relevant IPCablecom Recommendations for firewall

Description	Recommendation
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.cmss

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.

This application-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. It SHOULD also protect against the following 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 our purposes, 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. To authenticate the rule set file download, the authentication algorithm defined in 7.3.3.3.2 MUST be used with the corresponding hash and file name management parameters defined in 11.3.5.2 below.

Using the management interface of the Security MIB, the operator configures the security policy rule set file parameters listed in 11.3.5.2 and then follows the procedure defined in 7.3.3.3.2 to download and authenticate the file. If the download is a success, the security policy rule set file MUST be "activated" on the firewall. If the authentication fails, the policy rule set MUST be discarded.

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. Once the cabhSecFwPolicyFileURL object has been updated, it MUST trigger the file download. The PS MUST use an [RFC 1350] compliant TFTP client to download the firewall configuration file.
- **cabhSecFwPolicyFileHash** – Defines the SHA-1 digest for the corresponding rule set file.
- **cabhSecFwPolicyFileOperStatus** – InProgress(1) indicates that a rule set file download is under way, either as a result of a version mismatch at provisioning or as a result of an upgradeFromMgt request. CompleteFromProvisioning(2) indicates that the last rule set file upgrade was a result of version mismatch at provisioning. CompleteFromMgt(3) indicates that the last rule set file upgrade was a result of setting the FirewallPolicyFileAdminStatus object to upgradeFromMgt. Failed(4) indicates that the last attempted download failed, ordinarily due to TFTP time-out.
- **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. 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.
- **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 the active firewall policy or firewall configuration parameters.

The choice of which types of firewall events actually get logged is configured through the Security MIB interface as described in 11.3.5.2.

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. This Event Priority field MUST be set to priority 6 for Types 1, 2, and 3 firewall 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 Types 1 and 2 firewall events exceeds an alert threshold for a given alert period (in days). The alert threshold and alert period are configurable by the operator. The PS element accumulates the number of Types 1 and 2 firewall events that have occurred over the past number days 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.
- **cabhSecFwEventType2Enable** – Enables or disables logging of type 2 firewall event messages.
- **cabhSecFwEventType3Enable** – Enables or disables logging of type 3 firewall event messages.
- **cabhSecFwEventAttackAlertThreshold** – If the number of type 1 or 2 hacker attacks exceeds this threshold in the period defined by the cabhSecFwEventAttackAlertPeriod object, a firewall event message MUST be logged with priority level 4.
- **cabhSecFwEventAttackAlertPeriod** – Indicates the period to be used in past days for the cabhSecFwEventAttackAlertThreshold object.

11.3.6 MIBs

The 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 PS MUST support the following software download support MIBs defined in ITU-T Rec. J.112, Annex B.O]:

- **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 [RSA1].
  - **docsBpi2CmDeviceCmCert** – The X.509 DER-encoded device certificate.
  - **docsBpi2CmDeviceManufCert** – The X.509 DER-encoded manufacturer CA certificate that signed the device certificate.

The PS MUST support the following configuration download support MIB:

- **cabhPsDevProvConfigHash** – SHA-1 hash of the entire content of the configuration file, taken as a byte string.

11.3.7 Secure software download

The PS element in a device MUST be capable of remotely downloading a software image over the network. 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 for 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 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 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 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 PS) 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 a 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 Public Key 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.

The optional downloading of the CVC Root CA Public Key, CVC CA Certificate, and/or the Manufacturer CA Certificate as a part of the code file makes it possible to clearly discriminate the code image from other parameters in the code download file. It also makes it possible to change the Root CVC Public Key, the CVC CA Certificate, the Manufacturer CA Certificates or SignedData parameters in the code download file without disrupting or changing the code image that the PS element will receive. This permits the PS element to verify that the code image has not been altered even though the code download file changed due to changes in the CVC Root CA Public Key, the CVC CA Certificate, the Manufacturer CA Certificates or SignedData parameters.

11.3.7.1 Software download into PS elements

Since 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.

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 PKCS#7 compliant structure that has been defined in a specific format for use with PS elements. The code file MUST comply with [PKCS#7] and MUST be DER encoded. The code file MUST match the structure shown in Table 45.

When downloading the CVC Root CA Public Key and/or CA Certificates (e.g., a CVC CA Certificate and/or a Manufacturer CA Certificate) as a part of the code file, the certificates MAY be contained in the RootCAPublicKey field and/or the CACerts fields respectively as specified in the Table 45 respectively, and separated from the actual code image contained in the CodeImage field.

Table 45/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 {	
DownloadParameters {	Mandatory TLV format (Type 28). (Length is zero if there is no sub-TLVs.)
RootCAPublicKey ()	Optional TLV for the CL CVC Root CA Public Key formatted according to RSA-Public-Key format (Type 4).
CACerts ()	Optional TLV for one or more DER-encoded CA Certificate(s) each formatted according to the CA-Certificate TLV format (Type 17).
}	
CodeImage ()	Upgrade code image
} *end SignedContent*	

11.3.7.2.1.1 Signed Data

The code download file will contain the information in a PKCS#7 Signed Data content type as shown in Table 46. Though maintaining compliance to [PKCS#7], the structure used has been restricted in format to ease the processing performed by the PS to validate the signature. The PKCS#7 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 PKCS#7 signature if the PKCS#7 Signed Data does not match the DER-encoded structure.

Table 46/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)
signing Time	UTCTime(GMT),YYMMDDhhmmssZ
messageDigest	(digest of the content as defined in [PKCS#7])
digestEncryptionAlgorithm	rsaEncryption
EncryptedDigest	
} *end mfg signer info*	
CoSignerInfo {	(OPTIONAL; required for co-signatures)
version	version = 1
issuerandserialnumber	
issuername	
CountryName	US
organizationName	CableLabs
CommonName	CableLabs CVC Root CA
certificateSerialNumber	<coSigner CVC serial number>
digestAlgorithm	SHA-1
AuthenticatedAttributes	
contentType	data (contentType of signedContent)
signing Time	UTCTime (GMT),YYMMDDhhmmssZ

Table 46/J.191 – PKCS#7 Signed Data

PKCS#7 field	Description
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 – a CVC Root CA Public Key and CA Certificates (e.g., a CVC CA Certificate and/or a 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, the CVC Root CA Public Key is intended to replace the CVC Root CA Public Key currently stored in the PS element. If the code download and installation is successful, then the PS element MUST replace its currently stored CVC Root CA Public Key with the CVC Root CA Public Key received in the signed content field. This new CVC Root CA Public key will then be used for subsequent CVC verification.

If included in the signed content field, the CA Certificate(s) is intended to replace the CA Certificate(s) currently stored in the PS element. For example, if the code download and installation is successful and the CACert contained a Manufacturer CA Certificate, then the PS element MUST replace its currently stored Manufacturer Certificate(s) with the Manufacturer Certificate(s) received in the signed content field.

11.3.7.2.1.3 Code signing keys

The PKCS#7 digital signature uses the RSA Encryption Algorithm with SHA-1 [FIPS 186]. The RSA key modulus for code signing is 2048 bits in length. The PS element MUST be able to verify code file signatures that are signed using this modulus size. The public exponent is F4 (65537 decimal).

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 47 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 47.

Table 47/J.191 – X.509-compliant code verification certificate

X.509 Certificate	Description
Certificate {	
version	2 (i.e., X.509 version 3)
serialNumber	integer, 8-octets (i.e., unique number assigned by the root CA)
signature	SHA-1 RSA, null parameters
issuer	
countryName	US
organizationName	CableLabs
commonName	CableLabs CVC Root CA
validity	
notBefore	utcTime (GMT), YYMMDDhhmmssZ (i.e., Time of issue)
notAfter	utcTime (GMT), YYMMDDhhmmssZ
subject	
countryName	<Country Name>
organizationName	<Company Name>
commonName	<Common Name>
subjectPublicKeyInfo	
algorithm	RSA encryption, null parameters
subjectPublicKey	2048-bit modulus
extensions	
KeyUsage	<Key usage>
authorityKeyIdentifier	<Authority key identifier>
signatureAlgorithm	SHA-1 RSA, null parameters
signature Value	<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 UTV 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 CVCs that are received and verified by the PS element.

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 CVC chains up to the CVC Root before accepting a code file. When the PS element's configuration file contains a valid co-signer's CVC, it is used to initialize the device with a co- signer. Once validated, the name of the CVC's subject organizationName will become the code co- signer assigned to the PS element. In order for a PS element to subsequently accept a code image, the co-signer in addition to the device manufacturer MUST have signed the code file.
- Both a Manufacturer's CVC and a co-signer's CVC. The PS element MUST verify that both CVCs chain up to the CVC Root before accepting a code file.

Before the PS element will enable its ability to upgrade code files on the network, it MUST receive a valid CVC in a configuration file. In addition, when the PS element's configuration file does not contain a valid CVC, and its ability to upgrade code files has been disabled, the PS element MUST reject any information in a CVC subsequently delivered via SNMP.

The organization name of the PS element manufacturer and the manufacturer's time-varying control values MUST always be present in the PS element. If the PS element is initialized to accept code co-signed by an additional code-signer, the name of the organization and their corresponding time-varying control values MUST be stored and maintained while operational. Space MUST be allocated in the PS element's memory for the following co-signer's control values:

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 PS 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 CVCs 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 CVC CA Public Key held by the PS element.

4) validate the CVC CA signature using the 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 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 CVC CA Public Key held by the PS element.

4) validate the CVC issuer signature using the 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. All certificate parameters EXCEPT for the validity start time are no longer needed and SHOULD be discarded.

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 CVC CA. The CVC MUST be exactly as specified in 11.3.7.3. The CVC CA MUST NOT sign any CVC unless it is identical to

the format specified in that clause. Before signing a CVC, the CVC CA MUST verify that the certificate request is authentic.

The 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 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.

To conserve storage space, the PS element MAY internally represent the code co-signer's name in an alternate format as long as all information is maintained and the original format can 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 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 reused. 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 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 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 application 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 over-written from an SNMP manager or via the PS element configuration file.

The default state of docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2} until it is over-written by ignoreProvisioningUpgrade{3} following a successful SNMP-initiated software upgrade or otherwise altered by the management station. docsDevSwOperStatus MUST persist across resets to report the outcome of the last software upgrade attempt.

If a PS element suffers a loss of power or resets during SNMP-initiated upgrade, the PS element MUST resume the upgrade without requiring manual intervention and when the PS element resumes the upgrade process:

- docsDevSwAdminStatus MUST be Upgrade-from-mgt{1};
- docsDevSwFilename MUST be the filename of the software image to be upgraded;
- docsDevSwServer MUST be the address of the TFTP server containing the software upgrade image to be upgraded;
- docsDevSwOperStatus MUST be inProgress{1};
- docsDevSwCurrentVers MUST be the current version of software that is operating on the device.

In case where the PS element reaches the maximum number of retries (max retries = 3) resulting from multiple loss of powers or resets during an SNMP-initiated upgrade, the PS element's status MUST adhere to the following requirements after it is registered:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2};
- docsDevSwFilename MUST be the filename of the software that failed the upgrade process;
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process;
- docsDevSwOperStatus MUST be other{5};
- docsDevSwCurrentVer MUST be the current version of software that is operating on the device.

If a PS element exhausts the required number of TFTP retries by issuing a total of 16 consecutive retries, the PS element MUST fall back to last known working image and proceed to an operational state and adhere to the following requirements:

- docDevSwAdminStatus MUST be allowProvisioningUpgrade{2};
- docDevSwFilename 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 docsDevOperStatus 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 device the:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2};
- docsDevSwFilename MUST be the filename of the software that failed the upgrade;
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process;
- docsDevSwOperStatus MUST be other{5};
- docsDevSwCurrentVer MUST be the current version of software that is operating on the device.

In the case where PS element determines that the download image is damaged or corrupted, the PS element MUST reject the newly downloaded image. The PS element MAY re-attempt to download if the MAX number of TFTP sequence retries has not been reached. If the PS element chooses not to retry and the MAX number of TFTP sequence 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:

- docDevSwAdminStatus MUST be allowProvisioningUpgrade{2};
- docDevSwFilename MUST be the filename of the software that failed the upgrade process;
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process;
- docsDevSwOperStatus MUST be failed{4};
- docsDevSwCurrentVer MUST be the current version of software that is operating on the device;

After the PS element has completed the configuration-file-initiated secure software upgrade, the PS element MUST reboot and become operational with the correct software image. After the PS element is registered the:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2};

- docsDevSwFilename MAY be the filename of the software currently operating on the device;
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the device;
- docsDevSwOperStatus MUST be completeFromProvisioning{2};
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the device;

11.3.7.9 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 CVC CA Public Key held by the PS element. In turn, the CVC CA Certificate signature is validated by the 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. Once trust has been established in the manufacturer's CVK, the remaining certificate parameters EXCEPT for the validity start time are no longer needed and SHOULD be discarded.
4) The PS element MUST verify the manufacturer's code file signature.
   a) The PS element MUST perform a new SHA-1 hash over the SignedContent. If the value of the messageDigest doesn't match the new hash, the PS element MUST consider the signature on the code file as invalid;
   b) If the signature does not verify, all components of the code file (including the code image), and any values derived from the verification process MUST be rejected and SHOULD be immediately discarded.
5) If the manufacturer signature verifies and a co-signing agent signature is required:
   a) The PS element MUST validate the co-signer's signature information by verifying that the:
      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 CVC CA Public Key held by the PS element. In turn, the CVC CA certificate signature is validated by the 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. Once trust has been established in the co-signer's CVK, the remaining certificate parameters EXCEPT for the validity start time are no longer needed and SHOULD be discarded.

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 may 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 will 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 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 application 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, standard [FIPS 140-2]. In particular, 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.

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:
  - Remote Connection Speed Test;
  - Remote Ping Test.
- 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 CableHome Testing Portal (CTP) provides Remote Connection Speed Test and Remote Ping Test 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 (see Figure 28).

- 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 time-out 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.

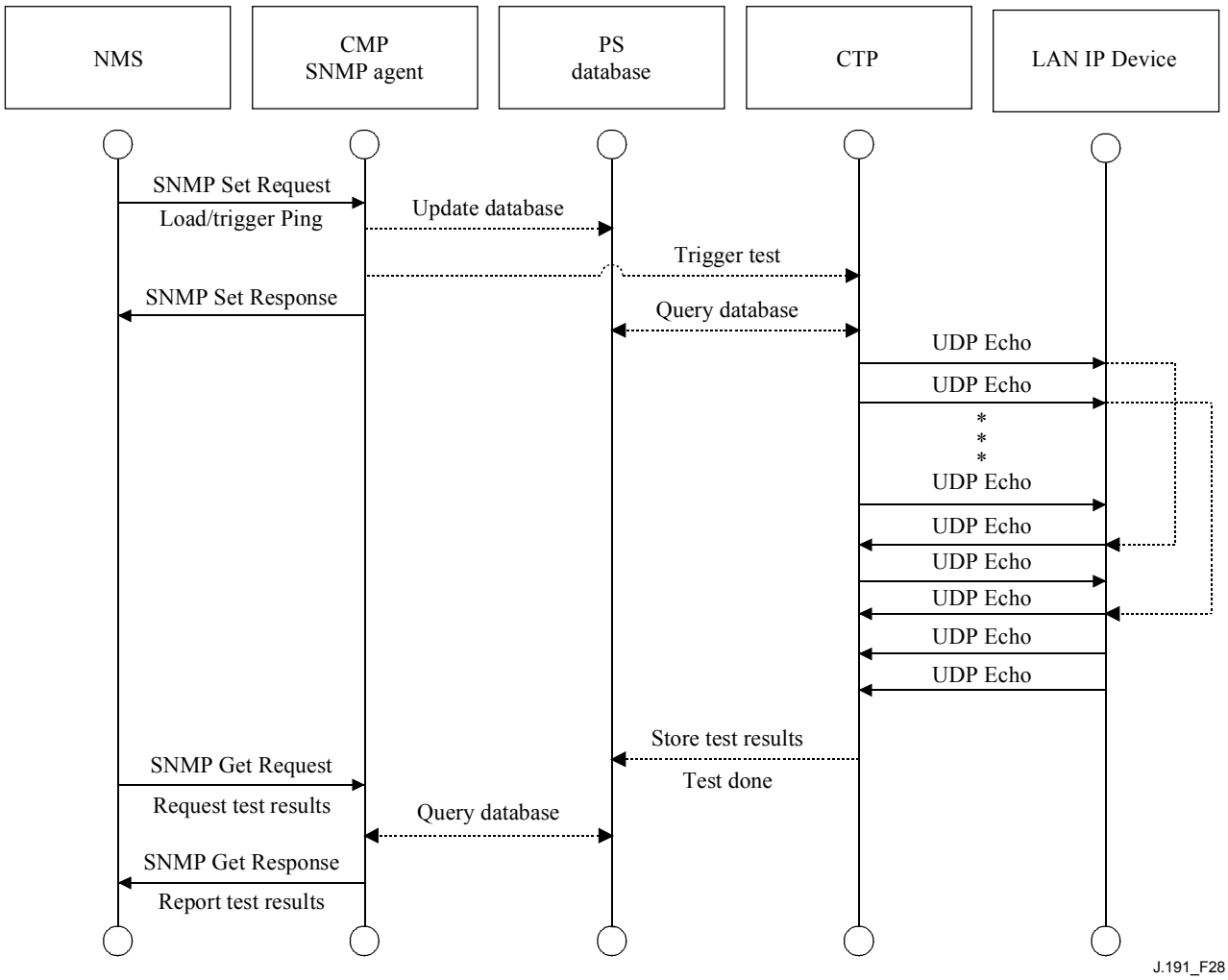


Figure 28/J.191 – Connection speed test sequence diagram

12.2.1.2 Remote Ping Test

The Remote Ping Test can be useful in validating connectivity state, performance levels, and identifying possible configuration errors (see Figure 29).

- 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.

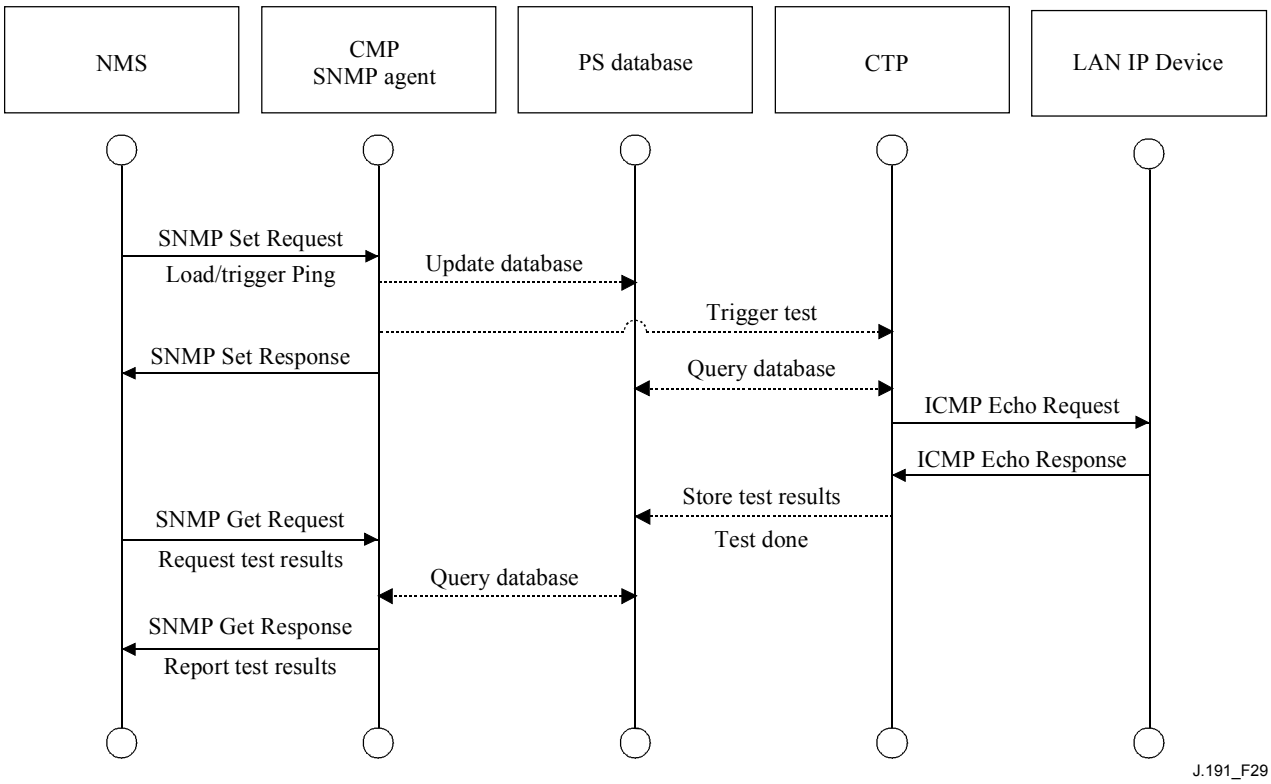


Figure 29/J.191 – Remote ping test sequence diagram

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 Remote Ping tests) executed by the NMS issuing SNMP Set Request messages with the appropriate parameters, to the PS WAN-Man address.

Figure 30 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.

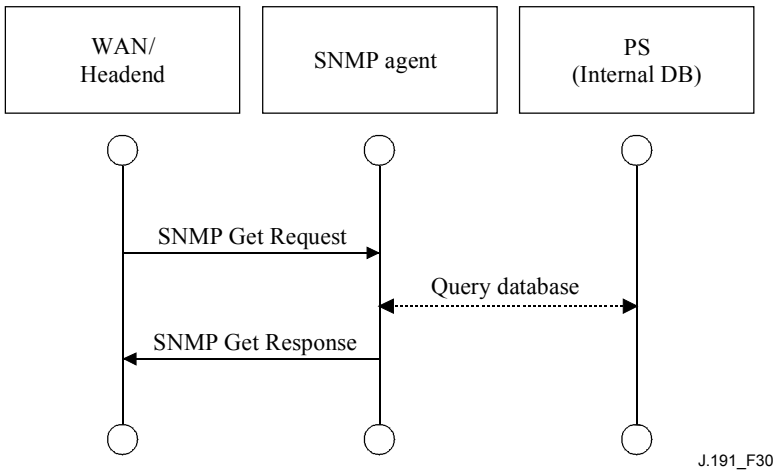


Figure 30/J.191 – PS database access from the PS WAN-Man interface sequence diagram

12.3.2 Reconfiguration

12.3.2.1 PS Software Download

The following example in Figure 31 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.

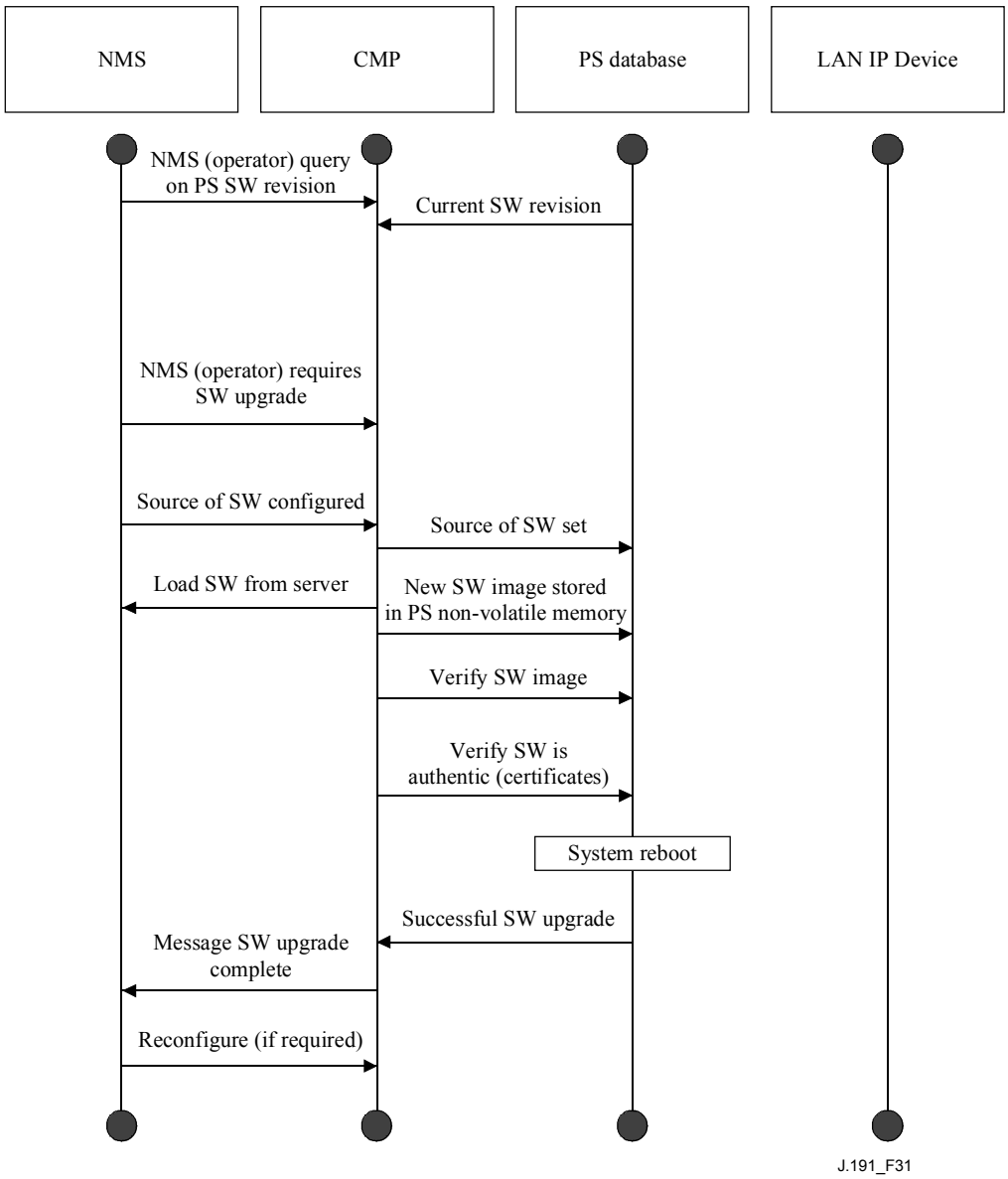


Figure 31/J.191 – PS Software Download sequence diagram

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.

12.3.2.2 PS Configuration File Download

The following example in Figure 32 illustrates a reconfiguration of a PS in SNMP Provisioning Mode, via configuration 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 configuration file, it will record the number of TLV tuples processed and skipped in the appropriate MIB objects.

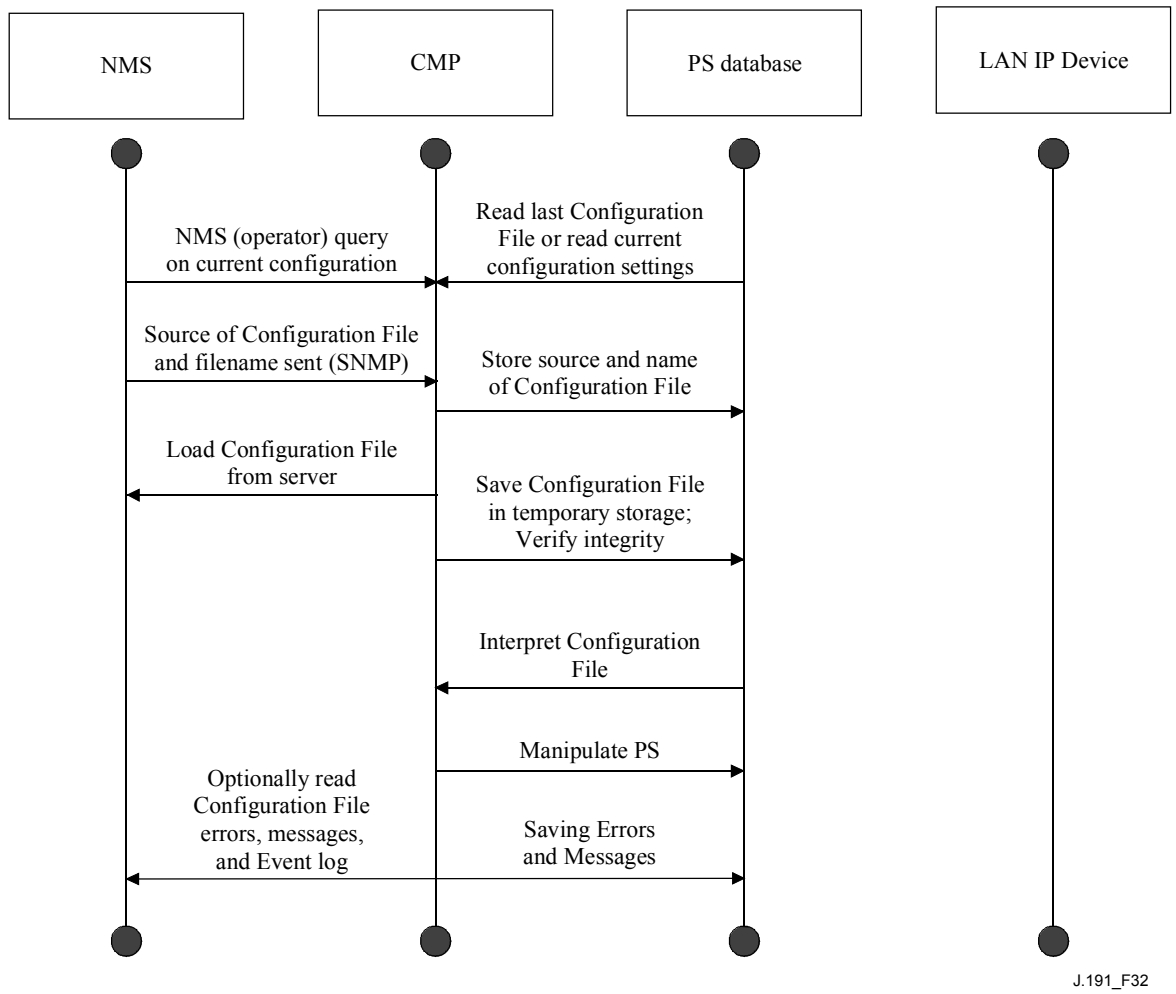


Figure 32/J.191 – PS reconfiguration (Configuration File Download) sequence diagram

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 33.

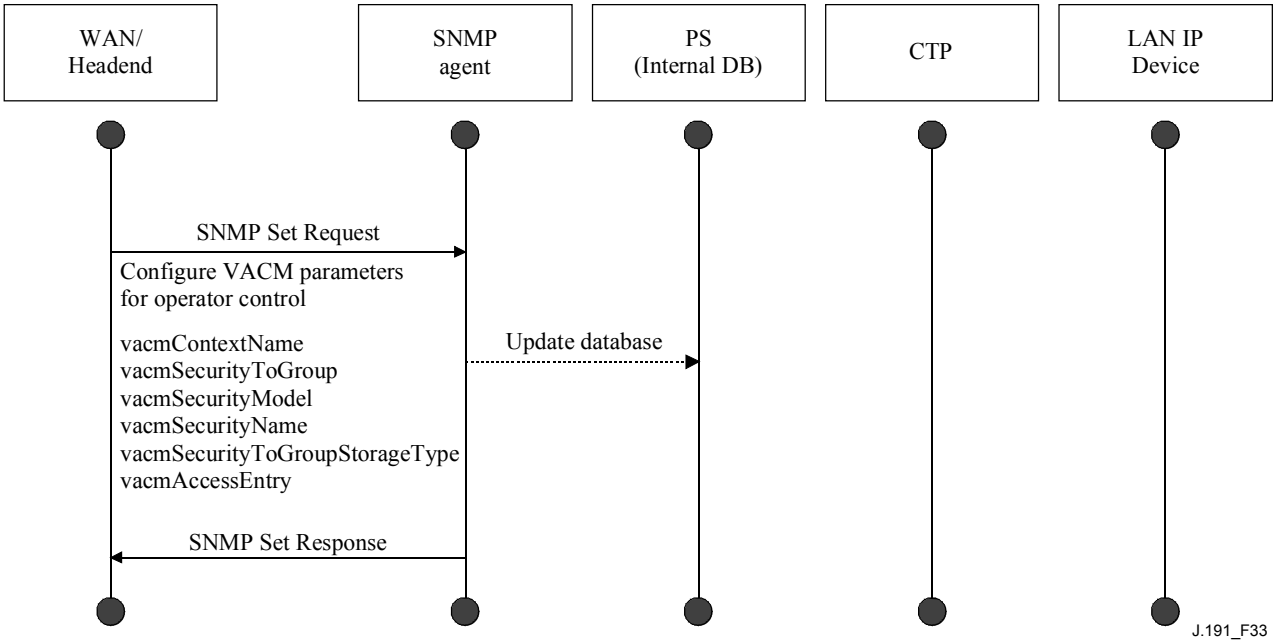


Figure 33/J.191 – PS configuration (VACM Parameters) sequence

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.

The following example in Figure 34 illustrates configuring the PS database to store events in local log files. Local log events are of two types: local non-volatile and local volatile. The NMS will read the content of the local log and write that content to the headend event logging system. A PS reboot causes only the volatile events to be cleared from the PS database. Nonvolatile events persist across reboots.

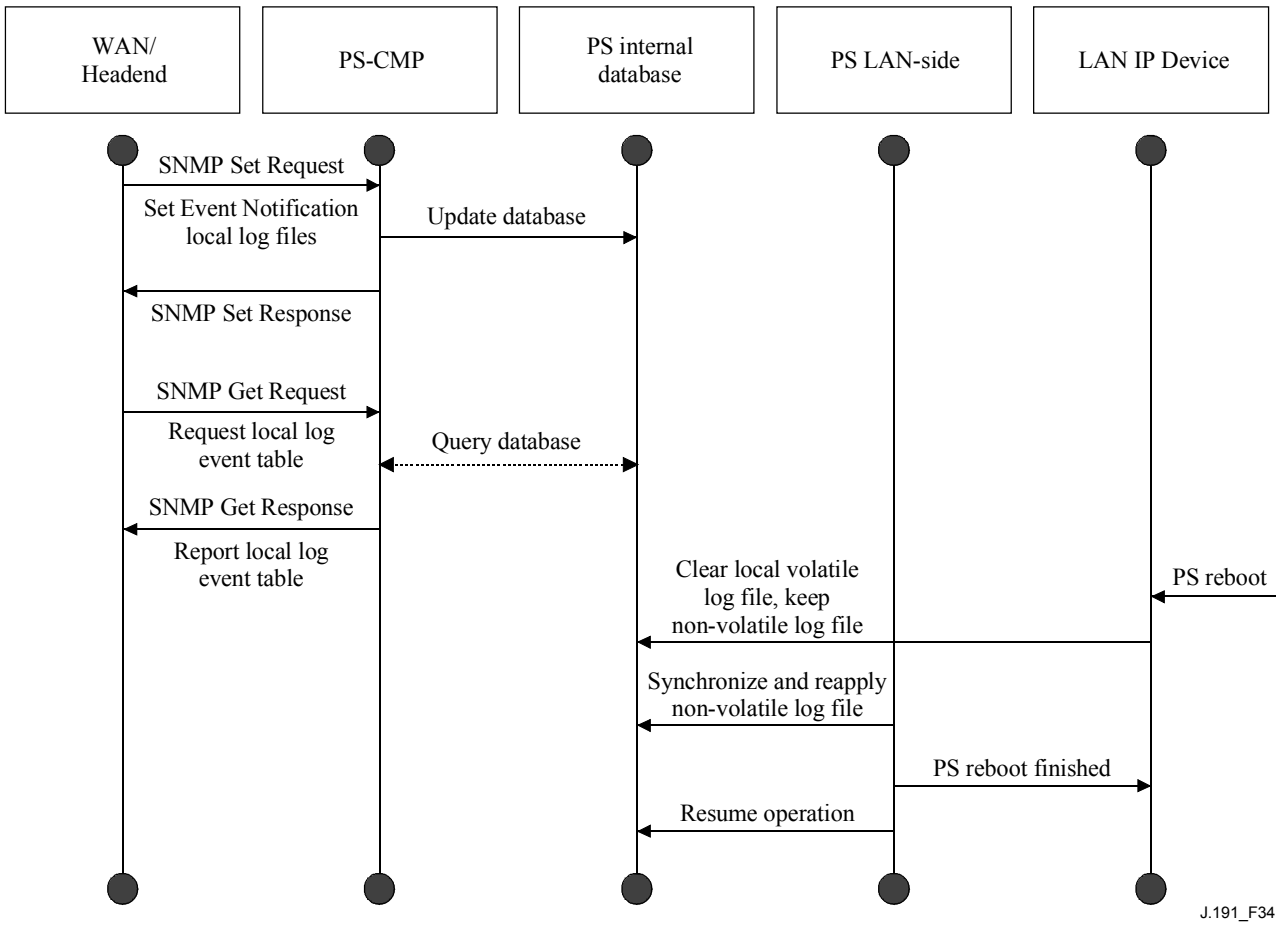


Figure 34/J.191 – PS configuration (event control) sequence

The next scenario (Figure 35) 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.

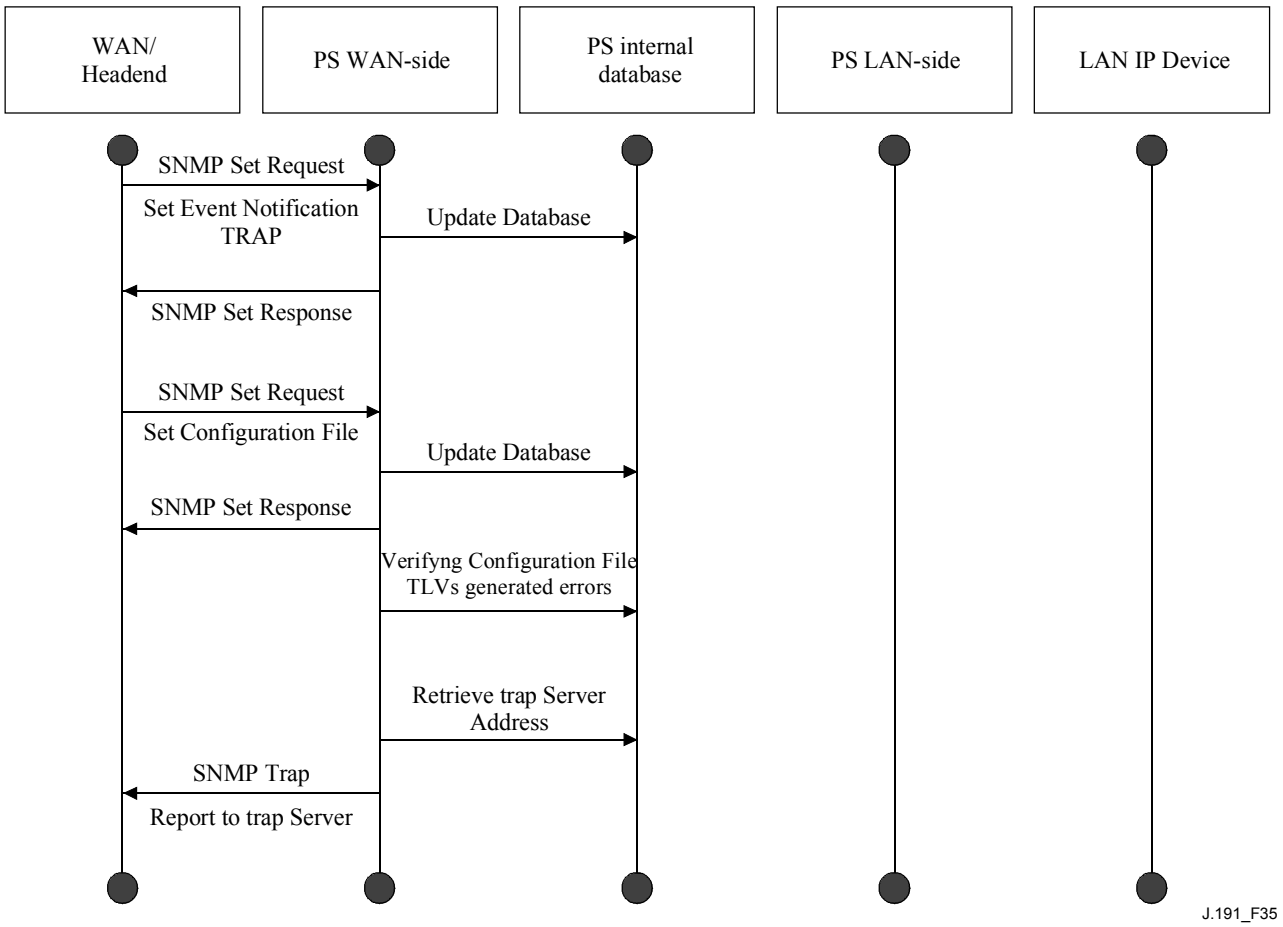


Figure 35/J.191 – PS Configuration File download (with invalid TLVs) sequence

The next example (Figure 36) 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.

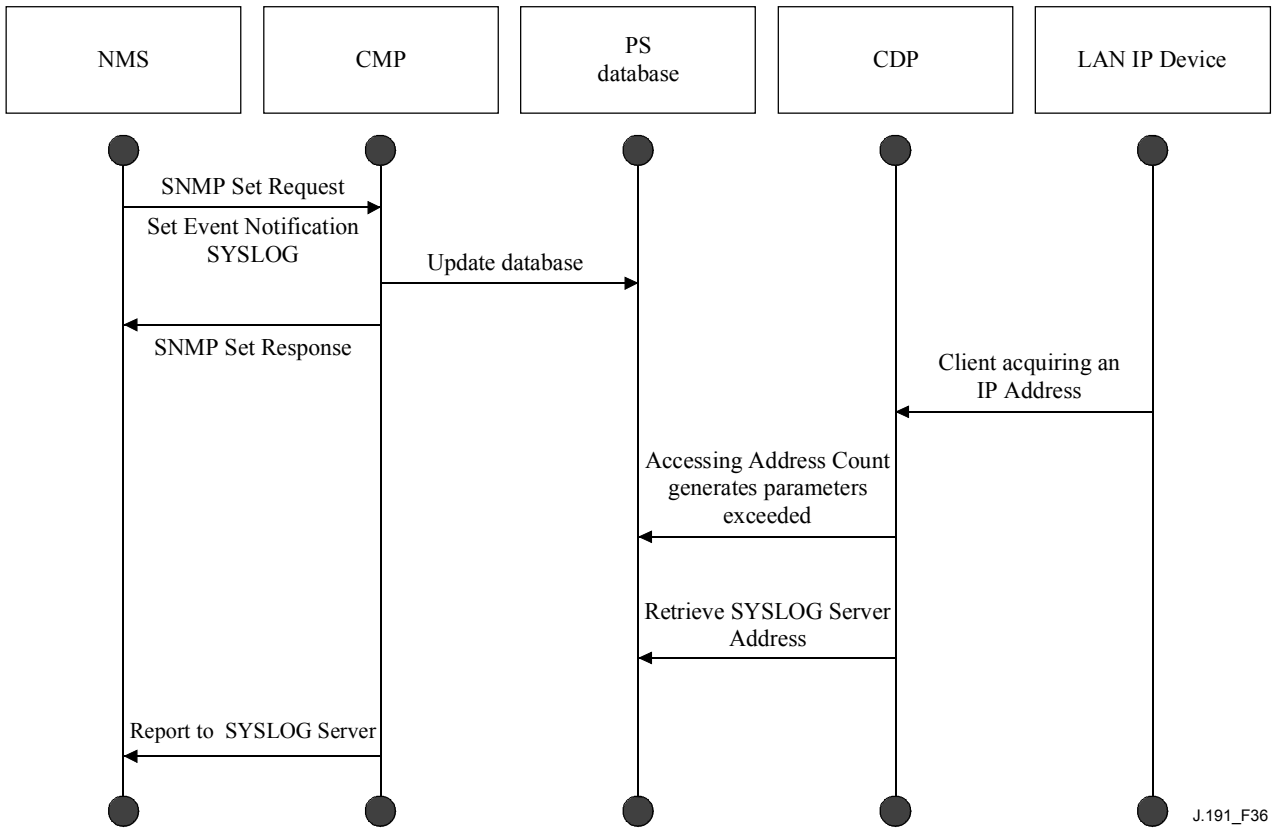


Figure 36/J.191 – LAN IP device address acquisition (request exceeds provisioned count) sequence

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 Figure 37 for a description of the CMP Event Throttling and Limiting mechanism.

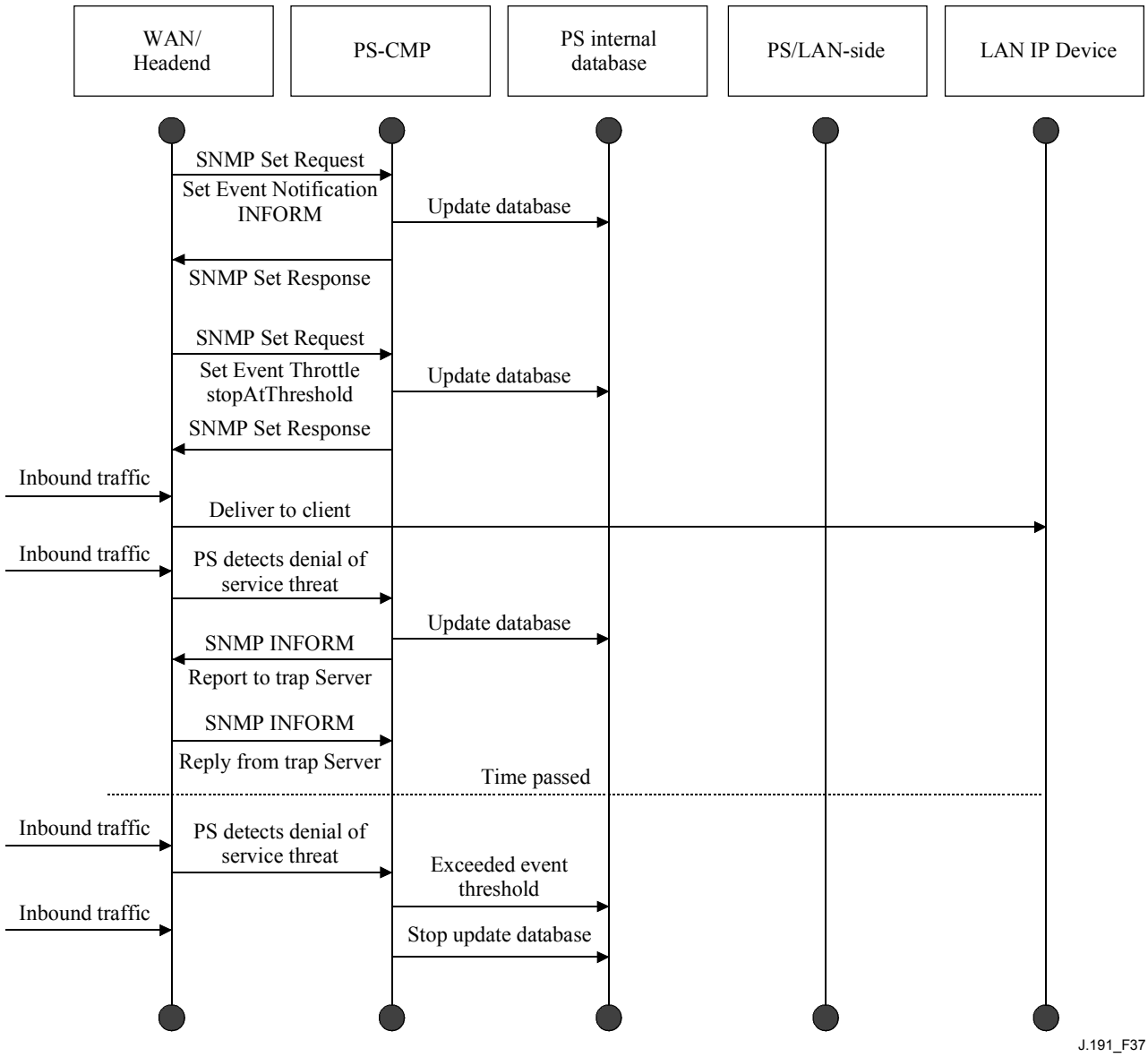


Figure 37/J.191 – CMP event throttling and limiting operation

The example shown in Figure 37 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.

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. 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 functionality is separate and distinct from PS provisioning, and is out of scope of this Recommendation. The reader is referred to DOCSIS specifications for descriptions of cable modem provisioning.

The functional elements with which the PS element interacts during the provisioning processes listed above are identified in Figure 38. The Key Distribution Center (KDC) functional element is shown with a broken outline since it is used in SNMP Provisioning Mode but not in DHCP Provisioning Mode. The other functional elements are used in both provisioning modes.

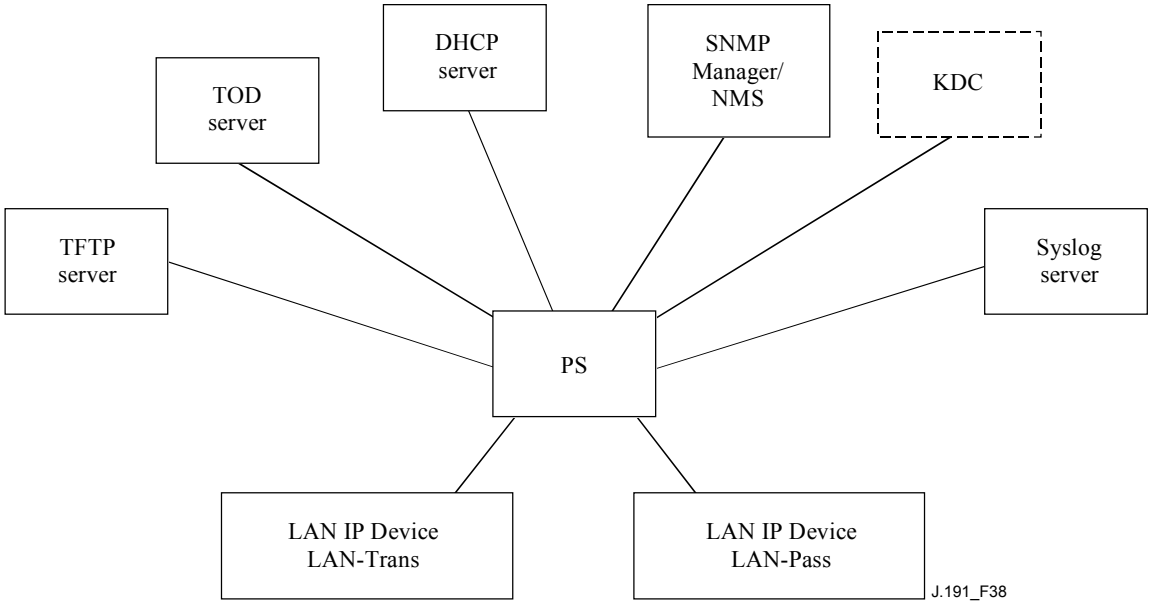


Figure 38/J.191 – Provisioning functional elements

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 2571], [RFC 2572], [RFC 2574], and [RFC 2575]. The Key Distribution Center (KDC) manages authorization and encryption keys for establishing trust between networked elements, and implements rules defined in [RFC 1949]. The System Log (SYSLOG) server handles event messages generated by the PS and by LAN IP Devices in the home. The PS implements clients for these 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 5.7 and 7.1.1 introduce two provisioning modes supported by the Portal Service element: DHCP Provisioning Mode and SNMP Provisioning Mode. In this clause, each of the two modes is presented in more detail. Figure 39 illustrates a possible event flow for the two provisioning modes. The key point of Figure 39 is the switch used by the PS to determine the provisioning mode in which it is to operate.

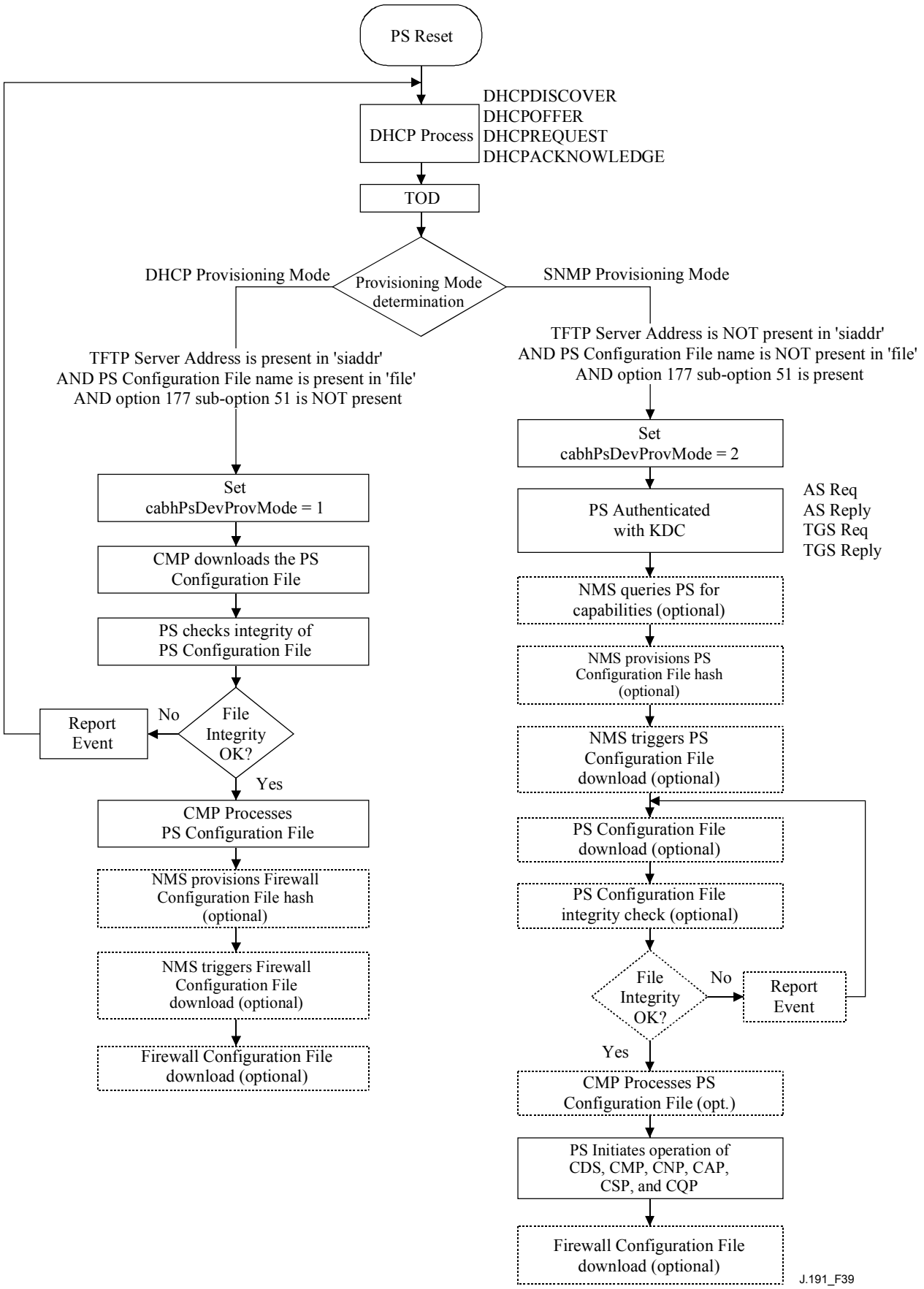


Figure 39/J.191 – Provisioning modes

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 sub-option 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 an infrastructure that does not include advanced IPCablecom features.

SNMP Provisioning Mode in the PS is triggered when the DHCP server in the cable network does NOT provide values for 'siaddr' and 'file', and when the cable network DHCP server DOES send DHCP option 177 sub-option 51. SNMP Provisioning Mode is intended to enable the PS to take advantage of advanced features of a IPCablecom infrastructure.

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 7.2.2.2.1 describes two 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. After the initial PS Configuration File download triggered by the DHCP message fields, the NMS may initiate post-provisioning configuration by issuing an SNMP Set Request to cabhPsDevProvConfigHash and cabhPsDevProvConfigFile MIB object as described in 7.3.

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 40 and Table 48 describe the sequence of messages needed to initialize a PS operating in DHCP Provisioning Mode. The provisioning for the PS MUST NOT occur before the cable modem provisioning process.

The optional process of downloading a Firewall Configuration File is shown with shading in Figure 40.

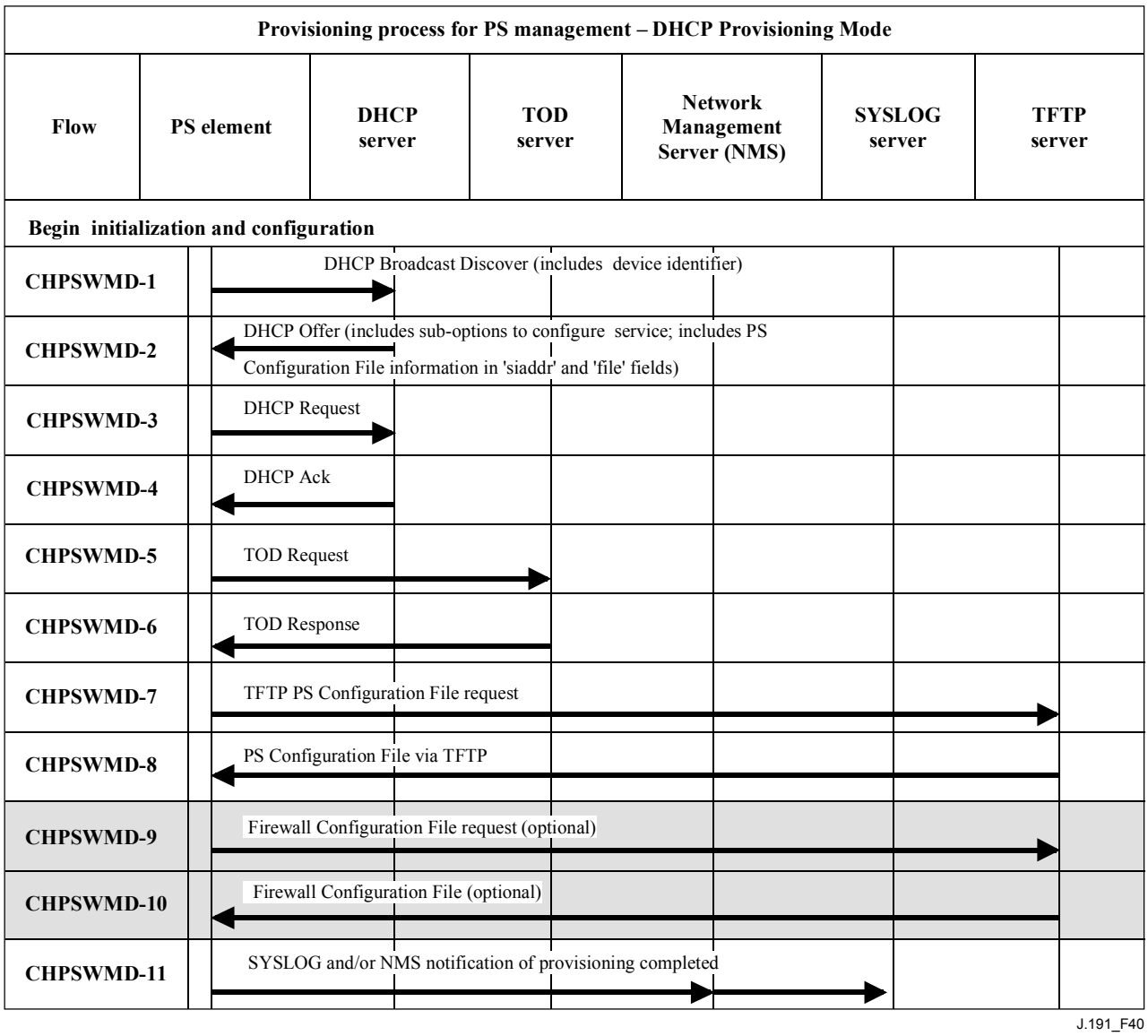


Figure 40/J.191 – Provisioning process for PS management – DHCP provisioning mode

Table 48 describes the individual messages CHPSWMD-1 to CHPSWMD-11 shown in Figure 40.

Table 48/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) MUST send a broadcast DHCP DISCOVER message. The DHCP DISCOVER broadcast by the CDP (CDC) MUST include mandatory options listed in Table 21. 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). After 5 retries 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 option code 177 with sub-option 51 AND is expected to include 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 OFFER (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 MUST store 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.
CHPSWMD-5	*Time of Day (TOD) Request per [RFC 868]* The PS MUST issue 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).

Table 48/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-7	*TFTP Request* The PS operating in DHCP Provisioning Mode MUST send 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 of the Configuration File is calculated and compared to the value appended to the PS Configuration File name (see 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/FQDN of the firewall Configuration File TFTP server, the firewall Configuration File filename, the hash of the firewall Configuration File, and the encryption key (if the firewall Configuration File is encrypted) are included in the PS Configuration File if there is a firewall Configuration File to be loaded, and this is the method selected to specify it.	CHPSWM-8 MUST occur after CHPSWM-7 completion.	If the TFTP download fails, report an error and return to CHPSWMD-7 (continue to retry PS Configuration File download). If processing of the PS Configuration File produces an error, continue with CHPSWMD-9 and report the error as an event. If the Provisioning Timer expires before PS Configuration File is successfully downloaded, the PS MUST report an error and return to CHPSWMD-1.
CHPSWMD-9	*TFTP Request – Firewall Configuration File (Optional)* If the PS receives firewall Configuration File information (firewall TFTP server and firewall Configuration File name) in the PS Configuration File, the PS sends the firewall Configuration TFTP server a TFTP Get Request to request a firewall Configuration File (see 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 CHPSMWD-10 and continue with step CHPSWMD-11.	If CHPSWMD-9 occurs, it MUST occur after CHPSWMD-8 completion.	If TFTP fails, continue with PS operation but report an error and continue to retry CHPSWMD-9.

Table 48/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-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. If encrypted, the file is decrypted. 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 48/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-11	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 the PS received valid parameters for docsDevNmAccessGroup identifying the Trap Receiver (docsDevNmAccessIP) and configuring the provisioning complete trap (cabhPsDevInitTrap) for 'read only with Traps' (set docsDevNmAccess control to '4'. Refer to [RFC 2669].), the PS MUST send a provisioning failed trap (cabhPsDevInitRetryTrap) to the Trap 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'.		

The PS Provisioning Timer MUST NOT be reset to the starting value from cabhPsDevProvTimer until the PS Provisioning Timer expires AND the value of cabhPsDevProvState is still inProgress (2) OR the PS is reset.

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 7.2.3.3 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 41 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 WAN-Man interface of a PS operating in SNMP Provisioning Mode MUST occur via the sequence depicted in Figure 41 and described in detail in Table 49. Optional steps are shown with a shaded background in Figure 41. These optional steps may be done immediately following step CHPSWMS-15, at a later time, or not at all.

Table 49 describes the individual steps of the provisioning process depicted in Figure 41.

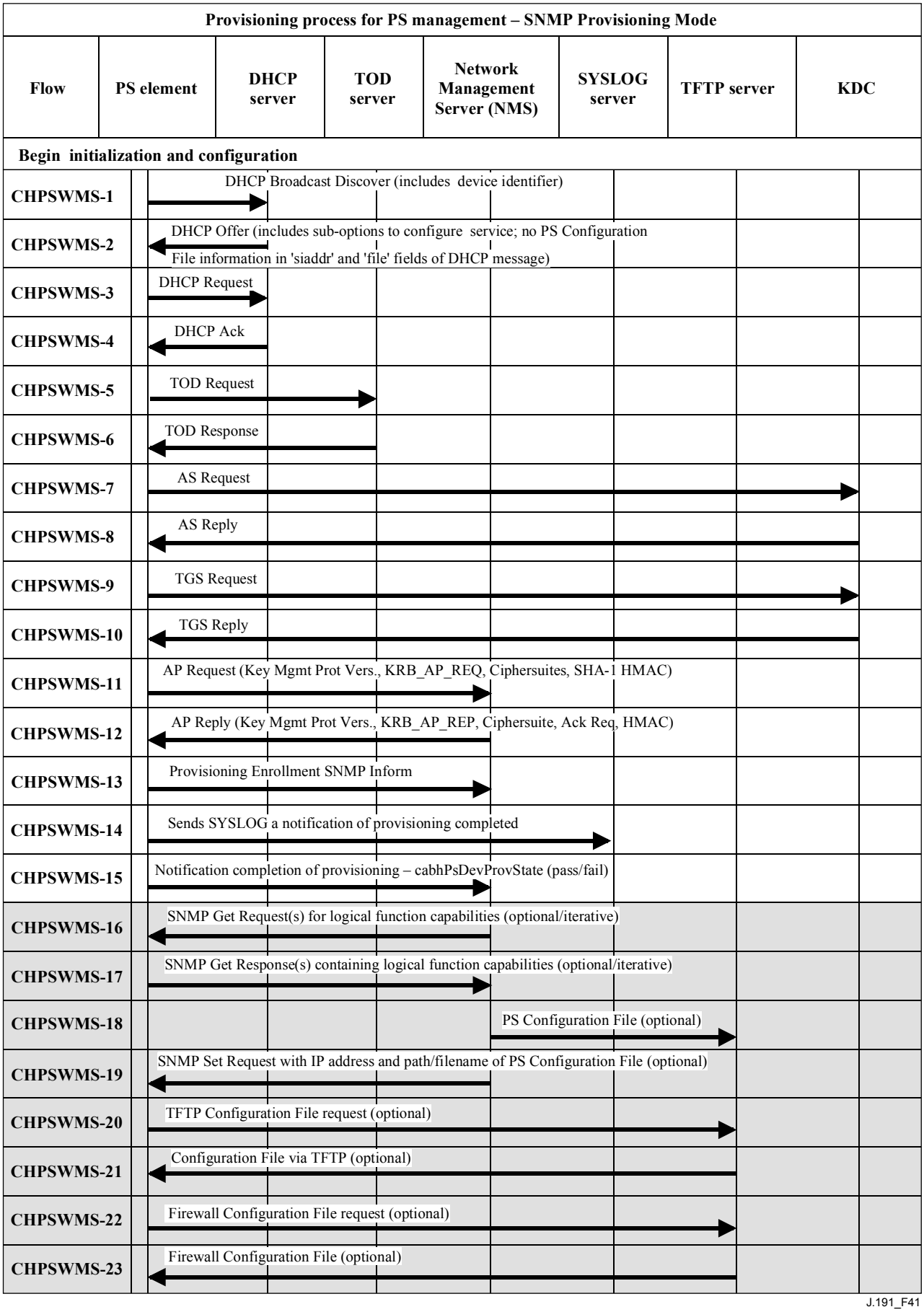


Figure 41/J.191 – Provisioning process for PS management – SNMP provisioning mode

Table 49/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) MUST send a broadcast DHCP DISCOVER message. The DHCP DISCOVER broadcast by the CDP (CDC) MUST include mandatory options listed in Table 21. 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 failure per DHCP protocol, report an error and continue to retry DHCP Broadcast Discover until successful (return to CHPSWMS-1). After 5 retries the PS initiates 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 sub-option 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 OFFER (see 7.2.3.3).	CHPSWMS-2 MUST occur after CHPSWMS-1 completion.	If failure per DHCP protocol, return to CHPSWMS-1 and report an error.
CHPSWMS-3	*DHCP REQUEST* The CDP MUST send 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 CDP a DHCP ACK message which contains the IPv4 address of the PS. The PS MUST store the Time of Day server address in cabhPsDevTimeServerAddr.	CHPSWMS-4 MUST occur after CHPSWMS-3 completion.	If failure per DHCP protocol, return to CHPSWMS-1 and report an error.
CHPSWMS-5	*Time of Day (TOD) Request per [RFC 868]* The PS sends a TOD request to the address stored in cabhPsDevServerTime as required in 7.4.2	CHPSWMS-5 MUST occur after CHPSWMS-4 completion.	Continue with CHPSWMS-6.

Table 49/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-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*[a)] The PS MUST send the AS Request message to the operator KDC to request a Kerberos ticket.	CHPSWMS-7 MUST occur after CHPSWMS-5 completion. CHPSWMS-7 MAY occur before CHPSWMS-6 completion.	Return to CHPSWMS-1.
CHPSWMS-8	*AS Reply* The AS Reply Message is received from the operator KDC containing the Kerberos ticket.	CHPSWMS-8 MUST occur after CHPSWMS-7 completion.	Return to CHPSWMS-1.
CHPSWMS-9	*TGS Request* If PS obtained Ticket Granting Ticket (TGT) in PS WAN-Man Interface provisioning process step CHPSWMS-10, the TGS Request message MUST be sent to the operator KDC.	CHPSWMS-9 MUST occur after CHPSWMS-8 completion.	Return to CHPSWMS-1.
CHPSWMS-10	*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.
CHPSWMS-11	*AP Request* The AP Request message MUST be sent to the Provisioning Server to request the keying information for SNMPv3.	CHPSWMS-11 MUST occur after CHPSWMS-10 completion.	Return to CHPSWMS-1.
CHPSWMS-12	*AP Reply* The AP Reply message is received from the Provisioning Server containing the keying information for SNMPv3. NOTE – The SNMPv3 keys MUST be established and the associated SNMPv3 tables populated before the next step. The keys and tables are established using the information in the AP Reply.	CHPSWMS-12 MUST occur after CHPSWMS-11 completion.	Return to CHPSWMS-1.

Table 49/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-13	*SNMP Inform* The PS MUST send the NMS an SNMPv3 INFORM (cabhPsDevProvEnrollTrap) requesting enrollment. The IP address of this PROVISIONING SNMP ENTITY is contained in the DHCP OFFER message.	CHPSWMS-13 MUST occur after CHPSWMS-12 completion.	Return to CHPSWMS-1.
CHPSWMS-14	*SYSLOG notification* If the PS received a SYSLOG server address in the DHCP OFFER, the PS MUST send the SYSLOG a "provisioning complete" notification. This notification will include the pass-fail result of the provisioning operation. The general format of this notification is as defined in 6.5.1.	CHPSWMS-14 MUST occur after CHPSWMS-13 completion.	
CHPSWMS-15	*SNMP Inform* The PS MUST send the NMS an SNMP INFORM (cabhPsDevInitTrap) containing a "provisioning complete" notification. FAIL occurs when the Configuration File processing fails. Otherwise the provisioning state is PASS. The PS MUST update the value of cabhPsDevProvState with status 'pass' (1) when provisioning flow steps CHPSWMS-1 through CHPSWMS-23 complete successfully. The PS MUST update the value of cabhPsDevProvState with status 'fail' (3) AND report an event indicating provisioning process failure if the PS Provisioning Timer expires before the value of cabhPsDevProvState is updated with status 'pass'.	CHPSWMS-15 MUST occur after CHPSWMS-14 completion.	If the SNMP Inform fails, the provisioning server may not know the provisioning process has completed unless it polls the cabhPsProvisioningState object.

Table 49/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
	Optional Steps		
CHPSWMS-16	*SNMP Get*[b)] If any additional device capabilities are needed by the provisioning system, the provisioning system requests these from the PS via SNMPv3 Get Requests. (Iterative:) The NMS sends the PS one or more SNMPv3 GET requests to obtain any needed PS capability information. The Provisioning Application may use a GETBulk request to obtain several pieces of information in a single message.	CHPSWMS-16 is not expected to occur before CHPSWMS-15 completion.	Return to CHPSWMS-1.
CHPSWMS-17	*SNMP Get Response* (Iterative:) The PS MUST reply to the NMS Get Request or Get Bulk request messages with a Get Response for each Get Request.After all the Gets, or the Get Bulk, finish, the NMS sends the requested data to the provisioning application.	CHPSWMS-17 MUST occur after CHPSWMS-16 completion.	N/A
CHPSWMS-18	*Configuration File Create* (Optional:) The provisioning system uses information from PS provisioning steps CHPSWMS-14 and CHPSWMS-15 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.	CHPSWMS-18 MUST occur after CHPSWMS-17 completion.	N/A

Table 49/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-19	*SNMP Set* The provisioning system might instruct the NMS to send an SNMP Set message to the PS containing the IP Address of the TFTP server, the PS Configuration File filename and the hash of the configuration file as described in 7.3.3.2 (SNMP Provisioning Mode). Optionally, the IP Address of the Firewall Configuration File TFTP server, the Firewall Configuration File filename, the hash of the Firewall Configuration File, and the encryption key (if the Firewall Configuration File is encrypted) are included in the SNMP Set if there is a Firewall Configuration File to be loaded, and this method is selected to specify it.	CHPSWMS-19 MUST occur after CHPSWMS-18 completion.	Return to CHPSWMS-1 if the Set was received, but there was a processing error.
CHPSWMS-20	*TFTP Request* If the NMS triggers the PS to download a PS Configuration File as described in 7.3.3.2, the PS MUST send the TFTP server a TFTP Get Request to request the specified PS Configuration File.	CHPSWMS-20 MUST occur after CHPSWMS-19 completion.	Continue with CHPSWMS-21.
CHPSWMS-21	*TFTP server sends Configuration File* After the PS receives the PS Configuration File, the PS calculates the hash of the PS Configuration File and compares it to the value received in step CHPSWMS-19. The PS then processes the PS Configuration File. Refer to 7.3.3 for PS Configuration File contents. Optionally, the IP Address/FQDN of the Firewall Configuration File TFTP server, the Firewall Configuration File filename, the hash of the Firewall Configuration File, and the encryption key (if the Firewall Configuration File is encrypted) 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.	CHPSWMS-21 MUST occur after CHPSWMS-20 completion.	If the TFTP download fails, report an error, proceed to CHPSWMS-23, and continue to retry CHPSWMS-20 (continue to retry PS Configuration File download). If processing of the Configuration File produces an error, continue and report the error as an event.
CHPSWMS-22	*TFTP Request – Firewall Configuration File* (Optional) The PS sends the Firewall Configuration TFTP server a TFTP Get Request to request the specified Firewall Configuration data file.	If CHPSWMS-22 occurs, it MUST occur after CHPSWMS-21 completion.	Return to CHPSWMS-1.

Table 49/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-23	*TFTP server sends Firewall Configuration File* The TFTP server sends the PS a TFTP Response containing the requested file. After the PS receives the Firewall Configuration File, the PS calculates the hash of the Firewall Configuration File and compares it to the value received in step CHPSWMS-21. If encrypted, the file is decrypted. The file is then processed. Refer to 7.3.3 for description of PS Configuration File contents.	CHPSWMS-23 MUST occur after CHPSWMS-22 completion.	If the TFTP download fails, continue with PS operation but report an error and continue to retry CHPSWMS-22. If processing of the firewall configuration file produces an error, continue and report the error as an event.
[a] Steps CHPSWMS-7-CHPSWMS-10 are optional in some cases. Refer to clause 11 for details. [b] The SNMP Get and following SNMP Get Response operations are optional, depending on whether additional information is required to form a PS Configuration File, and also depending on whether a PS Configuration File is needed.			

13.3.1 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.

DHCP Provisioning Mode

The provisioning timer MUST begin counting down when step CHPSWMD-1 begins. If the PS Provisioning Timer expires before step CHPSWMD-12 is executed, the CDC MUST set cabhPsDevProvState to status '3' (failure), the provisioning process MUST return to step CHPSWMD-1, AND the PS must generate the appropriate event, and reset the PS Provisioning Timer to the value of cabhPsDevProvTimer.

SNMP Provisioning Mode

The provisioning timer MUST begin counting down when step CHPSWMS-1 begins. If the PS Provisioning Timer expires before step CHPSWMS-23 is executed, the CDC MUST set cabhPsDevProvState to status '3' (failure), the provisioning process MUST return to step CHPSWMS-1, the PS MUST report the appropriate event, AND the PS MUST reset the PS Provisioning Timer to the value of cabhPsDevProvTimer.

13.3.3 Provisioning enrollment/provisioning complete informs

For the PS operating in SNMP Provisioning Mode only, the provisioning enrollment inform, defined in Annex B, 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), defined in Annex B, indicates whether the provisioning sequence has completed successfully or not.

13.4 SYSLOG provisioning

The syslog server IP address MUST be provisioned through the DHCP process. The syslog event will not be sent if the syslog server IP address is not configured.

13.4.1 Provisioning state and error reporting

As indicated in Tables 48 and 49, failure of the steps in the provisioning process generally results in the process restarting at the first step, CHPSWMD-1 or CHPSWMS-1.

13.5 PS WAN-Data provisioning process

The PS requests zero or more WAN-Data network address(es) from the DHCP server in the cable network to be used for the exchange of data between elements connected to the Internet and LAN IP Devices.

There is no difference in PS WAN-Data operation between the DHCP and SNMP Provisioning Modes.

The following diagrams illustrate the message flows that are to be used to accomplish the provisioning of PS WAN-Data addresses.

If the provisioning process for the PS WAN-Data address(es) occurs, it MUST follow the sequence depicted in Figure 42 and described in detail in Table 50.

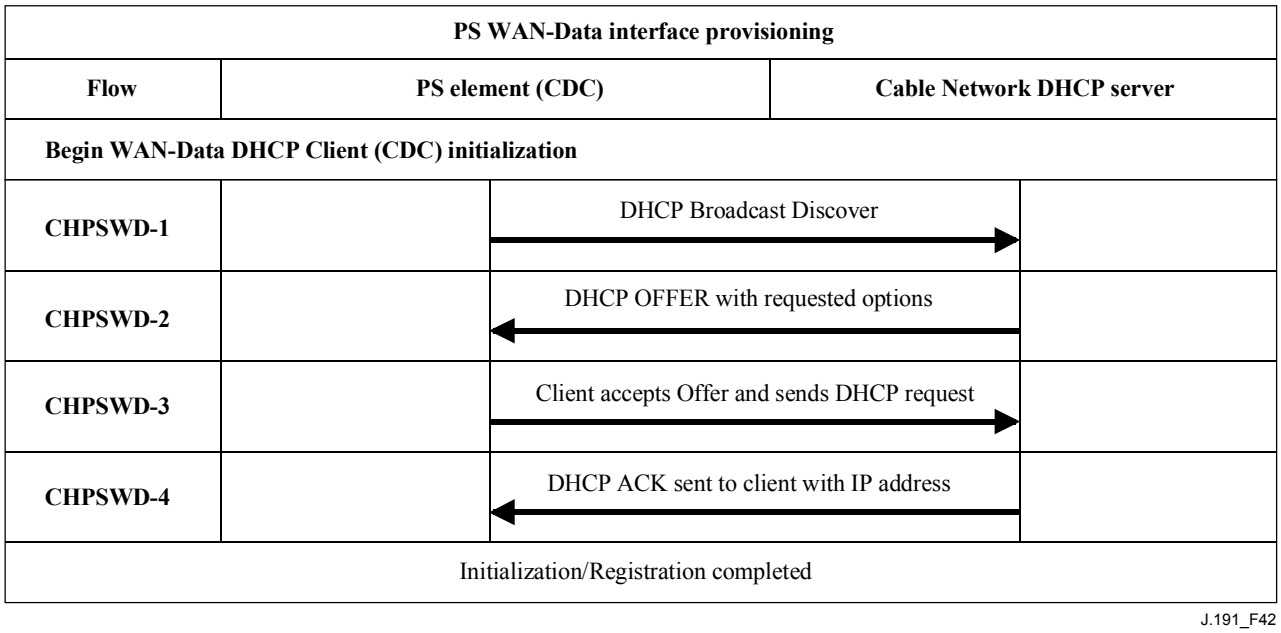


Figure 42/J.191 – PS WAN-data provisioning process

Table 50/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 MUST send a broadcast DHCP DISCOVER message including the mandatory options listed in Table 21.	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 MUST send to the appropriate DHCP server a DHCP REQUEST message 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.

13.6 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 43. Additional detail about the process is provided in Table 51.

The provisioning process for the LAN IP Device in the LAN-Trans realm MUST occur via the sequence depicted in Figure 43 and described in detail in Table 51.

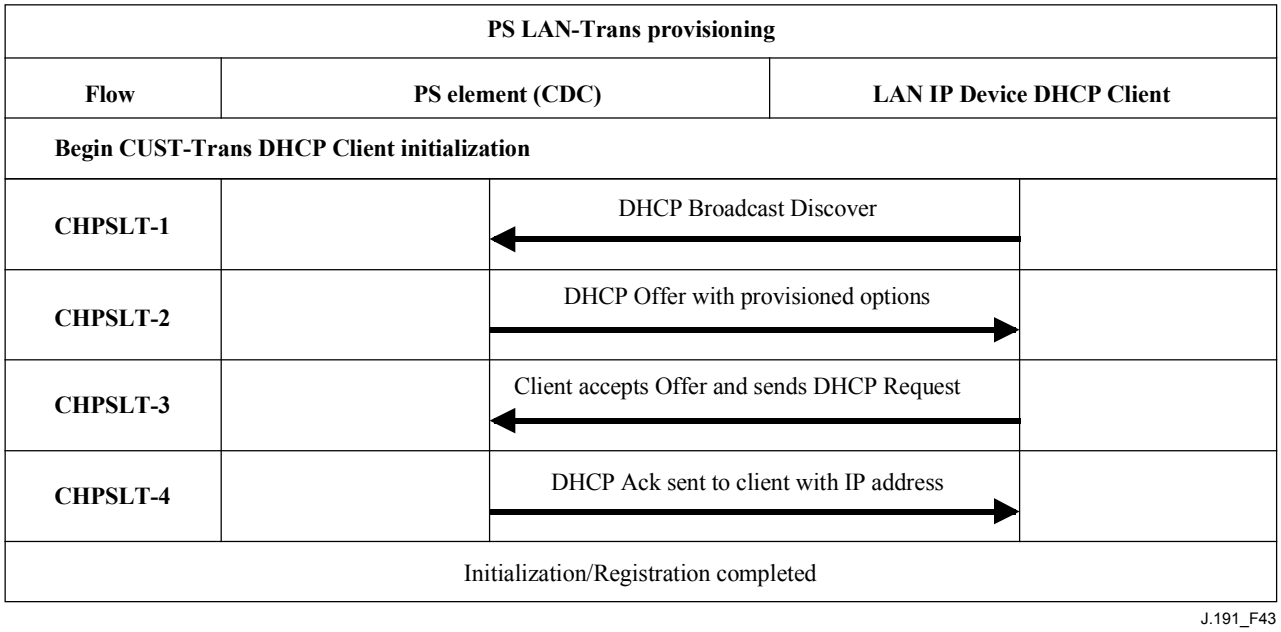


Figure 43/J.191 – Provisioning process for LAN IP device in LAN-Trans realm

Table 51/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[a] sends a broadcast DHCP DISCOVER message on its local LAN[b].	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[c].	Proceed to CHPSLT-4.	If failure, the client will time out per DHCP protocol and CHPSLT-1 will be repeated.
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.

[a] If the client is aware of its previous IP address (e.g., following reboot), it may omit the DHCP DISCOVER and proceed with step 3.

[b] If the client is located on a non-broadcast network it is expected to unicast the message to the DHCP Server.

[c] If the client is located on a non-broadcast network it is expected that it will unicast the message to the PS.

13.6.1 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 18.

13.7 Provisioning process: DHCP client in the LAN-Pass realm

Some customer 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 (cabhCapPassthroughTable). Figure 44 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 44 and described in detail in Table 52.

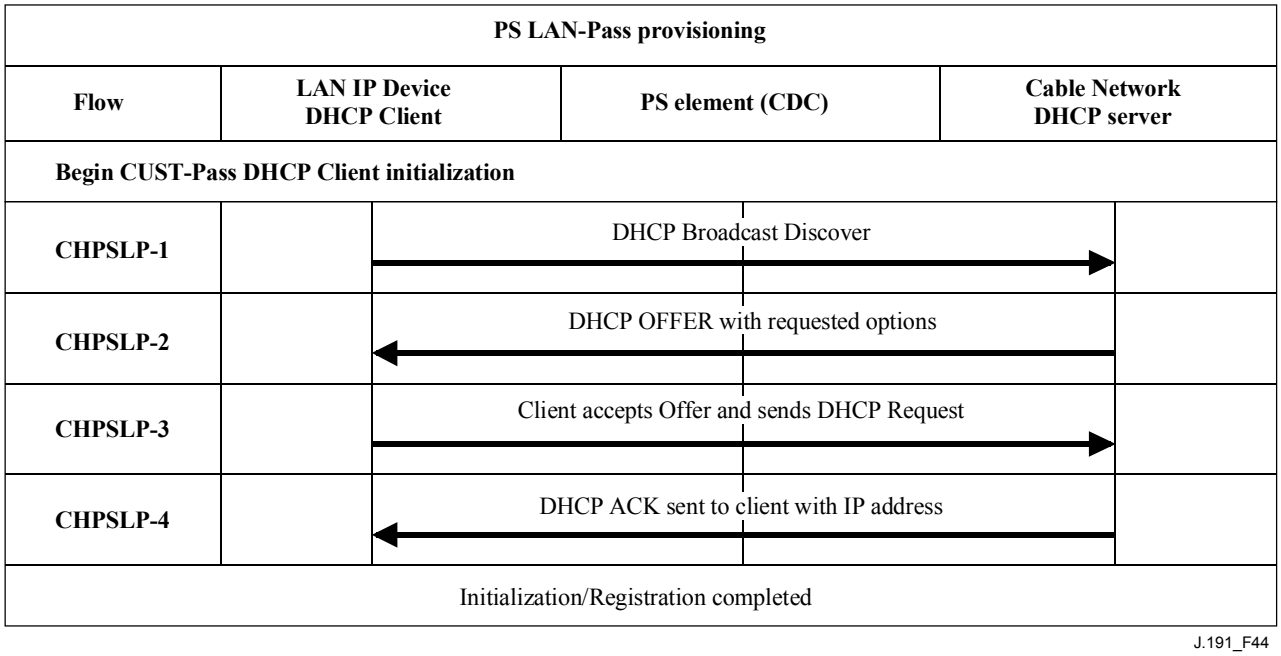


Figure 44/J.191 – Provisioning process for LAN IP device in the LAN-Pass realm

Table 52/J.191 – Flow descriptions for LAN-Pass provisioning process

Flow step	Client Pass-through address provisioning	Normal sequence	Failure sequence
CHPSLP-1	*DHCP Broadcast Discover* The LAN IP Device broadcasts a DHCP DISCOVER message on its local LAN[a)] 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 52/J.191 – Flow descriptions for LAN-Pass provisioning process

Flow step	Client Pass-through 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.
[a] 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 MIB objects required, as indicated in 6.3.7.

MIB NAME/Parameter	Max-Access	Persistent
mib-2		
system		
sysDescr	read-only	Yes
sysObjectID	read-only	Yes
sysUpTime	read-only	No
sysContact	read-only	Yes
sysName	read-only	Yes
sysLocation	read-only	Yes
sysServices	read-only	Yes
interfaces [RFC 2863]		
ifNumber	read-only	No
ifTable/ifEntry		
ifIndex	read-only	No
ifDescr	read-only	No
ifType	read-only	No
ifMtu	read-only	No
ifSpeed	read-only	No

ifPhysAddress	read-only	No
ifAdminStatus	read-write	No
ifOperStatus	read-only	No
ifLastChange	read-only	No
ifInOctets	read-only	No
ifInUcastPkts	read-only	No
ifInNUcastPkts	read-only	No
ifInDiscards	read-only	No
ifInErrors	read-only	No
ifInUnknownProtos	read-only	No
ifOutOctets	read-only	No
ifOutUcastPkts	read-only	No
ifOutNUcastPkts	read-only	No
ifOutDiscards	read-only	No
ifOutErrors	read-only	No
ifOutQLen	read-only	No
ifSpecifc	read-only	No
ip [RFC 2011]		
ipForwarding	read-write	No
ipDefaultTTL	read-write	No
ipInReceives	read-only	No
ipInHdrErrors	read-only	No
ipInAddrErrors	read-only	No
ipForwDatagrams	read-only	No
ipInUnknownProtos	read-only	No
ipInDiscards	read-only	No
ipInDelivers	read-only	No
ipOutRequests	read-only	No
ipOutDiscards	read-only	No
ipOutNoRoutes	read-only	No
ipReasmTimeout	read-only	No
ipReasmReqds	read-only	No
ipReasmOKs	read-only	No
ipReasmFails	read-only	No
ipFragOKs	read-only	No
ipFragFails	read-only	No
ipFragCreates	read-only	No
ipNetToMediaTable/ipNetToMediaEntry		
ipNetToMediaIfIndex	read-create	No
ipNetToMediaPhyAddress	read-create	No
ipNetToMediaNetAddress	read-create	No
ipNetToMediaType	read-create	No
icmp		

icmpInMsgs	read-only	No
icmpInErrors	read-only	No
icmpInDestUnreachs	read-only	No
icmpInTimeExcds	read-only	No
icmpInParmProbs	read-only	No
icmpInSrcQuenchs	read-only	No
icmpInRedirects	read-only	No
icmpInEchos	read-only	No
icmpInEchosReps	read-only	No
icmpInTimestamps	read-only	No
icmpInTimestampsReps	read-only	No
icmpInAddrMasks	read-only	No
icmpInAddrMaskReps	read-only	No
icmpOutMsgs	read-only	No
icmpOutErrors	read-only	No
icmpOutDestUnreachs	read-only	No
icmpOutTimeExcds	read-only	No
icmpOutParmProbs	read-only	No
icmpOutSrcQuenchs	read-only	No
icmpOutRedirects	read-only	No
icmpOutEchos	read-only	No
icmpOutEchosReps	read-only	No
icmpOutTimestamps	read-only	No
icmpOutTimestampReps	read-only	No
icmpOutAddrMasks	read-only	No
icmpOutAddrMaskReps	read-only	No
udp [RFC 2013]		
udpInDatagrams	read-only	No
udpNoPorts	read-only	No
udpInErrors	read-only	No
udpOutDatagrams	read-only	No
udpTable/udpEntry		
udpLocalAddress	read-only	No
udpLocalPort	read-only	No
transmission [RFC draft-ietf-ipcdn-bpiplus-mib-06.txt]		
docsIfMib		
docsBpi2MIB		
docsBpi2MIBObjects		
docsBpi2CmObjects		
docsBpi2CmCertObjects		
docsBpi2CmDeviceCertTable/docsBpi2CmDeviceCertEntry		
docsBpi2CmDeviceCmCert	read-write	Yes
docsBpi2CmDeviceManufCert	read-only	Yes

docsBpi2CodeDownloadControl		
docsBpi2CodeDownloadStatusCode	read-only	Yes
docsBpi2CodeDownloadStatusString	read-only	Yes
docsBpi2CodeMfgOrgName	read-only	Yes
docsBpi2CodeMfgCodeAccessStart	read-only	Yes
docsBpi2CodeMfgCvcAccessStart	read-only	Yes
docsBpi2CodeCoSignerOrgName	read-only	Yes
docsBpi2CodeCoSignerCodeAccessStart	read-only	Yes
docsBpi2CodeCoSignerCvcAccessStart	read-only	Yes
docsBpi2CodeCvcUpdate	read-write	Yes
snmp [RFC 1907]		
snmpInPkts	read-only	No
snmpOutPkts	read-only	No
snmpInBadVersions	read-only	No
snmpInBadCommunityNames	read-only	No
snmpInBadCommunityUses	read-only	No
snmpInASNParseErrs	read-only	No
snmpInTooBigs	read-only	No
snmpInNoSuchNames	read-only	No
snmpInBadValues	read-only	No
snmpInReadOnlys	read-only	No
snmpInGenErrs	read-only	No
snmpInTotalReqVars	read-only	No
snmpInTotalSetVars	read-only	No
snmpInGetRequests	read-only	No
snmpInGetNexts	read-only	No
snmpInSetRequests	read-only	No
snmpInGetResponses	read-only	No
snmpInTraps	read-only	No
snmpOutTooBigs	read-only	No
snmpOutNoSuchNames	read-only	No
snmpOutBadValues	read-only	No
snmpOutGenErrs	read-only	No
snmpOutGetRequests	read-only	No
snmpOutGetNexts	read-only	No
snmpOutSetRequests	read-only	No
snmpOutGetResponses	read-only	No
snmpOutTraps	read-only	No
snmpEnableAuthenTraps	read-write	No
snmpSilentDrops	read-only	No
snmpProxyDrops	read-only	No
ifMIB [RFC 2863]		
ifMIBOjects		

ifXTable/ifXEntry		
ifName	read-only	No
ifInMulticastPkts	read-only	No
ifInBroadcastPkts	read-only	No
ifOutMulticastPkts	read-only	No
ifOutBroadcastPkts	read-only	No
ifHCInOctets	read-only	No
ifHCInUcastPkts	read-only	No
ifHCInMulticastPkts	read-only	No
ifHCInBroadcastPkts	read-only	No
ifHCOutOctets	read-only	No
ifHCOutUcastPkts	read-only	No
ifHCOutMulticastPkts	read-only	No
ifHCOutBroadcastPkts	read-only	No
ifLinkUpDownTrapEnable	read-write	No
ifHighSpeed	read-only	No
ifPromiscuousMode	read-write	No
ifConnectorPresent	read-only	No
ifAlias	read-write	No
ifCounterDiscontinuitityTime	read-only	No
docsDev [RFC 2669]		
docsDevMIBObjects		
docsDevNmAccessTable/docsDevNmAccessEntry		
docsDevNmAccessIndex	not-accessible	No
docsDevNmAccessIp	read-create	No
docsDevNmAccessIpMask	read-create	No
docsDevNmAccessCommunity	read-create	No
docsDevNmAccessControl	read-create	No
docsDevNmAccessInterfaces	read-create	No
docsDevNmAccessStatus	read-create	No
docsDevSoftware		
docsDevSwServer	read-write	Yes
docsDevSwFilename	read-write	Yes
docsDevSwAdminStatus	read-write	Yes
docsDevSwOperStatus	read-only	Yes
docsDevSwCurrentVers	read-only	Yes
docsDevEvent		
docsDevEvControl	read-write	No
docsDevEvSyslog	read-write	No
docsDevEvThrottleAdminStatus	read-write	No
docsDevEvThrottleInhibited	read-only	No
docsDevEvThrottleThreshold	read-write	No
docsDevEvThrottleInterval	read-write	No

docsDevEvControlTable/docsDevEvControlEntry		
docsDevEvPriority	not-accessible	No
docsDevEvReporting	read-write	No
docsDevEventTable/docsDevEventEntry		
docsDevEvIndex	not-accessible	Yes
docsDevEvFirstTime	read-only	Yes
docsDevEvLastTime	read-only	Yes
docsDevEvCounts	read-only	Yes
docsDevEvLevel	read-only	Yes
docsDevEvId	read-only	Yes
docsDevEvText	read-only	Yes
private		
enterprises		
cableLabs		
clabProject		
clabProjCableHome		
cabhPsDevMib		
cabhPsDevBase		
cabhPsDevDateTime	read-write	No
cabhPsDevResetNow	read-write	No
cabhPsDevSerialNumber	read-only	Yes
cabhPsDevHardwareVersion	read-only	Yes
cabhPsdevMacAddress	read-only	Yes
cabhPsDevTypeIdentifier	read-only	Yes
cabhPsDevResetDefaults	read-write	No
cabhPsDevWanManClientId	read-write	Yes
cabhPsDevTodSyncStatus	read-only	No
cabhPsDevProvMode	read-only	No
cabhPsDevDwnldMode	read-only	No
cabhPsDevProv		
cabhPsDevProvisioningTimer	read-write	Yes
cabhPsDevProvConfigFile	read-write	No
cabhPsDevProvConfigHash	read-write	No
cabhPsDevProvConfigFileSize	read-only	No
cabhPsDevProvConfigTLVProcessed	read-only	No
cabhPsDevProvConfigTLVRejected	read-only	No
cabhPsDevProvSolicitedKeyTimeout	read-write	Yes
cabhPsDevProvState	read-only	No
cabhPsDevProvAuthState	read-only	No
cabhPsDevProvCorrelationId	read-only	No
cabhPsDevServerType	read-only	No
cabhPsDevServerTime	read-only	No
cabhSecMib		

cabhSecFwObjects		
cabhSecFwBase		
cabhSecFwPolicyFileEnable	read-write	Yes
cabhSecFwPolicyFileURL	read-write	No
cabhSecFwPolicyFileHash	read-write	No
cabhSecFwPolicyFileOperStatus	read-only	No
cabhSecFwPolicyFileCurrentVersion	read-write	Yes
cabhSecFwLogCtl		
cabhSecFwEventType1Enable	read-write	Yes
cabhSecFwEventType2Enable	read-write	Yes
cabhSecFwEventType3Enable	read-write	Yes
cabhSecFwEventAttachAlertThreshold	read-write	Yes
cabhSecFwEventAttackAlertPeriod	read-write	Yes
cabhCapMib		
cabhCapObjects		
cabhCapBase		
cabhCapTcpTimeWait	read-write	Yes
cabhCapUdpTimeWait	read-write	Yes
cabhCapIcmpTimeWait	read-write	Yes
cabhCapPrimaryMode	read-write	Yes
cabhCapSetToFactory	read-write	No
cabhCapMap		
cabhCapMappingTable/cabhCapMappingEntry		
cabhCapMappingWanAddrType	not-accessible	Yes[1]
cabhCapMappingWanAddrType	not-accessible	Yes[1]
cabhCapMappingWanPort	not-accessible	Yes[1]
cabhCapMappingLanAddrType	not-accessible	Yes[1]
cabhCapMappingLanAddrType	not-accessible	Yes[1]
cabhCapMappingLanPort	not-accessible	Yes[1]
cabhCapMappingMode	read-only	Yes[1]
cabhCapMappingMethod	read-only	Yes[1]
cabhCapMappingProtocol	read-only	Yes[1]
cabhCapPassthroughTable/cabhCapPassthroughEntry		
cabhCapPassthroughMACAddr	not-accessible	Yes
cabhCapPassthroughRowStatus	read-create	No
cabhCdpMib		
cabhCdpObjects		
cabhCdpBase		
cabhCdpSetToFactory	read-write	No

---

[1] 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.

cabhCdpLanTransCurCount	read-only	No
cabhCdpLanTransThreshold	read-write	Yes
cabhCdpLanTransAction	read-write	Yes
cabhCdpAddr		
cabhCdpLanAddrTable/cabhCdpLanAddrEntry		
cabhCdpLanAddrIpType	not-accessible	Yes
cabhCdpLanAddrIp	not-accessible	Yes
cabhCdpLanAddrClientID	read-only	Yes
cabhCdpLanAddrCreateTime	read-only	Yes
cabhCdpLanAddrExpireTime	read-only	Yes
cabhCdpLanAddrMethod	read-only	Yes
cabhCdpLanAddrHostName	read-only	Yes
cabhCdpLanAddrRowStatus	read-create	No
cabhCdpWanDataAddrTable/cabhCdpWanDataAddrEntry		
cabhCdpWanDataAddrIndex	not-accessible	Yes
cabhCdpWanDataAddrClientId	read-create	Yes
cabhCdpWanDataAddrIpType	read-create	No
cabhCdpWanDataAddrIp	read-create	No
cabhCdpWanDataAddrAddrRenewalTime	read-create	No
cabhCdpWanDataAddrRowStatus	read-create	No
cabhCdpWanDataAddrServerTable/cabhCdpWanDataAddrServerEntry		
cabhCdpWanDataAddrDnsIpType	not-accessible	No
cabhCdpWanDataAddrDnsIp	not-accessible	No
cabhCdpWanDataAddrDnsRowStatus	read-create	No
cabhCdpServer		
cabhCdpLanPoolStartType	read-write	Yes
cabhCdpLanPoolStart	read-write	Yes
cabhCdpLanPoolEndType	read-write	Yes
cabhCdpLanPoolEnd	read-write	Yes
cabhCdpServerSubnetMaskType	read-write	Yes
cabhCdpServerSubnetMask	read-write	Yes
cabhCdpServerTimeOffset	read-write	Yes
cabhCdpServerRouterType	read-write	Yes
cabhCdpServerRouter	read-write	Yes
cabhCdpServerDnsAddressType	read-write	Yes
cabhCdpServerDnsAddress	read-write	Yes
cabhCdpServerSyslogAddressType	read-write	Yes
cabhCdpServerSyslogAddress	read-write	Yes
cabhCdpServerDomainName	read-write	Yes
cabhCdpServerTTL	read-write	Yes
cabhCdpServerInterfaceMTU	read-write	Yes
cabhCdpServerVendorSpecific	read-write	Yes
cabhCdpServerLeaseTime	read-write	Yes

cabhCdpServerDhcpAddressType	read-write	Yes
cabhCdpServerDhcpAddress	read-write	Yes
cabhCtpMib		
cabhCtpObjects		
cabhCtpBase		
cabhCtpReset	read-write	No
cabpCtpConnSpeed		
cabhCtpConnSrcIpType	read-write	No
cabhCtpConnSrcIp	read-write	No
cabhCtpConnDestIpType	read-write	No
cabhCtpConnDestIp	read-write	No
cabhCtpConnProto	read-write	No
cabhCtpConnPort	read-write	No
cabhCtpConnNumPkts	read-write	No
cabhCtpConnPktSize	read-write	No
cabhCtpConnTimeOut	read-write	No
cabhCtpConnControl	read-write	No
cabhCtpConnStatus	read-only	No
cabhCtpConnPktsSent	read-only	No
cabhCtpConnPktsRecv	read-only	No
cabhCtpConnAvgRTT	read-only	No
cabhCtpConnMaxRTT	read-only	No
cabhCtpConnMinRTT	read-only	No
cabhCtpConnNumIcmpError	read-only	No
cabhCtpConnIcmpError	read-only	No
cabhCtpPing		
cabhCtpPingSrcIpType	read-write	No
cabhCtpPingSrcIp	read-write	No
cabhCtpPingDestIpType	read-write	No
cabhCtpPingDestIp	read-write	No
cabhCtpPingProto	read-write	No
cabhCtpPingNumPkts	read-write	No
cabhCtpPingPktSize	read-write	No
cabhCtpPingTimeBetween	read-write	No
cabhCtpPingTimeOut	read-write	No
cabhCtpPingControl	read-write	No
cabhCtpPingStatus	read-only	No
cabhCtpPingNumSent	read-only	No
cabhCtpPingNumRecv	read-only	No
experimental		
snmpUSMDHObjectsMIB [RFC 2786]		
usmDHKeyObjects		
usmDHPublicObjects		

usmDHPParameters	read-write	No
usmDHUserKeyTable/usmDHUserKeyEntry		
usmDHUserAuthKeyChange	read-create	No
usmDHUserOwnAuthKeyChange	read-create	No
usmDHUserPrivKeyChange	read-create	No
usmDHUserOwnPrivKeyChange	read-create	No
usmDHKickstartGroup		
usmDHKickstartTable/usmDHKickstartEntry		
usmDHKickstartIndex	not-accessible	No
usmDHKickstartMyPublic	read-only	No
usmDHKickstartMgrPublic	read-only	No
usmDHKickstartSecurityName	read-only	No
snmpV2		
snmpModules		
snmpMIB		
snmpMIBObjects		
snmpSet		
snmpSetSerialNo	read-write	No
snmpFrameworkMIB [RFC 2571]		
snmpEngine		
snmpEngineID	read-only	Yes
snmpEngineBoots	read-only	Yes
snmpEngineTime	read-only	No
snmpEngineMaxMessageSize	read-only	Yes
snmpMPDMIB [RFC 2572]		
snmpMPDObjects		
snmpMPDStats		
snmpUnknownSecurityModels	read-only	No
snmpInvalidMsgs	read-only	No
snmpUnknownPDUHandlers	read-only	No
snmpTargetMIB [RFC 2573]		
snmpTargetObjects		
snmpTargetSpinLock	read-write	No
snmpTargetAddrTable/snmpTargetAddrEntry		
snmpTargetAddrName	not-accessible	No
snmpTargetAddrTDomain	read-create	No
snmpTargetAddrTAddress	read-create	No
snmpTargetAddrTimeout	read-create	No
snmpTargetAddrRetryCount	read-create	No
snmpTargetAddrTagList	read-create	No
snmpTargetAddrParams	read-create	No
snmpTargetAddrStorageType	read-create	No
snmpTargetAddrRowStatus	read-create	No

snmpTargetParamsTable/snmpTargetParamsEntry		
snmpTargetParamsName	not-accessible	No
snmpTargetParamsMPModel	read-create	No
snmpTargetParamsSecurityModel	read-create	No
snmpTargetParamsSecurityName	read-create	No
snmpTargetParamsSecurityLevel	read-create	No
snmpTargetParamsStorageType	read-create	No
snmpTargetParamsRowStatus	read-create	No
snmpUnavailableContexts	read-only	No
snmpUnknownContexts	read-only	No
snmpNotificationMIB [RFC 2573]		
snmpNotifyObjects		
snmpNotifyTable/snmpNotifyEntry		
snmpNotifyName	not-accessible	No
snmpNotifyTag	read-create	No
snmpNotifyType	read-create	No
snmpNotifyStorageType	read-create	No
snmpNotifyRowStatus	read-create	No
snmpNotifyFilterProfileTable/snmpNotifyFilterProfileEntry		
snmpNotifyFilterProfileName	read-create	No
snmpNotifyFilterProfileStorType	read-create	No
snmpNotifyFilterProfileRowStatus	read-create	No
snmpNotifyFilterTable/snmpNotifyFilterEntry		
snmpNotifyFilterSubtree	not-accessible	No
snmpNotifyFilterMask	read-create	No
snmpNotifyFilterType	read-create	No
snmpNotifyFilterStorageType	read-create	No
snmpNotifyFilterRowStatus	read-create	No
snmpUsmMIB [RFC 2574]		
usmStats		
usmStatsUnsupportedSecLevels	read-only	No
usmStatsNotInTimeWindows	read-only	No
usmStatsUnknownUserNames	read-only	No
usmStatsUnknownEngineIDs	read-only	No
usmStatsWrongDigests	read-only	No
usmStatsDecryptionErrors	read-only	No
usmUser		
usmUserSpinLock	read-write	No
usmUserTable/usmUserEntry		
usmUserEngineID	not-accessible	No
usmUserName	not-accessible	No
usmUserSecurityName	read-only	No
usmUserCloneFrom	read-create	No

usmUserAuthProtocol	read-create	No
usmUserAuthKeyChange	read-create	No
usmUserOwnAuthKeyChange	read-create	No
usmUserPrivProtocol	read-create	No
usmUserPrivKeyChange	read-create	No
usmUserOwnPrivKeyChange	read-create	No
usmUserPublic	read-create	No
usmUserStorageType	read-create	No
usmUserStatus	read-create	No
SNMP-VIEW-BASED-ACM-MIB [RFC 2575]		
snmpVacmMIB		
vacmMIBObjects		
vacmContextTable/vacmContextEntry		
vacmContextName	read-only	No
vacmSecurityToGroupTable/vacmSecurityToGroupEntry		
vacmSecurityModel	not-accessible	No
vacmSecurityName	not-accessible	No
vacmGroupName	read-create	No
vacmSecurityToGroupStorageType	read-create	No
vacmSecurityToGroupStatus	read-create	No
vacmAccessTable/vacmAccessEntry		
vacmAccessContextPrefix	not-accessible	No
vacmAccessSecurityModel	not-accessible	No
vacmAccessSecurityLevel	not-accessible	No
vacmAccessContextMatch	read-create	No
vacmAccessReadViewName	read-create	No
vacmAccessWriteViewName	read-create	No
vacmAccessNotifyViewName	read-create	No
vacmAccessStorageType	read-create	No
vacmAccessStatus	read-create	No
vacmMIBViews		
vacmViewSpinLock	read-write	No
vacmViewTreeFamilyTable/vacmViewTreeFamilyEntry		
vacmViewTreeFamilyViewName	not-accessible	No
vacmViewTreeFamilySubtree	not-accessible	No
vacmViewTreeFamilyMask	read-create	No
vacmViewTreeFamilyType	read-create	No
vacmViewTreeFamilyStorageType	read-create	No
vacmViewTreeFamilyStatus	read-create	No
snmpCommunityMIB [RFC 2576]		
snmpCommunityMIBObjects		
snmpCommunityTable/snmpCommunityEntry		
snmpCommunityIndex	not-accessible	No

snmpCommunityName	read-create	No
snmpCommunitySecurityName	read-create	No
snmpCommunityContextEngineID	read-create	No
snmpCommunityContextName	read-create	No
snmpCommunityTransportTag	read-create	No
snmpCommunityStorageType	read-create	No
snmpCommunityStatus	read-create	No
snmpTargetAddrExtTable/snmpTargetAddrExtEntry		
snmpTargetAddrTMask	read-create	No
snmpTargetAddrMMS	read-create	No
snmpTrapAddress	accessible-for-notify	No
snmpTrapCommunity	accessible-for-notify	No

Annex B

Format and content for event, SYSLOG and SNMP traps

Table B.1 summarizes the format and content for local log event entries, syslog messages, and SNMP traps.

Each row in Table B.1 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 the SNMP trap format is defined in this annex, following Table B.1.

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 Table B.1 are in decimal format.

To better illustrate Table B.1, 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

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	Event ID	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	
Init	DHCP	Critical	DHCP failed – Response doesn't contain ALL the valid fields as described in the Recommendation		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 Message Integrity check MIC	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	
TFTP Success							
Init	TFTP	Notice	TFTP success		D10.0	68001000	

Table B.1/J.191 – Defined events

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	Event ID	Trap name
TLV parsing							
Init	TLV parsing	Notice	TLV-28 – Unrecognized OID		I401.0	73040100	cabhPsDev InitTLVUnknownTrap
Init	TLV parsing	Notice	Unknown TLV <P1>	For SYSLOG only: <P1> = the complete TLV in hexadecimal	I401.1	73040101	cabhPsDev InitTLVUnknownTrap
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	cabhPsDev InitTrap
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	cabhPsDev InitRetryTrap
SW upgrade init							
SW upgrade	SW upgrade init	Notice	SW Download INIT – Via NMS	For SYSLOG only, append: SW file: <P1> – SW server: <P2>. P1 = SW file name and P2 = TFTP server IP address	E101.0	69010100	cabhPsDev SwUpgrade InitTrap
SW upgrade	SW upgrade init	Notice	SW Download INIT – Via Config file <P1>	P1 = CM config file name for SYSLOG only, append: SW file: <P2> – SW server: < P3>. P2 = SW file name and P3 = TFTP server IP address	E102.0	69010200	cabhPsDev SwUpgrade InitTrap
SW upgrade general failure							
SW upgrade	SW upgrade general failure	Error	SW Upgrade Failed during download – Max retry exceed (3)	For SYSLOG only, append: SW file: <P1> – SW server: <P2>. P1 = SW file name and P2 = TFTP server IP address	E103.0	69010300	cabhPsDev SwUpgrade FailTrap

Table B.1/J.191 – Defined events

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	Event ID	Trap name
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
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

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	Event ID	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							
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					D100.0	68010000	
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
TOD failure after provisioning complete							
TOD	TOD	Warning	TOD Request sent – No response received		D04.3	68000403	cabhPsDev TODFailTrap
TOD	TOD	Warning	TOD Response received – Invalid data format		D04.4	68000404	cabhPsDev TODFailTrap
Verification of code file					E200		
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

Table B.1/J.191 – Defined events

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	Event ID	Trap name
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
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 CVCFailTrap
SW upgrade	Verification of CVC	Error	Configuration File CVC Validation Failure – TFTP Server: <P1> – Config File: <P2>	P1 = TFTP server IP Address P2 = Config File Name	E207.0	69020700	cabhPsDev SwUpgrade CVCFailTrap
SW upgrade	Verification of CVC	Error	Improper SNMP CVC Format – SNMP manager: <P1>	P1 = IP Address of SNMP manager	E208.0	69020800	cabhPsDev SwUpgrade CVCFailTrap
SW upgrade	Verification of CVC	Error	SNMP CVC Validation Failure – SNMP manager: <P1>	P1 = IP Address of SNMP manager	E209.0	69020900	cabhPsDev SwUpgrade CVCFailTrap

Table B.1/J.191 – Defined events

Process	Sub-process	PS priority	Event text	Message notes and details	Error code set	Event ID	Trap name
CDP Events					P		
CDP	CDS	Notice	Attempt to allocate more LAN TRANS IP addresses than allowed		P01.0	80000100	cabhPsDev CDPTrap
CSP Events							
CSP	Firewall	Notice	Firewall Type 1 and Type 2 Hacker Threshold Exceed		P101.0	80010100	cabhPsDev CSPTrap
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
CSP	Firewall	Notice	Firewall configuration has changed	P1 = description of change in firewall configuration parameters	P120.0	80012000	cabhPsDev CSPTrap
CAP Events							
CAP	C-NAT	Notice	CAP unable to make C-NAT mapping. No WAN-data IP address available		P201.0	800201.00	cabhPsDev CAPTrap
CAP	C-NAPT	Notice	CAP unable to make C-NAPT mapping. No WAN IP address available		P250.0	800250.00	cabhPsDev CAPTrap

B.1 Trap descriptions

```
cabhPsDevInitTLVUnknownTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress }
      STATUS current
      DESCRIPTION
            "Event due to detection of an unknown TLV during the TLV parsing
            process. The values of docsDevEvLevel, docsDevId, and DocsDevEvText
            are from the entry which logs this event in the docsDevEventTable.
            The ifPhysAddress value is the MAC address of the PS. This part of
            information is uniformed across all PS traps."
      := { cabhPsDevTraps 1 }

cabhPsDevInitTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress,
                  docsDevServerConfigFile,
                  number of TLVs,
                  number of skipped TLVs }
      STATUS current
      DESCRIPTION
            "An event to report the initialization process is complete as
            detected in the PS. The values of docsDevEvLevel, docsDevId, and
            docsDevEvText are from the entry which logs this event in the
            docsDevEventTable. The value of ifPhysAddress indicates the MAC
            address of the PS.
            DocsDevServerConfigFile is the name of the configuration file used.
            As well as the number of TLVs in the config file and the number of
            skipped TLVs. If no configuration file was used, set all three to
            'none'.
            This part of information is uniformed across all PS traps."
      ::= { cabhPsDevTraps 2 }

cabhPsDevInitRetryTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress }
      STATUS current
      DESCRIPTION
            "An event to report the failure happened during the initialization
            process and detected in the PS. The values of docsDevEvLevel,
            docsDevId, and docsDevEvText are from the entry which logs this
            event in the docsDevEventTable. The value of ifPhysAddress indicates
            the MAC address of the PS.
            This part of information is uniformed across all PS traps."
      ::= { cabhPsDevTraps 3 }
cabhPsDevDHCPFailTrap NOTIFICATION-TYPE

      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress,
                  docsDevServerDhcp }
      STATUS current
      DESCRIPTION
            "An event to report the failure of a DHCP server. The value of
            docsDevServerDhcp is the IP address of the DHCP server."
      ::= { cabhPsDevTraps 4 }
```

```
cabhPsDevSwUpgradeInitTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress,
                  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."
      ::= { cabhPsDevTraps 5 }

cabhPsDevSwUpgradeFailTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress,
                  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."
      ::= { cabhPsDevTraps 6 }

cabhPsDevSwUpgradeSuccessTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress,
                  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."
      ::= { cabhPsDevTraps 7 }

cabhPsDevSwUpgradeCVCFailTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress }
      STATUS current
      DESCRIPTION
            "An event to report the failure of the verification of code file
            happened during a secure software upgrade attempt."
      ::= { cabhPsDevTraps 8 }

cabhPsDevTODFailTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress,
                  docsDevServerTime }
      STATUS current
```

```
      DESCRIPTION
            "An event to report the failure of a Time of Day server. The value
            of docsDevServerTime indicates the server IP address."
      ::= { cabhPsDevTraps 9 }

cabhPsDevCDPTrap NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress,
                  addressThreshold }
      STATUS current
      DESCRIPTION
            "To report an event with the DHCP Portal."
      ::= { cabhPsDevTraps 10 }

cabhPsDevCSPTrap   NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress }
      STATUS current
      DESCRIPTION
            "To report an event with the Security Portal."
      ::= { cabhPsDevTraps 11 }

cabhPsDevCAPTrap   NOTIFICATION-TYPE
      OBJECTS { docsDevEvLevel,
                  docsDevEvId,
                  docsDevEvText,
                  ifPhysAddress }
      STATUS current
      DESCRIPTION
            "To report an event with the Security Portal."
      ::= { cabhPsDevTraps 12 }
```

Annex C

Security threats and preventative measures

When developing 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 security threats to subscribers and operators have been identified:

– **Theft of Service**: Theft of service comes in two forms: unauthorized access to cable services, and unauthorized duplication of service content.

  Unauthorized access involves a subscriber or 3rd party (such as a neighbor) having access to cable services for which they have not paid. Devices could be "cloned" or modified to appear as a qualified device at the subscriber's home. This could also degrade service delivery performance as these devices consume additional transport resources on the HFC and home links.

  Unauthorized duplication usually involves a subscriber or 3rd party (such as a neighbor) making illegal copies of service content. In some cases these copies are distributed to other consumers without the approval of the operator or content provider.

– **Denial of Service (DOS) Attacks**: Denial of service attacks can occur when a 3rd party entity (attacker, disgruntled customer, etc.) disrupts the normal communication and delivery of services between operators and their subscribers. Offending data transmissions coming from what appears to be a valid device/source, could be injected into the home link and severely degrade its normal functions. These offending data transmissions could also extend to the operator's HFC network, causing performance problems there.

– **Service Confidentiality**: The service confidentiality threat involves a 3rd party (neighbors, attacker, etc.) monitoring/receiving information about a subscriber and the services they use. This could result in passwords or device configuration information being stolen, allowing attackers to gain further access to a subscriber's network resources and confidential files/data.

There are a number of different methods that can be used to prevent the security threats mentioned above. Unfortunately, one method cannot prevent them all, but a combination may be the best line of defense. The following preventative measures can be used:

– **Authentication**: Authentication involves the verification that the sending and receiving entities are as claimed. This includes the service source, the receiving device, and the subscriber.

  Authentication helps prevent theft of service by validating end devices and users, but it does not prevent content from being illegally copied, or prevent unauthorized access by 3rd parties who are monitoring the link. It does do a good job at preventing DOS attacks because traffic can be rejected if it does not come from a valid source. By itself authentication does not provide any service confidentiality support, encryption must be used.

– **Copy Protection**: Copy protection methods limit the ability of a receiving device to make unauthorized copies of service content.

  Copy protection helps prevent theft of service by limiting how many copies can be made, but it does not prevent unauthorized access to services. It also does not prevent DOS or service confidentiality protection. In general, this preventive measure is implemented at higher application layers.

– **Data Encryption**: Data encryption prevents the unauthorized disclosure/access of data.

  Data encryption does an excellent job at providing data confidentiality and protection against theft of service. Encryption prevents making data unable to read without the correct decrypting key; however, it does not validate the source/receiving entities and it does not provide copy protection after the data has been decrypted. It also does not prevent DOS attacks.

– **Firewall**: Firewall applications prevent network traffic from passing from one domain to another unless it meets certain criteria set by the subscriber or operator. In home applications, firewalls are typically located on residential gateway devices that connect the HFC network to the home.

  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 side of the firewall. It also does not provide theft of service protection.

– **Management Message Security**: This method of prevention involves authentication and encryption of network management messages only. Network management messages are used for device configuration, network monitoring/control, service provisioning, and Quality of Service (QoS) reservations.

Management message security provides a good mechanism to prevent DOS attacks by authenticating and encrypting management messages. The 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 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.

Annex E

MIBs

E.1 Portal Service (PS) MIB

The PS 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,
     DisplayString,
     PhysAddress,
     DateAndTime,
     TEXTUAL-CONVENTION
                                        FROM SNMPv2-TC
     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
                         FROM CABH-CDP-MIB

     clabProjCableHome
                                        FROM CLAB-DEF-MIB;

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

cabhPsDevMib MODULE-IDENTITY
     LAST-UPDATED  "0112190000Z" -- December 19, 2001
     ORGANIZATION  "Cable    NMP Group"
     CONTACT-INFO
     "Kevin Luehrs
     Postal: Cable Television Laboratories, Inc.
                 400 Centennial Parkway
                 Louisville, Colorado 80027-1266
       U.S.A.
       Phone: +1 303-661-9100
       Fax: +1 303-661-9199
       E-mail: k.luehrs@cablelabs.com"
     DESCRIPTION
       "This MIB module supplies the basic management objects for the PS Device.
       The PS device parameter describe general PS Device attributes and
       behavior characteristics. Most the PS Device MIB is needed for
       configuration download.

-- Textual conventions
     X509Certificate ::= TEXTUAL-CONVENTION
       STATUS current
       DESCRIPTION
          "An X.509 digital certificate encoded as an ASN.1 DER object."
       SYNTAX OCTET STRING (SIZE (0..4096))

--
-- assumes SNMPv3
-- load management is per DOCSIS 1.1 only
--

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 time zone information."
     ::= { cabhPsDevBase 1 }

cabhPsDevResetNow OBJECT-TYPE
     SYNTAX        TruthValue
     MAX-ACCESS    read-write
     STATUS        current
     DESCRIPTION
       "Setting this object to true(1) causes the device to reset.
       Reading this object always returns false(2). When cabhPsDevResetNow is
       set to true, the following actions occur:
       1) Clear all statistics in PS.
       2) Clear trace logs.
       3) Clear all security associations.
       4) Initialize all configuration parameters
       5) Delete all address translations
       6) Delete all FQDN to IP mappings
       7) Delete all stored ARP translations
       8) The provisioning flow is started at step PS - 1."
     ::= { cabhPsDevBase 2 }

cabhPsDevSerialNumber OBJECT-TYPE
     SYNTAX        DisplayString (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        DisplayString (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 }

cabhPsDevMacAddress OBJECT-TYPE
     SYNTAX        PhysAddress
     MAX-ACCESS    read-only
     STATUS        current
     DESCRIPTION
       "The PS WAN-MAN MAC address. Typically, the PS WAN-MAN and PS
       WAN-DATA addresses will be identical. The client identifiers
       will not be the same so that each may be assigned a different
       IP address."
     ::= { cabhPsDevBase 5 }

cabhPsDevTypeIdentifier OBJECT-TYPE
     SYNTAX        DisplayString
     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 6 }
```

```
cabhPsDevResetDefaults OBJECT-TYPE
     SYNTAX        TruthValue
     MAX-ACCESS    read-write
     STATUS        current
     DESCRIPTION
       "Setting this object to True sets all PS parameters to the
       factory defaults"
     ::= { cabhPsDevBase 7 }

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 8 }

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."
     REFERENCE
       " "
     DEFVAL { false }
     ::= { cabhPsDevBase 9 }

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 receives PS Configuration File information (server
       address and file name) in the DHCP message issued by the DHCP server in
       the cable network, the PS sets this object to DHCPmode(1). If the PS
       receives DHCP option 177 sub-option 51 AND does not receive PS
       Configuration File information in the DHCP message the PS
       receives from the DHCP server in the cable network, the PS
       sets this object to SNMPmode(2)."
     ::= { cabhPsDevBase 10 }

cabhPsDevDwnldMode OBJECT-TYPE
     SYNTAX   INTEGER
     {
       standard    (1),
       enhanced    (2)
     }
     MAX-ACCESS    read-only
     STATUS        current
     DESCRIPTION
       "This is the download mode that the PS will used."
       ::= { 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
       time-out 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        DisplayString(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 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 }

cabhPsDevProvConfigTLVProcessed OBJECT-TYPE
     SYNTAX        INTEGER (0..16383)
     MAX-ACCESS    read-only
     STATUS        current
     DESCRIPTION
       "Number of TLVs processed in config file."
     ::={ cabhPsDevProv 5 }

cabhPsDevProvConfigTLVRejected OBJECT-TYPE
     SYNTAX        INTEGER (0..16383)
     MAX-ACCESS    read-only
     STATUS        current
     DESCRIPTION
       "Number of TLVs rejected in config file."
     ::={ cabhPsDevProv 6 }

cabhPsDevProvSolicitedKeyTimeout OBJECT-TYPE
     SYNTAX        Integer32 (15..600)
     UNITS         "seconds"
```

```
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
      "This time-out 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 7 }

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 ompletion of the initialization
      flow. InProgress occurs from PS initialization start to PS
      initialization end."
      ::= { cabhPsDevProv 8 }

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 9 }

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 10 }

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 11 }

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 12 }

--
--   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,
      cabhPsDevMacAddress
    }
    STATUS current
    DESCRIPTION
      "Event due to detection of unknown TLV during the TLV parsing process.
      The values of docsDevEvLevel, docsDevId, and docsDevEvText are from the
      entry which logs this event in the docsDevEventTable. The value of
      cabhPsDevMacAddressindicates the 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,
      cabhPsDevMacAddress,
      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,
    cabhPsDevMacAddress
    }
    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,
       cabhPsDevMacAddress,
       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,
       cabhPsDevMacAddress,
       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,
       cabhPsDevMacAddress,
       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,
       cabhPsDevMacAddress,
       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,
       cabhPsDevMacAddress
     }
     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
     }
     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
     }
     STATUS    current
     DESCRIPTION
       "An event to report the failure of PS to obtain all needed WAN-Data
       IP Addresses.
       cabhCdpWanDataAddrClientId indicates the ClientId for which the failure
       occurred."
     ::= { cabhPsNotification 10 }

cabhPsDevCdpThresholdTrap NOTIFICATION-TYPE
     OBJECTS {
       docsDevEvLevel,
       docsDevEvId,
       docsDevEvText,
       cabhPsDevMacAddress,
       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,
       cabhPsDevMacAddress
     }
     STATUS    current
```

```
      DESCRIPTION
        "To report an event with the Cable Security Portal."
      ::= { cabhPsNotification 12 }

cabhPsDevCapTrap NOTIFICATION-TYPE
      OBJECTS {
        docsDevEvLevel,
        docsDevEvId,
        docsDevEvText,
        cabhPsDevMacAddress
      }
      STATUS    current
      DESCRIPTION
        "To report an event with the Cable Address Portal."
      ::= { cabhPsNotification 13 }

cabhPsDevProvEnrollTrap NOTIFICATION-TYPE
      OBJECTS {
        cabhPsDevHardwareVersion,
        docsDevSwCurrentVers,
        cabhPsDevTypeIdentifier,
        cabhPsDevMacAddress,
        cabhPsDevProvCorrelationId
      }
      STATUS    current
      DESCRIPTION
        "This inform is issued to initiate the process provisioning."
      REFERENCE
        "Inform as defined in RFC 1902."
      ::= { cabhPsNotification 14 }

-- 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 3 }

cabhPsGroup OBJECT-GROUP
      OBJECTS {
        cabhPsDevDateTime,
        cabhPsDevResetNow,
        cabhPsDevSerialNumber,
        cabhPsDevHardwareVersion,
        cabhPsDevMacAddress,
        cabhPsDevTypeIdentifier,
           cabhPsDevResetDefaults,
           cabhPsDevWanManClientId,
           cabhPsDevTodSyncStatus,
        cabhPsDevProvMode,
        cabhPsDevDwnldMode,

        cabhPsDevProvisioningTimer,
           cabhPsDevProvConfigFile,
           cabhPsDevProvConfigHash,
```

```
          cabhPsDevProvConfigFileSize,
          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, cabhPsDevProvEnrollTrap }
     STATUS   current
     DESCRIPTION
          "These notifications deal with change in status of PS Device."
     ::= { cabhPsGroups 2 }

END
```

E.2 CableHome Testing 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:
--
--============================================================================

cabhCtpMib MODULE-IDENTITY
     LAST-UPDATED  "0112190000Z" -- December 19, 2001
     ORGANIZATION       "Cable NMP Group"
     CONTACT-INFO
       "Kevin Luehrs
       Postal: Cable Television Laboratories, Inc.
          400 Centennial Parkway
          Louisville, Colorado 80027-1266
          U.S.A.
          Phone: +1 303-661-9100
          Fax: +1 303-661-9199
          E-mail: k.luehrs@cablelabs.com"
     DESCRIPTION
       "This MIB module defines the diagnostic controls
       offered by the CableHome Testing Portal (CTP).
       Acknowledgements:
       "
     ::= { clabProjCableHome 5 }

-- Textual conventions

--
-- assumes SNMPv3
-- SW load management is per DOCSIS 1.1 only
--

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.
--

cabhCtpReset OBJECT-TYPE
       SYNTAX      TruthValue
       MAX-ACCESS  read-write
     STATUS        current
     DESCRIPTION
       "Setting this object to true(1) causes all testing to be
       terminated. Reading this object always returns false(2).
       When cabhCtpReset is set to true, the following actions occur:
       1) Terminate any diagnostic tests in progress.
       2) Clear all diagnostic statistics."
     ::= { 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.
        Typically the address with be the value in cabhCdpServerRouter. The
        default address is 192.168.0.1."
        REFERENCE
           "Specification Section 6.4.3.2"
        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 used as the destination address for the Connection
        Speed Test."
        ::= { 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 }

cabhCtpConnPort OBJECT-TYPE
        SYNTAX      INTEGER (1..65535)
        MAX-ACCESS  read-write
      STATUS        current
      DESCRIPTION
        "The port used for the Connection Speed Test."
      DEFVAL {7}
      ::= { cabhCtpConnSpeed 6 }

cabhCtpConnNumPkts OBJECT-TYPE
      SYNTAX        INTEGER (1..255)
      MAX-ACCESS    read-write
      STATUS        current
```

```
     DESCRIPTION
       "The number of packets to send."
       DEFVAL {1}
       ::= { cabhCtpConnSpeed 7 }

cabhCtpConnPktSize OBJECT-TYPE
       SYNTAX      INTEGER (64..1518)
       MAX-ACCESS  read-write
     STATUS        current
     DESCRIPTION
       "The size of the test frames."
       REFERENCE
       ""
       ::= { cabhCtpConnSpeed 8 }

cabhCtpConnTimeOut OBJECT-TYPE
       SYNTAX      INTEGER (0..600000)          -- Max 10 minutes
       UNITS       "milliseconds"
       MAX-ACCESS  read-write
     STATUS        current
     DESCRIPTION
       "The time-out value for the response. A value of zero indicates
       no time out and can be used for TCP only."
       DEFVAL {600000}
     ::= { cabhCtpConnSpeed 9 }

cabhCtpConnControl OBJECT-TYPE
          SYNTAX   INTEGER {
                        notRun        (1),
                        start         (2),
                        abort         (3)
                        }
       MAX-ACCESS  read-write
     STATUS        current
     DESCRIPTION
       "The control for Connection Speed Test. The value notRun
       is used to indicate never executed. This parameter should
       only be set via SNMP."
       DEFVAL { notRun(1) }
     ::= { cabhCtpConnSpeed 10 }

cabhCtpConnStatus OBJECT-TYPE
       SYNTAX      INTEGER {
                        running       (1),
                        complete      (2),
                        aborted       (3)
                   }
       MAX-ACCESS  read-only
     STATUS        current
     DESCRIPTION
       "The Status of the currently/last executed test."
       DEFVAL { complete(2) }
       ::= { cabhCtpConnSpeed 11 }

cabhCtpConnPktsSent OBJECT-TYPE
       SYNTAX      INTEGER (0..255)
     MAX-ACCESS    read-only
     STATUS        current
     DESCRIPTION
       "The number of packets sent."
       ::= { cabhCtpConnSpeed 12 }
```

```
cabhCtpConnPktsRecv OBJECT-TYPE
       SYNTAX       INTEGER (0..255)
       MAX-ACCESS   read-only
     STATUS         current
     DESCRIPTION
       "The number for packet received."
     ::= { cabhCtpConnSpeed 13 }

cabhCtpConnAvgRTT OBJECT-TYPE
     SYNTAX         INTEGER (0..600000)
     UNITS          "milliseconds"
     MAX-ACCESS     read-only
     STATUS         current
     DESCRIPTION
       "The resulting average of round-trip times for
       acknowledged packets."
       ::= { cabhCtpConnSpeed 14 }

cabhCtpConnMaxRTT OBJECT-TYPE
       SYNTAX       INTEGER (0..600000)
       UNITS        "milliseconds"
       MAX-ACCESS   read-only
     STATUS         current
     DESCRIPTION
       "The resulting maximum of round-trip times for
       acknowledged packets."
       ::= { cabhCtpConnSpeed 15 }

cabhCtpConnMinRTT OBJECT-TYPE
       SYNTAX       INTEGER (0..600000)
       UNITS        "milliseconds"
       MAX-ACCESS   read-only
     STATUS         current
     DESCRIPTION
       "The resulting minimum of round-trip times for
       acknowledged packets."
       ::= { cabhCtpConnSpeed 16 }

cabhCtpConnNumIcmpError OBJECT-TYPE
       SYNTAX       INTEGER (0..255)
       MAX-ACCESS   read-only
     STATUS         current
     DESCRIPTION
       "Number of ICMP errors."
       ::= { cabhCtpConnSpeed 17 }

cabhCtpConnIcmpError OBJECT-TYPE
       SYNTAX       INTEGER (0..255)
       MAX-ACCESS   read-only
     STATUS         current
     DESCRIPTION
       "The last ICMP error."
     ::= { cabhCtpConnSpeed 18 }

--
--   Parameters and Results for Ping Command
--

cabhCtpPingSrcIpType OBJECT-TYPE
       SYNTAX       InetAddressType
       MAX-ACCESS   read-write
     STATUS         current
```

```
      DESCRIPTION
        "The IP Address Type used as the source address for the Ping Test."
        ::= { 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. Typically the address will be the value of
        PS WanMan IP address. The address 192.168.0.x is used."
        ::= { cabhCtpPing 2 }

cabhCtpPingDestIpType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  read-write
      STATUS        current
      DESCRIPTION
        "The Destination IP Address Type used as the destination address for
        the Ping Test."
        ::= { 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 }

cabhCtpPingProto OBJECT-TYPE
        SYNTAX      INTEGER {
                         icmp (1),
                         }
        MAX-ACCESS  read-write
      STATUS        current
      DESCRIPTION
        "The protocol used to gather topology info."
        DEFVAL { icmp }
      ::= { cabhCtpPing 5 }

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 6 }

cabhCtpPingPktSize OBJECT-TYPE
        SYNTAX      INTEGER (64..1518)
        MAX-ACCESS  read-write
      STATUS        current
      DESCRIPTION
        "The size of the test frames."
        DEFVAL {64}
        ::= { cabhCtpPing 7 }
```

```
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 8 }

cabhCtpPingTimeOut OBJECT-TYPE
       SYNTAX       INTEGER (0..600000)
       UNITS        "milliseconds"
       MAX-ACCESS   read-write
       STATUS       current
     DESCRIPTION
       "The time-out for ping response of sending a single ping."
       DEFVAL { 5000 }  -- 5 seconds
       ::= { cabhCtpPing 9 }

cabhCtpPingControl OBJECT-TYPE
       SYNTAX       INTEGER {
                         notRun         (1),
                         start          (2),
                         abort          (3)
                         }
       MAX-ACCESS   read-write
     STATUS         current
     DESCRIPTION
       "The control for Ping Test. The value notRun
       is used to indicate never executed."
       DEFVAL { notRun(1) }
       ::= { cabhCtpPing 10 }

cabhCtpPingStatus OBJECT-TYPE
       SYNTAX       INTEGER {
                         running        (1),
                         complete       (2),
                         aborted        (3)
                         }
       MAX-ACCESS   read-only
     STATUS         current
     DESCRIPTION
       "The Status of the currently/last executed test."
       ::= { cabhCtpPing 11 }

cabhCtpPingNumSent OBJECT-TYPE
       SYNTAX       INTEGER (0..255)
       MAX-ACCESS   read-only
     STATUS         current
     DESCRIPTION
       "The number of pings sent."
       DEFVAL { complete(2) }
       ::= { cabhCtpPing 12 }

cabhCtpPingNumRecv OBJECT-TYPE
       SYNTAX       INTEGER (0..255)
       MAX-ACCESS   read-only
     STATUS         current
     DESCRIPTION
       "The number of pings received."
       ::= { cabhCtpPing 13 }
```

```
--============================================================================

--
-- 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 {
       cabhCtpReset,
          cabhCtpConnSrcIpType,
          cabhCtpConnSrcIp,
          cabhCtpConnDestIpType,
          cabhCtpConnDestIp,
          cabhCtpConnProto,
          cabhCtpConnPort,
          cabhCtpConnNumPkts,
          cabhCtpConnPktSize,
          cabhCtpConnTimeOut,
          cabhCtpConnControl,
          cabhCtpConnStatus,
          cabhCtpConnPktsSent,
          cabhCtpConnPktsRecv,
          cabhCtpConnAvgRTT,
          cabhCtpConnMinRTT,
          cabhCtpConnMaxRTT,
          cabhCtpConnNumIcmpError,
          cabhCtpConnIcmpError,

          cabhCtpPingSrcIpType,
          cabhCtpPingSrcIp,
          cabhCtpPingDestIpType,
          cabhCtpPingDestIp,
          cabhCtpPingProto,
          cabhCtpPingNumPkts,
          cabhCtpPingPktSize,
          cabhCtpPingTimeBetween,
          cabhCtpPingTimeOut,
```

```
          cabhCtpPingControl,
          cabhCtpPingStatus,
          cabhCtpPingNumSent,
          cabhCtpPingNumRecv
        }
    STATUS   current
    DESCRIPTION
      "Group of objects for Cable CTP MIB."
      ::= { cabhCtpGroups 1 }

END
```

E.3 Security MIB

The Security 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-BPI2MIB
      clabProjCableHome     FROM CLAB-DEF-MIB;

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

cabhSecMib MODULE-IDENTITY
    LAST-UPDATED  "0112200000Z" -- December 20, 2001
    ORGANIZATION       "Cable NMP Group"
    CONTACT-INFO
      "Kevin Luehrs
      Postal: Cable Television Laboratories, Inc.
             400 Centennial Parkway
             Louisville, Colorado 80027-1266
      U.S.A.
      Phone: +1 303-661-9100
      Fax: +1 303-661-9199
      E-mail: k.luehrs@cablelabs.com"
    DESCRIPTION
      "This MIB module supplies the basic management objects
      for the Security Portal Services.

      Acknowledgements:
      "
    ::= { clabProjCableHome 2 }
```

```
-- Textual conventions

--
-- assumes SNMPv3
-- SW load management is per DOCSIS 1.1 only
--

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
--   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
       "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 hash length
       160 bits."
     ::= { 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 time-out."

    ::= { cabhSecFwBase 4 }

cabhSecFwPolicyFileCurrentVersion OBJECT-TYPE
    SYNTAX        SnmpAdminString
    -- MAX-ACCESS read-only
    -- Write access added to allow factory configuration
    MAX-ACCESS    read-write
    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
      "Enables or disables logging of type 1 firewall event messages."
    ::= { cabhSecFwLogCtl 1 }

cabhSecFwEventType2Enable 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 2 firewall event messages."
    ::= { 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."
    ::= { 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."
     ::= { cabhSecFwLogCtl 4 }

cabhSecFwEventAttackAlertPeriod OBJECT-TYPE
     SYNTAX      INTEGER              (0..65535)
     MAX-ACCESS  read-write
     STATUS current
     DESCRIPTION
       "Indicates the period to be used (in days) for the
       cabhSecFwEventAttackAlertThreshold."
     ::= { cabhSecFwLogCtl 5 }
cabhSecCertPsCert OBJECT-TYPE
     SYNTAX      X509Certificate
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
       "The X.509 DER-encoded PS certificate."
     REFERENCE
       "Security Specification Section 11.3.2.2"
     ::= { cabhSecCertObjects 1 }

--
-- notification group is for future extension.
--

cabhSecNotification OBJECT IDENTIFIER ::= { cabhSecMib 3 0 }
cabhSecConformance OBJECT IDENTIFIER ::= { cabhSecMib 4 }
cabhSecCompliances OBJECT IDENTIFIER ::= { cabhSecConformance 1 }
cabhSecGroupsOBJECT 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 {
       cabhSecFwGroup
     }

::= { 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 MIB

The Definition MIB MUST be implemented as defined below.

```
CLAB-DEF-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    enterprises
                                FROM SNMPv2-SMI;

cableLabs MODULE-IDENTITY
    LAST-UPDATED  "0201310000Z" -- January 31, 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 }

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,
       DisplayString,
     RowStatus,
     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:
--
--
--
--=============================================================================

cabhCdpMib MODULE-IDENTITY
     LAST-UPDATED  "0112190000Z" -- December 19, 2001
     ORGANIZATION  "Cable NMP Group"
     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 CDP and the CAP portions of the PS database.

       Acknowledgements:
       "
     ::= { clabProjCableHome 4 }

-- Textual conventions
CabhCdpLanTransDhcpClientId ::= TEXTUAL-CONVENTION
     STATUS   current
     DESCRIPTION
       "LAN-Trans DHCP option61 information."
     SYNTAX OCTET STRING (SIZE (1..80))
```

```
--
-- assumes SNMPv3
-- SW load management is per DOCSIS 1.1 only
--

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 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 and all
       current mappings to use the factory default settings at
       the next lease renewal time. Reading this object always
       returns false(2). When cabhCdpDhcpReset is set to true,
       the following actions occur:
       1) Reset all default CDS DHCP options to the factory
       defaults.
       2) The CDS will offer the factory default DHCP options
       at the next lease renewal time.

       The objects set to factory defaults are:
          cabhCdpLanTransThreshold,
          cabhCdpLanTransAction,

          cabhCdpLanPoolStart,
          cabhCdpLanPoolEnd,
          cabhCdpServerSubnetMask,
          cabhCdpServerTimeOffset,
          cabhCdpServerRouter,
          cabhCdpServerDnsAddress,
          cabhCdpServerSyslogAddress,
          cabhCdpServerDomainName,
          cabhCdpServerTTL,
          cabhCdpServerInterfaceMTU,
          cabhCdpServerVendorSpecific,
          cabhCdpServerLeaseTime,
          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 WAN side addresses."
       REFERENCE
          ""
     ::= { cabhCdpBase 2 }
```

```
cabhCdpLanTransThreshold OBJECT-TYPE
     SYNTAX    INTEGER (1..65533)
     MAX-ACCESS    read-write
     STATUS    current
     DESCRIPTION
       "The threshold number of LAN-Trans IP addresses
       allocated or assigned above which an alarm
       condition MUST be generated. Whenever an attempt
       to allocate an LAN-Trans IP address when
       cabhCdpLanTransCurCount is greater than or equal
       to cabhCdpLanTransThreshold, an event is generated.
       For class C addresses, 253 is used as default. For
       class B addresses, 65533 is used as a default. In
       either case, this setting disables the feature."
       REFERENCE
          ""
       DEFVAL { 65533 }
     ::= { 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 }

--
--   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.
--
--============================================================================

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,
       cabhCdpLanAddrCreateTime   TimeStamp,
       cabhCdpLanAddrExpireTime   TimeStamp,
       cabhCdpLanAddrMethod            INTEGER,
       cabhCdpLanAddrHostName          DisplayString,
       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."
     ::= { 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."
     ::= { cabhCdpLanAddrEntry 2 }

cabhCdpLanAddrClientID OBJECT-TYPE
     SYNTAX         CabhCdpLanTransDhcpClientId
     MAX-ACCESS     read-only
     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."
     ::= { cabhCdpLanAddrEntry 3 }

cabhCdpLanAddrCreateTime 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 set only 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 }

cabhCdpLanAddrExpireTime 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        DisplayString(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 requesting a WAN-Data IP Address
        via DHCP."
      ::= { cabhCdpWanDataAddrEntry 2 }

cabhCdpWanDataAddrIpType OBJECT-TYPE
      SYNTAX         InetAddressType
      MAX-ACCESS     read-create
      STATUS         current
      DESCRIPTION
        "The address type assigned on the WAN-Data side."
      ::= { cabhCdpWanDataAddrEntry 3 }

cabhCdpWanDataAddrIp OBJECT-TYPE
      SYNTAX         InetAddress
      MAX-ACCESS     read-create
      STATUS         current
      DESCRIPTION
        "The address assigned on the WAN-Data side."
      ::= { cabhCdpWanDataAddrEntry 4 }

cabhCdpWanDataAddrRenewalTime OBJECT-TYPE
      SYNTAX         Integer32
      MAX-ACCESS     read-create
      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 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."
       ::= { 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 }

cabhCdpServerSubnetMaskType OBJECT-TYPE
       SYNTAX      InetAddressType
     MAX-ACCESS    read-write
     STATUS        current
     DESCRIPTION
       "Type of LAN-Trans Subnet Mask."
     DEFVAL { ipv4 }
     ::= { cabhCdpServer 5 }

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 6 }

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 7 }
```

```
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 8 }

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 9 }

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 10 }

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 11 }

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 12 }

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 13 }
```

```
cabhCdpServerDomainName OBJECT-TYPE
       SYNTAX      DisplayString(SIZE(0..128))
     MAX-ACCESS    read-write
     STATUS        current
     DESCRIPTION
       "Option value 15 - Domain name of LAN-Trans address realm."
       DEFVAL {""}
       ::= { cabhCdpServer 14 }

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 15 }

cabhCdpServerInterfaceMTU OBJECT-TYPE
       SYNTAX      INTEGER (68..4096)
       MAX-ACCESS  read-write
     STATUS        current
     DESCRIPTION
       "Option value 26 - LAN-Trans Interface MTU."
       DEFVAL { 1500 }
       ::= { cabhCdpServer 16 }

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 17 }

cabhCdpServerLeaseTime OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS            "seconds"
       MAX-ACCESS  read-write
     STATUS        current
     DESCRIPTION
       "Option value 51 - LAN-Trans default Lease Time (seconds)."
       DEFVAL { 60 }
       ::= { cabhCdpServer 18 }

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 19 }

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 20 }

--
-- 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,

          cabhCdpLanAddrIpType,
          cabhCdpLanAddrIp,
          cabhCdpLanAddrClientID,
          cabhCdpLanAddrCreateTime,
          cabhCdpLanAddrExpireTime,
          cabhCdpLanAddrMethod,
          cabhCdpLanAddrHostName,
          cabhCdpLanAddrRowStatus,

          cabhCdpWanDataAddrIndex,
          cabhCdpWanDataAddrClientId,
          cabhCdpLanAddrIpType,
          cabhCdpWanDataAddrIp,
          cabhCdpWanDataAddrRenewalTime,
          cabhCdpWanDataAddrRowStatus,

          cabhCdpWanDataAddrDnsIpType,
          cabhCdpWanDataAddrDnsIp,
```

```
          cabhCdpWanDataAddrDnsRowStatus,

          cabhCdpLanPoolStartType,
          cabhCdpLanPoolStart,
          cabhCdpLanPoolEndType,
          cabhCdpLanPoolEnd,
          cabhCdpServerSubnetMaskType,
          cabhCdpServerSubnetMask,
          cabhCdpServerTimeOffset,
          cabhCdpServerRouterType,
          cabhCdpServerRouterType,
          cabhCdpServerRouter,
          cabhCdpServerDnsAddressType,
          cabhCdpServerDnsAddress,
          cabhCdpServerSyslogAddressType,
          cabhCdpServerSyslogAddress,
          cabhCdpServerDomainName,
          cabhCdpServerTTL,
          cabhCdpServerInterfaceMTU,
          cabhCdpServerVendorSpecific,
          cabhCdpServerLeaseTime,
          cabhCdpServerDhcpAddressType,
          cabhCdpServerDhcpAddress
              }
     STATUS   current
     DESCRIPTION
       "Group of objects for Cable CDB 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:
--
--
--============================================================================

cabhCapMib MODULE-IDENTITY
     LAST-UPDATED  "0112190000Z" -- December 19, 2001
     ORGANIZATION  "Cable NMP Group"
     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 CDP and the CAP portions of the PS database.

       Acknowledgements:
       "
     ::= { 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
                   }

--
-- assumes SNMPv3
-- SW load management is per DOCSIS 1.1 only
--

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
       "The maximum time to wait before assuming TCP
       session is terminated."
```

```
       REFERENCE
          ""
       DEFVAL { 240 }        -- 4 minutes
     ::= { cabhCapBase 1 }

cabhCapUdpTimeWait OBJECT-TYPE
     SYNTAX      Unsigned32
       UNITS     "seconds"
     MAX-ACCESS  read-write
     STATUS      current
     DESCRIPTION
       "The maximum time to wait before assuming UDP
       session is terminated."
       REFERENCE
          ""
       DEFVAL { 86400 } -- 1 day
     ::= { cabhCapBase 2 }

cabhCapIcmpTimeWait OBJECT-TYPE
     SYNTAX      Unsigned32
       UNITS     "seconds"
     MAX-ACCESS  read-write
     STATUS      current
     DESCRIPTION
       "The maximum time to wait before assuming Icmp
       session is terminated."
       REFERENCE
          ""
       DEFVAL { 86400 } -- 1 day
     ::= { 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 the all the tables in the CAP
       to be cleared, and all CAP objects with defaults to be reset back to
       their default values."
     ::= { 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 { cabhCapMappingWanAddrType, cabhCapMappingWanAddr,
     cabhCapMappingWanPort,
       cabhCapMappingLanAddrType, cabhCapMappingLanAddr, cabhCapMappingLanPort}
     ::= { cabhCapMappingTable 1 }

CabhCapMappingEntry ::= SEQUENCE {
     cabhCapMappingWanAddrType   InetAddressType,
       cabhCapMappingWanAddr          InetAddress,
       cabhCapMappingWanPort          INTEGER,
       cabhCapMappingLanAddrType      InetAddressType,
       cabhCapMappingLanAddr          InetAddress,
       cabhCapMappingLanPort          INTEGER,
       cabhCapMappingMode         CabhCapPacketMode,
       cabhCapMappingMethod           INTEGER,
       cabhCapMappingProtocol              INTEGER
       }

cabhCapMappingWanAddrType OBJECT-TYPE
     SYNTAX      InetAddressType
     MAX-ACCESS    not-accessible
     STATUS      current
     DESCRIPTION
       "The IP address type assigned on the WAN side. IP version 4
       is typically used."
     ::= { cabhCapMappingEntry 1 }

cabhCapMappingWanAddr OBJECT-TYPE
     SYNTAX      InetAddress
     MAX-ACCESS    not-accessible
     STATUS        current
     DESCRIPTION
       "The IP address assigned on the WAN side. IP version 4
       is typically used."
     ::= { cabhCapMappingEntry 2 }

cabhCapMappingWanPort OBJECT-TYPE
     SYNTAX      INTEGER (1..65535)
     MAX-ACCESS    not-accessible
     STATUS        current
     DESCRIPTION
       "The TCP/UDP port number on the WAN side."
     ::= { cabhCapMappingEntry 3 }

cabhCapMappingLanAddrType OBJECT-TYPE
     SYNTAX        InetAddressType
     MAX-ACCESS    not-accessible
     STATUS        current
     DESCRIPTION
       "The IP address type assigned on the LAN side. IP version 4
       is typically used."
     ::= { cabhCapMappingEntry 4 }

cabhCapMappingLanAddr OBJECT-TYPE
     SYNTAX        InetAddress
     MAX-ACCESS    not-accessible
     STATUS        current
```

```
     DESCRIPTION
       "The IP address assigned on the LAN side. IP version 4
       is typically used."
     ::= { cabhCapMappingEntry 5 }

cabhCapMappingLanPort OBJECT-TYPE
     SYNTAX         INTEGER (1..65535)
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
       "The TCP/UDP port number on the LAN side."
     ::= { cabhCapMappingEntry 6 }

cabhCapMappingMode OBJECT-TYPE
     SYNTAX         CabhCapPacketMode
     MAX-ACCESS          read-only
     STATUS         current
     DESCRIPTION
       "The type of packet-handling mode for this mapping. Note that this
       information could be gleaned from the IP address and Port information for
       this mapping."
     ::= { 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-only
     STATUS    current
     DESCRIPTION
       "The protocol for this mapping."
     ::= { cabhCapMappingEntry 9 }

--------------------------------------------------------------------------
--
--   cabhCapPassthroughTable (CAP Passthrough Table)
--
--   The cabhCapPassthroughTable contains the MAC Addresses for all LAN-IP
     Devices which will be configured as pass-through.
--
--------------------------------------------------------------------------

cabhCapPassthroughTable OBJECT-TYPE
     SYNTAX         SEQUENCE OF CabhCapPassthroughEntry
     MAX-ACCESS     not-accessible
     STATUS         current
     DESCRIPTION
       "This table contains MAC addresses for LAN-IP Devices which are
       configured as passthrough mode."
```

```
     ::= { cabhCapMap 2 }

cabhCapPassthroughEntry OBJECT-TYPE
     SYNTAX        CabhCapPassthroughEntry
     MAX-ACCESS    not-accessible
     STATUS        current
     DESCRIPTION
       "List of MAC addresses for LAN-IP Devices which are configured as
       passthrough mode."
     INDEX {cabhCapPassthroughMACAddr }
     ::= { cabhCapPassthroughTable 1 }

CabhCapPassthroughEntry ::= SEQUENCE {
     cabhCapPassthroughMACAddr        PhysAddress,
     cabhCapPassthroughRowStatus           RowStatus
     }

cabhCapPassthroughMACAddr OBJECT-TYPE
     SYNTAX        PhysAddress
     MAX-ACCESS    not-accessible
     STATUS        current
     DESCRIPTION
       "MAC Address of LAN-IP Device to be configured as passthrough mode."
     ::= {cabhCapPassthroughEntry 1 }

cabhCapPassthroughRowStatus OBJECT-TYPE
     SYNTAX        RowStatus
     MAX-ACCESS    read-create
     STATUS        current
     DESCRIPTION
       "The RowStatus interlock for creation and deletion
       of cabhCapPassthroughTable entry."
     ::= { cabhCapPassthroughEntry 2 }

--
-- 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 3 }
```

```
cabhCapGroup OBJECT-GROUP
     OBJECTS {
       cabhCapTcpTimeWait,
       cabhCapUdpTimeWait,
       cabhCapIcmpTimeWait,
       cabhCapPrimaryMode,

--     cabhCapMappingWanAddrType,
--     cabhCapMappingWanAddr,
--     cabhCapMappingWanPort,
--     cabhCapMappingLanAddrType,
--     cabhCapMappingLanAddr,
--     cabhCapMappingLanPort,
       cabhCapMappingMode,
     cabhCapMappingMethod,
       cabhCapMappingProtocol,

--     cabhCapPassthroughMacAddr
       cabhCapPassthroughRowStatus
       }
     STATUS    current
     DESCRIPTION
       "Group of objects for CDB 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 and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2003