TECHNICAL REPORT

TR-098

Internet Gateway Device Data Model for TR-069

Issue: 1 Amendment 2
Issue Date: September 2008

Notice

The Broadband Forum is a non-profit corporation organized to create guidelines for broadband network system development and deployment. This Broadband Forum Technical Report has been approved by members of the Forum. This Broadband Forum Technical Report is not binding on the Broadband Forum, any of its members, or any developer or service provider. This Broadband Forum Technical Report is subject to change, but only with approval of members of the Forum.

This Broadband Forum Technical Report is provided "as is," with all faults. Any person holding a copyright in this Broadband Forum Technical Report, or any portion thereof, disclaims to the fullest extent permitted by law any representation or warranty, express or implied, including, but not limited to –

(a) any warranty of merchantability, fitness for a particular purpose, non-infringement, or title;
(b) any warranty that the contents of this Broadband Forum Technical Report are suitable for any purpose, even if that purpose is known to the copyright holder;
(c) any warranty that the implementation of the contents of the documentation will not infringe any third party patents, copyrights, trademarks or other rights.

This Broadband Forum publication may incorporate intellectual property. The Broadband Forum encourages but does not require declaration of such intellectual property. For a list of declarations made by Broadband Forum member companies, please see http://www.broadband-forum.org.

Broadband Forum Technical Reports may be copied, downloaded, stored on a server or otherwise re-distributed in their entirety only. The text of this notice must be included in all copies.

Issue History

Issue Number	Issue Date	Issue Editor	Changes
Issue 1	September 2005	Jeff Bernstein, 2Wire Barbara Stark, BellSouth	Issue 1
Issue 1 Amendment 1	November 2006	Jeff Bernstein, 2Wire John Blackford, 2Wire Mike Digdon, SupportSoft Heather Kirksey, Motive William Lupton, 2Wire Anton Okmianski, Cisco	Clarification of original document
Issue 1 Amendment 2	September 2008	Christele Bouchat, Alcatel William Lupton, 2Wire	Clarification of Amendment 1; Data model extensions (v1.4); See section 1 for further details

Technical comments or questions about this Technical Report should be directed to:

Editors	William Lupton Christele Bouchat	2Wire Alcatel-Lucent	wlupton@2wire.com christele.bouchat@alcatel-lucent.be
DSLHome™ Technical Working Group Chairs	Greg Bathrick Heather Kirksey	PMC-Sierra Motive	Greg_Bathrick@pmc-sierra.com hkirksey@motive.com

Table of Contents

List of Figures

List of Tables

Summary

Defines the Internet Gateway Device data model for the CPE WAN Management Protocol (TR-069).

1 Introduction

This document describes the Internet Gateway Device data model for the CPE WAN Management Protocol (CWMP). TR-069 defines the generic requirements of the management protocol methods which can be applied to any TR-069 CPE. It is intended to support a variety of different functionalities to manage a collection of CPE, including the following primary capabilities:

- Auto-configuration and dynamic service provisioning
- Software/firmware image management
- Status and performance monitoring
- Diagnostics

The ability to manage the home network remotely has a number of benefits including reducing the costs associated with activation and support of broadband services, improving time-to-market for new products and services, and improving the user experience.

If TR-069 defines the generic methods for any device, other documents (such as this one) specify the managed objects, or data models, which are collections of objects and parameters on which the generic methods act to configure, diagnose, and monitor the state of specific devices and services.

The following figure places TR-069 and this document in the end-to-end management architecture:

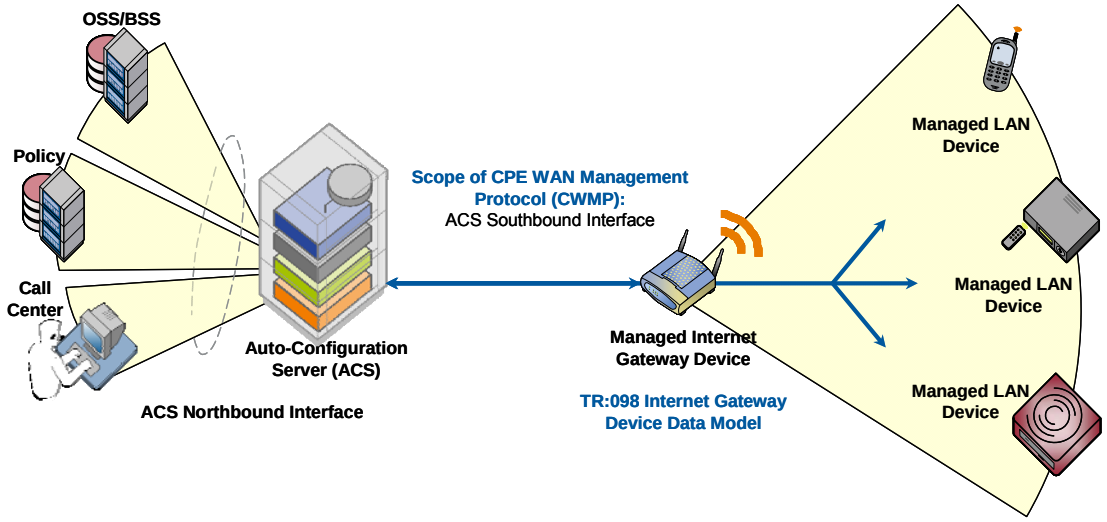

Figure 1 – Positioning in the End-to-End Architecture

The ACS is a server that resides in the network and manages devices in the subscriber premises. It uses the methods, or RPCs, defined to TR-069 to get and set the state of the device, initiate diagnostic tests, download and upload files, and manage events. This document defines those objects applicable to management of an Internet Gateway Device delivering broadband service.

The Internet Gateway Device data model follows the conventions defined in [3] for versioning of data models and the use of profiles.

This document, TR-098 Amendment 2, defines version 1.4 of the IGD data model. It updates and enhances TR-098 Amendment 1 in a number of ways, including:

- Enhanced management of LAN hosts, and addition of DHCP conditional serving capabilities,
- Improvements to management or QoS, routing, and bridging,
- Significant WiFi improvements, including configuration of WMM and U-APSD, and various fixes to the existing WiFi data model,
- PPPoE and NAT management enhancements,
- Enhancements to DSL and Ethernet statistics, including support for VDSL2.

1.1 Terminology

The following terminology is used throughout the series of documents defining the CPE WAN Management Protocol.

ACS	Auto-Configuration Server. This is a component in the broadband network responsible for auto-configuration of the CPE for advanced services.
ATM	Asynchronous Transfer Mode.
B-NT	Broadband-Network Termination. A specific type of Broadband CPE used in DSL networks.
CBR	Constant Bitrate.
CPE	Customer Premises Equipment; refers to any TR-069-compliant device and therefore covers both Internet Gateway Devices and LAN-side end devices.
CWMP	CPE WAN Management Protocol. Defined in [2], CWMP is a communication protocol between an ACS and CPE that defines a mechanism for secure auto-configuration of a CPE and other CPE management functions in a common framework.
Data Model	A hierarchical set of Parameters that define the managed objects accessible via TR-069 for a particular device or service.
Device	Used interchangeably with CPE.
Event	An indication that something of interest has happened that requires the CPE to notify the ACS.
ICMP	Internet Control Message Protocol.
IGD	Used interchangeably with Internet Gateway Device.
Internet Gateway Device	A CPE device, typically a broadband router, that acts as a gateway between the WAN and the LAN.
IPTV	Internet Protocol Television.
ISP	Internet Service Provider.
Parameter	A name-value pair representing a manageable CPE parameter made accessible to an ACS for reading and/or writing.
PVC	Permanent Virtual Circuit.
QoS	Quality of Service.
RG	Residential Gateway.
RPC	Remote Procedure Call.
RTP	Real-time Transport Protocol; RFC 3550 [46].
SAR	Segmentation and Reassembly.

VBR Variable Bitrate. An "-rt" suffix indicates "real time".

VoIP Voice over Internet Protocol.

1.2 Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1].

The key words "DEPRECATED" and "OBSOLETED" in this document are to be interpreted as defined in [3].

2 Data Model Definition

2.1 General Notation

Parameter names use a hierarchical form similar to a directory tree. The name of a particular Parameter is represented by the concatenation of each successive node in the hierarchy separated with a "." (dot), starting at the trunk of the hierarchy and leading to the leaves. When specifying a partial path, indicating an intermediate node in the hierarchy, the trailing "." (dot) is always used as the last character.

Parameter names MUST be treated as case sensitive.

In some cases, where multiple instances of an object can occur, the placeholder node name "{i}" is shown. In actual use, this placeholder is to be replaced by an instance number, which MUST be a positive integer (≥ 1). Because in some cases object instances can be deleted, instance numbers will in general not be contiguous.

2.2 Data Types

The parameters defined in this specification make use of a limited subset of the default SOAP data types [4]. The complete set of parameter data types along with the notation used to represent these types is listed in Table 1.

Table 1 – Data types

Type	Description
object	A container for parameters and/or other objects. The full path name of a parameter is given by the parameter name appended to the full path name of the object it is contained within.
string	For strings listed in this specification, a maximum allowed length can be listed using the form string(N), where N is the maximum string length in characters. A "k" or "K" suffix is interpreted as a 1024 (not 1000) multiplier, e.g. 32k means 32768. For all strings a maximum length is either explicitly indicated or implied by the size of the elements composing the string. For strings in which the content is an enumeration, the longest enumerated value determines the maximum length. If a string does not have an explicitly indicated maximum length or is not an enumeration, the default maximum is 16 characters. When transporting a string value within an XML document, any characters which are special to XML MUST be escaped as specified by the XML specification [12]. Additionally, any characters other than printable ASCII characters, i.e. any characters whose decimal ASCII representations are outside the (inclusive) range 32-126, SHOULD be escaped as specified by the XML specification.
int	Integer in the range –2147483648 to +2147483647, inclusive. For some int types listed, a value range is given using the form int[Min:Max], where the Min and Max values are inclusive. If either Min or Max are missing, this indicates no limit. A "k" or "K" suffix is interpreted as a 1024 (not 1000) multiplier, e.g. 32k means 32768.

Type	Description
unsignedInt	Unsigned integer in the range 0 to 4294967295, inclusive. For some unsignedInt types listed, a value range is given using the form unsignedInt[Min:Max], where the Min and Max values are inclusive. If either Min or Max are missing, this indicates no limit. A "k" or "K" suffix is interpreted as a 1024 (not 1000) multiplier, e.g. 32k means 32768.
boolean	Boolean, where the allowed values are "0", "1", "true", and "false". The values "1" and "true" are considered interchangeable, where both equivalently represent the logical value *true*. Similarly, the values "0" and "false" are considered interchangeable, where both equivalently represent the logical value *false*.
dateTime	The subset of the ISO 8601 date-time format defined by the SOAP dateTime type. All times MUST be expressed in UTC (Universal Coordinated Time) unless explicitly stated otherwise in the definition of a parameter of this type. If absolute time is not available to the CPE, it SHOULD instead indicate the relative time since boot, where the boot time is assumed to be the beginning of the first day of January of year 1, or 0001-01-01T00:00:00. For example, 2 days, 3 hours, 4 minutes and 5 seconds since boot would be expressed as 0001-01-03T03:04:05. Relative time since boot MUST be expressed using an untimezoned representation. Any untimezoned value with a year value less than 1000 MUST be interpreted as a relative time since boot. If the time is unknown or not applicable, the following value representing "Unknown Time" MUST be used: 0001-01-01T00:00:00Z. Any dateTime value other than one expressing relative time since boot (as described above) MUST use timezoned representation (that is, it MUST include a timezone suffix).
base64	Base64 encoded binary (no line-length limitation). A maximum allowed length can be listed using the form base64(N), where N is the maximum length in characters after Base64 encoding. A "k" or "K" suffix is interpreted as a 1024 (not 1000) multiplier, e.g. 32k means 32768.

All IPv4 addresses and subnet masks are represented as strings in IPv4 dotted-decimal notation. All IPv6 addresses and subnet masks MUST be represented using any of the 3 standard textual representations as defined in RFC 3513 [45], sections 2.2.1, 2.2.2 and 2.2.3. Both lower-case and upper-case letters can be used. Use of the lower-case letters is RECOMMENDED. Examples of valid IPv6 address textual representations:

- 1080:0:0:800:ba98:3210:11aa:12dd
- 1080::800:ba98:3210:11aa:12dd
- 0:0:0:0:0:0:13.1.68.3

Unspecified or inapplicable IP addresses and subnet masks MUST be represented as empty strings unless otherwise specified by the parameter definition.

All MAC addresses are represented as strings of 12 hexadecimal digits (digits 0-9, letters A-F or a-f) displayed as six pairs of digits separated by colons. Unspecified or inapplicable MAC addresses MUST be represented as empty strings unless otherwise specified by the parameter definition.

For unsignedInt parameters that are used for statistics, e.g. for byte counters, the actual value of the statistic might be greater than the maximum value that can be represented as an unsignedInt. Such values SHOULD wrap around through zero. The term "packet" is to be interpreted as the transmission unit appropriate to the protocol layer in question, e.g. an IP packet or an Ethernet frame.

For strings that are defined to contain comma-separated lists, the format is defined as follows. Between every pair of successive items in a comma-separated list there MUST be a separator. The separator MUST include exactly one comma character, and MAY also include one or more space characters before or after the comma. The entire separator, including any space characters, MUST NOT be considered part of the list items it separates. The last item in a comma-separated list MUST NOT be followed with a separator. Individual items in a comma-separated list MUST NOT include a space or comma character within them. If an item definition requires the use of spaces or commas, that definition MUST specify the use of an escape mechanism that prevents the use of these characters.

For string parameters whose value is defined to contain the full hierarchical name of an object, the representation of the object name MUST NOT include a trailing "dot." An example of a parameter of this kind in the InternetGatewayDevice data model is InternetGatewayDevice.Layer3Forwarding.Default-ConnectionService. For this parameter, the following is an example of a properly formed value:

InternetGatewayDevice.WANDevice.1.WANConnectionDevice.2.WANPPPConnection.1

2.3 Vendor-Specific Parameters

A vendor MAY extend the standardized parameter list with vendor-specific parameters and objects. Vendor-specific parameters and objects MAY be defined either in a separate naming hierarchy or within the standardized naming hierarchy.

The name of a vendor-specific parameter or object not contained within another vendor-specific object MUST have the form:

X_<VENDOR>_VendorSpecificName

In this definition <VENDOR> is a unique vendor identifier, which MAY be either an OUI or a domain name. The OUI or domain name used for a given vendor-specific parameter MUST be one that is assigned to the organization that defined this parameter (which is not necessarily the same as the vendor of the CPE or ACS). An OUI is an organizationally unique identifier as defined in [5], which MUST formatted as a six-hexadecimal-digit string using all upper-case letters and including any leading zeros. A domain name MUST be upper case with each dot (".") replaced with a hyphen or underscore.

The VendorSpecificName MUST be a valid string as defined in 2.2, and MUST NOT contain a "." (period) or a space character.

Note – the use of the string "X_" to indicate a vendor-specific parameter implies that no standardized parameter can begin with "X_".

The name of a vendor-specific parameter or object that is contained within another vendor-specific object which itself begins with the prefix described above need not itself include the prefix.

The full path name of a vendor-specific parameter or object MUST NOT exceed 256 characters in length.

Below are some example vendor-specific parameter and object names:

InternetGatewayDevice.UserInterface.X_012345_AdBanner

InternetGatewayDevice.LANDevice.1.X_012345_LANInfraredInterfaceConfig.2.Status

X_GAMECO-COM_GameDevice.Info.Type

When appropriate, a vendor MAY also extend the set of values of an enumeration. If this is done, the vendor-specified values MUST be in the form "X_<VENDOR>_VendorSpecificValue". The total length of such a string MUST NOT exceed 31 characters.

2.4 InternetGatewayDevice Data Model

Table 2 defines version 1.4 of the InternetGatewayDevice data model. This definition is a superset of previously defined versions, 1.3, 1.2, 1.1 and 1.0. The table lists the objects defined for an Internet Gateway Device, and the corresponding parameters within those objects.

For a given implementation of this data model, the CPE MUST indicate support for the highest version number of any object or parameter that it supports. For example, even if the CPE supports only a single parameter that was introduced in version 1.4, then it would have to indicate support for version 1.4. The version number associated with each object and parameter is shown in the Version column of Table 2.

Table 2 – Definition of InternetGatewayDevice:1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.	object	-	The top-level object for an Internet Gateway Device.	-	1.0
DeviceSummary	string(1024)	-	As defined in [3].	-	1.1
LANDeviceNumberOfEntries	unsignedInt	-	Number of instances of LANDevice.	-	1.0
WANDeviceNumberOfEntries	unsignedInt	-	Number of instances of WANDevice.	-	1.0
InternetGatewayDevice.Capabilities.	object	-	The capabilities of the device. This is a constant read-only object, meaning that only a firmware upgrade will cause these values to be altered.	-	1.3
InternetGatewayDevice.Capabilities.-PerformanceDiagnostic.	object	-	The capabilities of the Performance Diagnostics (DownloadDiagnostics and UploadDiagnostics) for the device.	-	1.3
DownloadTransports	string	-	Comma-separated list of supported Download-Diagnostics transport protocols for a CPE device. Each item in the list is an enumeration of: "HTTP" "FTP" (OPTIONAL)	-	1.3
UploadTransports	string	-	Comma-separated list of supported Upload-Diagnostics transport protocols for a CPE device. Each item in the list is an enumeration of: "HTTP" "FTP" (OPTIONAL)	-	1.3
InternetGatewayDevice.DeviceInfo.	object	-	This object contains general device information.	-	1.0
Manufacturer	string(64)	-	The manufacturer of the CPE (human readable string).	-	1.0

[1] The full name of a Parameter is the concatenation of the object name shown in the yellow header with the individual Parameter name.

[2] "W" indicates the parameter MAY be writable (if "W" is not present, the parameter is defined as read-only). For an object, "W" indicates object instances can be Added or Deleted.

[3] The default value of the parameter on creation of an object instance via TR-069. If the default value is an empty string, this is represented by the symbol <Empty>. A hyphen indicates that no default value is specified. For a parameter in which no default value is specified, on creation of a parent object instance, the CPE MUST set the parameter to a value that is valid according to the definition of that parameter.

[4] The Version column indicates the minimum data model version REQUIRED to support the associated Parameter or Object.

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ManufacturerOUI	string(6)	-	Organizationally unique identifier of the device manufacturer. Represented as a six hexadecimal-digit value using all upper-case letters and including any leading zeros. The value MUST be a valid OUI as defined in [5]. This value MUST remain fixed over the lifetime of the device, including across firmware updates.	-	1.0
ModelName	string(64)	-	Model name of the CPE (human readable string).	-	1.0
Description	string(256)	-	A full description of the CPE device (human readable string).	-	1.0
ProductClass	string(64)	-	Identifier of the class of product for which the serial number applies. That is, for a given manufacturer, this parameter is used to identify the product or class of product over which the SerialNumber parameter is unique. This value MUST remain fixed over the lifetime of the device, including across firmware updates.	-	1.0
SerialNumber	string(64)	-	Serial number of the CPE. This value MUST remain fixed over the lifetime of the device, including across firmware updates.	-	1.0
HardwareVersion	string(64)	-	A string identifying the particular CPE model and version.	-	1.0
SoftwareVersion	string(64)	-	A string identifying the software version currently installed in the CPE. To allow version comparisons, this element SHOULD be in the form of dot-delimited integers, where each successive integer represents a more minor category of variation. For example, 3.0.21 where the components mean: Major.Minor.Build.	-	1.0
ModemFirmwareVersion	string(64)	-	A string identifying the version of the modem firmware currently installed in the CPE. This is applicable only when the modem firmware is separable from the overall CPE software.	-	1.0
EnabledOptions	string(1024)	-	Comma-separated list of the OptionName of each Option that is currently enabled in the CPE. The OptionName of each is identical to the OptionName element of the OptionStruct described in [2]. Only those options are listed whose State indicates the option is enabled.	-	1.0
AdditionalHardwareVersion	string(64)	-	A comma-separated list of any additional versions. Represents any additional hardware version information the vendor might wish to supply.	-	1.0
AdditionalSoftwareVersion	string(64)	-	A comma-separated list of any additional versions. Represents any additional software version information the vendor might wish to supply.	-	1.0
SpecVersion	string(16)	-	Represents the version of the specification implemented by the device. Currently 1.0 is the only available version. The value of this parameter MUST equal “1.0”. This parameter is DEPRECATED because its value is fixed and it therefore serves no purpose. However, it is a Forced Inform parameter and therefore cannot be OBSOLETED.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ProvisioningCode	string(64)	W	Identifier of the primary service provider and other provisioning information, which MAY be used by the ACS to determine service provider-specific customization and provisioning parameters. If non-empty, this argument SHOULD be in the form of a hierarchical descriptor with one or more nodes specified. Each node in the hierarchy is represented as a 4-character sub-string, containing only numerals or upper-case letters. If there is more than one node indicated, each node is separated by a "." (dot). Examples: “TLCO” or “TLCO.GRP2”.	-	1.0
UpTime	unsignedInt	-	Time in seconds since the CPE was last restarted.	-	1.0
FirstUseDate	dateTime	-	Date and time in UTC that the CPE first both successfully established an IP-layer network connection and acquired an absolute time reference using NTP or equivalent over that network connection. The CPE MAY reset this date after a factory reset. If NTP or equivalent is not available, this parameter, if present, SHOULD be set to the Unknown Time value.	-	1.0
DeviceLog	string(32K)	-	Vendor-specific log(s).	-	1.0
VendorConfigFileNumberOfEntries	unsignedInt	-	Number of instances of VendorConfigFile.	-	1.0
InternetGatewayDevice.DeviceInfo.Vendor-ConfigFile.{i}.	object	-	Every instance of this object is a Vendor Configuration File, and contains parameters associated with the Vendor Configuration File. This table of Vendor Configuration Files is for information only and does not allow the ACS to operate on these files in any way. Whenever the CPE successfully downloads a configuration file as a result of the Download RPC with the FileType argument of “3 Vendor Configuration File”, the CPE MUST update this table. If the name of the file (determined as described in the definition of the Name parameter) differs from that of any existing instance, then the CPE MUST create a new instance to represent this file. If instead, the name of the file is identical to that of an existing instance, then the CPE MUST update the content of the existing instance with the new version, date, and (optionally) description of the file.	-	1.0
Name	string(64)	-	Name of the vendor configuration file. If the CPE is able to obtain the name of the configuration file from the file itself, then the value of this parameter MUST be set to that name. Otherwise, if the CPE can extract the file name from the URL used to download the configuration file, then the value of this parameter MUST be set to that name. Otherwise, the value of this parameter MUST be set to the value of the TargetFileName argument of the Download RPC used to download this configuration file.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Version	string(16)	-	A string identifying the configuration file version currently used in the CPE. If the CPE is able to obtain the version of the configuration file from the file itself, then the value of this parameter MUST be set to the obtained value. Otherwise, the value of this parameter MUST be empty.	-	1.0
Date	dateTime	-	Date and time when the content of the current version of this vendor configuration file was first applied by the CPE.	-	1.0
Description	string(256)	-	A description of the vendor configuration file (human-readable string).	-	1.0
InternetGatewayDevice.DeviceConfig.	object	-	This object contains general configuration parameters.	-	1.0
PersistentData	string(256)	W	Arbitrary user data that MUST persist across CPE reboots.	-	1.0
ConfigFile	string(32K)	W	A dump of the currently running configuration on the CPE. This parameter enables the ability to backup and restore the last known good state of the CPE. It returns a vendor-specific document that defines the state of the CPE. The document MUST be capable of restoring the CPE's state when written back to the CPE using SetParameterValues. An alternative to this parameter, e.g. when the configuration file is larger than the parameter size limit, is to use the Upload and Download RPCs with a FileType of "1 Vendor Configuration File".	-	1.0
InternetGatewayDevice.ManagementServer.	object	-	This object contains parameters relating to the CPE's association with an ACS.	-	1.0
EnableCWMP	boolean	W	Enables and disables the CPE's support for CWMP. False means that CWMP support in the CPE is disabled, in which case the device MUST NOT send any Inform messages to the ACS or accept any Connection Request notifications from the ACS. True means that CWMP support on the CPE is enabled. The factory default value MUST be True. The subscriber can re-enable the CPE's CWMP support either by performing a factory reset or by using a LAN-side protocol to change the value of this parameter back to True.	-	1.4
URL	string(256)	W	URL, as defined in [8], for the CPE to connect to the ACS using the CPE WAN Management Protocol. This parameter MUST be in the form of a valid HTTP or HTTPS URL [6]. The "host" portion of this URL is used by the CPE for validating the ACS certificate when using SSL or TLS. Note that on a factory reset of the CPE, the value of this parameter might be reset to its factory value. If an ACS modifies the value of this parameter, it SHOULD be prepared to accommodate the situation that the original value is restored as the result of a factory reset.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Username	string(256)	W	Username used to authenticate the CPE when making a connection to the ACS using the CPE WAN Management Protocol. This username is used only for HTTP-based authentication of the CPE. Note that on a factory reset of the CPE, the value of this parameter might be reset to its factory value. If an ACS modifies the value of this parameter, it SHOULD be prepared to accommodate the situation that the original value is restored as the result of a factory reset.	-	1.0
Password	string(256)	W	Password used to authenticate the CPE when making a connection to the ACS using the CPE WAN Management Protocol. This password is used only for HTTP-based authentication of the CPE. When read, this parameter returns an empty string, regardless of the actual value. Note that on a factory reset of the CPE, the value of this parameter might be reset to its factory value. If an ACS modifies the value of this parameter, it SHOULD be prepared to accommodate the situation that the original value is restored as the result of a factory reset.	-	1.0
PeriodicInformEnable	boolean	W	Whether or not the CPE MUST periodically send CPE information to the ACS using the Inform method call.	-	1.0
PeriodicInformInterval	unsignedInt [1:]	W	The duration in seconds of the interval for which the CPE MUST attempt to connect with the ACS and call the Inform method if PeriodicInformEnable is True.	-	1.0
PeriodicInformTime	dateTime	W	An absolute time reference in UTC to determine when the CPE will initiate the periodic Inform method calls. Each Inform call MUST occur at this reference time plus or minus an integer multiple of the PeriodicInformInterval. PeriodicInformTime is used only to set the “phase” of the periodic Informs. The actual value of PeriodicInformTime can be arbitrarily far into the past or future. For example, if PeriodicInformInterval is 86400 (a day) and if PeriodicInformTime is set to UTC midnight on some day (in the past, present, or future) then periodic Informs will occur every day at UTC midnight. These MUST begin on the very next midnight, even if PeriodicInformTime refers to a day in the future. The Unknown Time value defined in section 2.2 indicates that no particular time reference is specified. That is, the CPE MAY locally choose the time reference, and needs only to adhere to the specified PeriodicInformInterval. If absolute time is not available to the CPE, its periodic Inform behavior MUST be the same as if the PeriodicInformTime parameter was set to the Unknown Time value.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ParameterKey	string(32)	-	ParameterKey provides the ACS a reliable and extensible means to track changes made by the ACS. The value of ParameterKey MUST be equal to the value of the ParameterKey argument from the most recent successful SetParameterValues, AddObject, or DeleteObject method call from the ACS. The CPE MUST set ParameterKey to the value specified in the corresponding method arguments if and only if the method completes successfully and no fault response is generated. If a method call does not complete successfully (implying that the changes requested in the method did not take effect), the value of ParameterKey MUST NOT be modified. The CPE MUST only modify the value of ParameterKey as a result of SetParameterValues, AddObject, DeleteObject, or due to a factory reset. On factory reset, the value of ParameterKey MUST be set to empty.	-	1.0
ConnectionRequestURL	string(256)	-	HTTP URL, as defined in [8], for an ACS to make a Connection Request notification to the CPE. In the form: http://host:port/path The "host" portion of the URL MAY be the IP address for the management interface of the CPE in lieu of a host name.	-	1.0
ConnectionRequestUsername	string(256)	W	Username used to authenticate an ACS making a Connection Request to the CPE.	-	1.0
ConnectionRequestPassword	string(256)	W	Password used to authenticate an ACS making a Connection Request to the CPE. When read, this parameter returns an empty string, regardless of the actual value.	-	1.0
UpgradesManaged	boolean	W	Indicates whether or not the ACS will manage upgrades for the CPE. If True, the CPE SHOULD NOT use other means other than the ACS to seek out available upgrades. If False, the CPE MAY use other means for this purpose. Note that an autonomous upgrade (reported via an "10 AUTONOMOUS TRANSFER COMPLETE" Inform Event code) SHOULD be regarded as a managed upgade if it is performed according to ACS-specified policy.	-	1.0
KickURL	string(256)	-	Present only for a CPE that supports the Kicked RPC method. LAN-accessible URL, as defined in [8], from which the CPE can be "kicked" to initiate the Kicked RPC method call. MUST be an absolute URL including a host name or IP address as would be used on the LAN side of the CPE.	-	1.0
DownloadProgressURL	string(256)	-	Present only for a CPE that provides a LAN-side web page to show progress during a file download. LAN-accessible URL, as defined in [8], to which a web-server associated with the ACS MAY redirect a user's browser on initiation of a file download to observer the status of the download.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DefaultActiveNotificationThrottle	unsignedInt	W	This parameter is used to control throttling of active notifications sent by the CPE to the ACS. It defines the minimum number of seconds that the CPE MUST wait since the end of the last session with the ACS before establishing a new session for the purpose of delivering an active notification. In other words, if CPE needs to establish a new session with the ACS for the sole purpose of delivering an active notification, it MUST delay establishing such a session as needed to ensure that the minimum time since the last session completion has been met. The time is counted since the last successfully completed session, regardless of whether or not it was used for active notifications or other purposes. However, if connection to the ACS is established for purposes other than just delivering active notifications, including for the purpose of retrying a failed session, such connection MUST NOT be delayed based on this parameter value, and the pending active notifications MUST be communicated during that connection. The time of the last session completion does not need to be tracked across reboots.	-	1.4
UDPConnectionRequestAddress	string(256)	-	Address and port to which an ACS MAY send a UDP Connection Request to the CPE (see Annex G of [2]). This parameter is represented in the form of an Authority element as defined in [8]. The value MUST be in one of the following two forms: host:port host When STUNEnable is True, the “host” and “port” portions of this parameter MUST represent the public address and port corresponding to the NAT binding through which the ACS can send UDP Connection Request messages (once this information is learned by the CPE through the use of STUN). When STUNEnable is False, the “host” and “port” portions of the URL MUST represent the local IP address and port on which the CPE is listening for UDP Connection Request messages. The second form of this parameter MAY be used only if the port value is equal to “80”.	-	1.2
UDPConnectionRequestAddressNotification-Limit	unsignedInt	W	The minimum time, in seconds, between Active Notifications resulting from changes to the UDP-ConnectionRequestAddress (if Active Notification is enabled).	-	1.2
STUNEnable	boolean	W	Enables or disables the use of STUN by the CPE. This applies only to the use of STUN in association with the ACS to allow UDP Connection Requests.	-	1.2
STUNServerAddress	string(256)	W	Host name or IP address of the STUN server for the CPE to send Binding Requests if STUN is enabled via STUNEnable. If empty and STUNEnable is True, the CPE MUST use the address of the ACS extracted from the host portion of the ACS URL.	-	1.2

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
STUNServerPort	unsignedInt [0:65535]	W	Port number of the STUN server for the CPE to send Binding Requests if STUN is enabled via STUNEnable. By default, this SHOULD be the equal to the default STUN port, 3478.	-	1.2
STUNUsername	string(256)	W	If non-empty, the value of the STUN USERNAME attribute to be used in Binding Requests (only if message integrity has been requested by the STUN server). If empty, the CPE MUST NOT send STUN Binding Requests with message integrity.	-	1.2
STUNPassword	string(256)	W	The value of the STUN Password to be used in computing the MESSAGE-INTEGRITY attribute to be used in Binding Requests (only if message integrity has been requested by the STUN server). When read, this parameter returns an empty string, regardless of the actual value.	-	1.2
STUNMaximumKeepAlivePeriod	int[-1:]	W	If STUN Is enabled, the maximum period, in seconds, that STUN Binding Requests MUST be sent by the CPE for the purpose of maintaining the binding in the Gateway. This applies specifically to Binding Requests sent from the UDP Connection Request address and port. A value of -1 indicates that no maximum period is specified.	-	1.2
STUNMinimumKeepAlivePeriod	unsignedInt	W	If STUN Is enabled, the minimum period, in seconds, that STUN Binding Requests can be sent by the CPE for the purpose of maintaining the binding in the Gateway. This limit applies only to Binding Requests sent from the UDP Connection Request address and port, and only those that do not contain the BINDING-CHANGE attribute. This limit does not apply to retransmissions following the procedures defined in [9].	-	1.2
NATDetected	boolean	-	When STUN is enabled, this parameter indicates whether or not the CPE has detected address and/or port mapping in use. A True value indicates that the received MAPPED-ADDRESS in the most recent Binding Response differs from the CPE's source address and port. When STUNEnable is False, this value MUST be False.	-	1.2
ManageableDeviceNumberOfEntries	unsignedInt	-	Number of entries in the ManageableDevice table.	-	1.2
ManageableDeviceNotificationLimit	unsignedInt	W	The minimum time, in seconds, between Active Notifications resulting from changes to the ManageableDeviceNumberOfEntries (if Active Notification is enabled).	-	1.2
InternetGatewayDevice.ManagementServer.-ManageableDevice.{i}.	object	-	Each entry in this table corresponds to a distinct LAN Device that supports Device-Gateway Association according to Annex F of [2] as indicated by the presence of the DHCP option specified in that Annex.	-	1.2
ManufacturerOUI	string(6)	-	Organizationally unique identifier of the Device manufacturer as provided to the Gateway by the Device. Represented as a six hexadecimal-digit value using all upper-case letters and including any leading zeros. The value MUST be a valid OUI as defined in [5].	-	1.2
SerialNumber	string(64)	-	Serial number of the Device as provided to the Gateway by the Device.	-	1.2

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ProductClass	string(64)	-	Identifier of the class of product for which the Device's serial number applies as provided to the Gateway by the Device. If the Device does not provide a Product Class, then this parameter MUST be left empty.	-	1.2
Host	string(1024)	-	Comma-separated list of the full path names of all Host table entries, whether active or inactive, that correspond to this physical LAN Device. As such entries are added to or removed from the Host tables, the value of this parameter MUST be updated accordingly. For example: "InternetGatewayDevice.LANDevice.1.Hosts.Host.1,InternetGatewayDevice.LANDevice.1.Hosts.Host.5"	-	1.4
InternetGatewayDevice.Time.	object	-	This object contains parameters relating an NTP or SNTP time client in the CPE.	-	1.0
Enable	boolean	W	Enables or disables the NTP or SNTP time client.	-	1.4
Status	string	-	Status of Time support on the CPE. Enumeration of: "Disabled" "Unsynchronized" "Synchronized" "Error_FailedToSynchronize" "Error" (OPTIONAL) The "Unsynchronized" value indicates that the CPE's absolute time has not yet been set. The "Synchronized" value indicates that the CPE has acquired accurate absolute time; its current time is accurate. The "Error_FailedToSynchronize" value indicates that the CPE failed to acquire accurate absolute time; its current time is not accurate. The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-	1.4
NTPServer1	string(64)	W	First NTP timeserver. Either a host name or IP address.	-	1.0
NTPServer2	string(64)	W	Second NTP timeserver. Either a host name or IP address.	-	1.0
NTPServer3	string(64)	W	Third NTP timeserver. Either a host name or IP address.	-	1.0
NTPServer4	string(64)	W	Fourth NTP timeserver. Either a host name or IP address.	-	1.0
NTPServer5	string(64)	W	Fifth NTP timeserver. Either a host name or IP address.	-	1.0
CurrentLocalTime	dateTime	-	The current date and time in the CPE's local time zone.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
LocalTimeZone	string(6)	W	The local time zone offset from UTC, ignoring daylight savings time adjustments, in the form: +hh:mm -hh:mm For example, this will always be "-08:00" for California, "+00:00" or "-00:00" for the United Kingdom, and "+01:00" for France. This parameter is OBSOLETED because the information that it represents is fully covered by LocalTimeZoneName.	-	1.0
LocalTimeZoneName	string(64)	W	Name of the local time zone (human readable string). The name SHOULD be encoded according to IEEE 1003.1 (POSIX). The following is an example value: "EST+5EDT,M4.1.0/2,M10.5.0/2"	-	1.0
DaylightSavingsUsed	boolean	W	Whether or not daylight savings time is in use in the CPE's local time zone. This parameter is OBSOLETED because the information that it represents is fully covered by LocalTimeZoneName.	-	1.0
DaylightSavingsStart	dateTime	W	Current local date and time at which the switch to daylight savings time occurs. If daylight savings time is not used, this value is ignored. This parameter is OBSOLETED because the information that it represents is fully covered by LocalTimeZoneName.	-	1.0
DaylightSavingsEnd	dateTime	W	Current local date and time at which the switch from daylight savings time will occur. If daylight savings time is not used, this value is ignored. This parameter is OBSOLETED because the information that it represents is fully covered by LocalTimeZoneName.	-	1.0
InternetGatewayDevice.UserInterface.	object	-	This object contains parameters relating to the user interface of the CPE.	-	1.0
UserDatabaseSupported	boolean	-	Present only if the CPE provides a password-protected LAN-side user interface. Indicates whether or not the CPE supports a user database that provides per-user passwords that can be used for accessing the local user interface.	-	1.4
SharedPassword	boolean	W	Present only if the CPE provides a password-protected LAN-side user interface. Indicates whether or not a single shared password MUST protect the local user interface, or whether per-user passwords can be used. If either UserDatabaseSupported or Password-UserSelectable is False, the CPE MUST ignore the value of this parameter.	-	1.4
PasswordRequired	boolean	W	Present only if the CPE provides a password-protected LAN-side user interface. Indicates whether or not the local user interface MUST require a password to be chosen by the user. If False, the choice of whether or not a password is used is left to the user.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
PasswordUserSelectable	boolean	W	Present only if the CPE provides a password-protected LAN-side user interface and supports LAN-side Auto-Configuration. Indicates whether or not a password to protect the local user interface of the CPE MAY be selected by the user directly, or MUST be equal to the password used by the LAN-side Auto-Configuration protocol.	-	1.0
UpgradeAvailable	boolean	W	Indicates that a CPE upgrade is available, allowing the CPE to display this information to the user.	-	1.0
WarrantyDate	dateTime	W	Indicates the date and time in UTC that the warranty associated with the CPE is to expire.	-	1.0
ISPName	string(64)	W	The name of the customer's ISP.	-	1.0
ISPHelpDesk	string(32)	W	The help desk phone number of the ISP.	-	1.0
ISPHomePage	string(256)	W	The URL of the ISP's home page.	-	1.0
ISPHelpPage	string(256)	W	The URL of the ISP's on-line support page.	-	1.0
ISPLogo	base64 (5460)	W	Base64 encoded GIF or JPEG image. The binary image is constrained to 4095 bytes or less.	-	1.0
ISPLogoSize	unsignedInt [0:4095]	W	Un-encoded binary image size in bytes. If ISPLogoSize input value is 0 then the ISPLogo is cleared. ISPLogoSize can also be used as a check to verify correct transfer and conversion of Base64 string to image size.	-	1.0
ISPMailServer	string(256)	W	The URL of the ISP's mail server.	-	1.0
ISPNewsServer	string(256)	W	The URL of the ISP's news server.	-	1.0
TextColor	string(6)	W	The color of text on the GUI screens in RGB hexidecimal notation (e.g., FF0088).	-	1.0
BackgroundColor	string(6)	W	The color of the GUI screen backgrounds in RGB hexidecimal notation (e.g., FF0088).	-	1.0
ButtonColor	string(6)	W	The color of buttons on the GUI screens in RGB hexidecimal notation (e.g., FF0088).	-	1.0
ButtonTextColor	string(6)	W	The color of text on buttons on the GUI screens in RGB hexidecimal notation (e.g., FF0088).	-	1.0
AutoUpdateServer	string(256)	W	The server the CPE can check to see if an update is available for direct download to it. This MUST NOT be used by the CPE if the InternetGateway-Device.ManagementServer.UpgradesManaged parameter is True.	-	1.0
UserUpdateServer	string(256)	W	The server where a user can check via a web browser if an update is available for download to a PC. This MUST NOT be used by the CPE if the InternetGatewayDevice.ManagementServer.-UpgradesManaged parameter is True.	-	1.0
ExampleLogin	string(40)	W	An example of a correct login, according to ISP-specific rules.	-	1.0
ExamplePassword	string(30)	W	An example of a correct password, according to ISP-specific rules.	-	1.0
AvailableLanguages	string(256)	-	Comma-separated list of user-interface languages that are available, where each language is specified according to RFC 3066 [40].	-	1.4
CurrentLanguage	string(16)	W	Current user-interface language, specified according to RFC 3066 [40].	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.CaptivePortal.	object	-	This object contains parameters relating to the captive portal configuration on the CPE. The captive portal configuration defines the CPE's WAN-destined HTTP (port 80) traffic redirect behavior. When the captive portal is disabled, WAN-destined HTTP (port 80) traffic MUST be permitted to all destinations. When the captive portal is enabled, WAN-destined HTTP (port 80) traffic MUST be permitted only to destinations listed in the AllowedList; traffic to all other destinations MUST be redirected to the CaptivePortalURL.	-	1.4
Enable	boolean	W	Enables or disables the captive portal.	-	1.4
Status	string	-	Indicates the status of the captive portal. Enumeration of: "Disabled" "Enabled" "Error_URLEmpty" (CaptivePortalURL is empty) "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-	1.4
AllowedList	string (10000)	W	Comma-separated list of IP addresses to which HTTP (port 80) traffic MUST always be permitted, regardless of whether the captive portal is enabled. Each entry in the list MUST be either an IP address or an IP subnet specified using variable length subnet mask (VLSM) syntax. An IP subnet is specified as an IP address followed (with no intervening white space) by "/n", where n is an integer in the range 0-32; this is equivalent to a subnet mask consisting of n 1s followed by 32 minus n 0s. For example, 1.2.3.4 specifies a single IP address, and 1.2.3.4/24 specifies a class C subnet with subnet mask 255.255.255.0. The maximum length of a single entry (plus comma) is 19 characters so 10000 bytes is sufficient for more than 500 IP addresses and/or IP subnets.	-	1.4
CaptivePortalURL	string(2000)	W	Captive portal URL to which WAN-destined HTTP (port 80) traffic to destinations not listed in the AllowedList will be redirected. The captive portal URL MUST be an HTTP (not HTTPS) URL. The CPE MUST permit the captive portal URL to be set to an empty string, which has the effect of disabling the captive portal (if Enable is True and the captive portal URL is an empty string, Status MUST be "Error_URLEmpty").	-	1.4
InternetGatewayDevice.Layer3Forwarding.	object	-	This object allows the handling of the routing and forwarding configuration of the device.	-	1.0
DefaultConnectionService	string(256)	W	Specifies the default WAN interface. The content is the full hierarchical parameter name of the default layer 3 connection object. Example: "InternetGatewayDevice.WANDevice.1.WAN-ConnectionDevice.2.WANPPPConnection.1".	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ForwardNumberOfEntries	unsignedInt	-	Number of forwarding instances.	-	1.0
InternetGatewayDevice.Layer3Forwarding.-Forwarding.{i}.	object	W	Layer 3 forwarding table. In addition to statically configured routes, this table MUST include dynamic routes learned through layer 3 routing protocols, including RIP, OSPF, DHCP, and IPCP. The CPE MAY reject attempts to delete or modify a dynamic route entry. For each incoming packet, the layer 3 forwarding decision is conceptually made as follows: • Only table entries with a matching ForwardingPolicy are considered, i.e. those that either do not specify a ForwardingPolicy, or else specify a ForwardingPolicy that matches that of the incoming packet. • For the remaining table entries, those for which the source address/mask matches are sorted by longest prefix, i.e. with the most specific networks first (an unspecified source address is a wild-card and always matches, with a prefix length of zero). • For the remaining table entries, those for which the destination address/mask matches are sorted by longest prefix, i.e. with the most specific networks first (an unspecified destination address is a wild-card and always matches, with a prefix length of zero). • The first of the remaining table entries is applied to the packet.	-	1.0
Enable	boolean	W	Enables or disables the forwarding entry. On creation, an entry is disabled by default.	False	1.0
Status	string	-	Indicates the status of the forwarding entry. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	"Disabled"	1.0
StaticRoute	boolean	-	If True, this route is a Static route.	True	1.4
Type	string	W	Indicates the type of route. Enumeration of: "Default" "Network" "Host" This parameter is DEPRECATED because its value could conflict with DestIPAddress and/or DestSubnetMask.	"Host"	1.0
DestIPAddress	string	W	Destination address. An empty string or a value of "0.0.0.0" indicates no destination address is specified. A Forwarding table entry for which DestIPAddress and DestSubnetMask are both empty or "0.0.0.0" is a default route.	<Empty>	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestSubnetMask	string	W	Destination subnet mask. An empty string or a value of "0.0.0.0" indicates no destination subnet mask is specified. If a destination subnet mask is specified, the DestSubnetMask is ANDed with the destination address before comparing with the DestIPAddress. Otherwise, the full destination address is used as is. A Forwarding table entry for which DestIPAddress and DestSubnetMask are both empty or "0.0.0.0" is a default route.	<Empty>	1.0
SourceIPAddress	string	W	Source address. An empty string or a value of "0.0.0.0" indicates no source address is specified.	<Empty>	1.0
SourceSubnetMask	string	W	Source subnet mask. An empty string or a value of "0.0.0.0" indicates no source subnet mask is specified. If a source subnet mask is specified, the SourceSubnetMask is ANDed with the source address before comparing with the SourceIPAddress. Otherwise, the full source address is used as is.	<Empty>	1.0
ForwardingPolicy	int[-1:]	W	Identifier of a set of classes or flows that have the corresponding ForwardingPolicy value as defined in the QueueManagement object. A value of -1 indicates no ForwardingPolicy is specified. If specified, this forwarding entry is to apply only to traffic associated with the specified classes and flows.	-1	1.0
GatewayIPAddress	string	W	IP address of the gateway. Only one of GatewayIPAddress and Interface SHOULD be configured for a route. If both are configured, GatewayIPAddress and Interface MUST be consistent with each other.	<Empty>	1.0
Interface	string(256)	W	Specifies the egress interface associated with this entry. The content is the full hierarchical parameter name of the layer 3 connection object. Example: "InternetGatewayDevice.WANDevice.1.WAN-ConnectionDevice.2.WANPPPConnection.1". Only one of GatewayIPAddress and Interface SHOULD be configured for a route. If both are configured, GatewayIPAddress and Interface MUST be consistent with each other. For a route that was configured by setting GatewayIPAddress but not Interface, read access to Interface MUST return the full hierarchical parameter name for the route's egress interface.	-	1.0
ForwardingMetric	int[-1:]	W	Forwarding metric. A value of -1 indicates this metric is not used.	-1	1.0
MTU	unsignedInt [1:1540]	W	The maximum allowed size of an Ethernet frame for this route.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.Layer2Bridging.	object	-	Layer 2 bridging configuration. Specifies bridges between layer 2 LAN and/or WAN interfaces. Bridges can be defined to include layer 2 filter criteria to selectively bridge traffic between interfaces. This object can be used to configure both 802.1D [13] and 802.1Q [14] bridges. Not all 802.1D and 802.1Q features are modeled, and some additional features not present in either 802.1D or 802.1Q are modeled. If the Layer2Bridging object is implemented, the view that it provides of the CPE's underlying bridging configuration MUST be consistent with the view provided by any LANDevice and WAN**Conn-ection objects. The implications of this are explained in Annex A.6.	-	1.1
MaxBridgeEntries	unsignedInt	-	The maximum number of entries available in the Bridge table.	-	1.1
MaxDBridgeEntries	unsignedInt	-	The maximum number of 802.1D [13] entries available in the Bridge table. A positive value for this parameter implies support for 802.1D. There is no guarantee that this many 802.1D Bridges can be configured. For example, the CPE might not be able simultaneously to support both 802.1D and 802.1Q Bridges.	-	1.4
MaxQBridgeEntries	unsignedInt	-	The maximum number of 802.1Q [14] entries available in the Bridge table. A positive value for this parameter implies support for 802.1Q. There is no guarantee that this many 802.1Q Bridges can be configured. For example, the CPE might not be able simultaneously to support both 802.1D and 802.1Q Bridges.	-	1.4
MaxVLANEntries	unsignedInt	-	The maximum number of 802.1Q [14] VLANs supported per Bridge table entry.	-	1.4
MaxFilterEntries	unsignedInt	-	The maximum number of entries available in the Filter table.	-	1.1
MaxMarkingEntries	unsignedInt	-	The maximum number of entries available in the Marking table.	-	1.1
BridgeNumberOfEntries	unsignedInt	-	Number of entries in the Bridge table.	-	1.1
FilterNumberOfEntries	unsignedInt	-	Number of entries in the Filter table.	-	1.1
MarkingNumberOfEntries	unsignedInt	-	Number of entries in the Marking table.	-	1.1
AvailableInterfaceNumberOfEntries	unsignedInt	-	Number of entries in the AvailableInterface table.	-	1.1
InternetGatewayDevice.Layer2Bridging.-Bridge.{i}.	object	W	Bridge table. Each entry in this table represents a single physical 802.1D [13] or 802.1Q [14] bridge. If the Bridge Port table is supported, it explicitly defines the Bridge's interfaces. Otherwise, they are implicitly defined via the union of the Filter-Interface / MarkingInterface parameters for all the Filter and Marking table entries that are associated with the Bridge.	-	1.1
BridgeKey	unsignedInt	-	Unique key for each Bridge table entry.	-	1.1
BridgeStandard	boolean	W	Selects the standard supported by this Bridge table entry. Enumeration of: "802.1D" ([13]) "802.1Q" ([14])	"802.1D"	1.4
BridgeEnable	boolean	W	Enables or disables this Bridge table entry.	False	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
BridgeStatus	string	-	The status of this Bridge table entry. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	"Disabled"	1.1
BridgeName	string(64)	W	Human-readable name for this Bridge table entry.	<Empty>	1.1
VLANID	unsignedInt [0:4094]	W	For an 802.1D [13] Bridge, which has no concept of VLANs, the value of this parameter MUST be 0. For an 802.1Q [14] Bridge, this is the Bridge's default VLAN ID, i.e. the VLAN ID that applies to Filter table entries with VLANIDFilter=-1. For an 802.1Q Bridge, the value of this parameter MUST NOT be 0.	-	1.1
PortNumberOfEntries	unsignedInt	-	Number of entries in the Bridge Port table.	0	1.4
VLANNumberOfEntries	unsignedInt	-	Number of entries in the Bridge VLAN table.	0	1.4
InternetGatewayDevice.Layer2Bridging.-Bridge.{i}.Port.{i}.	object	W	Bridge Port table. If this table is supported, it MUST contain an entry for each Bridge Port.	-	1.4
PortEnable	boolean	W	Enables or disables this Bridge Port table entry.	False	1.4
PortInterface	string(16)	W	The interface associated with this Bridge Port table entry. Represents a bridge port as defined in 802.1D [13] and 802.1Q [14]. To associate this Bridge Port with an interface listed in the AvailableInterface table, the Port-Interface value is set to the value of the corresponding AvailableInterfaceKey.	<Empty>	1.4
PortState	string	-	Bridge Port state as defined in 802.1D [13] and 802.1Q [14]. Enumeration of: "Disabled" "Blocking" "Listening" "Learning" "Forwarding" "Broken"	"Disabled"	1.4
PVID	int[1:4094]	W	Default Port VLAN ID as defined in 802.1Q [14]. For an 802.1D [13] Bridge, this parameter MUST be ignored.	1	1.4
AcceptableFrameTypes	string	W	Bridge Port acceptable frame types as defined in 802.1Q [14]. Enumeration of: "AdmitAll" "AdmitOnlyVLANTagged" (OPTIONAL) "AdmitOnlyPrioUntagged" (OPTIONAL) For an 802.1D [13] Bridge, the value of this parameter MUST be "AdmitAll".	"AdmitAll"	1.4
IngressFiltering	boolean	W	Enables or disables Ingress Filtering as defined in 802.1Q [14]. For an 802.1D [13] Bridge, the value of this parameter MUST be False.	False	1.4
InternetGatewayDevice.Layer2Bridging.-Bridge.{i}.VLAN.{i}.	object	W	Bridge VLAN table. If this table is supported, if MUST contain an entry for each VLAN known to the Bridge. This table only applies to an 802.1Q [14] Bridge.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
VLANEnable	boolean	W	Enables or disables this VLAN table entry.	False	1.4
VLANName	string(64)	W	Human-readable name for this VLAN table entry.	<Empty>	1.4
VLANID	int[1:4094]	W	VLAN ID of the entry,	-	1.4
InternetGatewayDevice.Layer2Bridging.Filter.-{i}.	object	W	Filter table containing filter entries each of which is associated with one Bridge as specified by a Bridge table entry. For both 802.1D [13] and 802.1Q [14] Bridges, this table is used for the following: 1. If the Bridge Port table is not supported, it implicitly specifies the Bridge interfaces (in collaboration with the Marking table). 2. It specifies destination MAC address classification rules. For an 802.1Q Bridge, this table is also used for the following: 1. For each VLAN ID, it specifies the interfaces that are in the VLAN's Member Set. 2. If the Bridge Port table is not supported, it specifies the Port VLAN ID (PVID) for each interface. This table also supports several concepts that are not covered by either 802.1D or 802.1Q: 1. It allows a given packet to be admitted to multiple Bridges. 2. It supports Ethertype and source MAC address classification rules.	-	1.1
FilterKey	unsignedInt	-	Unique key for each Filter table entry.	-	1.1
FilterEnable	boolean	W	Enables or disables this Filter table entry.	False	1.1
FilterStatus	string	-	The status of this Filter table entry. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition. If the Bridge Port table is supported, but none of its entries correspond to FilterInterface, or if such an entry exists but is disabled, FilterStatus MUST NOT indicate Enabled. If the Bridge VLAN table is supported, but none of its entries correspond to VLANIDFilter, or if such an entry exists but is disabled, FilterStatus MUST NOT indicate Enabled.	"Disabled"	1.1
FilterBridgeReference	int[-1:]	W	The BridgeKey value of the Bridge table entry associated with this Filter. A value of -1 indicates the Filter table entry is not associated with a Bridge (and has no effect).	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ExclusivityOrder	unsignedInt	W	Whether or not the Filter definition is exclusive of all others. And if the entry is exclusive, order of precedence. A value of 1 or greater indicates an Exclusive Filter, where the value 1 indicates the first entry to be considered (highest precedence). A value of 0 indicates a Non-Exclusive Filter (OPTIONAL). For each packet, if the packet matches any Exclusive Filters, the packet is assigned to the Bridge associated with the highest precedence Exclusive Filter to which it matches (lowest ExclusivityOrder value). If and only if the packet does not match any Exclusive Filters, the packet is assigned to all Bridges associated with each Non-Exclusive Filter for which it matches the defining criteria. If a packet matches no Filter, it is discarded. When the ExclusivityOrder is set to match that of an existing Exclusive Filter (1 or greater), the value for the existing entry and all higher numbered entries is incremented (lowered in precedence) to ensure uniqueness of this value. A deletion or change in ExclusivityOrder of an Exclusive Filter causes ExclusivityOrder values of other Exclusive Filters (values 1 or greater) to be compacted. Note that the use of Exclusive Filters to associate a layer 3 router interface with LAN and/or WAN interfaces via a Bridge entry overrides and updates the association between layer 3 and layer 2 objects implied by the InternetGatewayDevice object hierarchy. Support for Non-Exclusive Filter entries, i.e. entries with an ExclusivityOrder value of 0, is OPTIONAL because 802.1D [13] and 802.1Q [14] do not consider the case of a packet potentially being admitted to more than one bridge.	-	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
FilterInterface	string(16)	W	The interface or interfaces associated with this Filter table entry. The bridge corresponding to this Filter table entry is defined to admit packets on ingress to the bridge from the specified interfaces that meet all of the criteria specified in the Filter table entry. The following values are defined. To associate this Filter with a single interface listed in the AvailableInterface table, the FilterInterface value is set to the value of the corresponding AvailableInterfaceKey. "AllInterfaces" indicates that this Filter is associated with all LAN and WAN interfaces listed in the AvailableInterface table (all entries of InterfaceType LANInterface or WANInterface). This value is DEPRECATED because of the configuration complexity it requires. "LANInterfaces" indicates that this Filter is associated with all LAN interfaces listed in the AvailableInterface table (all entries of InterfaceType LANInterface). This value is DEPRECATED because of the configuration complexity it requires. "WANInterfaces" indicates that this Filter is associated with all WAN interfaces listed in the AvailableInterface table (all entries of InterfaceType WANInterface). This value is DEPRECATED because of the configuration complexity it requires. An empty string indicates the Filter table entry is not associated with any interface (and has no effect)	<Empty>	1.1
VLANIDFilter	int[-1:4094]	W	The 802.1Q [14] VLAN ID associated with this Filter table entry. A value of -1 indicates that the default VLAN ID for the Bridge MUST be used instead (as specified by InternetGatewayDevice.Layer2Bridging.Bridge.{i}.-VLANID for the Bridge table entry associated with this Filter table entry). For an 802.1Q bridge, the value of this parameter MUST NOT be 0, and it is interpreted as follows (more than one condition can apply, e.g. a single Filter table entry might both add an interface to a VLAN's Member Set, and define a classification rule). 1. If the Bridge Port table is not supported, it is a candidate to be the 802.1Q Port VLAN ID (PVID) for the interface associated with this Filter. Where there is more than one such candidate for a given interface, the PVID MUST be selected according to the ExclusivityOrder precedence rules. 2. The interface associated with this Filter is added to the VLAN's 802.1Q Member Set. 3. If a classification parameter, e.g. DestMACAddr-essFilterList, is specified, a classification rule for this VLAN ID is added. For an 802.1D [13] Bridge, which has no concept of VLANs, the VLAN ID MUST be 0. This is most easily achieved by allowing this parameter to retain its default value of -1 and relying on the fact that the default VLAN ID for an 802.1D bridge will always be 0.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
AdmitOnlyVLANTagged	boolean	W	802.1Q [14] Acceptable Frame Types criterion. This parameter is DEPRECATED, because it only partly models 802.1Q Acceptable Frame Types (and Ingress Filtering). The Bridge Port table SHOULD be used instead and, if supported, MUST take precedence over this parameter. If True, the Bridge admits only packets tagged with VLAN IDs that include the ingress interface in their 802.1Q Member Sets. If False, the Bridge admits both packets tagged with VLAN IDs that include the ingress interface in their 802.1Q Member Sets, and any Untagged or PriorityOnly packets. All Untagged or PriorityOnly packets are associated on ingress with the interface's Port VLAN ID (PVID). See the description of VLANIDFilter for an explanation of how the Member Set and PVID are determined. If more than one Filter table entry is associated with a given interface, the value of AdmitOnlyVLANTagged MUST be the same for all such entries. For an 802.1D [13] Bridge, which has no concept of VLANs, the value of this parameter MUST be False.	False	1.1
EthertypeFilterList	string(256)	W	Classification criterion. Comma-separated list of unsigned integers, each representing an Ethertype value. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on Ethertype.	<Empty>	1.1
EthertypeFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge is defined to admit only those packets that match one of the EthertypeFilterList entries (in either the Ethernet or SNAP Type header). If the EthertypeFilterList is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge is defined to admit all packets except those packets that match one of the EthertypeFilterList entries (in either the Ethernet or SNAP Type header). If the EthertypeFilterList is empty, packets are admitted regardless of Ethertype. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on Ethertype.	True	1.1
SourceMACAddressFilterList	string(512)	W	Classification criterion. Comma-separated list of MAC Addresses. Each list entry MAY optionally specify a bit-mask, where matching of a packet's MAC address is only to be done for bit positions set to one in the mask. If no mask is specified, all bits of the MAC Address are to be used for matching. For example, the list might be: "01:02:03:04:05:06, 1:22:33:00:00:00/FF:FF:FF:00:00:00, 88:77:66:55:44:33" Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	<Empty>	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SourceMACAddressFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge admits only those packets whose source MAC Address matches one of the SourceMACAddressFilterList entries. If the SourceMACAddressFilterList is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge admits all packets except those packets whose source MAC Address matches one of the SourceMACAddressFilterList entries. If the SourceMACAddressFilterList is empty, packets are admitted regardless of MAC address. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	True	1.1
DestMACAddressFilterList	string(512)	W	Classification criterion. Comma-separated list of MAC Addresses. Each list entry MAY optionally specify a bit-mask, where matching of a packet's MAC address is only to be done for bit positions set to one in the mask. If no mask is specified, all bits of the MAC Address are to be used for matching. For example, the list might be: "01:02:03:04:05:06, 1:22:33:00:00:00/FF:FF:FF:00:00:00, 88:77:66:55:44:33"	<Empty>	1.1
DestMACAddressFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge admits only those packets whose destination MAC Address matches one of the DestMACAddressFilterList entries. If the DestMACAddressFilterList is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge admits all packets except those packets whose destination MAC Address matches one of the DestMACAddressFilterList entries. If the WANSourceMACAddressFilterList is empty, packets are admitted regardless of MAC address.	True	1.1
SourceMACFromVendorClassIDFilter	string(256)	W	Classification criterion. A string used to identify one or more devices via DHCP for which MAC address filtering would subsequently apply. A device is considered matching if its DHCP Vendor Class Identifier (Option 60 as defined in RFC 2132 [28]) in the most recent DHCP lease acquisition or renewal matches the specified value according to the match criterion in SourceMACFromVendorClassIDMode. Case sensitive. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	<Empty>	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SourceMACFromVendorClassIDFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge admits only those packets whose source MAC Address matches that of a LAN device previously identified as described in SourceMACFromVendorClassIDFilter. If SourceMACFromVendorClassIDFilter is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge admits all packets except those packets whose source MAC Address matches that of a LAN device previously identified as described in SourceMACFromVendorClassID-Filter. If the SourceMACFromVendorClassIDFilter is empty, packets are admitted regardless of MAC address. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	True	1.1
SourceMACFromVendorClassIDMode	string	W	SourceMACFromVendorClassIDFilter pattern match criterion. Enumeration of: "Exact" "Prefix" "Suffix" "Substring" For example, if SourceMACFromVendorClassID-Filter is "Example" then an Option 60 value of "Example device" will match with SourceMACFrom-VendorClassIDMode values of "Prefix" or "Substring", but not with "Exact" or "Suffix". Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	"Exact"	1.4
DestMACFromVendorClassIDFilter	string(256)	W	Classification criterion. A string used to identify one or more devices via DHCP for which MAC address filtering would subsequently apply. A device is considered matching if its DHCP Vendor Class Identifier (Option 60 as defined in RFC 2132 [28]) in the most recent DHCP lease acquisition or renewal matches the specified value according to the match criterion in DestMACFromVendorClassIDMode. Case sensitive.	<Empty>	1.1
DestMACFromVendorClassIDFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge admits only those packets whose destination MAC Address matches that of a LAN device previously identified as described in DestMACFromVendorClassIDFilter. If DestMACFromVendorClassIDFilter is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge admits all packets except those packets whose destination MAC Address matches that of a LAN device previously identified as described in DestMACFromVendorClassID-Filter. If the DestMACFromVendorClassIDFilter is empty, packets are admitted regardless of MAC address.	True	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestMACFromVendorClassIDMode	string	W	DestMACFromVendorClassIDFilter pattern match criterion. Enumeration of: "Exact" "Prefix" "Suffix" "Substring" For example, if DestMACFromVendorClassIDFilter is "Example" then an Option 60 value of "Example device" will match with DestMACFromVendor-ClassIDMode values of "Prefix" or "Substring", but not with "Exact" or "Suffix".	"Exact"	1.4
SourceMACFromClientIDFilter	string(256)	W	Classification criterion. A string used to identify one or more devices via DHCP for which MAC address filtering would subsequently apply. A device is considered matching if its DHCP Client Identifier (Option 61 as defined in RFC 2132 [28]) in the most recent DHCP lease acquisition or renewal was equal to the specified value. The option value is binary, so an exact match is REQUIRED. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	<Empty>	1.1
SourceMACFromClientIDFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge admits only those packets whose source MAC Address matches that of a LAN device previously identified as described in SourceMACFromClientIDFilter. If SourceMAC-FromClientIDFilter is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge admits all packets except those packets whose source MAC Address matches that of a LAN device previously identified as described in SourceMACFromClientIDFilter. If the SourceMACFromClientIDFilter is empty, packets are admitted regardless of MAC address. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	True	1.1
DestMACFromClientIDFilter	string(256)	W	Classification criterion. A string used to identify one or more devices via DHCP for which MAC address filtering would subsequently apply. A device is considered matching if its DHCP Client Identifier (Option 61 as defined in RFC 2132 [28]) in the most recent DHCP lease acquisition or renewal was equal to the specified value. The option value is binary, so an exact match is REQUIRED.	<Empty>	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestMACFromClientIDFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge admits only those packets whose destination MAC Address matches that of a LAN device previously identified as described in DestMACFromClientIDFilter. If Dest-MACFromClientIDFilter is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge admits all packets except those packets whose destination MAC Address matches that of a LAN device previously identified as described in DestMACFromClientIDFilter. If the DestMACFromClientIDFilter is empty, packets are admitted regardless of MAC address.	True	1.1
SourceMACFromUserClassIDFilter	string(256)	W	Classification criterion. A string used to identify one or more devices via DHCP for which MAC address filtering would subsequently apply. A device is considered matching if its DHCP User Class Identifier (Option 77 as defined in RFC 3004 [39]) in the most recent DHCP lease acquisition or renewal was equal to the specified value. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	<Empty>	1.1
SourceMACFromUserClassIDFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge admits only those packets whose source MAC Address matches that of a LAN device previously identified as described in SourceMACFromUserClassIDFilter. If Source-MACFromUserClassIDFilter is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge admits all packets except those packets whose source MAC Address matches that of a LAN device previously identified as described in SourceMACFromUserClassID-Filter. If the SourceMACFromUserClassIDFilter is empty, packets are admitted regardless of MAC address. Note that neither 802.1D [13] nor 802.1Q [14] support classification based on source MAC address.	True	1.1
DestMACFromUserClassIDFilter	string(256)	W	Classification criterion. A string used to identify one or more devices via DHCP for which MAC address filtering would subsequently apply. A device is considered matching if its DHCP User Class Identifier (Option 77 as defined in RFC 3004 [39]) in the most recent DHCP lease acquisition or renewal was equal to the specified value.	<Empty>	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestMACFromUserClassIDFilterExclude	boolean	W	If False, on ingress to the interfaces associated with this Filter, the Bridge admits only those packets whose destination MAC Address matches that of a LAN device previously identified as described in DestMACFromUserClassIDFilter. If DestMACFromUserClassIDFilter is empty, no packets are admitted. If True, on ingress to the interfaces associated with this Filter, the Bridge admits all packets except those packets whose destination MAC Address matches that of a LAN device previously identified as described in DestMACFromUserClassIDFilter. If the DestMACFromUserClassIDFilter is empty, packets are admitted regardless of MAC address.	True	1.1
InternetGatewayDevice.Layer2Bridging.-Marking.{i}.	object	W	Marking table identifying non-default layer 2 marking behavior for packets on egress from the specified interfaces. This table is not relevant to 802.1D [13] Bridges, which are not VLAN-aware. For 802.1Q [14] Bridges, this table is used for the following: 1. It specifies whether VLAN tags are to be removed on egress. This table also supports several concepts that are not covered by 802.1Q: 1. It allows the VLAN ID to be changed on egress. 2. It allows the Ethernet Priority to be changed on egress.	-	1.1
MarkingKey	unsignedInt	-	Unique key for each Marking table entry.	-	1.1
MarkingEnable	boolean	W	Enables or disables this Marking table entry.	False	1.1
MarkingStatus	string	-	The status of this Marking table entry. Enumeration of: “Disabled” “Enabled” “Error” (OPTIONAL) The “Error” value MAY be used by the CPE to indicate a locally defined error condition. If the Bridge Port table is supported, but none of its entries correspond to MarkingInterface, or if such an entry exists but is disabled, MarkingStatus MUST NOT indicate Enabled.	“Disabled”	1.1
MarkingBridgeReference	int[-1:]	W	The BridgeKey value of the Bridge table entry associated with this Marking table entry. A value of -1 indicates the Marking table entry is not associated with a Bridge (and has no effect). The effect of a Marking table entry applies only to packets that have been admitted to the specified bridge (regardless of the ingress interface).	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
MarkingInterface	string(16)	W	The interface or interfaces associated with this Marking table entry for which the specified marking behavior is to apply on egress from the associated bridge. The following values are defined. To associate this Marking table entry with a single interface listed in the AvailableInterface table, the MarkingInterface value is set to the value of the corresponding AvailableInterfaceKey. "AllInterfaces" indicates that this Marking table entry is associated with all LAN and WAN interfaces listed in the AvailableInterface table (all entries of InterfaceType LANInterface or WANInterface). This value is DEPRECATED because of the configuration complexity it requires. "LANInterfaces" indicates that this Marking table entry is associated with all LAN interfaces listed in the AvailableInterface table (all entries of InterfaceType LANInterface). This value is DEPRECATED because of the configuration complexity it requires. "WANInterfaces" indicates that this Marking table entry is associated with all WAN interfaces listed in the AvailableInterface table (all entries of InterfaceType WANInterface). This value is DEPRECATED because of the configuration complexity it requires. An empty string indicates the Marking table entry table entry is not associated with any interface (and has no effect) If there is more than one enabled Marking table entry that specifies one or more of the same interfaces for the same bridge (identical values of MarkingBridgeReference), then for packets on egress from the specified bridge to those interfaces, the applied marking MUST be that specified in the Marking table entry among those in conflict with the lowest MarkingKey value. If an interface in a given bridge does not have a corresponding Marking table entry, the marking is left unchanged on egress.	<Empty>	1.1
VLANIDUntag	boolean	W	If True, on egress to the interfaces associated with this Marking table entry, all packets are sent Untagged. If False, on egress to the interfaces associated with this Marking table entry, all packets are sent Tagged with the VLAN ID of the VLAN in which the packet is being bridged.	False	1.1
VLANIDMark	int[-1:4094]	W	The 802.1Q [14] VLAN ID to be used on egress to the interfaces associated with this Marking table entry. A value of -1 indicates that the VLAN ID of the VLAN in which the packet is being bridged is to be used, i.e. no change. The value of this parameter MUST NOT be 0. If VLANIDUntag is True, then no VLAN marking is done since the tag containing the VLAN ID is removed. Note that 802.1Q does not support re-marking on egress.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
VLANIDMarkOverride	boolean	W	If False, on egress to the interfaces associated with this Marking table entry, the VLANIDMark, if specified, is applied only to PriorityOnly packets. If True, on egress to the interfaces associated with this Marking table entry, the VLANIDMark, if specified, is to be applied to all packets on this Bridge. If VLANIDUntag is True, then no VLAN marking is done since the tag containing the VLAN ID is removed. Note that 802.1Q [14] does not support re-marking on egress.	False	1.4
EthernetPriorityMark	int[-1:7]	W	Ethernet priority code (as defined in 802.1D [13]) to mark traffic with that falls into this Bridge on egress to the interfaces associated with this Marking table entry. A value of -1 indicates no change from the incoming packet or the mark assigned by the classifier. Note that 802.1Q [14] does not support re-marking on egress.	-1	1.1
EthernetPriorityOverride	boolean	W	If False, on egress to the interfaces associated with this Marking table entry, the EthernetPriorityMark, if specified, is applied only to packets of priority 0. If True, on egress to the interfaces associated with this Marking table entry, the EthernetPriorityMark, if specified, is to be applied to all packets on this Bridge. Note that 802.1Q [14] does not support re-marking on egress.	False	1.1
InternetGatewayDevice.Layer2-Bridging.AvailableInterface.{i}.	object	-	Table containing all LAN and WAN interfaces that are available to be referenced by the Bridge table. Only interfaces that can carry layer 2 bridged traffic are included.	-	1.1
AvailableInterfaceKey	unsignedInt	-	Unique key for each Interface entry.	-	1.1
InterfaceType	string	-	Whether the interface is a LAN-side or WAN-side interface, or a LAN-side or WAN-side connection to the Gateway's IP router. Enumeration of: "LANInterface" "WANInterface" "LANRouterConnection" "WANRouterConnection"	-	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InterfaceReference	string(256)	-	This table SHOULD contain a single entry for each *available* LAN and WAN interface. When such an interface is modeled in more than one place within the data model, the value of this parameter MUST be a comma-separated list of the full hierarchical parameter names of all of the corresponding objects. For example, if a given Ethernet interface is present within two LANDevice instances, the value of this parameter might be: "InternetGatewayDevice.LANDevice.1.-LANEthernetInterfaceConfig.1,Internet-GatewayDevice.LANDevice.2.LANEthernet-InterfaceConfig.1" Note that the remainder of the parameter description does not refer to the possibility that the parameter value is a comma-separated list. Nevertheless, the above requirement does apply. For a WAN interface, this parameter is the full hierarchical parameter name of a particular WAN-ConnectionDevice. A WANConnectionDevice is considered available (included in this table) *only* if it supports layer 2 bridged traffic. That is, this table MUST include only WANConnectionDevices that contain either a WANEthernetLinkConfig object, or that contain a WANDSLLinkConfig object for which the LinkType is "EoA". For example: "InternetGatewayDevice.WANDevice.1.WAN-ConnectionDevice.2" For a LAN interface, this parameter is the full hierarchical parameter name of a particular LAN**-InterfaceConfig object, or a WLANConfiguration object. This table SHOULD include one entry for each such object. For example: "InternetGatewayDevice.LANDevice.1.LAN-EthernetInterfaceConfig.2" For a WAN-side connection to the Gateway's IP router, this parameter is the full hierarchical parameter name of a particular WAN**Connection service. This table SHOULD include an entry for each layer 3 WAN connection. For example: "InternetGatewayDevice.WANDevice.1.WAN-ConnectionDevice.2.WANPPPConnection.1" For a LAN-side connection to the Gateway's IP router, this parameter is the full hierarchical parameter name of a particular LANDevice. This table SHOULD include an entry for each LANDevice, each of which is associated with a LAN-side layer 3 connection to the Gateway's IP router. For example: "InternetGatewayDevice.LANDevice.2"	-	1.1
InternetGatewayDevice.QueueManagement.	object	-	Queue management configuration object.	-	1.1
Enable	boolean	W	Enables or disables all queuing operation.	-	1.1
MaxQueues	unsignedInt	-	The maximum number of queues supported by the CPE. Calculated as the sum of the number of different queues pointed to by Classification table. For each entry in the Classification table, the count includes a queue for each egress interface to which the corresponding classified traffic could reach.	-	1.1
MaxClassificationEntries	unsignedInt	-	The maximum number of entries available in the Classification table.	-	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ClassificationNumberOfEntries	unsignedInt	-	The number of entries in the Classification table.	-	1.1
MaxAppEntries	unsignedInt	-	The maximum number of entries available in the App table.	-	1.1
AppNumberOfEntries	unsignedInt	-	The number of entries in the App table.	-	1.1
MaxFlowEntries	unsignedInt	-	The maximum number of entries available in the Flow table.	-	1.1
FlowNumberOfEntries	unsignedInt	-	The number of entries in the Flow table.	-	1.1
MaxPolicerEntries	unsignedInt	-	The maximum number of entries available in the Policer table.	-	1.1
PolicerNumberOfEntries	unsignedInt	-	The number of entries in the Policer table.	-	1.1
MaxQueueEntries	unsignedInt	-	The maximum number of entries available in the Queue table.	-	1.1
QueueNumberOfEntries	unsignedInt	-	The number of entries in the Queue table.	-	1.1
QueueStatsNumberOfEntries	unsignedInt	-	The number of entries in the QueueStats table.	-	1.4
DefaultForwardingPolicy	unsignedInt	W	Identifier of the forwarding policy associated with traffic not associated with any specified classifier.	-	1.1
DefaultTrafficClass	int[-1:]	W	Identifier of the traffic class associated with traffic not associated with any specified classifier. A value of -1 indicates a null traffic class.	-	1.4
DefaultPolicer	int[-1:]	W	Instance number of the Policer table entry for traffic not associated with any specified classifier. A value of -1 indicates a null policer.	-	1.1
DefaultQueue	unsignedInt	W	Instance number of the Queue table entry for traffic not associated with any specified classifier. A value of 0xffffffff (-1) indicates a null queue (permitted in data model versions 1.4 and later).	-	1.1
DefaultDSCPMark	int[-2:]	W	DSCP to mark traffic not associated with any specified classifier. A value of -1 indicates no change from the incoming packet. A value of -2 indicates automatic marking of DSCP based upon the EthernetPriority value of the incoming packet as defined in Annex A.	-	1.1
DefaultEthernetPriorityMark	int[-2:]	W	Ethernet priority code (as defined in 802.1D) to mark traffic not associated with any specified classifier. A value of -1 indicates no change from the incoming packet. A value of -2 indicates automatic marking of EthernetPriority based upon the DSCP value of the incoming packet as defined in Annex A.	-	1.1
AvailableAppList	string(1024)	-	Comma-separated list of URNs, each indicating a protocol supported for use as a ProtocolIdentifier in the App table. This list MAY include any of the URNs defined in Annex A as well as other URNs defined elsewhere.	-	1.1
InternetGatewayDevice.QueueManagement.-Classification.{i}.	object	W	Classification table.	-	1.1
ClassificationKey	unsignedInt	-	Unique key for each classification entry. This parameter is OBSOLETED because it serves no purpose (no other parameter references it).	-	1.1
ClassificationEnable	boolean	W	Enables or disables this classifier.	False	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ClassificationStatus	string	-	The status of this classifier. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	"Disabled"	1.1
ClassificationOrder	unsignedInt [1:]	W	Position of the classification entry in the order of precedence. A value of 1 indicates the first entry considered. For each packet, the highest ordered entry that matches the classification criteria is applied. All lower order entries are ignored. When this value is modified, if the value matches that of an existing entry, the Order value for the existing entry and all lower Order entries is incremented (lowered in precedence) to ensure uniqueness of this value. A deletion causes Order values to be compacted. When a value is changed, incrementing occurs before compaction. The value on creation of a Classification table entry MUST be one greater than the largest current value.	-	1.1
ClassInterface	string(256)	W	Classification criterion. Specifies the LAN or WAN ingress interface associated with this entry. The content is the full hierarchical parameter name of the particular WANDevice, WANConnectionDevice, WAN**-Connection, LANDevice, LAN**InterfaceConfig, or WLANConfiguration object. The following are WAN interface examples: "InternetGatewayDevice.WANDevice.2" "InternetGatewayDevice.WANDevice.1.WAN-ConnectionDevice.2.WANPPPConnection.1" The following are LAN interface examples: "InternetGatewayDevice.LANDevice.3" "InternetGatewayDevice.LANDevice.1.LAN-EthernetInterfaceConfig.2" "InternetGatewayDevice.LANDevice.1.WLAN-Configuration.3" The string "WAN" indicates this entry is to apply to traffic entering from any WAN interface. The string "LAN" indicates this entry is to apply to traffic entering from any LAN interface. The string "Local" indicates this entry is to apply to IP-layer traffic entering from a local source within the Internet Gateway Device. An empty value indicates this classification entry is to apply to all sources.	<Empty>	1.1
DestIP	string	W	Classification criterion. Destination IP address. An empty value indicates this criterion is not used for classification.	<Empty>	1.1
DestMask	string	W	Destination IP address mask. If non-empty, only the indicated network portion of the DestIP address is to be used for classification. An empty value indicates that the full DestIP address is to be used for classification.	<Empty>	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestIPExclude	boolean	W	If False, the class includes only those packets that match the (masked) DestIP entry, if specified. If True, the class includes all packets except those that match the (masked) DestIP entry, if specified.	False	1.1
SourceIP	string	W	Classification criterion. Source IP address. An empty value indicates this criterion is not used for classification.	<Empty>	1.1
SourceMask	string	W	Source IP address mask. If non-empty, only the indicated network portion of the SourceIP address is to be used for classification. An empty value indicates that the full SourceIP address is to be used for classification.	<Empty>	1.1
SourceIPExclude	boolean	W	If False, the class includes only those packets that match the (masked) SourceIP entry, if specified. If True, the class includes all packets except those that match the (masked) SourceIP entry, if specified.	False	1.1
Protocol	int[-1:]	W	Classification criterion. Protocol number. A value of -1 indicates this criterion is not used for classification.	-1	1.1
ProtocolExclude	boolean	W	If False, the class includes only those packets that match the Protocol entry, if specified. If True, the class includes all packets except those that match the Protocol entry, if specified.	False	1.1
DestPort	int[-1:]	W	Classification criterion. Destination port number. A value of -1 indicates this criterion is not used for classification.	-1	1.1
DestPortRangeMax	int[-1:]	W	Classification criterion. If specified, indicates the classification criterion is to include the port range from DestPort through DestPortRangeMax (inclusive). If specified, DestPortRangeMax MUST be greater than or equal to DestPort. A value of -1 indicates that no port range is specified.	-1	1.1
DestPortExclude	boolean	W	If False, the class includes only those packets that match the DestPort entry (or port range), if specified. If True, the class includes all packets except those that match the DestPort entry (or port range), if specified.	False	1.1
SourcePort	int[-1:]	W	Classification criterion. Source port number. A value of -1 indicates this criterion is not used for classification.	-1	1.1
SourcePortRangeMax	int[-1:]	W	Classification criterion. If specified, indicates the classification criterion is to include the port range from SourcePort through SourcePortRangeMax (inclusive). If specified, SourcePortRangeMax MUST be greater than or equal to SourcePort. A value of -1 indicates that no port range is specified.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SourcePortExclude	boolean	W	If False, the class includes only those packets that match the SourcePort entry (or port range), if specified. If True, the class includes all packets except those that match the SourcePort entry (or port range), if specified.	False	1.1
SourceMACAddress	string	W	Classification criterion. Source MAC Address. An empty value indicates this criterion is not used for classification.	<Empty>	1.1
SourceMACMask	string	W	Bit-mask for the MAC address, where matching of a packet's MAC address with the SourceMAC-Address is only to be done for bit positions set to one in the mask. A mask of FF:FF:FF:FF:FF:FF or an empty string indicates all bits of the Source-MACAdress are to be used for classification.	<Empty>	1.1
SourceMACExclude	boolean	W	If False, the class includes only those packets that match the (masked) SourceMACAddress entry, if specified. If True, the class includes all packets except those that match the (masked) SourceMACAddress entry, if specified.	False	1.1
DestMACAddress	string	W	Classification criterion. Destination MAC Address. An empty value indicates this criterion is not used for classification. The use of destination MAC address as a classification criterion is primarily useful only for bridged traffic.	<Empty>	1.1
DestMACMask	string	W	Bit-mask for the MAC address, where matching of a packet's MAC address with the DestMACAddress is only to be done for bit positions set to one in the mask. A mask of FF:FF:FF:FF:FF:FF or an empty string indicates all bits of the DestMACAdress are to be used for classification.	<Empty>	1.1
DestMACExclude	boolean	W	If False, the class includes only those packets that match the (masked) DestMACAddress entry, if specified. If True, the class includes all packets except those that match the (masked) DestMACAddress entry, if specified.	False	1.1
Ethertype	int[-1:]	W	Classification criterion. Ethertype as indicated in either the Ethernet or SNAP Type header. A value of -1 indicates this criterion is not used for classification.	-1	1.1
EthertypeExclude	boolean	W	If False, the class includes only those packets that match the Ethertype entry, if specified. If True, the class includes all packets except those that match the Ethertype entry, if specified.	False	1.1
SSAP	int[-1:]	W	Classification criterion. SSAP element in the LLC header. A value of -1 indicates this criterion is not used for classification.	-1	1.1
SSAPExclude	boolean	W	If False, the class includes only those packets that match the SSAP entry, if specified. If True, the class includes all packets except those that match the SSAP entry, if specified.	False	1.1
DSAP	int[-1:]	W	Classification criterion. DSAP element in the LLC header. A value of -1 indicates this criterion is not used for classification.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DSAPExclude	boolean	W	If False, the class includes only those packets that match the DSAP entry, if specified. If True, the class includes all packets except those that match the DSAP entry, if specified.	False	1.1
LLCControl	int[-1:]	W	Classification criterion. Control element in the LLC header. A value of -1 indicates this criterion is not used for classification.	-1	1.1
LLCControlExclude	boolean	W	If False, the class includes only those packets that match the LLCControl entry, if specified. If True, the class includes all packets except those that match the LLCControl entry, if specified.	False	1.1
SNAPOUI	int[-1:]	W	Classification criterion. OUI element in the SNAP header. A value of -1 indicates this criterion is not used for classification.	-1	1.1
SNAPOUIExclude	boolean	W	If False, the class includes only those packets that match the SNAPOUI entry, if specified. If True, the class includes all packets except those that match the SNAPOUI entry, if specified.	False	1.1
SourceVendorClassID	string(256)	W	Classification criterion. Used to identify one or more LAN devices, value of the DHCP Vendor Class Identifier (Option 60) as defined in RFC 2132 [28], matched according to the criterion in SourceVendorClassIDMode. Case sensitive. An empty value indicates this criterion is not used for classification.	<Empty>	1.1
SourceVendorClassIDExclude	boolean	W	If False, the class includes only those packets sourced from LAN devices that match the SourceVendorClassID entry, if specified. If True, the class includes all packets except those sourced from LAN devices that match the SourceVendorClassID entry, if specified.	False	1.1
SourceVendorClassIDMode	string	W	SourceVendorClassID pattern match criterion. Enumeration of: “Exact” “Prefix” “Suffix” “Substring” For example, if SourceVendorClassID is “Example” then an Option 60 value of “Example device” will match with SourceVendorClassID values of “Prefix” or “Substring”, but not with “Exact” or “Suffix”.	“Exact”	1.4
DestVendorClassID	string(256)	W	Classification criterion. Used to identify one or more LAN devices, value of the DHCP Vendor Class Identifier (Option 60) as defined in RFC 2132 [28], matched according to the criterion in DestVendorClassIDMode. Case sensitive. An empty value indicates this criterion is not used for classification.	<Empty>	1.1
DestVendorClassIDExclude	boolean	W	If False, the class includes only those packets destined for LAN devices that match the DestVendorClassID entry, if specified. If True, the class includes all packets except those destined for LAN devices that match the DestVendorClassID entry, if specified.	False	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestVendorClassIDMode	string	W	DestVendorClassID pattern match criterion. Enumeration of: "Exact" "Prefix" "Suffix" "Substring" For example, if DestVendorClassID is "Example" then an Option 60 value of "Example device" will match with DestVendorClassID values of "Prefix" or "Substring", but not with "Exact" or "Suffix".	"Exact"	1.4
SourceClientID	string(256)	W	Classification criterion. Used to identify one or more LAN devices, value of the DHCP Client Identifier (Option 61) as defined in RFC 2132 [28]. The option value is binary, so an exact match is REQUIRED. An empty value indicates this criterion is not used for classification.	<Empty>	1.1
SourceClientIDExclude	boolean	W	If False, the class includes only those packets sourced from LAN devices that match the SourceClientID entry, if specified. If True, the class includes all packets except those sourced from LAN devices that match the SourceClientID entry, if specified.	False	1.1
DestClientID	string(256)	W	Classification criterion. Used to identify one or more LAN devices, value of the DHCP Client Identifier (Option 61) as defined in RFC 2132 [28]. The option value is binary, so an exact match is REQUIRED. An empty value indicates this criterion is not used for classification.	<Empty>	1.1
DestClientIDExclude	boolean	W	If False, the class includes only those packets destined for LAN devices that match the DestClientID entry, if specified. If True, the class includes all packets except those destined for LAN devices that match the DestClientID entry, if specified.	False	1.1
SourceUserClassID	string(256)	W	Classification criterion. Used to identify one or more LAN devices, value of the DHCP User Class Identifier (Option 77) as defined in RFC 3004 [39]. An empty value indicates this criterion is not used for classification.	<Empty>	1.1
SourceUserClassIDExclude	boolean	W	If False, the class includes only those packets sourced from LAN devices that match the SourceUserClassID entry, if specified. If True, the class includes all packets except those sourced from LAN devices that match the SourceUserClassID entry, if specified.	False	1.1
DestUserClassID	string(256)	W	Classification criterion. Used to identify one or more LAN devices, value of the DHCP User Class Identifier (Option 77) as defined in RFC 3004 [39]. An empty value indicates this criterion is not used for classification.	<Empty>	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestUserClassIDExclude	boolean	W	If False, the class includes only those packets destined for LAN devices that match the DestUserClassID entry, if specified. If True, the class includes all packets except those destined for LAN devices that match the DestUserClassID entry, if specified.	False	1.1
SourceVendorSpecificInfo	string(256)	W	Classification criterion. Used to identify one or more LAN devices, value of the DHCP Vendor-specific Information (Option 125) as defined in RFC 3925 [47], matched according to the criteria in SourceVendorSpecificInfoEnterprise, SourceVendorSpecificInfoSubOption and SourceVendorSpecificInfoMode. Case sensitive. An empty value indicates this criterion is not used for classification.	<Empty>	1.4
SourceVendorSpecificInfoExclude	boolean	W	If False, the class includes only those packets sourced from LAN devices that match the SourceVendorSpecificInfo entry, if specified. If True, the class includes all packets except those sourced from LAN devices that match the SourceVendorSpecificInfo entry, if specified.	False	1.4
SourceVendorSpecificInfoEnterprise	unsignedInt	W	SourceVendorSpecificInfo Enterprise Number as defined in RFC 3925 [47]. The default value (0) is assigned to IANA and will probably need to be replaced with an appropriate enterprise number.	0	1.4
SourceVendorSpecificInfoSubOption	int[0:255]	W	SourceVendorSpecificInfo Sub Option Code as defined in RFC 3925 [47].	0	1.4
SourceVendorSpecificInfoMode	string	W	SourceVendorSpecificInfo pattern match criterion. Enumeration of: "Exact" "Prefix" "Suffix" "Substring"	"Exact"	1.4
DestVendorSpecificInfo	string(256)	W	Classification criterion. Used to identify one or more LAN devices, value of the DHCP Vendor-specific Information (Option 125) as defined in RFC 3925 [47], matched according to the criteria in DestVendorSpecificInfoEnterprise, DestVendorSpecificInfoSubOption and DestVendorSpecificInfoMode. An empty value indicates this criterion is not used for classification.	<Empty>	1.4
DestVendorSpecificInfoExclude	boolean	W	If False, the class includes only those packets destined for LAN devices that match the DestVendorSpecificInfo entry, if specified. If True, the class includes all packets except those destined for LAN devices that match the DestVendorSpecificInfo entry, if specified.	False	1.4
DestVendorSpecificInfoEnterprise	unsignedInt	W	DestVendorSpecificInfo Enterprise Number as defined in RFC 3925 [47]. The default value (0) is assigned to IANA and will probably need to be replaced with an appropriate enterprise number.	0	1.4
DestVendorSpecificInfoSubOption	int[0:255]	W	DestVendorSpecificInfo Sub Option Code as defined in RFC 3925 [47].	0	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestVendorSpecificInfoMode	string	W	DestVendorSpecificInfo pattern match criterion. Enumeration of: "Exact" "Prefix" "Suffix" "Substring"	"Exact"	1.4
TCPACK	boolean	W	Classification criterion. If False, this criterion is not used for classification. If True, this criterion matches with all TCP segments that have the ACK control bit set.	False	1.1
TCPACKExclude	boolean	W	If False, the class includes only those packets that match the TCPACK entry, if specified. If True, the class includes all packets except those that match the TCPACK entry, if specified.	False	1.1
IPLengthMin	unsignedInt	W	Classification criterion. Minimum IP Packet Length (including header) in bytes.	0	1.1
IPLengthMax	unsignedInt	W	Classification criterion. Maximum IP Packet Length (including header) in bytes. A value of zero indicates that no maximum is specified (an umlimited maximum length).	0	1.1
IPLengthExclude	boolean	W	If False, the class includes only those packets whose length (including header) falls within the inclusive range IPLengthMin through IPLengthMax. A value of zero for both IPLengthMin and IPLengthMax allows any length packet. An equal non-zero value of IPLengthMin and IPLengthMax allows only a packet with the exact length specified. If True, the class includes all packets except those whose length (including header) falls within the inclusive range IPLengthMin through IPLengthMax.	False	1.1
DSCPCheck	int[-1:]	W	Classification criterion. DiffServ codepoint (defined in RFC 2474 [30]). If set to a Class Selector Codepoint (defined in RFC 2474), all DSCP values that match the first 3 bits will be considered a valid match. A value of -1 indicates this criterion is not used for classification.	-1	1.1
DSCPExclude	boolean	W	If False, the class includes only those packets that match the DSCPCheck entry, if specified. If True, the class includes all packets except those that match the DSCPCheck entry, if specified.	False	1.1
DSCPMark	int[-2:]	W	Classification result. DSCP to mark traffic with that falls into this classification entry. A value of -1 indicates no change from the incoming packet. A value of -2 indicates automatic marking of DSCP based upon the EthernetPriority value of the incoming packet as defined in Annex A.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
EthernetPriorityCheck	int[-1:]	W	Classification criterion. Current Ethernet priority as defined in 802.1D. A value of -1 indicates this criterion is not used for classification.	-1	1.1
EthernetPriorityExclude	boolean	W	If False, the class includes only those packets that match the EthernetPriorityCheck entry, if specified. If True, the class includes all packets except those that match the EthernetPriorityCheck entry, if specified.	False	1.1
EthernetPriorityMark	int[-2:]	W	Classification result. Ethernet priority code (as defined in 802.1D) to mark traffic with that falls into this classification entry. A value of -1 indicates no change from the incoming packet. A value of -2 indicates automatic marking of EthernetPriority based upon the DSCP value of the incoming packet as defined in Annex A.	-1	1.1
VLANIDCheck	int[-1:]	W	Classification criterion. Current Ethernet VLAN ID as defined in 802.1Q. A value of -1 indicates this criterion is not used for classification.	-1	1.1
VLANIDExclude	boolean	W	If False, the class includes only those packets that match the VLANIDCheck entry, if specified. If True, the class includes all packets except those that match the VLANIDCheck entry, if specified.	False	1.1
OutOfBandInfo	int[-1:]	W	Classification criterion. Allows traffic to be distinguished based on out-of-band information such as physical port or application ID. Primarily intended for, but not restricted to, locally sourced traffic. If specified, this entry applies to traffic with matching out-of-band information. A value of -1 indicates this criterion is not used for classification.	-1	1.4
ForwardingPolicy	unsignedInt	W	Classification result. Identifier of the forwarding policy associated with traffic that falls in this classification.	0	1.1
TrafficClass	int[-1:]	W	Classification result. Identifier of the traffic class associated with traffic that falls in this classification. If specified, at least one Queue table entry MUST include this traffic class in its TrafficClass parameter (which is a comma-separated list). A value of -1 indicates a null traffic class. TrafficClass, ClassQueue and ClassApp are mutually exclusive and one of the three MUST be specified. If TrafficClass and ClassQueue are null, ClassApp MUST be specified, and vice versa.	-1	1.4
ClassPolicer	int[-1:]	W	Classification result. Instance number of the Policer table entry for traffic that falls in this classification. A value of -1 indicates a null policer.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ClassQueue	int[-1:]	W	Classification result. Instance number of the Queue table entry for traffic that falls in this classification. A value of -1 indicates a null queue. TrafficClass, ClassQueue and ClassApp are mutually exclusive and one of the three MUST be specified. If TrafficClass and ClassQueue are null, ClassApp MUST be specified, and vice versa.	-1	1.1
ClassApp	int[-1:]	W	Classification result. Instance number of the App table entry for traffic that falls in this classification. A value of -1 indicates a null App table entry. TrafficClass, ClassQueue and ClassApp are mutually exclusive and one of the three MUST be specified. If TrafficClass and ClassQueue are null, ClassApp MUST be specified, and vice versa.	-1	1.1
InternetGatewayDevice.QueueManagement.-App.{i}.	object	W	Application table.	-	1.1
AppKey	unsignedInt	-	Unique key for each App table entry. This parameter is OBSOLETED because it serves no purpose (no other parameter references it).	-	1.1
AppEnable	boolean	W	Enables or disables this App table entry.	False	1.1
AppStatus	string	-	The status of this App table entry. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	"Disabled"	1.1
ProtocolIdentifier	string(256)	W	URN identifying the protocol associated with the given application. A set of defined URNs is given in Annex A.	<Empty>	1.1
AppName	string(64)	W	Human-readable name associated with this entry in the App table.	<Empty>	1.1
AppDefaultForwardingPolicy	unsignedInt	W	Identifier of the forwarding policy associated with traffic associated with this App table entry, but not associated with any specified flow.	0	1.1
AppDefaultTrafficClass	int[-1:]	W	Identifier of the traffic class associated with traffic associated with this App table entry, but not associated with any specified flow. A value of -1 indicates a null traffic class. AppDefaultTrafficClass and AppDefaultQueue MUST NOT both be specified.	-1	1.4
AppDefaultPolicer	int[-1:]	W	Instance number of the Policer table entry for traffic associated with this App table entry, but not associated with any specified flow. A value of -1 indicates a null policer.	-1	1.1
AppDefaultQueue	int[-1:]	W	Instance number of the Queue table entry for traffic associated with this App table entry, but not associated with any specified flow. A value of -1 indicates a null queue. AppDefaultTrafficClass and AppDefaultQueue MUST NOT both be specified.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
AppDefaultDSCPMark	int[-2:]	W	DSCP to mark traffic associated with this App table entry, but not associated with any specified flow. A value of -1 indicates no change from the incoming packet. A value of -2 indicates automatic marking of DSCP based upon the EthernetPriority value of the incoming packet as defined in Annex A.	-1	1.1
AppDefaultEthernetPriorityMark	int[-2:]	W	Ethernet priority code (as defined in 802.1D) to mark traffic associated with this App table entry, but not associated with any specified flow. A value of -1 indicates no change from the incoming packet. A value of -2 indicates automatic marking of EthernetPriority based upon the DSCP value of the incoming packet as defined in Annex A.	-1	1.1
InternetGatewayDevice.QueueManagement.-Flow.{i}.	object	W	Flow table.	-	1.1
FlowKey	unsignedInt	-	Unique key for each Flow table entry. This parameter is OBSOLETED because it serves no purpose (no other parameter references it).	-	1.1
FlowEnable	boolean	W	Enables or disables this Flow table entry.	False	1.1
FlowStatus	string	-	The status of this Flow table entry. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	"Disabled"	1.1
FlowType	string(256)	W	URN identifying the type of flow to be associated with the specified queue and policer. A set of defined URNs is given in Annex A.	<Empty>	1.1
FlowTypeParameters	string(256)	W	List of name-value pairs representing additional criteria to identify the flow type. The use and interpretation is specific to the particular FlowType URN.Encoded using the "x-www-form-urlencoded" content type defined in [7].	<Empty>	1.1
FlowName	string(64)	W	Human-readable name associated with this entry in the Flow table.	<Empty>	1.1
AppIdentifier	int[-1:]	W	Instance number of the App table entry associated with this flow. A value of -1 indicates the flow table is not associated with any App table entry.	-1	1.1
FlowForwardingPolicy	unsignedInt	W	Identifier of the forwarding policy associated with this flow.	0	1.1
FlowTrafficClass	int[-1:]	W	Identifier of the traffic class associated with this flow. A value of -1 indicates a null traffic class. FlowTrafficClass and FlowQueue MUST NOT both be specified.	-1	1.4
FlowPolicer	int[-1:]	W	Instance number of the Policer table entry for traffic that falls in this flow. A value of -1 indicates a null policer.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
FlowQueue	int[-1:]	W	Instance number of the Queue table entry for traffic that falls in this flow. A value of -1 indicates a null queue. FlowTrafficClass and FlowQueue MUST NOT both be specified.	-1	1.1
FlowDSCPMark	int[-2:]	W	DSCP to mark traffic with that falls into this flow. A value of -1 indicates no change from the incoming packet. A value of -2 indicates automatic marking of DSCP based upon the EthernetPriority value of the incoming packet as defined in Annex A.	-1	1.1
FlowEthernetPriorityMark	int[-2:]	W	Ethernet priority code (as defined in 802.1D) to mark traffic with that falls into this flow. A value of -1 indicates no change from the incoming packet. A value of -2 indicates automatic marking of EthernetPriority based upon the DSCP value of the incoming packet as defined in Annex A.	-1	1.1
InternetGatewayDevice.QueueManagement.-Policer.{i}.	object	W	Policer table.	-	1.1
PolicerKey	unsignedInt	-	Unique key for each policer entry. This parameter is OBSOLETED because it serves no purpose (no other parameter references it).	-	1.1
PolicerEnable	boolean	W	Enables or disables this policer.	False	1.1
PolicerStatus	string	-	The status of this policer. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	"Disabled"	1.1
CommittedRate	unsignedInt	W	Committed rate allowed for this policer in bits-per-second.	0	1.1
CommittedBurstSize	unsignedInt	W	Committed Burstsize in bytes.	0	1.1
ExcessBurstSize	unsignedInt	W	Excess Burstsize in bytes. Applied for a SingleRateThreeColor meter.	0	1.1
PeakRate	unsignedInt	W	Peak rate allowed for this Meter in bits-per-second. Applied for TwoRateThreeColor meters.	0	1.1
PeakBurstSize	unsignedInt	W	Peak Burstsize in bytes. Applied for TwoRateThreeColor meters.	0	1.1
MeterType	string	W	Identifies the method of traffic measurement to be used for this policer. Enumeration of: "SimpleTokenBucket" "SingleRateThreeColor" "TwoRateThreeColor" SimpleTokenBucket makes use of CommittedRate and CommittedBurstSize. SingleRateThreeColor makes use of CommittedRate, CommittedBurstSize, and ExcessBurstSize as defined in RFC 2697 [36]. TwoRateThreeColor makes use of CommittedRate, CommittedBurstSize, PeakRate, and PeakBurstSize as defined in RFC 2698 [37].	"Simple-Token-Bucket"	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
PossibleMeterTypes	string	-	Comma-separated list of supported meter types. Each item is an enumeration of: "SimpleTokenBucket" "SingleRateThreeColor" "TwoRateThreeColor"	-	1.1
ConformingAction	string	W	Instructions for how to handle traffic that is conforming. Enumeration of: "Null" "Drop" "Count" (DEPRECATED) <DSCP Value> <:Ethernet Priority> <DSCP Value:EthernetPriority> Null corresponds with no action. A Count action (and only the Count action) increases the meter instance count statistics in the CountedPackets and CountedBytes parameters. Count actions are DEPRECATED because they can not be combined with other actions, e.g. marking actions. <DSCP Value> is an unsigned integer that corresponds with a mark action overwriting the traffic's DSCP with the configured DSCP. <:EthernetPriority> is a colon (":") followed by an unsigned integer (no white space). It corresponds with a mark action overwriting the traffic's Ethernet Priority with the configured Ethernet Priority. <DSCP Value:Ethernet Priority> is an unsigned integer followed by a colon (":") and a second unsigned integer (no white space). It corresponds with a mark action overwriting the traffic's DSCP and Ethernet Priority with the configured values. For example, "24" specifies a DSCP value of 24, ":3" specifies an Ethernet Priority of 3, and "24:3" specifies both.	"Null"	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
PartialConformingAction	string	W	Instructions for how to handle traffic that is partially conforming (colored yellow). Enumeration of: "Null" "Drop" "Count" (DEPRECATED) <DSCP Value> <:Ethernet Priority> <DSCP Value:EthernetPriority> Null corresponds with no action. A Count action (and only the Count action) increases the meter instance count statistics in the CountedPackets and CountedBytes parameters. Count actions are DEPRECATED because they can not be combined with other actions, e.g. marking actions. <DSCP Value> is an unsigned integer that corresponds with a mark action overwriting the traffic's DSCP with the configured DSCP.Only applies for three-color meters. <:EthernetPriority> is a colon (":") followed by an unsigned integer (no white space). It corresponds with a mark action overwriting the traffic's Ethernet Priority with the configured Ethernet Priority. <DSCP Value:Ethernet Priority> is an unsigned integer followed by a colon (":") and a second unsigned integer (no white space). It corresponds with a mark action overwriting the traffic's DSCP and Ethernet Priority with the configured values. For example, "24" specifies a DSCP value of 24, ":3" specifies an Ethernet Priority of 3, and "24:3" specifies both.	"Drop"	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
NonConformingAction	string	W	Instructions for how to handle traffic that is non-conforming. Enumeration of: "Null" "Drop" "Count" (DEPRECATED) <DSCP Value> <:Ethernet Priority> <DSCP Value:EthernetPriority> Null corresponds with no action. A Count action (and only the Count action) increases the meter instance count statistics in the CountedPackets and CountedBytes parameters. Count actions are DEPRECATED because they can not be combined with other actions, e.g. marking actions. <DSCP Value> is an unsigned integer that corresponds with a mark action overwriting the traffic's DSCP with the configured DSCP. <:EthernetPriority> is a colon (":") followed by an unsigned integer (no white space). It corresponds with a mark action overwriting the traffic's Ethernet Priority with the configured Ethernet Priority. <DSCP Value:Ethernet Priority> is an unsigned integer followed by a colon (":") and a second unsigned integer (no white space). It corresponds with a mark action overwriting the traffic's DSCP and Ethernet Priority with the configured values. For example, "24" specifies a DSCP value of 24, ":3" specifies an Ethernet Priority of 3, and "24:3" specifies both.	"Drop"	1.1
CountedPackets	unsignedInt	-	Number of Packets counted as result of a Count meter action. This parameter is DEPRECATED because the Count meter action is DEPRECATED.	0	1.1
CountedBytes	unsignedInt	-	Number of Bytes counted as result of a Count meter action. This parameter is DEPRECATED because the Count meter action is DEPRECATED.	0	1.1
TotalCountedPackets	unsignedInt	-	Total number of Packets counted by this policer, regardless of meter action.	0	1.4
TotalCountedBytes	unsignedInt	-	Total number of Bytes counted by this policer, regardless of meter action.	0	1.4
ConformingCountedPackets	unsignedInt	-	Number of conforming Packets counted by this policer, regardless of meter action.	0	1.4
ConformingCountedBytes	unsignedInt	-	Number of conforming Bytes counted by this policer, regardless of meter action.	0	1.4
PartiallyConformingCountedPackets	unsignedInt	-	Number of partially conforming Packets counted by this policer, regardless of meter action.	0	1.4
PartiallyConformingCountedBytes	unsignedInt	-	Number of partially conforming Bytes counted by this policer, regardless of meter action.	0	1.4
NonConformingCountedPackets	unsignedInt	-	Number of non-conforming Packets counted by this policer, regardless of meter action.	0	1.4
NonConformingCountedBytes	unsignedInt	-	Number of non-conforming Bytes counted by this policer, regardless of meter action.	0	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.QueueManagement.-Queue.{i}.	object	W	Queue table. This table can contain hardware queues. The CPE MAY refuse to allow hardware queues to be deleted.	-	1.1
QueueKey	unsignedInt	-	Unique key for each queue entry. This parameter is OBSOLETED because it serves no purpose (no other parameter references it).	-	1.1
QueueEnable	boolean	W	Enables or disables this queue.	False	1.1
QueueStatus	string	-	The status of this queue. Enumeration of: "Disabled" "Enabled" "Error" (OPTIONAL) The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	"Disabled"	1.1
TrafficClasses	string(256)	W	Comma-separated list of unsigned integers, identifying the set of traffic classes associated with this queue. If this list is empty then traffic can be sent to this queue only as a result of a direct reference from a Classification, App or Flow table entry, e.g. via the Classification table's ClassQueue parameter. If this list is non-empty then traffic can additionally be sent to this queue if a Classification, App or Flow table entry specifies a traffic class, e.g. via the Classification table's TrafficClass parameter. If more than one queue on a given egress interface is associated with a given traffic class, the implementation will choose which queue to send traffic of this class to.	<Empty>	1.4
QueueInterface	string(256)	W	Egress interfaces for which the specified queue MUST exist. This parameter MUST be in one of the following forms: The full hierarchical parameter name of the particular WANDevice, WANConnection-Device, WAN**Connection, LANDevice, LAN**InterfaceConfig, or WLANConfiguration object. The string "WAN", which indicates this entry applies to all WAN interfaces. The string "LAN", which indicates this entry applies to all LAN interfaces. An empty value, which indicates this classification entry is to apply to all interfaces. Packets classified into this queue that exit through any other interface MUST instead use the default queuing behavior specified in the Queue table entry referenced by InternetGatewayDevice.-QueueManagement.DefaultQueue. For the default queue itself (the Queue table entry referenced by InternetGatewayDevice.Queue-Management.DefaultQueue), the value of the QueueInterface parameter MUST be ignored. That is, the default queue MUST exist on all egress interfaces.	<Empty>	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
QueueBufferLength	unsignedInt	-	Number of bytes in the buffer. Queue buffer size for all egress interfaces for which this queue exists. If the buffer size is not the same for all such egress interfaces, this parameter MUST be 0.	-	1.1
QueueWeight	unsignedInt	W	Weight of this queue in case of WFQ or WRR, but only used for queues of equal precedence.	0	1.1
QueuePrecedence	unsignedInt [1:]	W	Precedence of this queue relative to others. Lower numbers imply greater precedence.	1	1.1
REDThreshold	unsignedInt [0:100]	W	Random Early Detection threshold, used only when DropAlgorithm is RED. This is the minimum threshold (min_th) and is measured as a percentage of the queue size. If the value is set to zero, the CPE MUST choose a sensible value, e.g. 5 (but the value MUST still read back as zero). In this version of the data model, there is no way to set the maximum threshold (max_th). The CPE MUST choose a sensible value, e.g. three times the minimum threshold. In this version of the data model, there is no way to set the RED weight (w_q). The CPE MUST choose a sensible value, e.g. 0.002.	0	1.1
REDPercentage	unsignedInt [0:100]	W	Random Early Detection percentage, used only when DropAlgorithm is RED. This is the maximum value of the packet marking probability (max_p). If the value is set to zero, the CPE MUST choose a sensible value, e.g. 10 (but the value MUST still read back as zero). In this version of the data model, there is no way to set the RED weight (w_q). The CPE MUST choose a sensible value, e.g. 0.002.	0	1.1
DropAlgorithm	string	W	Dropping algorithm used for this queue if congested. Enumeration of: “RED” (Random Early Detection [10]) “DT” (Drop Tail) “WRED” (Weighted RED) “BLUE” ([11])	“DT”	1.1
SchedulerAlgorithm	string	W	Scheduling Algorithm used by scheduler. Enumeration of: “WFQ” (Weighted Fair Queueing) “WRR” (Weighted Round Robin) “SP” (Strict Priority)	“SP”	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ShapingRate	int[-1:]	W	Rate to shape this queue's traffic to. For leaky bucket (constant rate shaping), this is the constant rate. For token bucket (variable rate shaping), this is the average rate. If <= 100, in percent of the rate of the highest rate-constrained layer over which the packet will travel on egress. If > 100, in bits per second. A value of -1 indicates no shaping. For example, for packets destined for a WAN DSL interface, if the egress will be on a PPP or IP link with a specified ShapingRate, the percentage is calculated relative to this rate. Otherwise, if the ATM layer is rate-constrained, then the rate is calculated relative to this rate. Otherwise, the rate is calculated relative to the physical-layer DSL rate.	-1	1.1
ShapingBurstSize	unsignedInt	W	Burst size in bytes. For both leaky bucket (constant rate shaping) and token bucket (variable rate shaping) this is the bucket size and is therefore the maximum burst size.	-	1.1
InternetGatewayDevice.QueueManagement.-QueueStats.{i}.	object	W	Queue statistics table. This table is managed by the ACS, which will create entries only for those {Queue, Interface} combinations for which statistics are to be collected.	-	1.4
Enable	boolean	W	Enables or disables this object.	False	1.4
Status	string	-	The status of this object. Enumeration of: "Disabled" "Enabled" (Enabled and {Queue,Interface} is valid) "Error" (Enabled but {Queue,Interface} is invalid)	"Disabled"	1.4
Queue	unsignedInt	W	Instance number of the Queue table entry with which this object is associated.	0	1.4
Interface	unsignedInt	W	Egress interface for which this object contains statistics. This parameter MUST be the full hierarchical parameter name of the particular WANDevice, WANConnectionDevice, WAN**-Connection, LANDevice, LAN**InterfaceConfig, or WLANConfiguration object.	<Empty>	1.4
OutputPackets	unsignedInt	-	Number of packets output through the queue.	0	1.4
OutputBytes	unsignedInt	-	Number of bytes output through the queue.	0	1.4
DroppedPackets	unsignedInt	-	Number of packets dropped by the queue.	0	1.4
DroppedBytes	unsignedInt	-	Number of bytes dropped by the queue.	0	1.4
QueueOccupancyPackets	unsignedInt	-	Queue occupancy in packets (gives a measure of queue latency).	0	1.4
QueueOccupancyPercentage	unsignedInt [0:100]	-	Queue occupancy as a percentage, i.e. 100 * queue occupancy in bytes / queue size in bytes (gives a measure of queue usage).	0	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.LANConfigSecurity.	object	-	This object contains generic device configuration information.	-	1.0
ConfigPassword	string(64)	W	A password to allow LAN access to protected auto-configuration services. If the CPE supports TR-064 (LAN-side DSL CPE Configuration Protocol), this parameter is to be used as the “dslf-config” password (as defined in TR-064). If the CPE has a user interface with password protection enabled, this parameter is also to be used as the user password for password-protected operations. However, this parameter MUST NOT be used to set the user password if the parameter InternetGatewayDevice.UserInterface.Password-UserSelectable is True. When read, this parameter returns an empty string, regardless of the actual value.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.IPPingDiagnostics.	object	-	This object provides access to an IP-layer ping test.	-	1.0
DiagnosticsState	string	W	Indicates availability of diagnostic data. One of: "None" "Requested" "Complete" "Error_CannotResolveHostName" "Error_Internal" "Error_Other" If the ACS sets the value of this parameter to Requested, the CPE MUST initiate the corresponding diagnostic test. When writing, the only allowed value is Requested. To ensure the use of the proper test parameters (the writable parameters in this object), the test parameters MUST be set either prior to or at the same time as (in the same SetParameterValues) setting the DiagnosticsState to Requested. When requested, the CPE SHOULD wait until after completion of the communication session with the ACS before starting the diagnostic. When the test is completed, the value of this parameter MUST be either Complete (if the test completed successfully), or one of the Error values listed above. If the value of this parameter is anything other than Complete, the values of the results parameters for this test are indeterminate. When the diagnostic initiated by the ACS is completed (successfully or not), the CPE MUST establish a new connection to the ACS to allow the ACS to view the results, indicating the Event code "8 DIAGNOSTICS COMPLETE" in the Inform message. After the diagnostic is complete, the value of all result parameters (all read-only parameters in this object) MUST be retained by the CPE until either this diagnostic is run again, or the CPE reboots. After a reboot, if the CPE has not retained the result parameters from the most recent test, it MUST set the value of this parameter to "None". Modifying any of the writable parameters in this object except for this one MUST result in the value of this parameter being set to "None". While the test is in progress, modifying any of the writable parameters in this object except for this one MUST result in the test being terminated and the value of this parameter being set to "None". While the test is in progress, setting this parameter to Requested (and possibly modifying other writable parameters in this object) MUST result in the test being terminated and then restarted using the current values of the test parameters.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Interface	string(256)	W	Specifies the WAN or LAN IP-layer interface over which the test is to be performed. This identifies the source IP address to use when performing the test. The content is the full hierarchical parameter name of the interface. The following is a WAN interface example: "InternetGatewayDevice.WANDevice.1.WAN-ConnectionDevice.2.WANPPPConnection.1" The following is a LAN interface example: "InternetGatewayDevice.LANDevice.1.LAN-HostConfigManagement.IPInterface.1" The value of this parameter MUST be either a valid interface or an empty string. An attempt to set this parameter to a different value MUST be rejected as an invalid parameter value. If an empty string is specified, the CPE MUST use the interface as directed by its routing policy (Forwarding table entries) to determine the appropriate interface.	-	1.0
Host	string(256)	W	Host name or address of the host to ping.	-	1.0
NumberOfRepetitions	unsignedInt[1:]	W	Number of repetitions of the ping test to perform before reporting the results.	-	1.0
Timeout	unsignedInt[1:]	W	Timeout in milliseconds for the ping test.	-	1.0
DataBlockSize	unsignedInt[1:65535]	W	Size of the data block in bytes to be sent for each ping.	-	1.0
DSCP	unsignedInt [0:63]	W	DiffServ codepoint to be used for the test packets. By default the CPE SHOULD set this value to zero.	-	1.0
SuccessCount	unsignedInt	-	Result parameter indicating the number of successful pings (those in which a successful response was received prior to the timeout) in the most recent ping test.	-	1.0
FailureCount	unsignedInt	-	Result parameter indicating the number of failed pings in the most recent ping test.	-	1.0
AverageResponseTime	unsignedInt	-	Result parameter indicating the average response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.	-	1.0
MinimumResponseTime	unsignedInt	-	Result parameter indicating the minimum response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.	-	1.0
MaximumResponseTime	unsignedInt	-	Result parameter indicating the maximum response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.	-	1.0
InternetGatewayDevice.TraceRouteDiagnos-tics.	object	-	This object is defines access to an IP-layer trace-route test for the specified IP interface.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DiagnosticsState	string	W	Indicates availability of diagnostic data. One of: "None" "Requested" "Complete" "Error_CannotResolveHostName" "Error_MaxHopCountExceeded" If the ACS sets the value of this parameter to Requested, the CPE MUST initiate the corresponding diagnostic test. When writing, the only allowed value is Requested. To ensure the use of the proper test parameters (the writable parameters in this object), the test parameters MUST be set either prior to or at the same time as (in the same SetParameterValues) setting the DiagnosticsState to Requested. When requested, the CPE SHOULD wait until after completion of the communication session with the ACS before starting the diagnostic. When the diagnostic initiated by the ACS is completed (successfully or not), the CPE MUST establish a new connection to the ACS to allow the ACS to view the results, indicating the Event code "8 DIAGNOSTICS COMPLETE" in the Inform message.	-	1.4
Interface	string(256)	W	Specifies the WAN or LAN IP-layer interface over which the test is to be performed. This identifies the source IP address to use when performing the test. The content is the full hierarchical parameter name of the interface. The following is a WAN interface example: "InternetGatewayDevice.WANDevice.1.WAN-ConnectionDevice.2.WANPPPConnection.1" The following is a LAN interface example: "InternetGatewayDevice.LANDevice.1.LAN-HostConfigManagement.IPInterface.1"	-	1.4
Host	string(256)	W	Host name or address of the host to find a route to	-	1.4
NumberOfTries	unsignedInt [1:3]	W	Number of tries per hop. Set prior to running Diagnostic. By default, the CPE SHOULD set this value to 3.	-	1.4
Timeout	unsignedInt [1:]	W	Timeout in milliseconds for the trace route test. By default the CPE SHOULD set this value to 5000.	-	1.4
DataBlockSize	unsignedInt [1:65535]	W	Size of the data block in bytes to be sent for each trace route. By default, the CPE SHOULD set this value to 38.	-	1.4
DSCP	unsignedInt [0:63]	W	DiffServ codepoint to be used for the test packets. By default the CPE SHOULD set this value to 0.	-	1.4
MaxHopCount	unsignedInt [1:64]	W	The maximum number of hop used in outgoing probe packets (max TTL). By default the CPE SHOULD set this value to 30.	-	1.4
ResponseTime	unsignedInt	-	Result parameter indicating the response time in milliseconds the most recent trace route test. If a route could not be determined, this value MUST be zero.	-	1.4
RouteHopsNumberOfEntries	unsignedInt	-	Number of entries in the RouteHops table.	-	1.4
InternetGateway-Device.TraceRouteDiagnostics.RouteHops.{i}.	object	-	Contains the array of results returned. If a route could not be determined, this array will be empty	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
HopHost	string(256)	-	Result parameter indicating the Host Name if DNS is able to resolve or IP Address of a hop along the discovered route.	-	1.4
HopHostAddress	string	-	If this parameter is non empty it will contain the last IP address of the host returned for this hop and the HopHost will contain the Host Name returned from the reverse DNS query.	-	1.4
HopErrorCode	unsignedInt	-	Contains the error code returned for this hop This code is directly from the ICMP CODE field.	-	1.4
HopRTTimes	string(16)	-	Contains the comma separated list of one or more round trip times in milliseconds (one for each repetition) for this hop.	-	1.4
InternetGatewayDevice.DownloadDiagnostics.	object	-	This object defines the diagnostics configuration for a HTTP and FTP DownloadDiagnostics Test. Files received in the DownloadDiagnostics do not require file storage on the CPE device.	-	1.3

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DiagnosticsState	string	W	Indicate the availability of diagnostic data. One of: "None" "Requested" "Completed" "Error_InitConnectionFailed" "Error_NoResponse " "Error_TransferFailed" "Error_PasswordRequestFailed" "Error_LoginFailed" "Error_NoTransferMode" "Error_NoPASV" "Error_IncorrectSize" "Error_Timeout" If the ACS sets the value of this parameter to Requested, the CPE MUST initiate the corresponding diagnostic test. When writing, the only allowed value is Requested. To ensure the use of the proper test parameters (the writable parameters in this object), the test parameters MUST be set either prior to or at the same time as (in the same SetParameterValues) setting the DiagnosticsState to Requested. When requested, the CPE SHOULD wait until after completion of the communication session with the ACS before starting the diagnostic. When the test is completed, the value of this parameter MUST be either Completed (if the test completed successfully), or one of the Error values listed above. If the value of this parameter is anything other than Completed, the values of the results parameters for this test are indeterminate. When the diagnostic initiated by the ACS is completed (successfully or not), the CPE MUST establish a new connection to the ACS to allow the ACS to view the results, indicating the Event code "8 DIAGNOSTICS COMPLETE" in the Inform message. After the diagnostic is complete, the value of all result parameters (all read-only parameters in this object) MUST be retained by the CPE until either this diagnostic is run again, or the CPE reboots. After a reboot, if the CPE has not retained the result parameters from the most recent test, it MUST set the value of this parameter to "None". Modifying any of the writable parameters in this object except for this one MUST result in the value of this parameter being set to "None". While the test is in progress, modifying any of the writable parameters in this object except for this one MUST result in the test being terminated and the value of this parameter being set to "None". While the test is in progress, setting this parameter to Requested (and possibly modifying other writable parameters in this object) MUST result in the test being terminated and then restarted using the current values of the test parameters.	-	1.3

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Interface	string(256)	W	Specifies the IP-layer interface over which the test is to be performed. The content is the full hierarchical parameter name of the interface. The value of this parameter MUST be either a valid interface or an empty string. An attempt to set this parameter to a different value MUST be rejected as an invalid parameter value. If an empty string is specified, the CPE MUST use the default routing interface.	-	1.3
DownloadURL	string(256)	W	The URL, as defined in [8], for the CPE to perform the download on. This parameter MUST be in the form of a valid HTTP [6] or FTP [25] URL. When using FTP transport, FTP binary transfer MUST be used. When using HTTP transport, persistent connections MUST be used and pipelining MUST NOT be used. When using HTTP transport the HTTP Authentication MUST NOT be used.	-	1.3
DSCP	unsignedInt[0:63]	W	The DiffServ code point for marking packets transmitted in the test. The default value SHOULD be zero.	-	1.3
EthernetPriority	unsignedInt[0:7]	W	Ethernet priority code for marking packets transmitted in the test (if applicable). The default value SHOULD be zero.	-	1.3
ROMTime	dateTime	-	Request time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the time at which the client sends the GET command. For FTP this is the time at which the client sends the RTRV command.	-	1.3
BOMTime	dateTime	-	Begin of transmission time in UTC, which MUST be specified to microsecond precision For example: 2008-04-09T15:01:05.123456 For HTTP this is the time at which the first data packet is received. For FTP this is the time at which the client receives the first data packet on the data connection.	-	1.3
EOMTime	dateTime	-	End of transmission in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the time at which the last data packet is received. For FTP this is the time at which the client receives the last packet on the data connection.	-	1.3
TestBytesReceived	unsignedInt	-	The test traffic received in bytes during the FTP/HTTP transaction including FTP/HTTP headers, between BOMTime and EOMTime,	-	1.3
TotalBytesReceived	unsignedInt	-	The total number of bytes received on the Interface between BOMTime and EOMTime.	-	1.3

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
TCPOpenRequestTime	dateTime	-	Request time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the time at which the TCP socket open (SYN) was sent for the HTTP connection. For FTP this is the time at which the TCP socket open (SYN) was sent for the data connection. Note: Interval of 1 microsecond SHOULD be supported.	-	1.3
TCPOpenResponseTime	dateTime	-	Response time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the time at which the TCP ACK to the socket opening the HTTP connection was received. For FTP this is the time at which the TCP ACK to the socket opening the data connection was received. Note: Interval of 1 microsecond SHOULD be supported.	-	1.3
InternetGatewayDevice.UploadDiagnostics.	object	-	This object defines the diagnostics configuration for a HTTP or FTP UploadDiagnostics test. Files sent by the UploadDiagnostics do not require file storage on the CPE device, and MAY be an arbitrary stream of bytes.	-	1.3

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DiagnosticsState	string	W	Indicate the availability of diagnostic data. One of: "None" "Requested" "Completed" "Error_InitConnectionFailed" "Error_NoResponse" "Error_PasswordRequestFailed" "Error_LoginFailed" "Error_NoTransferMode" "Error_NoPASV" "Error_NoCWD" "Error_NoSTOR" "Error_NoTransferComplete" If the ACS sets the value of this parameter to Requested, the CPE MUST initiate the corresponding diagnostic test. When writing, the only allowed value is Requested. To ensure the use of the proper test parameters (the writable parameters in this object), the test parameters MUST be set either prior to or at the same time as (in the same SetParameterValues) setting the DiagnosticsState to Requested. When requested, the CPE SHOULD wait until after completion of the communication session with the ACS before starting the diagnostic. When the test is completed, the value of this parameter MUST be either Completed (if the test completed successfully), or one of the Error values listed above. If the value of this parameter is anything other than Completed, the values of the results parameters for this test are indeterminate. When the diagnostic initiated by the ACS is completed (successfully or not), the CPE MUST establish a new connection to the ACS to allow the ACS to view the results, indicating the Event code "8 DIAGNOSTICS COMPLETE" in the Inform message. After the diagnostic is complete, the value of all result parameters (all read-only parameters in this object) MUST be retained by the CPE until either this diagnostic is run again, or the CPE reboots. After a reboot, if the CPE has not retained the result parameters from the most recent test, it MUST set the value of this parameter to "None". Modifying any of the writable parameters in this object except for this one MUST result in the value of this parameter being set to "None". While the test is in progress, modifying any of the writable parameters in this object except for this one MUST result in the test being terminated and the value of this parameter being set to "None". While the test is in progress, setting this parameter to Requested (and possibly modifying other writable parameters in this object) MUST result in the test being terminated and then restarted using the current values of the test parameters.	-	1.3

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Interface	string(256)	W	IP-layer interface over which the test is to be performed. The content is the full hierarchical parameter name of the interface. The value of this parameter MUST be either a valid interface or an empty string. An attempt to set this parameter to a different value MUST be rejected as an invalid parameter value. If an empty string is specified, the CPE MUST use the default routing interface.	-	1.3
UploadURL	string(256)	W	The URL, as defined in [8], for the CPE to Upload to. This parameter MUST be in the form of a valid HTTP [6] or FTP [25] URL. When using FTP transport, FTP binary transfer MUST be used. When using HTTP transport, persistent connections MUST be used and pipelining MUST NOT be used. When using HTTP transport the HTTP Authentication MUST NOT be used.	-	1.3
DSCP	unsignedInt[0:63]	W	DiffServ code point for marking packets transmitted in the test. The default value SHOULD be zero.	-	1.3
EthernetPriority	unsignedInt[0:7]	W	Ethernet priority code for marking packets transmitted in the test (if applicable). The default value SHOULD be zero.	-	1.3
TestFileLength	unsignedInt	W	The size of the file (in bytes) to be uploaded to the server. The CPE MUST insure the appropriate number of bytes are sent.	-	1.3
ROMTime	dateTime	-	Request time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the time at which the client sends the PUT command For FTP this is the time at which the STOR command is sent.	-	1.3
BOMTime	dateTime	-	Begin of transmission time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the time at which the first data packet is sent. For FTP this is the time at which the client receives the ready for transfer notification.	-	1.3
EOMTime	dateTime	-	End of transmission in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the time when the HTTP successful response code is received. For FTP this is the time when the client receives a transfer complete.	-	1.3
TotalBytesSent	unsignedInt	-	The total number of bytes sent on the Interface between BOMTime and EOMTime.	-	1.3

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
TCPOpenRequestTime	dateTime	-	Request time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the time at which the TCP socket open (SYN) was sent for the HTTP connection. For FTP this is the time at which the TCP socket open (SYN) was sent for the data connection Note: Interval of 1 microsecond SHOULD be supported.	-	1.3
TCPOpenResponseTime	dateTime	-	Response time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 For HTTP this is the Time at which the TCP ACK to the socket opening the HTTP connection was received. For FTP this is the Time at which the TCP ACK to the socket opening the Data connection was received. Note: Interval of 1 microsecond SHOULD be supported.	-	1.3
InternetGatewayDevice.UDPEchoConfig.	object	-	This object allows the CPE to be configured to perform the UDP Echo Service defined in [25] and UDP Echo Plus Service defined in Appendix A.1 of [50].	-	1.3
Enable	boolean	W	MUST be enabled to receive UDP echo. When enabled from a disabled state all related timestamps, statistics and UDP Echo Plus counters are cleared.	-	1.3
Interface	string(256)	W	IP-layer interface over which the CPE MUST listen and receive UDP echo requests on. The content is the full hierarchical parameter name of the interface. The value of this parameter MUST be either a valid interface or an empty string. An attempt to set this parameter to a different value MUST be rejected as an invalid parameter value. If an empty string is specified, the CPE MUST listen and receive UDP echo requests on all interfaces. Note: Interfaces behind a NAT MAY require port forwarding rules configured in the Gateway to enable receiving the UDP packets.	-	1.3
SourceIPAddress	string	W	The Source IP address of the UDP echo packet. The CPE MUST only respond to a UDP echo from this source IP address.	-	1.3
UDPPort	unsignedInt	W	The UDP port on which the UDP server MUST listen and respond to UDP echo requests.	-	1.3
EchoPlusEnabled	boolean	W	If True the CPE will perform necessary packet processing for UDP Echo Plus packets.	-	1.3
EchoPlusSupported	boolean	-	True if UDP Echo Plus is supported.	-	1.3
PacketsReceived	unsignedInt	-	Incremented upon each valid UDP echo packet received.	-	1.3
PacketsResponded	unsignedInt	-	Incremented for each UDP echo response sent.	-	1.3
BytesReceived	unsignedInt	-	The number of UDP received bytes including payload and UDP header after the UDPEchoConfig is enabled.	-	1.3

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
BytesResponded	unsignedInt	-	The number of UDP responded bytes, including payload and UDP header sent after the UDPEchoConfig is enabled.	-	1.3
TimeFirstPacketReceived	dateTime	-	Time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456, The time that the server receives the first UDP echo packet after the UDPEchoConfig is enabled.	-	1.3
TimeLastPacketReceived	dateTime	-	Time in UTC, which MUST be specified to microsecond precision. For example: 2008-04-09T15:01:05.123456 The time that the server receives the most recent UDP echo packet.	-	1.3
InternetGatewayDevice.LANDevice.{i}.	object	W	Each instance models a LAN side layer 3 IP interface. Each instance has children that correspond to the layer 2 interfaces that are connected to the Gateway's IP router via the modeled IP interface. If a LANDevice instance is deleted, the objects modeling those layer 2 interfaces that are as a result no longer connected to the Gateway's IP router will move to the InternetGateway-Device.LANInterfaces object. If the Layer2Bridging object is implemented, the view that it provides of the CPE's underlying bridging configuration MUST be consistent with the view provided by any LANDevice and WAN**Conn-ection objects. The implications of this are explained in Annex A.6.	-	1.0
LANEthernetInterfaceNumberOfEntries	unsignedInt	-	Number of instances of LANEthernetInterface-Config in this LANDevice.	0	1.0
LANUSBInterfaceNumberOfEntries	unsignedInt	-	Number of instances of LANUSBInterfaceConfig in this LANDevice.	0	1.0
LANWLANConfigurationNumberOfEntries	unsignedInt	-	Number of instances of WLANConfiguration in this LANDevice.	0	1.0
InternetGatewayDevice.LANDevice.{i}.LAN-HostConfigManagement.	object	-	This object enables reporting of LAN-related device information and setting and configuring LAN IP addressing. The DHCP parameters in this object define the behavior of the default DHCP server, i.e. the behavior for DHCP requests that do not match any of the DHCP conditional serving pool entries.	-	1.0
MACAddress	string	-	The MAC address associated with the IP interface modeled by this LANDevice instance. This is the MAC address that is returned in response to an ARP request for any of the IP interface's IP addresses. It is also the source MAC address in all IP traffic sent over the IP interface. If no single MAC address meets the above criteria, the value of this parameter MUST be the all-zero MAC address "00:00:00:00:00:00".	-	1.4
DHCPServerConfigurable	boolean	W	Enables the configuration of the DHCP server on the LAN interface. If this variable is set to False, the CPE SHOULD restore its default DHCP server settings.	True	1.0
DHCPServerEnable	boolean	W	Enables or disables the DHCP server on the LAN interface.	False	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DHCPRelay	boolean	-	Indicates if the DHCP server performs the role of a server (False) or a relay (True) on the LAN interface. This parameter is DEPRECATED because the functionality that it describes is not well-defined. The CPE MAY set it to the value that it thinks most appropriate, based on its configuration.	-	1.0
MinAddress	string	W	Specifies first address in the pool to be assigned by the DHCP server on the LAN interface. This parameter MUST have a valid value before the DHCP server can be enabled.	-	1.0
MaxAddress	string	W	Specifies last address in the pool to be assigned by the DHCP server on the LAN interface. This parameter MUST have a valid value before the DHCP server can be enabled.	-	1.0
ReservedAddresses	string(256)	W	Comma-separated list of addresses marked reserved from the address allocation pool.	<Empty>	1.0
SubnetMask	string	W	Specifies the client's network subnet mask. This parameter MUST have a valid value before the DHCP server can be enabled.	-	1.0
DNSServers	string(64)	W	Comma-separated list of DNS servers offered to DHCP clients. Support for more than three DNS Servers is OPTIONAL.	-	1.0
DomainName	string(64)	W	Sets the domain name to provide to clients on the LAN interface.	-	1.0
IPRouters	string(64)	W	Comma-separated list of IP addresses of routers on this subnet. Also known as default gateway. Support for more than one Router address is OPTIONAL.	-	1.0
DHCPLeaseTime	int[-1:]	W	Specifies the lease time in seconds of client assigned addresses. A value of -1 indicates an infinite lease.	86400	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
UseAllocatedWAN	string	W	Controls use of addresses from the associated WAN connection. Enumeration of: "Normal" "UseAllocatedSubnet" (DEPRECATED) "Passthrough" If Normal, the address pool is directly configured by the ACS. If UseAllocatedSubnet, behavior is the same as for Passthrough with an empty PassthroughMACAddress. For this reason, UseAllocatedSubnet is DEPRECATED. If Passthrough, and PassthroughMACAddress is empty, the configured values of the MinAddress, MaxAddress, SubnetMask and DNSServers parameters are ignored. The corresponding address pool values are instead taken from the WAN connection specified by AssociatedConnection. If Passthrough, and PassthroughMACAddress is non-empty, the LAN Host identified by PassthroughMACAddress is given a WAN IP address from the WAN connection specified by AssociatedConnection. Other LAN Hosts are treated as for Normal. Use of PassthroughMACAddress does not cover the case where more than one LAN Host is to be given a WAN IP address. This can be achieved by using a DHCP conditional serving pool.	"Normal"	1.0
AssociatedConnection	string(256)	W	Specifies the connection instance to be used for address allocation if UseAllocatedWAN is set to UseAllocatedSubnet or Passthrough. The content is the full hierarchical parameter name of a WAN-side layer 3 connection object. Example: "InternetGatewayDevice.WANDevice.1.WANConnectionDevice.2.WANPPPConnection.1". If UseAllocatedWAN is UseAllocatedSubnet or Passthrough, this parameter MUST have a valid value before the DHCP server can be enabled.	-	1.0
PassthroughLease	unsignedInt	W	DHCP lease time in seconds given to the LAN Host that is used to passthrough a WAN IP address if UseAllocatedWAN is Passthrough. Note: A temporary private IP address with short lease (for example, 1 min) might be given to the passthrough LAN Host before the WAN IP address is acquired.	600	1.0
PassthroughMACAddress	string	W	Hardware address of the LAN Host that is used to passthrough a WAN IP address if UseAllocatedWAN is Passthrough. Note: An empty address indicates that no specific LAN Host is designated, and results in the same behavior as the (DEPRECATED) UseAllocatedSubnet.	-	1.0
AllowedMACAddresses	string(512)	W	Represents a comma-separated list of hardware addresses that are allowed to connect to this connection if MACAddressControlEnabled is 1 for a given interface.	-	1.0
IPInterfaceNumberOfEntries	unsignedInt	-	The number of entries in the IPInterface table.	0	1.0
DHCPStaticAddressNumberOfEntries	unsignedInt	-	The number of entries in the DHCPStaticAddress table.	0	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DHCPOptionNumberOfEntries	unsignedInt	-	The number of entries in the DHCPOption table.	0	1.4
DHCPConditionalPoolNumberOfEntries	unsignedInt	-	The number of entries in the DHCPConditional-Pool table.	0	1.4
InternetGatewayDevice.LANDevice.{i}.LAN-HostConfigManagement.IPInterface.{i}.	object	W	IP address table with each object representing an IP address on the LANDevice IP interface. Support for more than one interface instance is OPTIONAL.	-	1.0
Enable	boolean	W	Enables or disables this entry. On creation, an entry is disabled by default.	False	1.0
IPInterfaceIPAddress	string	W	IP address of the LAN-side interface of the CPE.	<Empty>	1.0
IPInterfaceSubnetMask	string	W	Subnet mask of the LAN-side interface of the IGD.	<Empty>	1.0
IPInterfaceAddressingType	string	W	Represents the addressing method used to assign the LAN-side IP address of the CPE on this interface. Enumeration of: "DHCP" "Static" "AutoIP"	"DHCP"	1.0
InternetGatewayDevice.LANDevice.{i}.LAN-HostConfigManagement.DHCPStaticAddress.-{i}.	object	W	DHCP static address table. Entries in this table correspond to what RFC 2131 [27] calls "manual allocation", where a client's IP address is assigned by the network administrator, and DHCP is used simply to convey the assigned address to the client. Each instance of this object specifies a hardware address (MAC address) and an IP address within the pool. When serving from this pool, this IP address MUST, if available, be assigned to the DHCP client with this hardware address, and MUST NOT be assigned to any other client. Note that it is possible that an IP address in this table is present in one or more of the conditional serving pools, in which case it is possible that such an address will be assigned to a different client.	-	1.4
Enable	boolean	W	Enables or disables the DHCPStaticAddress table entry. Disabling an entry does not return the IP address to the pool.	False	1.4
Chaddr	string	W	Hardware address (MAC address) of the physical interface of the DHCP client. This parameter MUST have a valid value before the table entry can be enabled.	<Empty>	1.4
Yiaddr	string	W	IP address to be assigned by the DHCP server to the DHCP client with the specified hardware address (MAC address). This parameter MUST have a valid value before the table entry can be enabled.	<Empty>	1.4
InternetGatewayDevice.LANDevice.{i}.LAN-HostConfigManagement.DHCPOption.{i}.	object	W	This object specifies the DHCP options that MUST, if enabled, be returned to clients whose DHCP requests do not match any of the DHCP conditional serving pool entries.	-	1.4
Enable	boolean	W	Enables or disables this DHCPOption table entry.	False	1.4
Tag	unsignedInt [1:254]	W	Option tag as defined in RFC 2132 [28].	-	1.4
Value	base64(340)	W	Base64 encoded option value.	<Empty>	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.DHCPConditionalServingPool.{i}.	object	W	DHCP conditional serving pool table. Each instance of this object defines a DHCP conditional serving pool. Client requests are associated with pools based on criteria such as source interface, supplied DHCP options, and MAC address. If a DHCP request does not match any of the DHCP conditional serving pool entries, the handling of the request is determined by the default DHCP server behavior that is defined by the LANHostConfigManagement object. Overlapping pool ranges MUST be supported.	-	1.4
Enable	boolean	W	Enables or disables the DHCPConditionalServingPool entry.	False	1.4
PoolOrder	unsignedInt [1:]	W	Position of the pool entry in the order of precedence. A value of 1 indicates the first entry considered. For each DHCP request, the highest ordered entry that matches the association criteria is applied. All lower order entries are ignored. When this value is modified, if the value matches that of an existing entry, the Order value for the existing entry and all lower Order entries is incremented (lowered in precedence) to ensure uniqueness of this value. A deletion causes Order values to be compacted. When a value is changed, incrementing occurs before compaction. The value on creation of a DHCPConditionalServingPool table entry MUST be one greater than the largest current value.	-	1.4
SourceInterface	string(1024)	W	Pool association criterion. Specifies the layer 2 ingress interfaces associated with this entry. The content is a comma-separated list of the full hierarchical parameter names of the corresponding LAN**InterfaceConfig or WLANConfiguration objects. For example: ”InternetGatewayDevice.LANDevice.1.LANEthernetInterfaceConfig.2,InternetGatewayDevice.LANDevice.1.WLANConfiguration.3” An empty value indicates this entry is to apply to all layer 2 interface objects under this LANDevice instance.	<Empty>	1.4
VendorClassID	string(256)	W	Pool association criterion. Used to identify one or more LAN devices, value of the DHCP Vendor Class Identifier (Option 60) as defined in RFC 2132 [28], matched according to the criterion in VendorClassIDMode. Case sensitive. An empty value indicates this criterion is not used for conditional serving.	<Empty>	1.4
VendorClassIDExclude	boolean	W	If False, matching packets are those that match the VendorClassID entry, if specified. If True, matching packets are those that do not match the VendorClassID entry, if specified.	False	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
VendorClassIDMode	string	W	VendorClassID pattern match criterion. Enumeration of: "Exact" "Prefix" "Suffix" "Substring" For example, if VendorClassID is "Example" then an Option 60 value of "Example device" will match with VendorClassID values of "Prefix" or "Substring", but not with "Exact" or "Suffix".	"Exact"	1.4
ClientID	string(256)	W	Pool association criterion. Used to identify one or more LAN devices, value of the DHCP Client Identifier (Option 61) as defined in RFC 2132 [28]. The option value is binary, so an exact match is REQUIRED. An empty value indicates this criterion is not used for conditional serving.	<Empty>	1.4
ClientIDExclude	boolean	W	If False, matching packets are those that match the ClientID entry, if specified. If True, matching packets are those that do not match the ClientID entry, if specified.	False	1.4
UserClassID	string(256)	W	Pool association criterion. Used to identify one or more LAN devices, value of the DHCP User Class Identifier (Option 77) as defined in RFC 3004 [39]. An empty value indicates this criterion is not used for conditional serving.	<Empty>	1.4
UserClassIDExclude	boolean	W	If False, matching packets are those that match the UserClassID entry, if specified. If True, matching packets are those that do not match the UserClassID entry, if specified.	False	1.4
Chaddr	string	W	Pool association criterion. Hardware address (MAC address.) of the physical interface of the DHCP client. An empty value indicates this criterion is not used for conditional serving.	<Empty>	1.4
ChaddrMask	string	W	Bit-mask for the MAC address, where matching of a packet's MAC address with the Chaddr is only to be done for bit positions set to one in the mask. A mask of FF:FF:FF:FF:FF:FF or an empty string indicates all bits of the Chaddr are to be used for conditional serving classification.	<Empty>	1.4
ChaddrExclude	boolean	W	If False, matching packets are those that match the (masked) Chaddr entry, if specified. If True, matching packets are those that do not match the (masked) Chaddr entry, if specified.	False	1.4
LocallyServed	boolean	W	If True, then the local DHCP server will assign an IP address from the specific address pool specified in this object. If False, the DHCP server will send the request to the DHCPServerIPAddress configured for this pool.	True	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
MinAddress	string	W	Specifies first address in the pool to be assigned by the DHCP server on the LAN interface. This parameter is configurable only if UseAllocated WAN is Normal. This parameter MUST have a valid value before this pool can be enabled.	-	1.4
MaxAddress	string	W	Specifies last address in the pool to be assigned by the DHCP server on the LAN interface. This parameter is configurable only if UseAllocated WAN is Normal. This parameter MUST have a valid value before this pool can be enabled.	-	1.4
ReservedAddresses	string(512)	W	Comma-separated list of IP addresses marked reserved from the address allocation pool.	<Empty>	1.4
SubnetMask	string	W	Specifies the client's network subnet mask. This parameter is configurable only if UseAllocated WAN is Normal. This parameter MUST have a valid value before this pool can be enabled.	-	1.4
DNSServers	string(64)	W	Comma-separated list of DNS servers offered to DHCP clients. Support for more than three DNS Servers is OPTIONAL. This parameter is configurable only if UseAllocated WAN is Normal.	-	1.4
DomainName	string(64)	W	Sets the domain name to provide to clients on the LAN interface.	-	1.4
IPRouters	string(64)	W	Comma-separated list of IP addresses of routers on this subnet. Also known as default gateway. Support for more than one Router address is OPTIONAL.	-	1.4
DHCPLeaseTime	int[-1:]	W	Specifies the lease time in seconds of client assigned addresses. A value of -1 indicates an infinite lease.	86400	1.4
UseAllocatedWAN	string	W	Controls whether the MinAddress, MaxAddress, SubnetMask and DNSServers parameters are configurable or are taken from the associated WAN connection. Enumeration of: "Normal" "Passthrough" If Normal, the above-mentioned pool parameters are directly configured by the ACS. If Passthrough, the above-mentioned pool parameters cannot be configured by the ACS. Their values are instead taken from the WAN connection specified by AssociatedConnection.	"Normal"	1.4
AssociatedConnection	string(256)	W	Specifies the connection instance to be used for address allocation if UseAllocatedWAN is set to Passthrough. The content is the full hierarchical parameter name of a WAN-side layer 3 connection object. Example: "InternetGatewayDevice.WANDevice.1.WANConnectionDevice.2.WANPPPConnection.1". If UseAllocatedWAN is Passthrough, this parameter MUST have a valid value before this pool can be enabled.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DHCPServerIPAddress	string	W	IP address of the DHCP server, where the request has to be sent to when there is a conditional match with this pool and LocallyServed is False. If this parameter is not configured, then the DHCP request is dropped.	-	1.4
DHCPStaticAddressNumberOfEntries	unsignedInt	-	The number of entries in the DHCPStaticAddress table.	0	1.4
DHCPOptionNumberOfEntries	unsignedInt	-	The number of entries in the DHCPOption table.	0	1.4
InternetGatewayDevice.LANDevice.{i}.LAN-HostConfigManagement.DHCPConditional-ServingPool.{i}.DHCPStaticAddress.{i}.	object	W	DHCP static address table. Entries in this table correspond to what RFC 2131 [27] calls “manual allocation”, where a client's IP address is assigned by the network administrator, and DHCP is used simply to convey the assigned address to the client. Each instance of this object specifies a hardware address (MAC address) and an IP address within the pool. When serving from this pool, this IP address MUST, if available, be assigned to the DHCP client with this hardware address, and MUST NOT be assigned to any other client. Note that it is possible that an IP address in this table is present in the main pool and/or one or more of the other conditional serving pools, in which case it is possible that such an address will be assigned to a different client.	-	1.4
Enable	boolean	W	Enables or disables the DHCPStaticAddress table entry. Disabling an entry does not return the IP address to the pool.	False	1.4
Chaddr	string	W	Hardware address (MAC address) of the physical interface of the DHCP client. This parameter MUST have a valid value before the table entry can be enabled.	<Empty>	1.4
Yiaddr	string	W	IP address to be assigned by the DHCP server to the DHCP client with the specified hardware address (MAC address). This parameter MUST have a valid value before the table entry can be enabled.	<Empty>	1.4
InternetGatewayDevice.LANDevice.{i}.LAN-HostConfigManagement.DHCPConditional-ServingPool.{i}.DHCPOption.{i}.	object	W	This object specifies the DHCP options that MUST, if enabled, be returned to clients whose DHCP requests are associated with this pool.	-	1.4
Enable	boolean	W	Enables or disables this DHCPOption table entry.	False	1.4
Tag	unsignedInt [1:254]	W	Option tag as defined in RFC 2132 [28].	-	1.4
Value	base64(340)	W	Base64 encoded option value.	<Empty>	1.4
InternetGatewayDevice.LANDevice.{i}.LAN-EthernetInterfaceConfig.{i}.	object	-	This object models an Ethernet LAN connection on a CPE device. This object MUST be implemented for CPE that contain an Ethernet interface on the LAN side.	-	1.0
Enable	boolean	W	Enables or disables this interface.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Status	string	-	Indicates the status of this interface. Enumeration of: "Up" "NoLink" "Error" (OPTIONAL) "Disabled" The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-	1.0
Name	string(16)	-	The name of this layer 2 interface, chosen by the vendor, e.g. "eth0" or "eth0:1".	-	1.4
MACAddress	string	-	The physical address of the interface.	-	1.0
MACAddressControlEnabled	boolean	W	Indicates whether MAC Address Control is enabled or not on this interface. MAC Address Control limits the clients that connect to those that match a list of allowed MAC addresses specified in Internet-GatewayDevice.LANDevice.{i}.LANHostConfig-Management.AllowedMACAddresses.	-	1.0
MaxBitRate	string	W	The maximum upstream and downstream bit rate available to this connection. Enumeration of: "10" "100" "1000" "10000" "Auto"	-	1.0
DuplexMode	string	W	The duplex mode available to this connection. Enumeration of: "Half" "Full" "Auto"	-	1.0
InternetGatewayDevice.LANDevice.{i}.LAN-EthernetInterfaceConfig.{i}.Stats.	object	-	This object contains statistics for an Ethernet LAN interface on a CPE device. Note that these statistics refer to the link layer, not to the physical layer.	-	1.0
BytesSent	unsignedInt	-	The total number of bytes transmitted out of the interface, including framing characters. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
BytesReceived	unsignedInt	-	The total number of bytes received on the interface, including framing characters. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
PacketsSent	unsignedInt	-	The total number of packets transmitted out of the interface. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
PacketsReceived	unsignedInt	-	The total number of packets which were received on this interface. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
ErrorsSent	unsignedInt	-	The total number of outbound packets that could not be transmitted because of errors. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ErrorsReceived	unsignedInt	-	The total number of inbound packets that contained errors preventing them from being deliverable. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnicastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were not addressed to a multicast or broadcast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnicastPacketsReceived	unsignedInt	-	The total number of received packets which were not addressed to a multicast or broadcast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
DiscardPacketsSent	unsignedInt	-	The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
DiscardPacketsReceived	unsignedInt	-	The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
MulticastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were addressed to a multicast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
MulticastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a multicast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
BroadcastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were addressed to a broadcast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
BroadcastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a broadcast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnknownProtoPacketsReceived	unsignedInt	-	The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.LANDevice.{i}.LAN-USBInterfaceConfig.{i}.	object	-	This object models a USB LAN connection on a CPE device. This object MUST be implemented for CPE that contain a USB interface on the LAN side.	-	1.0
Enable	boolean	W	Enables or disables this interface.	-	1.0
Status	string	-	Indicates the status of this interface. Enumeration of: "Up" "NoLink" "Error" (OPTIONAL) "Disabled" The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-	1.0
Name	string(16)	-	The name of this layer 2 interface, chosen by the vendor, e.g. "usb0".	-	1.4
MACAddress	string	-	The physical address of the interface.	-	1.0
MACAddressControlEnabled	boolean	W	Indicates whether MAC Address Control is enabled or not on this interface. MAC Address Control limits the clients that connect to those that match a list of allowed MAC addresses specified in Internet-GatewayDevice.LANDevice.{i}.LANHostConfig-Management.AllowedMACAddresses.	-	1.0
Standard	string(6)	-	USB version supported by the device.	-	1.0
Type	string	-	Type of the USB interface. Enumeration of: "Host" "Hub" "Device"	-	1.0
Rate	string	-	Speed of the USB interface. Enumeration of: "Low" "Full" "High" (USB 2.0)	-	1.0
Power	string	-	Power configuration of the USB interface. Enumeration of: "Self" "Bus" "Unknown"	-	1.0
InternetGatewayDevice.LANDevice.{i}.LAN-USBInterfaceConfig.{i}.Stats.	object	-	This object contains statistics for a USB LAN interface on a CPE device. Note that these statistics refer to the link layer, not to the physical layer.	-	1.0
BytesSent	unsignedInt	-	The total number of bytes transmitted out of the interface, including framing characters. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
BytesReceived	unsignedInt	-	The total number of bytes received on the interface, including framing characters. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
CellsSent	unsignedInt	-	The total number of packets (cells) transmitted out of the interface. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
CellsReceived	unsignedInt	-	The total number of packets (cells) which were received on this interface. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
ErrorsSent	unsignedInt	-	The total number of outbound packets that could not be transmitted because of errors. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
ErrorsReceived	unsignedInt	-	The total number of inbound packets that contained errors preventing them from being deliverable. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnicastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were not addressed to a multicast or broadcast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnicastPacketsReceived	unsignedInt	-	The total number of received packets which were not addressed to a multicast or broadcast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
DiscardPacketsSent	unsignedInt	-	The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
DiscardPacketsReceived	unsignedInt	-	The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
MulticastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were addressed to a multicast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
MulticastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a multicast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
BroadcastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were addressed to a broadcast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
BroadcastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a broadcast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
UnknownProtoPacketsReceived	unsignedInt	-	The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.	object	-	This object models an 802.11 LAN connection on a CPE device. This object MUST be implemented for CPE that contain an 802.11 interface on the LAN side.	-	1.0
Enable	boolean	W	Enables or disables this interface. When there are multiple WLANConfiguration instances, e.g. each instance supports a different 802.11 standard or has a different security configuration, this parameter can be used to control which of the instances are currently enabled.	-	1.0
Status	string	-	Indicates the status of this interface. Enumeration of: "Up" "Error" (OPTIONAL) "Disabled" The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-	1.0
Name	string(16)	-	The name of this layer 2 interface, chosen by the vendor, e.g. "wlan0".	-	1.4
BSSID	string	-	The MAC address of the interface.	-	1.0
MaxBitRate	string(4)	W	The maximum upstream and downstream bit rate available to this connection in Mbps. Either "Auto", or the largest of the OperationalDataTransmitRates values.	-	1.0
Channel	unsignedInt [0:255]	W	The current radio channel used by the connection. To request automatic channel selection, set AutoChannelEnable to True. Whenever AutoChannelEnable is True, the value of the Channel parameter MUST be the channel selected by the automatic channel selection procedure.	-	1.0
AutoChannelEnable	boolean	W	Enable or disable automatic channel selection. Set to False to disable the automatic channel selection procedure, in which case the currently selected channel remains selected. Set to True to enable the automatic channel selection procedure. This procedure MUST automatically select the channel, and MAY also change it subsequently. AutoChannelEnable MUST automatically change to False whenever the channel is manually selected, i.e. whenever the Channel parameter is written. Whenever AutoChannelEnable is True, the value of the Channel parameter MUST be the channel selected by the automatic channel selection procedure.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SSID	string(32)	W	The current service set identifier in use by the connection. The SSID is an identifier that is attached to packets sent over the wireless LAN that functions as a "password" for joining a particular radio network (BSS). Note: If an access point wishes to be known by more than one SSID, it MUST provide a WLANConfiguration instance for each SSID.	-	1.0
BeaconType	string	W	The capabilities that are currently enabled on the access point (and that are announced via beacons if BeaconAdvertisementEnabled is True). Write access to this parameter enables and disables such capabilities. An attempt to set this parameter to one of the REQUIRED (mandatory) values MAY be rejected if (and only if) the requested capability is not available on this WLANConfiguration instance but is available on another WLANConfiguration instance within this Internet Gateway Device. For example, only basic 802.11 might be supported by one virtual AP, and only WPA might be supported by another virtual AP. A value of "None" means that no capabilities are currently enabled on the access point and that no stations will be able to associate with it. Enumeration of: "None" "Basic" "WPA" "11i" (OPTIONAL) "BasicandWPA" (OPTIONAL, OBSOLETED) "Basicand11i' (OPTIONAL, OBSOLETED) "WPAand11i" (OPTIONAL) "BasicandWPAand11i" (OPTIONAL, OBSOLETED) "11i" SHOULD be taken to refer to both the 802.11i specification and to the WPA2 specification (any WPA2-certified device will implement all mandatory parts of the 802.11i standard). The OBSOLETED values are those for Basic + WPA/WPA2 mixed modes, which are not permitted by the WPA specifications.	-	1.0
MACAddressControlEnabled	boolean	W	Indicates whether MAC Address Control is enabled or not on this interface. MAC Address Control limits the clients that connect to those that match a list of allowed MAC addresses specified in Internet-GatewayDevice.LANDevice.{i}.LANHostConfig-Management.AllowedMACAddresses.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Standard	string	-	Indicates which IEEE 802.11 standard this WLANConfiguration instance is configured for. Enumeration of: "a" "b" "g" (b and g clients supported) "g-only" (only g clients supported) "n" Where each value indicates support for only the indicated standard. If the device is configured simultaneously for more than one standard, a separate WLANConfiguration instance MUST be used for each supported standard.	-	1.0
WEPKeyIndex	unsignedInt [1:4]	W	The index of the default WEP key.	-	1.0
KeyPassphrase	string(63)	W	A passphrase from which the WEP keys are to be generated. This parameter is the same as the parameter InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.PreSharedKey.1.KeyPassphrase for the same instance of WLANConfiguration. When either parameter is changed, the value of the other is changed as well. If KeyPassphrase is written, all four WEP keys are immediately generated. The ACS SHOULD NOT set the passphrase and also set the WEP keys directly (the result of doing this is undefined). This MUST either be a valid key length divided by 8, in which case each byte contributes 8 bits to the key, or else MUST consist of Hex digits and be a valid key length divided by 4, in which case each byte contributes 4 bits to the key. Note: If a passphrase is used, all four WEP keys will be the same. When read, this parameter returns an empty string, regardless of the actual value.	-	1.0
WEPEncryptionLevel	string(64)	-	Comma-separated list of the supported key lengths. Each entry in the list is an enumeration of: "Disabled" "40-bit" "104-bit" Any additional vendor-specific values MUST start with the key length in bits. This parameter does not enforce a given encryption level but only indicates capabilities. The WEP encryption level for a given key is inferred from the key length.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
BasicEncryptionModes	string	W	Encryption modes that are available when basic 802.11 is enabled. "WEPEncryption" implies that all wireless clients can use WEP for data encryption. Enumeration of: "None" "WEPEncryption" If this WLANConfiguration instance does not support basic 802.11 then this parameter MUST NOT be present in this instance of the WLANConfiguration object.	-	1.0
BasicAuthenticationMode	string	W	Authentication modes that are available when basic 802.11 is enabled. Enumeration of: "None" (Open authentication) "EAPAuthentication" (OPTIONAL) "SharedAuthentication" (OPTIONAL) If this WLANConfiguration instance does not support basic 802.11 then this parameter MUST NOT be present in this instance of the WLANConfiguration object.	-	1.0
WPAEncryptionModes	string	W	Encryption modes that are available when WPA is enabled. Enumeration of: "WEPEncryption" (DEPRECATED) "TKIPEncryption" "WEPandTKIPEncryption" (DEPRECATED) "AESEncryption" (OPTIONAL) "WEPandAESEncryption" (OPTIONAL, DEPRECATED) "TKIPandAESEncryption" (OPTIONAL) "WEPandTKIPandAESEncryption" (OPTIONAL, DEPRECATED) If this WLANConfiguration instance does not support WPA then this parameter MUST NOT be present in this instance of the WLANConfiguration object. The DEPRECATED values are those that combine WEP with TKIP and/or AES, which is not permitted by the WPA specifications.	-	1.0
WPAAuthenticationMode	string	W	Authentication modes that are available when WPA is enabled. Enumeration of: "PSKAuthentication" "EAPAuthentication" (OPTIONAL) If this WLANConfiguration instance does not support WPA then this parameter MUST NOT be present in this instance of the WLANConfiguration object.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
IEEE11iEncryptionModes	string	W	Encryption modes that are available when 802.11i is enabled. Enumeration of: "WEPEncryption" (DEPRECATED) "TKIPEncryption" (OPTIONAL) "WEPandTKIPEncryption" (DEPRECATED) "AESEncryption" "WEPandAESEncryption" (OPTIONAL, DEPRECATED) "TKIPandAESEncryption" (OPTIONAL) "WEPandTKIPandAESEncryption" (OPTIONAL, DEPRECATED) If this WLANConfiguration instance does not support 802.11i then this parameter MUST NOT be present in this instance of the WLANConfiguration object. "IEEE11i" SHOULD be taken to refer to both the 802.11i specification and to the WPA2 specification (any WPA2-certified device will implement all mandatory parts of the 802.11i standard). The DEPRECATED values are those that combine WEP with TKIP and/or AES, which is not permitted by the WPA2 specifications.	-	1.0
IEEE11iAuthenticationMode	string	W	Authentication modes that are available when 802.11i is enabled. Enumeration of: "PSKAuthentication" "EAPAuthentication" (OPTIONAL) "EAPandPSKAuthentication" (OPTIONAL) If this WLANConfiguration instance does not support 802.11i then this parameter MUST NOT be present in this instance of the WLANConfiguration object. "IEEE11i" SHOULD be taken to refer to both the 802.11i specification and to the WPA2 specification (any WPA2-certified device will implement all mandatory parts of the 802.11i standard).	-	1.0
PossibleChannels	string (1024)	-	The possible radio channels for the wireless standard (a, b or g) and the regulatory domain. Comma-separated list. Ranges in the form "n-m" are permitted. For example, for 802.11b and North America, would be "1-11".	-	1.0
BasicDataTransmitRates	string(256)	W	Comma-separated list of the maximum access point data transmit rates in Mbps for unicast, multicast and broadcast frames. For example, a value of "1,2", indicates that unicast, multicast and broadcast frames can be transmitted at 1 Mbps and 2 Mbps.	-	1.0
OperationalDataTransmitRates	string(256)	W	Comma-separated list of the maximum access point data transmit rates in Mbps for unicast frames (a superset of BasicDataTransmitRates). Given the value of BasicDataTransmitRates from the example above, OperationalDataTransmitRates might be "1,2,5.5,11", indicating that unicast frames can additionally be transmitted at 5.5 Mbps and 11 Mbps.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
PossibleDataTransmitRates	string(256)	-	Comma-separated list of the data transmit rates for unicast frames at which the access point will permit a station to connect (a subset of OperationalData-TransmitRates). Given the values of BasicDataTransmitRates and OperationalDataTransmitRates from the examples above, PossibleDataTransmitRates might be "1,2,5.5", indicating that the AP will only permit connections at 1 Mbps, 2 Mbps and 5.5 Mbps, even though it could theoretically accept connections at 11 Mbps.	-	1.0
InsecureOOBAccessEnabled	boolean	W	Indicates whether insecure write access via mechanisms other than the CPE WAN Management Protocol is permitted to the parameters in this object.	-	1.0
BeaconAdvertisementEnabled	boolean	W	Indicates whether or not the access point is sending out beacons.	-	1.0
SSIDAdvertisementEnabled	boolean	W	Indicates whether or not beacons include the SSID name. This parameter has an effect only if BeaconAdvert-isementEnabled is True.	-	1.4
RadioEnabled	boolean	W	Indicates whether or not the access point radio is enabled.	-	1.0
TransmitPowerSupported	string(64)	-	Comma-separated list of the supported transmit power levels as percentages of full power. Each value MUST be an integer in the range 0 to 100 inclusive. For example, "0,25,50,75,100".	-	1.4
TransmitPower	unsignedInt	W	Indicates the current transmit power level as a percentage of full power. The value MUST be one of the values reported by the TransmitPowerSupp-orted parameter.	-	1.4
AutoRateFallBackEnabled	boolean	W	Indicates whether the access point can automatically reduce the data rate in the event of undue noise or contention.	-	1.0
LocationDescription	string (4096)	W	An XML description of information used to identify the access point by name and physical location. The CPE is not expected to parse this string, but simply to treat it as an opaque string. An empty string indicates no location has been set.	-	1.0
RegulatoryDomain	string(3)	W	802.11d Regulatory Domain String. First two octets are ISO/IEC 3166-1 two-character country code. The third octet is either " " (all environments), "O" (outside) or "I" (inside).	-	1.0
TotalPSKFailures	unsignedInt	-	The number of times pre-shared key (PSK) authentication has failed (relevant only to WPA and 802.11i).	-	1.0
TotalIntegrityFailures	unsignedInt	-	The number of times the MICHAEL integrity check has failed (relevant only to WPA and 802.11i)	-	1.0
ChannelsInUse	string (1024)	-	The channels that the access point determines to be currently in use (including any that it is using itself). Comma-separated list. Ranges in the form "n-m" are permitted.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DeviceOperationMode	string	W	The current access-point operating mode. The OPTIONAL modes permit the AP to be configured as a wireless bridge (to bridge two wired networks), repeater (a bridge that also serves wireless clients), or wireless station. Ad hoc stations are not supported. Enumeration of: "InfrastructureAccessPoint" "WirelessBridge" (OPTIONAL) "WirelessRepeater" (OPTIONAL) "WirelessStation" (OPTIONAL)	-	1.0
DistanceFromRoot	unsignedInt	W	The number of hops from the root access point to the wireless repeater or bridge.	-	1.0
PeerBSSID	string	W	The MAC address of the peer in wireless repeater or bridge mode.	-	1.0
AuthenticationServiceMode	string	W	Indicates whether another service is involved in client authentication (LinkAuthentication for a co-located authentication server; RadiusClient for an external RADIUS server). Enumeration of: "None" "LinkAuthentication" (OPTIONAL) "RadiusClient" (OPTIONAL)	-	1.0
WMMSupported	boolean	-	Indicates whether this interface supports WiFi Multimedia (WMM) Access Categories (AC).	-	1.4
UAPSDSupported	boolean	-	Indicates whether this interface supports WMM Unscheduled Automatic Power Save Delivery (U-APSD). Note: U-APSD support implies WMM support.	-	1.4
WMMEnable	boolean	W	Whether WMM support is currently enabled. When enabled, this is indicated in beacon frames	-	1.4
UAPSDEnable	boolean	W	Whether U-APSD support is currently enabled. When enabled, this is indicated in beacon frames. Note: U-APSD can only be enabled if WMM is also enabled.	-	1.4
TotalBytesSent	unsignedInt	-	The total number of bytes transmitted out of the interface, including framing characters. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
TotalBytesReceived	unsignedInt	-	The total number of bytes received on the interface, including framing characters. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
TotalPacketsSent	unsignedInt	-	The total number of packets transmitted out of the interface. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
TotalPacketsReceived	unsignedInt	-	The total number of packets which were received on this interface. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
TotalAssociations	unsignedInt	-	The number of devices currently associated with the access point. This corresponds to the number of entries in the AssociatedDevice table.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.Stats.	object	-	This object contains statistics for an 802.11 LAN interface on a CPE device. Note that these statistics refer to the link layer, not to the physical layer. Note that this object does not include the total byte and packet statistics, which are, for historical reasons, in the parent object.	-	1.4
ErrorsSent	unsignedInt	-	The total number of outbound packets that could not be transmitted because of errors. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
ErrorsReceived	unsignedInt	-	The total number of inbound packets that contained errors preventing them from being deliverable. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnicastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were not addressed to a multicast or broadcast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnicastPacketsReceived	unsignedInt	-	The total number of received packets which were not addressed to a multicast or broadcast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
DiscardPacketsSent	unsignedInt	-	The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
DiscardPacketsReceived	unsignedInt	-	The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
MulticastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were addressed to a multicast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
MulticastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a multicast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
BroadcastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were addressed to a broadcast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
BroadcastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a broadcast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnknownProtoPacketsReceived	unsignedInt	-	The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.WPS.	object	-	This object contains parameters related to WPS (Wi-Fi Protected Setup) [51] that apply to a CPE acting as an Access Point.	-	1.4
Enable	boolean	W	Enables or disables WPS functionality for this interface.	-	1.4
DeviceName	string(32)	-	User-friendly description of the device. This parameter corresponds directly to the "Device Name" attribute of the WPS specification [51].	-	1.4
DevicePassword	unsignedInt	W	Represents the DevicePassword used (commonly known as PIN). When read, this parameter returns an empty string, regardless of the actual value. This parameter corresponds directly to the "Device Password" attribute of the WPS specification [51].	-	1.4
UUID	string(36)	-	UUID of the device. This is represented as specified in RFC 4122 [48] but omitting the leading "urn:uuid:", e.g. "f81d4fae-7dec-11d0-a765-00a0c91e6bf6". This parameter corresponds directly to the "UUID-E" (enrollee) and "UUID-R" (registrar) attributes of the WPS specification [51]. Note that if the Access Point can act both as an enrollee and as a registrar then UUID-E and UUID-R will be the same as each other.	-	1.4
Version	unsignedInt	-	The Wi-Fi Protected Setup version supported by the device. This parameter corresponds directly to the "Version" attribute of the WPS specification [51].	-	1.4
ConfigMethodsSupported	string	-	Comma-separated list of the WPS configuration methods supported by the device. Each entry in the list is an enumeration of: "USBFlashDrive" "Ethernet" "Label" "Display" "ExternalNFCToken" "IntegratedNFCToken" "NFCInterface" "PushButton" "Keypad" This parameter corresponds directly to the "Config Methods" attribute of the WPS specification [51].	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ConfigMethodsEnabled	string	W	Comma-separated list of the WPS configuration methods enabled on the device. Each entry in the list MUST be a member of the list reported by the ConfigMethodsSupported parameter. This parameter corresponds directly to the "Permitted Config Methods" attribute of the WPS specification [51].	-	1.4
SetupLockedState	string	-	Indicates if the AP Setup mode is enabled for configuration of the AP through an external registrar. The AP Setup mode can be disabled by the user, by the remote management or in case of a brute force attack against the AP's PIN (Wrong PIN provided to AP multiple times). Enumeration of: "Unlocked" "LockedByLocalManagement" "LockedByRemoteManagement" "PINRetryLimitReached" This parameter corresponds directly to the "AP Setup Locked" attribute of the WPS specification [51]. The factory default setting is "Unlocked"	-	1.4
SetupLock	boolean	W	When set to True, the Access Point will refuse to accept new external registrars; already established registrars will continue to be able to add new enrollees (the "SetupLockedState" becomes "LockedByRemoteManagement"). When set to False, the Access Point is enabled for configuration through an external registrar (the "SetupLockedState" becomes "Unlocked"). The factory default setting is False.	-	1.4
ConfigurationState	string	-	Description of the WPS status on the Wireless Access Point side. Enumeration of: "Not configured" (WLAN interface is unconfigured: out-of-the box configuration) "Configured" (WLAN interface is configured) This parameter corresponds directly to the "Wi-Fi Protected Setup State" attribute of the WPS specification [51].	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
LastConfigurationError	string	-	Shows the result of the last external registrar attempt to configure the Access Point. Enumeration of: "NoError" "DecryptionCRCFailure" "SignalTooWeak" "CouldntConnectToRegistrar" "RogueActivitySuspected" "DeviceBusy" "SetupLocked" "MessageTimeout" "RegistrationSessionTimeout" "DevicePasswordAuthFailure" If no external registrar has yet attempted to configure the Access Point, this parameter MUST have the value "No Error". The value of this parameter MUST persist across CPE reboots. This parameter corresponds directly to the "Configuration Error" attribute of the WPS specification [51].	-	1.4
RegistrarNumberOfEntries	unsignedInt	-	Number of entries in the Registrar table: number of Registrars that currently have an association with the Access Point. This parameter corresponds directly to the "Registrar Current" attribute of the WPS specification [51].	-	1.4
RegistrarEstablished	boolean	-	True if the Access Point has ever previously created an association with a Registrar. This parameter corresponds directly to the "Registrar Established" attribute of the WPS specification [51]. The factory default setting is False.	-	1.4
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.WPS.Registrar.{i}.	object	-	This table lists the Registrars associated with the Access Point. This table MUST persist across CPE reboots. The registrar UUID is the unique key. This object corresponds directly to the "Registrar List" attribute of the WPS specification [51].	-	1.4
Enable	boolean	W	If True, the registrar can be used by the Access Point for WPS procedures.	-	1.4
UUID	string(36)	-	UUID of the registrar. This is represented as specified in RFC 4122 [48] but omitting the leading "urn:uuid:", e.g. "f81d4fae-7dec-11d0-a765-00a0c91e6bf6". This parameter corresponds directly to the "UUID-R" attribute of the WPS specification [51].	-	1.4
DeviceName	string(32)	-	Device Name of the registrar. This parameter corresponds directly to the "Device Name" attribute of the WPS specification [51].	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.AssociatedDevice.{i}.	object	-	A table of the devices currently associated with the access point. The size of this table is given by InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.TotalAssociations. This object MUST be implemented for CPE that contain an 802.11 interface on the LAN side.	-	1.0
AssociatedDeviceMACAddress	string	-	The MAC address of an associated device.	-	1.0
AssociatedDeviceIPAddress	string(64)	-	The IP address or DNS name of an associated device.	-	1.0
AssociatedDeviceAuthenticationState	boolean	-	Whether an associated device has authenticated (True) or not (False).	-	1.0
LastRequestedUnicastCipher	string(256)	-	The unicast cipher that was most recently used for a station with a specified MAC address (802.11i only).	-	1.0
LastRequestedMulticastCipher	string(256)	-	The multicast cipher that was most recently used for a station with a specified MAC address (802.11i only).	-	1.0
LastPMKId	string(256)	-	The pairwise master key (PMK) that was most recently used for a station with a specified MAC address (802.11i only).	-	1.0
LastDataTransmitRate	string(4)	-	The data transmit rate that was most recently used for a station with a specified MAC address.	-	1.4
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.WEPKey.{i}.	object	-	This is a table of WEP keys. The size of this table is fixed with exactly 4 entries (with instance numbers 1 through 4). This object MUST be implemented for CPE that contain an 802.11 interface on the LAN side.	-	1.0
WEPKey	string(128)	W	A WEP key expressed as a hexadecimal string. The WEP encryption level for a given key is inferred from the key length, e.g. 10 characters for 40-bit encryption, or 26 characters for 104-bit encryption (keys do not all have to be of the same length, although they will be if the CPE uses KeyPassphrase to generate them). If KeyPassphrase is written, all four WEP keys are immediately generated. The ACS SHOULD NOT set the passphrase and also set the WEP keys directly (the result of doing this is undefined). When read, this parameter returns an empty string, regardless of the actual value.	-	1.0
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.PreSharedKey.{i}.	object	-	This is a table of preshared keys. The size of this table is fixed with exactly 10 entries (with instance numbers 1 through 10). This object MUST be implemented for CPE that contain an 802.11 interface on the LAN side.	-	1.0
PreSharedKey	string(64)	W	A literal WPA PSK expressed as a hexadecimal string. The first table entry contains the default PreSharedKey (InternetGatewayDevice.LAN-Device.{i}.WLANConfiguration.{i}.PreSharedKey.1.-PreSharedKey). If KeyPassphrase is written, the PSK is immediately generated. The ACS SHOULD NOT set the passphrase and also set the PSK directly (the result of doing this is undefined). When read, this parameter returns an empty string, regardless of the actual value.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
KeyPassphrase	string(63)	W	A passphrase from which the PSK is to be generated. The first table entry is the same as the parameter InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.KeyPassphrase for the same instance of WLANConfiguration. When either parameter is changed, the value of the other is changed as well. If KeyPassphrase is written, the PSK is immediately generated. The ACS SHOULD NOT set the passphrase and also set the PSK directly (the result of doing this is undefined). The key is generated as specified by WPA, which uses PBKDF2 from PKCS #5: Password-based Cryptography Specification Version 2.0 (RFC 2898 [38]). When read, this parameter returns an empty string, regardless of the actual value.	-	1.0
AssociatedDeviceMACAddress	string	W	The MAC address associated with a preshared key, or an empty string if no MAC address is associated with the key.	-	1.0
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.APWMMParameter.{i}.	object	-	This is a table of WMM parameters for traffic that originates at the wireless access point, i.e. for outgoing traffic. The size of this table is fixed, with 4 entries (with instance numbers 1 through 4). Instance numbers MUST be assigned as follows: 1: BE AC (Best Effort) 2: BK AC (Background) 3: VI AC (Video) 4: VO AC (Voice)	-	1.4
AIFSN	unsignedInt [2:15]	W	Arbitration Inter Frame Spacing (Number). This is the number of time slots in the arbitration interframe space.	-	1.4
ECWMin	unsignedInt [0:15]	W	Exponent of Contention Window (Minimum). This encodes the Values of CWMin as an exponent: CWMin = 2^ECWMin – 1. For example, if ECWMin is 8, then CWMin is 2^8 – 1, or 255,	-	1.4
ECWMax	unsignedInt [0:15]	W	Exponent of Contention Window (Maximum). This encodes the Values of CWMax as an exponent: CWMax = 2^ECWMax – 1. For example, if ECWMax is 8, then CWMax is 2^8 – 1, or 255,	-	1.4
TXOP	unsignedInt [0:255]	W	Transmit Opportunity, in multiples of 32 microseconds.	-	1.4
AckPolicy	boolean	W	Ack Policy, where False="Do Not Acknowledge" and True="Acknowledge".	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.LANDevice.{i}.WLAN-Configuration.{i}.STAWMMParameter.{i}.	object	-	This is a table of WMM parameters for traffic that originates at the wireless station, i.e. for incoming traffic. The size of this table is fixed, with 4 entries (with instance numbers 1 through 4). Instance numbers MUST be assigned as follows: 1: BE AC (Best Effort) 2: BK AC (Background) 3: VI AC (Video) 4: VO AC (Voice)	-	1.4
AIFSN	unsignedInt [2:15]	W	Arbitration Inter Frame Spacing (Number). This is the number of time slots in the arbitration interframe space.	-	1.4
ECWMin	unsignedInt [0:15]	W	Exponent of Contention Window (Minimum). This encodes the Values of CWMin as an exponent: CWMin = 2^ECWMin – 1. For example, if ECWMin is 8, then CWMin is 2^8 – 1, or 255,	-	1.4
ECWMax	unsignedInt [0:15]	W	Exponent of Contention Window (Maximum). This encodes the Values of CWMax as an exponent: CWMax = 2^ECWMax – 1. For example, if ECWMax is 8, then CWMax is 2^8 – 1, or 255,	-	1.4
TXOP	unsignedInt [0:255]	W	Transmit Opportunity, in multiples of 32 microseconds.	-	1.4
AckPolicy	boolean	W	Ack Policy, where False="Do Not Acknowledge" and True="Acknowledge".	-	1.4
InternetGatewayDevice.LANDevice.{i}.Hosts.	object	-	This object provides information about each of the hosts on the LAN, including those whose IP address was allocated by the CPE using DHCP as well as hosts with statically allocated IP addresses.	-	1.0
HostNumberOfEntries	unsignedInt	-	Number of entries in the Host table.	-	1.0
InternetGatewayDevice.LANDevice.{i}.Hosts.-Host.{i}.	object	-	Host table.	-	1.0
IPAddress	string	-	Current IP Address of the host.	-	1.0
AddressSource	string	-	Indicates whether the IP address of the host was allocated by the CPE using DHCP, was assigned to the host statically, or was assigned using automatic IP address allocation. Enumeration of: "DHCP" "Static" "AutoIP"	-	1.0
LeaseTimeRemaining	int[-1:]	-	DHCP lease time remaining in seconds. A value of -1 indicates an infinite lease. The value MUST be 0 (zero) if the AddressSource is not DHCP.	-	1.0
MACAddress	string	-	MAC address of the host.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Layer2Interface	string(256)	-	This parameter is the full hierarchical parameter name of a particular LAN**InterfaceConfig object or a WLANConfiguration object. For example: InternetGatewayDevice.LANDevice.-1.LANEthernetInterfaceConfig.2. In the case of an embedded Ethernet switch, the Layer2Interface parameter references the LANEthernetInterfaceConfig object that corresponds to the switch port the device is connected to (a LANEthernetInterfaceConfig instance for each switch port). In the case of an embedded WLAN access point, the Layer2Interface parameter references the WLANConfiguration object that corresponds to the SSID the device is connected to (if the access point supports multiple SSIDs, then each SSID is a separate instance).	-	1.4
VendorClassID	string(256)	-	Vendor Class Identifier DHCP option (Option 60) of the host. It MAY be defined when AddressSource is DHCP. An empty value indicates this option is not used.	-	1.4
ClientID	string(256)	-	Client Identifier DHCP option (Option 61) for the specific IP connection of the client. The option value is binary, so an exact match is REQUIRED. It MAY be defined when AddressSource is DHCP. An empty value indicates this option is not used.	-	1.4
UserClassID	string(256)	-	User Class Identifier DHCP option (Option 77) of the host. It MAY be defined when AddressSource is DHCP. An empty value indicates this option is not used.	-	1.4
HostName	string(64)	-	The device's host name or empty string if unknown.	-	1.0
InterfaceType	string	-	Type of physical interface through which this host is connected to the CPE. Enumeration of: "Ethernet" "USB" "802.11" "HomePNA" "HomePlug" "MoCA" "Other"	-	1.0
Active	boolean	-	Whether or not the host is currently present on the LAN. The method of presence detection is a local matter to the CPE. The ability to list inactive hosts is OPTIONAL. If the CPE includes inactive hosts in this table, this variable MUST be set to False for each inactive host. The length of time an inactive host remains listed in this table is a local matter to the CPE.	-	1.0
InternetGatewayDevice.LANInterfaces.	object	-	This object contains LAN-side layer 1/2 interfaces that are not currently connected to the Gateway's IP router and which therefore do not currently reside within a LANDevice instance.	-	1.4
LANEthernetInterfaceNumberOfEntries	unsignedInt	-	Number of instances of LANEthernetInterface-Config in this object.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
LANUSBInterfaceNumberOfEntries	unsignedInt	-	Number of instances of LANUSBInterfaceConfig in this object.	-	1.4
LANWLANConfigurationNumberOfEntries	unsignedInt	-	Number of instances of WLANConfiguration in this object object.	-	1.4
InternetGatewayDevice.LANInterfaces.LAN-EthernetInterfaceConfig.{i}.	object	-	This object models an Ethernet LAN connection on a CPE device. The object definition is identical to that for Internet-GatewayDevice.LANDevice.{i}.LANEthernetInter-faceConfig.{i}.	-	1.4
InternetGatewayDevice.LANInterfaces.LAN-USBInterfaceConfig.{i}.	object	-	This object models a USB LAN connection on a CPE device. The object definition is identical to that for Internet-GatewayDevice.LANDevice.{i}.LANUSBInterface-Config.{i}.	-	1.4
InternetGatewayDevice.LANInterfaces.WLAN-Configuration.{i}.	object	-	This object models an 802.11 LAN connection on a CPE device. The object definition is identical to that for Internet-GatewayDevice.LANDevice{i}.WLANConfigurat-ion.{i}.	-	1.4
InternetGatewayDevice.WANDevice.{i}.	object	-	Each instance contains all objects associated with a particular physical WAN interface.	-	1.0
WANConnectionNumberOfEntries	unsignedInt	-	Number of instances of WANConnectionDevice in this WANDevice.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-CommonInterfaceConfig.	object	-	This object models WAN interface properties common across all connection instances.	-	1.0
EnabledForInternet	boolean	W	Used to enable or disable access to and from the Internet across all connection instances.	-	1.0
WANAccessType	string	-	Specifies the WAN access (modem) type. Enumeration of: "DSL" "Ethernet" "POTS"	-	1.0
Layer1UpstreamMaxBitRate	unsignedInt	-	Specifies the maximum upstream theoretical bit rate for the WAN device in bits per second. This describes the maximum possible rate given the type of interface assuming the best-case operating environment, regardless of the current operating rate. For example, if the physical interface is 100BaseT, this value would be 100000000, regardless of the current operating rate.	-	1.0
Layer1DownstreamMaxBitRate	unsignedInt	-	Specifies the maximum downstream theoretical bit rate for the WAN device in bits per second. This describes the maximum possible rate given the type of interface assuming the best-case operating environment, regardless of the current operating rate. For example, if the physical interface is 100BaseT, this value would be 100000000, regardless of the current operating rate.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
PhysicalLinkStatus	string	-	Indicates the state of the physical connection (link) from WANDevice to a connected entity. Enumeration of: "Up" "Down" "Initializing" "Unavailable"	-	1.0
WANAccessProvider	string(256)	-	Name of the Service Provider providing link connectivity on the WAN.	-	1.0
TotalBytesSent	unsignedInt	-	The cumulative counter for total number of bytes sent upstream across all connection service instances on the WAN device.	-	1.0
TotalBytesReceived	unsignedInt	-	The cumulative counter for total number of bytes received downstream across all connection service instances on the WAN device.	-	1.0
TotalPacketsSent	unsignedInt	-	The cumulative counter for total number of packets (IP or PPP) sent upstream across all connection service instances on the WAN device.	-	1.0
TotalPacketsReceived	unsignedInt	-	The cumulative counter for total number of packets (IP or PPP) received downstream across all connection service instances on the WAN device.	-	1.0
MaximumActiveConnections	unsignedInt	-	Indicates the maximum number of active connections the CPE can simultaneously support.	-	1.0
NumberOfActiveConnections	unsignedInt	-	Number of WAN connection service instances currently active on this WAN interface.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-CommonInterfaceConfig.Connection.{i}.	object	-	Active connection table.	-	1.0
ActiveConnectionDeviceContainer	string(256)	-	Specifies a WAN connection device object associated with this connection instance. The content is the full hierarchical parameter name of the WAN connection device. Example: "Internet-GatewayDevice.WANDevice.1.WANConnection-Device.2".	-	1.0
ActiveConnectionServiceID	string(256)	-	Specifies a WAN connection object associated with this connection instance. The content is the full hierarchical parameter name of the layer 3 connection object. Example: "InternetGateway-Device.WANDevice.1.WANConnectionDevice.2.-WANPPPConnection.1".	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-DSLInterfaceConfig.	object	-	This object models physical layer properties specific to a single physical connection of a DSL modem used for Internet access on a CPE. This object is intended for a CPE with a DSL modem WAN interface, and is exclusive of any other WAN*InterfaceConfig object within a given WAN-Device instance.	-	1.0
Enable	boolean	W	Enables or disables the link.	-	1.0
Status	string	-	Status of the DSL physical link. Enumeration of: "Up" "Initializing" "EstablishingLink" "NoSignal" "Error" (OPTIONAL) "Disabled" The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
LinkEncapsulationSupported	string	-	Indicates which link encapsulation standards and recommendations are supported by the B-NT. Comma-separated list of strings. Each element in the list is one of: "G.992.3_Annex_K_ATM" "G.992.3_Annex_K_PTM" "G.993.2_Annex_K_ATM" "G.993.2_Annex_K_PTM" "G.994.1" (Auto)	-	1.4
LinkEncapsulationRequested	string	W	Indicates the link encapsulation standard requested by the B-NT. The value MUST be one of the standards reported by the LinkEncapsulation-Supported parameter.	-	1.4
LinkEncapsulationUsed	string	-	Indicates the link encapsulation standard that the B-NT is using for the connection. Enumeration of: "G.992.3_Annex_K_ATM" "G.992.3_Annex_K_PTM" "G.993.2_Annex_K_ATM" "G.993.2_Annex_K_PTM" When the standard identifies ATM encapsulation then the InternetGatewayDevice.WANDevice.{i}.-WANConnectionDevice.{i}. WANDSLLinkConfig object MUST be used. When the standard identifies PTM encapsulation then the InternetGatewayDevice.WANDevice.{i}.-WANConnectionDevice.{i}. WANPTMLinkConfig object MUST be used.	-	1.4
ModulationType	string	-	Indicates the type of modulation used on the connection. Enumeration of: "ADSL_G.dmt" "ADSL_G.lite" "ADSL_G.dmt.bis" "ADSL_re-adsl" "ADSL_2plus" "ADLS_four" "ADSL_ANSI_T1.413" "G.shdsl" "IDSL" "HDSL" "SDSL" "VDSL" This parameter, which was inherited from WANDSLLinkConfig, is DEPRECATED because it is in general not clear which standards correspond to which of the the above enumerated values. It is RECOMMENDED that the StandardUsed parameter be used to indicate which standard is in use.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
StandardsSupported	string	-	Indicates which DSL standards and recommend-ations are supported by the B-NT. Comma-separated list of strings. Each element in the list is one of: "G.992.1_Annex_A" "G.992.1_Annex_B" "G.992.1_Annex_C" "T1.413" "T1.413i2" "ETSI_101_388" "G.992.2" "G.992.3_Annex_A" "G.992.3_Annex_B" "G.992.3_Annex_C" "G.992.3_Annex_I" "G.992.3_Annex_J" "G.992.3_Annex_L" "G.992.3_Annex_M" "G.992.4" "G.992.5_Annex_A" "G.992.5_Annex_B" "G.992.5_Annex_C" "G.992.5_Annex_I" "G.992.5_Annex_J" "G.992.5_Annex_M" "G.993.1" "G.993.1_Annex_A" "G.993.2_Annex_A" "G.993.2_Annex_B" "G.993.2_Annex_C"	-	1.4
StandardUsed	string	-	Indicates the standard that the B-NT is using for the connection. The value MUST be a member of the list reported by the StandardsSupported parameter.	-	1.4
LineEncoding	string	-	The line encoding method used in establishing the Layer 1 DSL connection between the CPE and the DSLAM. Note: Generally speaking, this variable does not change after provisioning. Enumeration of: "DMT" "CAP" "2B1Q" "43BT" "PAM" "QAM"	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
AllowedProfiles	string	-	Indicates which VDSL2 profiles are allowed on the line. Comma-separated list of strings. Each element in the list is one of: "8a" "8b" "8c" "8d" "12a" "12b" "17a" "17b" "30a" Note: In G.997.1, this parameter is called PROFILES. See ITU-T Recommendation G.997.1. Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be an empty string.	-	1.4
CurrentProfile	string	-	Indicates which VDSL2 profile is currently in use on the line. The value MUST be a member of the list reported by the AllowedProfiles parameter, or an empty string as noted below. Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be an empty string.	-	1.4
PowerManagementState	string	-	The power management state of the line. Enumeration of: "L0" "L1" "L3" "L4"	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SuccessFailureCause	unsignedInt	-	The success failure cause of the initialization. An enumeration of the following integer values: 0: Successful 1: Configuration error. This error occurs with inconsistencies in configuration parameters, e.g. when the line is initialized in an xDSL Transmission system where an xTU does not support the configured Maximum Delay or the configured Minimum or Maximum Data Rate for one or more bearer channels. 2: Configuration not feasible on the line. This error occurs if the Minimum Data Rate cannot be reached on the line with the Minimum Noise Margin, Maximum PSD level, Maximum Delay and Maximum Bit Error Ratio for one or more bearer channels. 3: Communication problem. This error occurs, for example, due to corrupted messages or bad syntax messages or if no common mode can be selected in the G.994.1 handshaking procedure or due to a timeout.. 4: No peer xTU detected. This error occurs if the peer xTU is not powered or not connected or if the line is too long to allow detection of a peer xTU. 5: Any other or unknown Initialization Failure cause.	-	1.4
LastStateTransmittedDownstream	unsignedInt	-	This parameter represents the last successful transmitted initialization state in the downstream direction in the last full initialization performed on the line. Initialization states are defined in the individual xDSL Recommendations and are counted from 0 (if G.994.1 is used) or 1 (if G.994.1 is not used) up to Showtime. This parameter needs to be interpreted along with the xDSL Transmission System. This parameter is available only when, after a failed full initialization, the line diagnostics procedures are activated on the line.	-	1.4
LastStateTransmittedUpstream	unsignedInt	-	This parameter represents the last successful transmitted initialization state in the upstream direction in the last full initialization performed on the line. Initialization states are defined in the individual xDSL Recommendations and are counted from 0 (if G.994.1 is used) or 1 (if G.994.1 is not used) up to Showtime. This parameter needs to be interpreted along with the xDSL Transmission System. This parameter is available only when, after a failed full initialization, the line diagnostics procedures are activated on the line.	-	1.4
UPBOKLE	unsignedInt [0:1280]	-	This parameter contains the estimated electrical loop length expressed in dB at 1MHz, kl_e (see O-UPDATE in § 12.2.4.2.1.2/G.993.2). The value SHALL be coded as an unsigned 16 bit number in the range 0 (coded as 0) to 128 dB (coded as 1280) in steps of 0.1 dB.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
MREFPSDds	base64(196)	-	This parameter SHALL contain the set of breakpoints exchanged in the MREFPSDds fields of the O-PRM message of G.993.2. Base64 encoded of the binary representation defined in Table 12-19/G.993.2 (maximum length is 145 octets, which requires 196 bytes for Base64 encoding).	-	1.4
MREFPSDus	base64(196)	-	This parameter SHALL contain the set of breakpoints exchanged in the MREFPSDus fields of the R-PRM message of G.993.2. Base64 encoded of the binary representation defined in Table 12-19/G.993.2 (maximum length is 145 octets, which requires 196 bytes for Base64 encoding).	-	1.4
LIMITMASK	unsignedInt	-	Indicates the enabled VDSL2 Limit PSD mask of the selected PSD mask class. Bit mask as specified in ITU-T Recommendation G.997.1. Note: For a VDSL2-capable multimode device operating in a mode other than VDSL2, the value of this parameter SHOULD be set to 0.	-	1.4
US0MASK	unsignedInt	-	Indicates the allowed VDSL2 US0 PSD masks for Annex A operation. Bit mask as specified in see ITU-T Recommendation G.997.1. Note: For a VDSL2-capable multimode device operating in a mode other than VDSL2, the value of this parameter SHOULD be set to 0.	-	1.4
DataPath	string	-	Indicates whether the data path is fast (lower latency) or interleaved (lower error rate). Enumeration of: "Interleaved" "Fast" "None" Note1: This parameter is only applicable to G.992.1. Note2: For an ADSL1-capable multimode device operating in a mode other than ADSL1, the value of this parameter SHOULD be set to "None".	-	1.0
InterleaveDepth	unsignedInt	-	ADSL1 Interleaved depth. This variable is only applicable to ADSL1 and only if DataPath = Interleaved. Otherwise, the value of this parameter MUST be zero.	-	1.0
LPATH	unsignedInt	-	Reports the index of the latency path supporting the bearer channel. For single-latency connections, LPATH = 0. Note: See ITU-T Recommendation G.997.1.	-	1.4
INTLVDEPTH	int	-	Reports the interleaver depth D for the latency path indicated in LPATH. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to -1.	-	1.4
INTLVBLOCK	int	-	Reports the interleaver block length in use on the latency path indicated in LPATH. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to -1.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ActualInterleavingDelay	unsignedInt	-	Reports the actual delay, in milliseconds, of the latency path due to interleaving. Note: In G.997.1, this parameter is called "Actual Interleaving Delay." See ITU-T Recommendation G.997.1.	-	1.4
ACTINP	int	-	Reports the actual impulse noise protection (INP) provided by the latency path indicated in LPATH. The value is the actual INP in the L0 (i.e., Showtime) state. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to -1.	-	1.4
INPREPORT	boolean	-	Reports whether the value reported in ACTINP was computed assuming the receiver does not use erasure decoding. Valid values are 0 (computed per the formula assuming no erasure decoding) and 1 (computed by taking into account erasure decoding capabilities of receiver). Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to False.	-	1.4
NFEC	int	-	Reports the size, in octets, of the Reed-Solomon codeword in use on the latency path indicated in LPATH. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to -1.	-	1.4
RFEC	int	-	Reports the number of redundancy bytes per Reed-Solomon codeword on the latency path indicated in LPATH. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to -1.	-	1.4
LSYMB	int	-	Reports the number of bits per symbol assigned to the latency path indicated in LPATH. This value does not include overhead due to trellis coding. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to -1.	-	1.4
TRELLISds	int	-	Reports whether trellis coding is enabled in the downstream direction. A value of 1 indicates that trellis coding is in use, and a value of 0 indicates that the trellis is disabled. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to -1.	-	1.4
TRELLISus	int	-	Reports whether trellis coding is enabled in the upstream direction. A value of 1 indicates that trellis coding is in use, and a value of 0 indicates that the trellis is disabled. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to -1.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ACTSNRMODEds	unsignedInt	-	Reports whether the OPTIONAL virtual noise mechanism is in use in the downstream direction. A value of 1 indicates the virtual noise mechanism is not in use, and a value of 2 indicates the virtual noise mechanism is in use. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
ACTSNRMODEus	unsignedInt	-	Reports whether the OPTIONAL virtual noise mechanism is in use in the upstream direction. A value of 1 indicates the virtual noise mechanism is not in use, and a value of 2 indicates the virtual noise mechanism is in use. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
VirtualNoisePSDds	base64(132)	-	Reports the virtual noise PSD for the downstream direction. Base64 encoded of the binary representation defined in G.997.1 by the parameter called TXREFVNds (maximum length is 97 octets, which requires 132 bytes for Base64 encoding). See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to an empty string.	-	1.4
VirtualNoisePSDus	base64(68)	-	Reports the virtual noise PSD for the upstream direction. Base64 encoded of the binary representation defined in G.997.1by the parameter called TXREFVNus (maximum length is 49 octets, which requires 68 bytes for Base64 encoding). See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set an empty string.	-	1.4
ACTUALCE	unsignedInt	-	Reports the actual cyclic extension, as the value of m, in use for the connection. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 99.		1.4
LineNumber	int[1:]	-	Signifies the line pair that the modem is using to connection. LineNumber = 1 is the innermost pair.	-	1.0
UpstreamCurrRate	unsignedInt	-	The current physical layer aggregate data rate (expressed in Kbps) of the upstream DSL connection. Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, it MUST have the value 4294967295 (the maximum for its data type).	-	1.0
DownstreamCurrRate	unsignedInt	-	The current physical layer aggregate data rate (expressed in Kbps) of the downstream DSL connection. Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, it MUST have the value 4294967295 (the maximum for its data type).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
UpstreamMaxRate	unsignedInt	-	The current attainable rate (expressed in Kbps) of the upstream DSL channel. Note: This parameter is related to the G.997.1 parameter ATTNDRus, which is measured in bits/s. See ITU-T Recommendation G.997.1.	-	1.0
DownstreamMaxRate	unsignedInt	-	The current attainable rate (expressed in Kbps) of the downstream DSL channel. Note: This parameter is related to the G.997.1 parameter ATTNDRds, which is measured in bits/s. See ITU-T Recommendation G.997.1.	-	1.0
UpstreamNoiseMargin	int	-	The current signal-to-noise ratio margin (expressed in 0.1 dB) in the upstream direction. Note: In G.997.1, this parameter is called SNRMus. See ITU-T Recommendation G.997.1.	-	1.0
DownstreamNoiseMargin	int	-	The current signal-to-noise ratio margin (expressed in 0.1 dB) in the downstream direction. Note: In G.997.1, this parameter is called SNRMds. See ITU-T Recommendation G.997.1.	-	1.0
SNRMpbus	string(24)	-	The current signal-to-noise ratio margin of each upstream band. Comma-separated list of values. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4
SNRMpbds	string(24)	-	The current signal-to-noise ratio margin of each band. Comma-separated list of values. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4
INMIATOds	unsignedInt [3:511]	-	The Impulse Noise Monitoring (INM) Inter Arrival Time (IAT) Offset, measured in DMT symbols, that the xTU receiver uses to determine in which bin of the IAT histogram the IAT is reported. Note: In G.997.1, this parameter is called INMIATO. See ITU-T Recommendation G.997.1.	-	1.4
INMIATSds	unsignedInt [0:7]	-	The Impulse Noise Monitoring (INM) Inter Arrival Time (IAT) Step that the xTU receiver uses to determine in which bin of the IAT histogram the IAT is reported. Note: In G.997.1, this parameter is called INMIATS. See ITU-T Recommendation G.997.1.	-	1.4
INMCCds	unsignedInt [0:64]	-	The Impulse Noise Monitoring (INM) Cluster Continuation value, measured in DMT symbols, that the xTU receiver uses in the cluster indication process. Note: In G.997.1, this parameter is called INMCC. See ITU-T Recommendation G.997.1.	-	1.4
INMINPEQMODEds	unsignedInt [0:3]	-	The Impulse Noise Monitoring (INM) Equivalent Impulse Noise Protection (INP) Mode that the xTU receiver uses in the computation of the Equivalent INP. Note: In G.997.1, this parameter is called INM_INPEQ_MODE. See ITU-T Recommendation G.997.1.	-	1.4
UpstreamAttenuation	int	-	The current upstream signal loss (expressed in 0.1 dB).	-	1.0
DownstreamAttenuation	int	-	The current downstream signal loss (expressed in 0.1 dB).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
UpstreamPower	int	-	The current output power at the CPE's DSL interface (expressed in 0.1 dBmV).	-	1.0
DownstreamPower	int	-	The current received power at the CPE's DSL interface (expressed in 0.1 dBmV).	-	1.0
ATURVendor	string(8)	-	ATU-R vendor identifier as defined in G.994.1 and T1.413. In the case of G.994.1 this corresponds to the four-octet provider code, which MUST be represented as eight hexadecimal digits. Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, it MUST have the value “00000000”.	-	1.0
ATURCountry	string(4)	-	T.35 country code of the ATU-R vendor as defined in G.994.1, where the two-octet value defined in G.994.1 MUST be represented as four hexadecimal digits. Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, it MUST have the value “0000”.	-	1.0
ATURANSIStd	unsignedInt	-	ATU-R T1.413 Revision Number as defined in T1.413 Issue 2. When T1.413 modulation is not in use, the parameter value SHOULD be 0.	-	1.0
ATURANSIRev	unsignedInt	-	ATU-R Vendor Revision Number as defined in T1.413 Issue 2. When T1.413 modulation is not in use, the parameter value SHOULD be 0.	-	1.0
ATUCVendor	string(8)	-	ATU-C vendor identifier as defined in G.994.1 and T1.413. In the case of G.994.1 this corresponds to the four-octet provider code, which MUST be represented as eight hexadecimal digits. Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, it MUST have the value “00000000”.	-	1.0
ATUCCountry	string(4)	-	T.35 country code of the ATU-C vendor as defined in G.994.1, where the two-octet value defined in G.994.1 MUST be represented as four hexadecimal digits. Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, it MUST have the value “0000”.	-	1.0
ATUCANSIStd	unsignedInt	-	ATU-C T1.413 Revision Number as defined in T1.413 Issue 2. When T1.413 modulation is not in use, the parameter value SHOULD be 0.	-	1.0
ATUCANSIRev	unsignedInt	-	ATU-C Vendor Revision Number as defined in T1.413 Issue 2. When T1.413 modulation is not in use, the parameter value SHOULD be 0.	-	1.0
TotalStart	unsignedInt	-	Number of seconds since the beginning of the period used for collection of Total statistics. Statistics SHOULD continue to be accumulated across CPE reboots, though this might not always be possible.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ShowtimeStart	unsignedInt	-	Number of seconds since the most recent DSL Showtime – the beginning of the period used for collection of Showtime statistics. Showtime is defined as successful completion of the DSL link establishment process. The Showtime statistics are those collected since the most recent establishment of the DSL link.	-	1.0
LastShowtimeStart	unsignedInt	-	Number of seconds since the second most recent DSL Showtime—the beginning of the period used for collection of LastShowtime statistics. If the CPE has not retained information about the second most recent Showtime (e.g., on reboot), the start of LastShowtime statistics MAY temporarily coincide with the start of Showtime statistics.	-	1.0
CurrentDayStart	unsignedInt	-	Number of seconds since the beginning of the period used for collection of CurrentDay statistics. The CPE MAY align the beginning of each CurrentDay interval with days in the UTC time zone, but does not have to do so. Statistics SHOULD continue to be accumulated across CPE reboots, though this might not always be possible.	-	1.0
QuarterHourStart	unsignedInt	-	Number of seconds since the beginning of the period used for collection of QuarterHour statistics. The CPE MAY align the beginning of each QuarterHour interval with real-time quarter-hour intervals, but does not have to do so. Statistics SHOULD continue to be accumulated across CPE reboots, though this might not always be possible.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-DSLInterfaceConfig.TestParams.	object	-	This object contains the DSL test parameters that are available during the L0 (i.e., Showtime) state.	-	1.4
HLOGGds	unsignedInt	-	Number of sub-carriers per sub-carrier group in the downstream direction for HLOGpsds. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
HLOGGus	unsignedInt	-	Number of sub-carriers per sub-carrier group in the upstream direction for HLOGpsus. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
HLOGpsds	string(2559)	-	Downstream logarithmic channel characteristics per sub-carrier group. Comma-separated list of values. The maximum number of elements is 256 for G.992.3, and 512 for G.992.5. For G.993.2, the number of elements will depend on the value of HLOGGds but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to “None.” Note: HLOGpsds is measured during initialization and is not updated during Showtime.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
HLOGpsus	string(2559)	-	Upstream logarithmic channel characteristics per sub-carrier group. Comma-separated list of values. The maximum number of elements is 64 for G.992.3 and G.992.5. For G.993.2, the number of elements will depend on the value of HLOGGus but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None." Note: HLOGpsus is measured during initialization and is not updated during Showtime.	-	1.4
HLOGMTds	unsignedInt	-	Indicates the number of symbols over which HLOGpsds was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
HLOGMTus	unsignedInt	-	Indicates the number of symbols over which HLOGpsus was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
QLNGds	unsignedInt	-	Number of sub-carriers per sub-carrier group in the downstream direction for QLNpsds. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
QLNGus	unsignedInt	-	Number of sub-carriers per sub-carrier group in the upstream direction for QLNpsus. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
QLNpsds	string(2047)	-	Downstream quiet line noise per subcarrier group. Comma-separated list of values. The maximum number of elements is 256 for G.992.3 and G.992.5. For G.993.2, the number of elements will depend on the value of QLNGds but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None." Note: QLNpsds is measured during initialization and is not updated during Showtime.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
QLNpsus	string(2047)	-	Upstream quiet line noise per subcarrier group. Comma-separated list of values. The maximum number of elements is 64 for G.992.3 and G.992.5. For G.993.2, the number of elements will depend on the value of QLNGus but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None." Note: QLNpsus is measured during initialization and is not updated during Showtime.	-	1.4
QLNMTds	unsignedInt	-	Indicates the number of symbols over which QLNpsds was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
QLNMTus	unsignedInt	-	Indicates the number of symbols over which QLNpsus was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
SNRGds	unsignedInt	-	Number of sub-carriers per sub-carrier group in the downstream direction for SNRpsds. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
SNRGus	unsignedInt	-	Number of sub-carriers per sub-carrier group in the upstream direction for SNRpsus. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
SNRpsds	string(2047)	-	Downstream SNR per subcarrier group. Comma-separated list of values. The maximum number of elements is 256 for G.992.3, and 512 for G.992.5. For G.993.2, the number of elements will depend on the value of SNRGds but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None." Note: SNRpsds is first measured during initialization and is updated during Showtime.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SNRpsus	string(2047)	-	Upstream SNR per subcarrier group. Comma-separated list of values. The maximum number of elements is 64 for G.992.3 and G.992.5. For G.993.2, the number of elements will depend on the value of SNRGus but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None." Note: SNRpsus is first measured during initialization and is updated during Showtime.	-	1.4
SNRMTds	unsignedInt	-	Indicates the number of symbols over which SNRpsds was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
SNRMTus	unsignedInt	-	Indicates the number of symbols over which SNRpsus was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
LATNds	string(24)	-	Downstream line attenuation per usable band, as computed during initialization. Comma-separated list of values. Number of elements is dependent on the number of downstream bands but will exceed one only for G.993.2. Interpretation of LATNpbds is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4
LATNus	string(24)	-	Upstream line attenuation per usable band, as computed during initialization. Comma-separated list of values. Number of elements is dependent on the number of upstream bands but will exceed one only for G.993.2. Interpretation of LATNpbus is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4
SATNds	string(24)	-	Downstream signal attenuation per usable band, as computed during the L0 (i.e., Showtime) state. Comma-separated list of values. Number of elements is dependent on the number of downstream bands but will exceed one only for G.993.2. Interpretation of SATNpbds is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4
SATNus	string(24)	-	Upstream signal attenuation per usable band, as computed during the L0 (i.e., Showtime) state. Comma-separated list of values. Number of elements is dependent on the number of downstream bands but will exceed one only for G.993.2. Interpretation of SATNpbus is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.	object	-	This object contains statistics for a WAN DSL physical interface.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.Total.	object	-	This object contains DSL total statistics.	-	1.0
ReceiveBlocks	unsignedInt	-	Total number of successfully received blocks, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
TransmitBlocks	unsignedInt	-	Total number of successfully transmitted blocks, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
CellDelin	unsignedInt	-	Total number of cell-delineation errors (total seconds with NCD or LCD failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
LinkRetrain	unsignedInt	-	Total number of link-retrain errors (Full Initialization Count as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
InitErrors	unsignedInt	-	Total number of initialization errors (LINIT failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LInit	boolean	-	LInit is a flag to signal that a failure occurred as defined in G.997.1.	-	1.4
InitTimeouts	unsignedInt	-	Total number of initialization timeout errors.	-	1.0
LossOfFraming	unsignedInt	-	Total number of loss-of-framing errors (LOF failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LOF	boolean		This parameter corresponds to LOF as defined in ITU-T Rec. G.997.1. LOF is a flag to signal that a failure occurred.	-	1.4
ErroredSecs	unsignedInt	-	Total number of errored seconds (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCErroredSecs	unsignedInt	-	Total number of errored seconds detected by the ATU-C (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
SeverelyErroredSecs	unsignedInt	-	Total number of severely errored seconds (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ATUCSeverelyErroredSecs	unsignedInt	-	Total number of severely errored seconds detected by the ATU-C (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
FECErrors	unsignedInt	-	Total number of FEC errors detected (FEC-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCFECErrors	unsignedInt	-	Total number of FEC errors detected by the ATU-C (FEC-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
HECErrors	unsignedInt	-	Total number of HEC errors detected (HEC-P as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCHECErrors	unsignedInt	-	Total number of HEC errors detected by the ATU-C (HEC-PFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
CRCErrors	unsignedInt	-	Total number of CRC errors detected (CV-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCCRCErrors	unsignedInt	-	Total number of CRC errors detected by the ATU-C (CV-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-DSLInterfaceConfig.Stats.Showtime.	object	-	This object contains DSL statistics accumulated since the most recent DSL Showtime.	-	1.0
ReceiveBlocks	unsignedInt	-	Number of successfully received blocks since the most recent DSL Showtime, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
TransmitBlocks	unsignedInt	-	Number of successfully transmitted blocks since the most recent DSL Showtime, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
CellDelin	unsignedInt	-	Number of cell-delineation errors since the most recent DSL Showtime (total seconds with NCD or LCD failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
LinkRetrain	unsignedInt	-	Number of link-retrain errors since the most recent DSL Showtime (Full Initialization Count as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
InitErrors	unsignedInt	-	Number of initialization errors since the most recent DSL Showtime (LINIT failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LInit	boolean	-	LInit is a flag to signal that a failure occurred as defined in G.997.1.	-	1.4
InitTimeouts	unsignedInt	-	Number of initialization timeout errors since the most recent DSL Showtime.	-	1.0
LossOfFraming	unsignedInt	-	Number of loss-of-framing errors since the most recent DSL Showtime (LOF failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LOF	boolean		This parameter corresponds to LOF as defined in ITU-T Rec. G.997.1. LOF is a flag to signal that a failure occurred.	-	1.4
ErroredSecs	unsignedInt	-	Number of errored seconds since the most recent DSL Showtime (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCErroredSecs	unsignedInt	-	Number of errored seconds since the most recent DSL Showtime detected by the ATU-C (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
SeverelyErroredSecs	unsignedInt	-	Number of severely errored seconds since the most recent DSL Showtime (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ATUCSeverelyErroredSecs	unsignedInt	-	Number of severely errored seconds since the most recent DSL Showtime detected by the ATU-C (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
FECErrors	unsignedInt	-	Number of FEC errors detected since the most recent DSL Showtime (FEC-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCFECErrors	unsignedInt	-	Number of FEC errors detected by the ATU-C since the most recent DSL Showtime (FEC-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
HECErrors	unsignedInt	-	Number of HEC errors detected since the most recent DSL Showtime (HEC-P as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCHECErrors	unsignedInt	-	Number of HEC errors detected by the ATU-C since the most recent DSL Showtime (HEC-PFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
CRCErrors	unsignedInt	-	Number of CRC errors detected since the most recent DSL Showtime (CV-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCCRCErrors	unsignedInt	-	Number of CRC errors detected by the ATU-C since the most recent DSL Showtime (CV-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WAN-DSLInterfaceConfig.Stats.LastShowtime.	object	-	This object contains DSL statistics accumulated since the second most recent DSL Showtime.	-	1.0
ReceiveBlocks	unsignedInt	-	Number of successfully received blocks since the second most recent DSL Showtime, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
TransmitBlocks	unsignedInt	-	Number of successfully transmitted blocks since the second most recent DSL Showtime, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
CellDelin	unsignedInt	-	Number of cell-delineation errors since the second most recent DSL Showtime (total seconds with NCD or LCD failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
LinkRetrain	unsignedInt	-	Number of link-retrain errors since the second most recent DSL Showtime (Full Initialization Count as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
InitErrors	unsignedInt	-	Number of initialization errors since the second most recent DSL Showtime (LINIT failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LInit	boolean	-	LInit is a flag to signal that a failure occurred as defined in G.997.1.	-	1.4
InitTimeouts	unsignedInt	-	Number of initialization timeout errors since the second most recent DSL Showtime.	-	1.0
LossOfFraming	unsignedInt	-	Number of loss-of-framing errors since the second most recent DSL Showtime (LOF failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LOF	boolean		This parameter corresponds to LOF as defined in ITU-T Rec. G.997.1. LOF is a flag to signal that a failure occurred.	-	1.4
ErroredSecs	unsignedInt	-	Number of errored seconds since the second most recent DSL Showtime (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCErroredSecs	unsignedInt	-	Number of errored seconds since the second most recent DSL Showtime detected by the ATU-C (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SeverelyErroredSecs	unsignedInt	-	Number of severely errored seconds since the second most recent DSL Showtime (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCSeverelyErroredSecs	unsignedInt	-	Number of severely errored seconds since the second most recent DSL Showtime detected by the ATU-C (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
FECErrors	unsignedInt	-	Number of FEC errors detected since the second most recent DSL Showtime (FEC-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCFECErrors	unsignedInt	-	Number of FEC errors detected by the ATU-C since the second most recent DSL Showtime (FEC-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
HECErrors	unsignedInt	-	Number of HEC errors detected since the second most recent DSL Showtime (HEC-P as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCHECErrors	unsignedInt	-	Number of HEC errors detected by the ATU-C since the second most recent DSL Showtime (HEC-PFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
CRCErrors	unsignedInt	-	Number of CRC errors detected since the second most recent DSL Showtime (CV-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ATUCCRCErrors	unsignedInt	-	Number of CRC errors detected by the ATU-C since the second most recent DSL Showtime (CV-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-DSLInterfaceConfig.Stats.CurrentDay.	object	-	This object contains DSL statistics accumulated during the current day.	-	1.0
ReceiveBlocks	unsignedInt	-	Number of successfully received blocks during the current day, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
TransmitBlocks	unsignedInt	-	Number of successfully transmitted blocks during the current day, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
CellDelin	unsignedInt	-	Number of cell-delineation errors during the current day (total seconds with NCD or LCD failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
LinkRetrain	unsignedInt	-	Number of link-retrain errors during the current day (Full Initialization Count as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
InitErrors	unsignedInt	-	Number of initialization errors during the current day (LINIT failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LInit	boolean	-	LInit is a flag to signal that a failure occurred as defined in G.997.1.	-	1.4
InitTimeouts	unsignedInt	-	Number of initialization timeout errors during the current day.	-	1.0
LossOfFraming	unsignedInt	-	Number of loss-of-framing errors during the current day (LOF failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LOF	boolean		This parameter corresponds to LOF as defined in ITU-T Rec. G.997.1. LOF is a flag to signal that a failure occurred.	-	1.4
ErroredSecs	unsignedInt	-	Number of errored seconds during the current day (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ATUCErroredSecs	unsignedInt	-	Number of errored seconds during the current day detected by the ATU-C (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
SeverelyErroredSecs	unsignedInt	-	Number of severely errored seconds during the current day (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCSeverelyErroredSecs	unsignedInt	-	Number of severely errored seconds during the current day detected by the ATU-C (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
FECErrors	unsignedInt	-	Number of FEC errors detected during the current day (FEC-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCFECErrors	unsignedInt	-	Number of FEC errors detected by the ATU-C during the current day (FEC-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
HECErrors	unsignedInt	-	Number of HEC errors detected during the current day (HEC-P as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCHECErrors	unsignedInt	-	Number of HEC errors detected by the ATU-C during the current day (HEC-PFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
CRCErrors	unsignedInt	-	Number of CRC errors detected during the current day (CV-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCCRCErrors	unsignedInt	-	Number of CRC errors detected by the ATU-C during the current day (CV-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-DSLInterfaceConfig.Stats.QuarterHour.	object	-	This object contains DSL statistics accumulated during the current quarter hour.	-	1.0
ReceiveBlocks	unsignedInt	-	Number of successfully received blocks during the current quarter hour, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
TransmitBlocks	unsignedInt	-	Number of successfully transmitted blocks during the current quarter hour, where a block is as defined in RFC 2662 [34]. This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
CellDelin	unsignedInt	-	Number of cell-delineation errors during the current quarter hour (total seconds with NCD or LCD failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
LinkRetrain	unsignedInt	-	Number of link-retrain errors during the current quarter hour (Full Initialization Count as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not available at either the G or the S/T interface.	-	1.0
InitErrors	unsignedInt	-	Number of initialization errors during the current quarter hour (LINIT failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LInit	boolean	-	LInit is a flag to signal that a failure occurred as defined in G.997.1.	-	1.4
InitTimeouts	unsignedInt	-	Number of initialization timeout errors during the current quarter hour.	-	1.0
LossOfFraming	unsignedInt	-	Number of loss-of-framing errors during the current quarter hour (LOF failures as defined in ITU-T Rec. G.997.1). This parameter is DEPRECATED because it is not defined in G.997.1.	-	1.0
LOF	boolean		This parameter corresponds to LOF as defined in ITU-T Rec. G.997.1. LOF is a flag to signal that a failure occurred.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ErroredSecs	unsignedInt	-	Number of errored seconds during the current quarter hour (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCErroredSecs	unsignedInt	-	Number of errored seconds during the current quarter hour detected by the ATU-C (ES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
SeverelyErroredSecs	unsignedInt	-	Number of severely errored seconds during the current quarter hour (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCSeverelyErroredSecs	unsignedInt	-	Number of severely errored seconds during the current quarter hour detected by the ATU-C (SES-L as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.4
FECErrors	unsignedInt	-	Number of FEC errors detected during the current quarter hour (FEC-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCFECErrors	unsignedInt	-	Number of FEC errors detected by the ATU-C during the current quarter hour (FEC-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
HECErrors	unsignedInt	-	Number of HEC errors detected during the current quarter hour (HEC-P as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ATUCHECErrors	unsignedInt	-	Number of HEC errors detected by the ATU-C during the current quarter hour (HEC-PFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
CRCErrors	unsignedInt	-	Number of CRC errors detected during the current quarter hour (CV-C as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
ATUCCRCErrors	unsignedInt	-	Number of CRC errors detected by the ATU-C during the current quarter hour (CV-CFE as defined in ITU-T Rec. G.997.1). Note: This parameter is OPTIONAL at the G and S/T interfaces in G.997.1 Amendment 1. If the parameter is implemented but no value is available, its value MUST be 4294967295 (the maximum for its data type).	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-EthernetInterfaceConfig.	object	-	This object models physical layer properties specific to a single Ethernet physical connection used for Internet access on a CPE. This object is intended for a CPE with an Ethernet WAN interface, and is exclusive of any other WAN*InterfaceConfig object within a given WAN-Device instance. Note that this object is *not* related to the Ethernet protocol layer sometimes used in associated with a DSL connection.	-	1.0
Enable	boolean	W	Enables or disables this interface.	-	1.0
Status	string	-	Indicates the status of this interface. Enumeration of: "Up" "NoLink" "Error" (OPTIONAL) "Disabled" The "Error" value MAY be used by the CPE to indicate a locally defined error condition.	-	1.0
MACAddress	string	-	The physical address of the interface.	-	1.0
MaxBitRate	string	W	The maximum upstream and downstream bit rate available to this connection. Enumeration of: "10" "100" "1000" "10000" "Auto"	-	1.0
DuplexMode	string	W	The duplex mode available to this connection. Enumeration of: "Half" "Full" "Auto"	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ShapingRate	int[-1:]	W	Rate to shape this connection's egress traffic to. For leaky bucket (constant rate shaping), this is the constant rate. For token bucket (variable rate shaping), this is the average rate. If <= 100, in percent of the rate of the highest rate-constrained layer over which the packet will travel on egress. If > 100, in bits per second. A value of -1 indicates no shaping. For example, for packets destined for a WAN DSL interface, if the ATM layer is rate-constrained, then the rate is calculated relative to this rate. Otherwise, the rate is calculated relative to the physical-layer DSL rate.	-	1.4
ShapingBurstSize	unsignedInt	W	Burst size in bytes. For both leaky bucket (constant rate shaping) and token bucket (variable rate shaping) this is the bucket size and is therefore the maximum burst size.	-	1.4
InternetGatewayDevice.WANDevice.{i}.WAN-EthernetInterfaceConfig.Stats.	object	-	This object contains statistics for an Ethernet WAN interface on a CPE device.	-	1.0
BytesSent	unsignedInt	-	Total number of bytes sent over the interface since the CPE was last reset.	-	1.0
BytesReceived	unsignedInt	-	Total number of bytes received over the interface since the CPE was last reset.	-	1.0
PacketsSent	unsignedInt	-	Total number of packets sent over the interface since the CPE was last reset.	-	1.0
PacketsReceived	unsignedInt	-	Total number of packets received over the interface since the CPE was last reset.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-DSLConnectionManagement.	object	-	This object is intended for a CPE with a DSL modem WAN interface. *Note – This object was originally created to allow WANConnection devices and services to be added dynamically in the IGD object model in TR-064 because UPnP Device Architecture 1.0 did not contain this capability natively. Because in TR-069 objects can be created and removed using the AddObject and DeleteObject RPCs, WANConnection interfaces can be managed using these TR-069 mechanisms directly. Therefore, unlike the TR-064 equivalent, the ConnectionService table within this object is Read-Only in the TR-069 InternetGatewayDevice data model context.* This object is OBSOLETED because it serves no purpose.	-	1.0
ConnectionServiceNumberOfEntries	unsignedInt	-	Number of table entries in the ConnectionService table. This parameter is OBSOLETED because it is within an OBSOLETED object. The CPE MAY return a value of 0 for this parameter, regardless of the number of connection services, in which case no ConnectionService instances will exist.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WANDSLConnectionManagement.ConnectionService.{i}.	object	-	This table contains an entry for each connection service. This object is OBSOLETED because it is within an OBSOLETED object.	-	1.0
WANConnectionDevice	string(256)	-	Specifies a WAN connection device object associated with this connection instance. The content is the full hierarchical parameter name of the WAN connection device. Example: "InternetGatewayDevice.WANDevice.1.WANConnectionDevice.2". This parameter is OBSOLETED because it is within an OBSOLETED object.	-	1.0
WANConnectionService	string(256)	-	Specifies a WAN connection object associated with this connection instance. The content is the full hierarchical parameter name of the layer 3 connection object. Example: "InternetGatewayDevice.WANDevice.1.WANConnectionDevice.2.WANPPPConnection.1". This parameter is OBSOLETED because it is within an OBSOLETED object.	-	1.0
DestinationAddress	string(256)	-	Destination address of the WANConnectionDevice entry. One of: PVC: VPI/VCI SVC: ATM connection name SVC: ATM address The "PVC:" or "SVC:" prefix is part of the parameter value and MUST be followed by 0 or 1 space characters. For example, possible values for this parameter are "PVC:8/23" or "PVC: 0/35". This parameter is OBSOLETED because it is within an OBSOLETED object.	-	1.0
LinkType	string(16)	-	Link Type of the WANConnectionDevice entry. One of Link Types as described in WANDSLLinkConfig. This parameter is OBSOLETED because it is within an OBSOLETED object.	-	1.0
ConnectionType	string(16)	-	Connection Type of the WANPPPConnection or WANIPConnection entry. One of PossibleConnectionTypes as described in WAN**Connection service. This parameter is OBSOLETED because it is within an OBSOLETED object.	-	1.0
Name	string(32)	-	User-readable name of the connection. This parameter is OBSOLETED because it is within an OBSOLETED object.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WAN-DSLDiagnostics.	object	-	This object is to provide diagnostic information for a CPE with an ADSL2 or ADSL2+ modem WAN interface, but MAY also be used for ADSL.	-	1.0
LoopDiagnosticsState	string	W	Indicates availability of diagnostic data. One of: "None" "Requested" "Complete" "Error_Internal" "Error_Other" If the ACS sets the value of this parameter to Requested, the CPE MUST initiate the corresponding diagnostic test, which brings down the DSL connection while the test is operating. When writing, the only allowed value is Requested. When requested, the CPE SHOULD wait until after completion of the communication session with the ACS before starting the diagnostic. When the test is completed, the value of this parameter MUST be either Complete (if the test completed successfully), or one of the Error values listed above. If the value of this parameter is anything other than Complete, the values of the results parameters for this test are indeterminate. When the diagnostic initiated by the ACS is completed, the CPE MUST establish a new connection to the ACS to allow the ACS to view the results, indicating the corresponding reason in the Inform message. After the diagnostic is complete, the value of all result parameters (all read-only parameters in this object instance) MUST be retained by the CPE until either this diagnostic is run again, or the CPE reboots. After a reboot, if the CPE has not retained the result parameters from the most recent test, it MUST set the value of this parameter to "None".	-	1.0
ACTPSDds	int	-	Downstream actual power spectral density. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.0
ACTPSDus	int	-	Upstream actual power spectral density. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.0
ACTATPds	int	-	Downstream actual aggregate transmitter power. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.0
ACTATPus	int	-	Upstream actual aggregate transmitter power. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
HLINSCds	int	-	Downstream linear representation scale. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.0
HLINSCus	int	-	Scaling used to represent the upstream linear channel characteristics. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
HLINGds	unsignedInt	-	Number of sub-carriers per sub-carrier group in the downstream direction for HLINpsds. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
HLINGus	unsignedInt	-	Number of sub-carriers per sub-carrier group in the downstream direction for HLINpsus. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
HLOGGds	unsignedInt	-	Number of sub-carriers per sub-carrier group in the downstream direction for HLOGpsds. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
HLOGGus	unsignedInt	-	Number of sub-carriers per sub-carrier group in the upstream direction for HLOGpsus. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
HLOGpsds	string(2559)	-	Downstream logarithmic channel characteristics per sub-carrier group. Comma-separated list of values. The maximum number of elements is 256 for G.992.3, and 512 for G.992.5. For G.993.2, the number of elements will depend on the value of HLOGGds but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None." Note: HLOGpsds is measured during initialization and is not updated during Showtime.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
HLOGpsus	string(2559)	-	Upstream logarithmic channel characteristics per sub-carrier group. Comma-separated list of values. The maximum number of elements is 64 for G.992.3 and G.992.5. For G.993.2, the number of elements will depend on the value of HLOGGus but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None." Note: HLOGpsus is measured during initialization and is not updated during Showtime.	-	1.4
HLOGMTds	unsignedInt	-	Indicates the number of symbols over which HLOGpsds was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
HLOGMTus	unsignedInt	-	Indicates the number of symbols over which HLOGpsus was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
LATNpbds	string(24)	-	Downstream line attenuation per usable band, as computed during initialization. Comma-separated list of values. Number of elements is dependent on the number of downstream bands but will exceed one only for G.993.2. Interpretation of LATNpbds is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4
LATNpbus	string(24)	-	Upstream line attenuation per usable band, as computed during initialization. Comma-separated list of values. Number of elements is dependent on the number of upstream bands but will exceed one only for G.993.2. Interpretation of LATNpbus is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4
SATNds	string(24)	-	Downstream signal attenuation per usable band, as computed during the L0 (i.e., Showtime) state. Comma-separated list of values. Number of elements is dependent on the number of downstream bands but will exceed one only for G.993.2. Interpretation of SATNpbds is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4
SATNus	string(24)	-	Upstream signal attenuation per usable band, as computed during the L0 (i.e., Showtime) state. Comma-separated list of values. Number of elements is dependent on the number of downstream bands but will exceed one only for G.993.2. Interpretation of SATNpbus is as defined in ITU-T Rec. G.997.1. Note: See ITU-T Recommendation G.997.1.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
HLINpsds	string (61430)	-	Downstream linear channel characteristics per subcarrier group. Comma-separated list of values. Maximum number of complex pairs is 256 for G.992.3, and 512 for G.992.5. For G.993.2, the number of pairs will depend on the value of HLINGds but will not exceed 512. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: HLIN is not applicable in PLOAM for G.992.1 or G.992.2. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None."	-	1.0
HLINpsus	string (61430)	-	Upstream linear channel characteristics per sub-carrier group. Comma-separated list of values. Maximum number of complex pairs is 64 for G.992.3, and G.992.5. For G.993.2, the number of pairs will depend on the value of HLINGus but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: HLIN is not applicable in PLOAM for G.992.1 or G.992.2. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None."	-	1.4
QLNGds	unsignedInt	-	Number of sub-carriers per sub-carrier group in the downstream direction for QLNpsds. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
QLNGus	unsignedInt	-	Number of sub-carriers per sub-carrier group in the upstream direction for QLNpsus. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
QLNpsds	string (61430)	-	Downstream quiet line noise per subcarrier group. Comma-separated list of values. Maximum number of elements is 256 for G.992.3, 512 for G.992.5. For G.993.2, the number of elements will depend on the value of QLNGds but will not exceed 512. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: QLN is not applicable in PLOAM for G.992.1 or G.992.2. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None."	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
QLNpsus	string (61430)	-	Upstream quiet line noise per subcarrier group. Comma-separated list of values. The maximum number of elements is 64 for G.992.3, and G.992.5. For G.993.2, the number of elements will depend on the value of QLNGus but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: QLN is not applicable in PLOAM for G.992.1 or G.992.2. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None."	-	1.4
QLNMTds	unsignedInt	-	Indicates the number of symbols over which QLNpsds was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
QLNMTus	unsignedInt	-	Indicates the number of symbols over which QLNpsus was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
SNRGds	unsignedInt	-	Number of sub-carriers per sub-carrier group in the downstream direction for SNRpsds. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
SNRGus	unsignedInt	-	Number of sub-carriers per sub-carrier group in the upstream direction for SNRpsus. Valid values are 1, 2, 4, and 8. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 1.	-	1.4
SNRpsds	string (61430)	-	Downstream SNR per subcarrier group. Comma-separated list of values. Maximum number of elements is 256 for G.992.3, 512 for G.992.5. For G.993.2, the number of elements will depend on the value of SNRGds but will not exceed 512. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Interpretation of the value is as defined in ITU-T Rec. G.997.1. Note: SNRps is not applicable in PLOAM for G.992.1 or G.992.2. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None."	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SNRpsus	string (61430)	-	Upstream SNR per subcarrier group. Comma-separated list of values. The maximum number of elements is 64 for G.992.3, and G.992.5. For G.993.2, the number of elements will depend on the value of SNRGus but will not exceed 512. Interpretation of the values is as defined in ITU-T Rec. G.997.1. Note: SNRps is not applicable in PLOAM for G.992.1 or G.992.2. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to "None."	-	1.4
SNRMTds	unsignedInt	-	Indicates the number of symbols over which SNRpsds was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
SNRMTus	unsignedInt	-	Indicates the number of symbols over which SNRpsus was measured. Note: See ITU-T Recommendation G.997.1. For a multimode device operating in a mode in which this parameter does not apply, the value of this parameter SHOULD be set to 0.	-	1.4
BITSpsds	string (61430)	-	Downstream bit allocation per subcarrier group. Comma-separated list of values. Maximum number of elements is 256 for G.992.3, 512 for G.992.5. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	-	1.0
BITSpsus	string (61430)	-	Upstream bit allocation per subcarrier group. Comma-separated list of values. Maximum number of elements is 256 for G.992.3, 512 for G.992.5. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	-	1.4
GAINSpsds	string (61430)	-	Downstream gain allocation per subcarrier group. Comma-separated list of integers. Maximum number of elements is 256 for G.992.3, 512 for G.992.5. Interpretation of the value is as defined in ITU-T Rec. G.997.1. This parameter is DEPRECATED.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.	object	W	Each instance contains objects associated with a given WAN link. In the case of DSL, each instance corresponds to either a single ATM VC or a PTM Ethernet link. On creation of a WANConnection-Device instance, there are initially no connection objects contained within. In the case of Ethernet (interface or link), only one WANConnectionDevice instance is supported.	-	1.0
WANIPConnectionNumberOfEntries	unsignedInt	-	Number of instances of WANIPConnection in this WANConnectionDevice.	-	1.0
WANPPPConnectionNumberOfEntries	unsignedInt	-	Number of instances of WANPPPConnection in this WANConnectionDevice.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANDSLLinkConfig.	object	-	This object models the ATM layer properties specific to a single physical connection of a DSL modem used for Internet access on a CPE. This object is intended for a CPE with a DSL modem WAN interface, and is exclusive of any other WAN*LinkConfig object within a given WAN-ConnectionDevice instance.	-	1.0
Enable	boolean	W	Enables or disables the link. On creation of a WANConnectionDevice, this object is disabled by default.	False	1.0
LinkStatus	string	-	Status of the link. Enumeration of: "Up" "Down" "Initializing" "Unavailable"	-	1.0
LinkType	string	W	Indicates the type of DSL connection and refers to the complete stack of protocol used for this connection. Enumeration of: "EoA" (RFC 2684 [35] bridged Ethernet over ATM) "IPoA" (RFC 2684 [35] routed IP over ATM) "PPPoA" (RFC 2364 [33] PPP over ATM) "PPPoE" (RFC 2516 [31] PPP over Ethernet on RFC 2684 [35] bridged Ethernet over ATM, DEPRECATED) "CIP" (RFC 2225 [29] Classical IP over ATM) "Unconfigured" The value "PPPoE" has always been DEPRECATED and "EoA" SHOULD be used instead (see Annex B). The ACS MUST NOT set this parameter to "PPPoE" and the CPE MUST reject attempts to do so.	"Unconfig-ured"	1.0
AutoConfig	boolean	-	Indicates if the CPE is currently using some auto configuration mechanisms for this connection. If this variable is True, all writable variables in this connection instance become read-only. Any attempt to change one of these variables SHOULD fail and an error SHOULD be returned.	-	1.0
ModulationType	string	-	Indicates the type of DSL modulation used on the interface associated with this connection (duplication from WANDSLInterfaceConfig). Enumeration of: "ADSL_G.dmt" "ADSL_G.lite" "ADSL_G.dmt.bis" "ADSL_re-adsl" "ADSL_2plus" "ADLS_four" "ADSL_ANSI_T1.413" "G.shdsl" "IDSL" "HDSL" "SDSL" "VDSL"	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
DestinationAddress	string(256)	W	Destination address of this link. One of: PVC: VPI/VCI SVC: ATM connection name SVC: ATM address The "PVC:" or "SVC:" prefix is part of the parameter value and MUST be followed by 0 or 1 space characters. For example, possible values for this parameter are "PVC:8/23" or "PVC: 0/35".	-	1.0
ATMEncapsulation	string	W	Identifies the connection encapsulation that will be used. Enumeration of: "LLC" "VCMUX"	-	1.0
FCSPreserved	boolean	W	This flag tells if a checksum SHOULD be added in the ATM payload. It does not refer to the checksum of one of the ATM cells or AALX packets. In case of LLC or VCMUX encapsulation, this ATM checksum is the FCS field described in RFC 2684 [35]. It is only applicable in the upstream direction.	-	1.0
VCSearchList	string(256)	W	Comma-separated ordered list of VPI/VCI pairs to search if a link using the DestinationAddress cannot be established. In the form: VPI1/VCI1, VPI2/VCI2, … Example: "0/35, 8/35, 1/35"	-	1.0
ATMAAL	string	-	Describes the ATM Adaptation Layer (AAL) currently in use on the PVC. Enumeration of: "AAL1" "AAL2" "AAL3" "AAL4" "AAL5"	-	1.0
ATMTransmittedBlocks	unsignedInt	-	The current count of successfully transmitted cells.	-	1.0
ATMReceivedBlocks	unsignedInt	-	The current count of successfully received cells.	-	1.0
ATMQoS	string	W	Describes the ATM Quality Of Service (QoS) being used on the VC. Enumeration of: "UBR" "CBR" "GFR" "VBR-nrt" "VBR-rt" "UBR+" "ABR"	-	1.0
ATMPeakCellRate	unsignedInt	W	Specifies the upstream peak cell rate in cells per second.	-	1.0
ATMMaximumBurstSize	unsignedInt	W	Specifies the upstream maximum burst size in cells.	-	1.0
ATMSustainableCellRate	unsignedInt	W	Specifies the upstream sustainable cell rate, in cells per second, used for traffic shaping.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
AAL5CRCErrors	unsignedInt	-	Count of the AAL5 layer cyclic redundancy check errors. This parameter is DEPRECATED because it overlaps with the ATMCRCErrors parameter. If present, it MUST have the same value as the ATMCRCErrors parameter if AAL5 is in use, or 0 if AAL5 is not in use.	-	1.0
ATMCRCErrors	unsignedInt	-	Count of the ATM layer cyclic redundancy check (CRC) errors. This refers to CRC errors at the ATM adaptation layer (AAL). The AAL in use is indicated by the ATMAAL parameter. The value of the ATMCRCErrors parameter MUST be 0 for AAL types that have no CRCs.	-	1.0
ATMHECErrors	unsignedInt	-	Count of the number of Header Error Check related errors at the ATM layer.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANATMF5Loopback-Diagnostics.	object	-	This object is provides access to an ATM-layer F5 OAM loopback test.	-	1.0
DiagnosticsState	string	W	Indicates availability of diagnostic data. One of: "None" "Requested" "Complete" "Error_Internal" "Error_Other" If the ACS sets the value of this parameter to Requested, the CPE MUST initiate the corresponding diagnostic test. When writing, the only allowed value is Requested. To ensure the use of the proper test parameters (the writable parameters in this object), the test parameters MUST be set either prior to or at the same time as (in the same SetParameterValues) setting the DiagnosticsState to Requested. When requested, the CPE SHOULD wait until after completion of the communication session with the ACS before starting the diagnostic. When the test is completed, the value of this parameter MUST be either Complete (if the test completed successfully), or one of the Error values listed above. If the value of this parameter is anything other than Complete, the values of the results parameters for this test are indeterminate. When the diagnostic initiated by the ACS is completed (successfully or not), the CPE MUST establish a new connection to the ACS to allow the ACS to view the results, indicating the Event code "8 DIAGNOSTICS COMPLETE" in the Inform message. After the diagnostic is complete, the value of all result parameters (all read-only parameters in this object instance) MUST be retained by the CPE until either this diagnostic is run again, or the CPE reboots. After a reboot, if the CPE has not retained the result parameters from the most recent test, it MUST set the value of this parameter to "None". Modifying any of the writable parameters in this object except for this one MUST result in the value of this parameter being set to "None". While the test is in progress, modifying any of the writable parameters in this object except for this one MUST result in the test being terminated and the value of this parameter being set to "None". While the test is in progress, setting this parameter to Requested (and possibly modifying other writable parameters in this object) MUST result in the test being terminated and then restarted using the current values of the test parameters.	"None"	1.0
NumberOfRepetitions	unsignedInt [1:]	W	Number of repetitions of the ping test to perform before reporting the results.	1	1.0
Timeout	unsignedInt [1:]	W	Timeout in milliseconds for the ping test.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
SuccessCount	unsignedInt	-	Result parameter indicating the number of successful pings (those in which a successful response was received prior to the timeout) in the most recent ping test.	0	1.0
FailureCount	unsignedInt	-	Result parameter indicating the number of failed pings in the most recent ping test.	0	1.0
AverageResponseTime	unsignedInt	-	Result parameter indicating the average response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.	0	1.0
MinimumResponseTime	unsignedInt	-	Result parameter indicating the minimum response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.	0	1.0
MaximumResponseTime	unsignedInt	-	Result parameter indicating the maximum response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value MUST be zero.	0	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANPTMLinkConfig.	object	-	This object models the PTM layer properties specific to a layer 2 interface of a DSL modem used for Internet access on a CPE. This object is intended for a CPE with a DSL modem WAN interface, and is exclusive of any other WAN*Link-Config object within a given WANConnection-Device instance. The PTM Link Layer object exists when the WANDSLInterfaceConfig LinkEncapsulation-Supported parameter includes any of: "G.992.3_Annex_K_PTM" "G.993.2_Annex_K_PTM" "G.994.1"	-	1.4
Enable	boolean	W	Enables or disables the link. On creation of a WANConnectionDevice, this object is disabled by default.	False	1.4
LinkStatus	string	-	Status of the link. Enumeration of: "Up" "Down" "Initializing" "Unavailable"	-	1.4
MACAddress	string	-	The physical address of the interface.	-	1.4
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANPTMLinkConfig.-Stats.	object	-	This object represents the statistics collected and returned over a PTM link.	-	1.4
BytesSent	unsignedInt	-	The total number of bytes transmitted out of the interface, including framing characters. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
BytesReceived	unsignedInt	-	The total number of bytes received on the interface, including framing characters. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
FramesSent	unsignedInt	-	The total number of packets (frames) transmitted out of the interface. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
FramesReceived	unsignedInt	-	The total number of packets (frames) which were received on this interface. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
OOSNearEnd	boolean	-	Indication that the CPE has detected the link is Out of Synchronization since the CPE rebooted or the interface was last enabled.	-	1.4
OOSFarEnd	boolean	-	Indication that the remote device has detected the link is Out of Synchronization since the CPE rebooted or the interface was last enabled.	-	1.4
ErrorsSent	unsignedInt	-	The total number of outbound packets that could not be transmitted because of errors. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
ErrorsReceived	unsignedInt	-	The total number of inbound packets that contained errors preventing them from being deliverable. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnicastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were not addressed to a multicast or broadcast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnicastPacketsReceived	unsignedInt	-	The total number of received packets which were not addressed to a multicast or broadcast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
DiscardPacketsSent	unsignedInt	-	The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
DiscardPacketsReceived	unsignedInt	-	The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
MulticastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were addressed to a multicast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
MulticastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a multicast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
BroadcastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were addressed to a broadcast address, including those that were discarded or not sent. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
BroadcastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a broadcast address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
UnknownProtoPacketsReceived	unsignedInt	-	The total number of packets received via the interface which were discarded because of an unknown or unsupported protocol. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANEthernetLinkConfig.	object	-	This object models the Ethernet link layer properties specific to a single physical connection used for Internet access on a CPE. This object is intended for a CPE with an Ethernet WAN interface, and is exclusive of any other WAN*Link-Config object within a given WANConnection-Device instance. Note that this object is *not* related to the Ethernet protocol layer sometimes used in associated with a DSL connection.	-	1.0
EthernetLinkStatus	string	-	Status of the Ethernet link. Enumeration of: "Up" "Down" "Unavailable"	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANPOTSLinkConfig.	object	-	This object models the POTS link layer properties specific to a single physical connection used for Internet access on a CPE. This object is intended for a CPE with a POTS WAN interface, and is exclusive of any other WAN*LinkConfig object within a given WANConnectionDevice instance.	-	1.0
Enable	boolean	W	Enables or disables the link. On creation of a WANConnectionDevice, this object is disabled by default.	False	1.0
LinkStatus	string	-	Status of the link. Enumeration of: "Up" "Down" "Dialing" "Connecting" "Unavailable"	-	1.0
ISPPhoneNumber	string(64)	W	Specifies a list of strings separated by semicolon (;), each string representing a phone number to connect to a particular ISP. The digits of the phone number follow the semantics of the ITU-T E.164 specification. Delimiters such as brackets or hyphens between the digits of a phone number are to be ignored by the CPE.	<Empty>	1.0
ISPInfo	string(64)	W	Information identifying the Internet Service Provider. The format of the string is vendor specific.	<Empty>	1.0
LinkType	string	W	This variable indicates the type of POTS link used for the dialup connection. Enumeration of: "PPP_Dialup"	"PPP-_Dialup"	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
NumberOfRetries	unsignedInt	W	The number of times the CPE SHOULD attempt an Internet connection setup before returning error.	-	1.0
DelayBetweenRetries	unsignedInt	W	The number of seconds the CPE SHOULD wait between attempts to setup an Internet connection.	-	1.0
Fclass	string	-	Specifies capabilities of the POTS modem – i.e., if it handles data (0), fax (1,2,2.0), voice (8), DSVD (80). Comma-separated list of the following enumeration: "0" "1" "2" "2.0" "8" "80"	-	1.0
DataModulationSupported	string	-	The modulation standard currently being used for data. Enumeration of: "V92" "V90" "V34" "V32bis" "V32"	-	1.0
DataProtocol	string	-	The protocol standard currently being used for data transfers. Enumeration of: "V42_LAPM" "V42_MNP4" "V14" "V80"	-	1.0
DataCompression	string	-	The compression technology implemented on the modem. Enumeration of: "V42bis" "MNP5"	-	1.0
PlusVTRCommandSupported	boolean	-	Capability for full duplex operation with data and voice.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANIPConnection.{i}.	object	W	This object enables configuration of IP connections on the WAN interface of a CPE. If the Layer2Bridging object is implemented, the view that it provides of the CPE's underlying bridging configuration MUST be consistent with the view provided by any LANDevice and WAN**Conn-ection objects. The implications of this are explained in Annex A.6.	-	1.0
Enable	boolean	W	Enables or disables the connection instance. On creation of a WANIPConnection instance, it is initially disabled.	False	1.0
Reset	boolean	W	When set to True, the device MUST tear down the existing IP connection represented by this object and establish a new one. The device MUST initiate the reset after completion of the current CWMP session. The device MAY delay resetting the connection in order to avoid interruption of a user service such as an ongoing voice call. When read, this parameter always returns False.	False	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ConnectionStatus	string	-	Current status of the connection. Enumeration of: "Unconfigured" "Connecting" "Connected" "PendingDisconnect" "Disconneting" (DEPRECATED) "Disconnecting" "Disconnected" The value "Disconneting" is DEPRECATED because it is a typo. The ACS MUST treat "Disconneting" and "Disconnecting" the same.	-	1.0
PossibleConnectionTypes	string	-	A comma-separated list indicating the types of connections possible for this connection instance. Each element of the list is an enumeration of: "Unconfigured" "IP_Routed" "IP_Bridged"	-	1.0
ConnectionType	string	W	Specifies the connection type of the connection instance. Enumeration of: "Unconfigured" "IP_Routed" "IP_Bridged"	-	1.0
Name	string(256)	W	User-readable name of this connection.	-	1.0
Uptime	unsignedInt	-	The time in seconds that this connection has been up.	-	1.0
LastConnectionError	string	-	The cause of failure for the last connection setup attempt. Enumeration of: "ERROR_NONE" "ERROR_COMMAND_ABORTED" "ERROR_NOT_ENABLED_FOR_INTERNET" "ERROR_USER_DISCONNECT" "ERROR_ISP_DISCONNECT" "ERROR_IDLE_DISCONNECT" "ERROR_FORCED_DISCONNECT" "ERROR_NO_CARRIER" "ERROR_IP_CONFIGURATION" "ERROR_UNKNOWN"	"ERROR-_NONE"	1.0
AutoDisconnectTime	unsignedInt	W	The time in seconds since the establishment of the connection after which connection termination is automatically initiated by the CPE. This occurs irrespective of whether the connection is being used or not. A value of 0 (zero) indicates that the connection is not to be shut down automatically.	-	1.0
IdleDisconnectTime	unsignedInt	W	The time in seconds that if the connection remains idle, the CPE automatically terminates the connection. A value of 0 (zero) indicates that the connection is not to be shut down automatically.	-	1.0
WarnDisconnectDelay	unsignedInt	W	Time in seconds the Status remains in the pending disconnect state before transitioning to disconnecting state to drop the connection.	-	1.0
RSIPAvailable	boolean	-	Indicates if Realm-specific IP (RSIP) is available as a feature on the CPE.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
NATEnabled	boolean	W	Indicates if Network Address Translation (NAT) is enabled for this connection.	-	1.0
AddressingType	string	W	The method used to assign an address to the WAN side interface of the CPE for this connection. Enumeration of: "DHCP" "Static"	-	1.0
ExternalIPAddress	string	W	This is the external IP address used by NAT for this connection. This parameter is configurable only if the AddressingType is Static.	-	1.0
SubnetMask	string	W	Subnet mask of the WAN interface. This parameter is configurable only if the AddressingType is Static.	-	1.0
DefaultGateway	string	W	The IP address of the default gateway for this connection. This parameter is configurable only if the AddressingType is Static.	-	1.0
DNSEnabled	boolean	W	Whether or not the device SHOULD attempt to query a DNS server across this connection.	True	1.0
DNSOverrideAllowed	boolean	W	Whether or not a manually set, non-empty DNS address can be overridden by a DNS entry received from the WAN.	False	1.0
DNSServers	string(64)	W	Comma-separated list of DNS server IP addresses for this connection. Support for more than three DNS Servers is OPTIONAL.	-	1.0
MaxMTUSize	unsignedInt [1:1540]	W	The maximum allowed size of an Ethernet frame from LAN-side devices.	-	1.0
MACAddress	string	W	The physical address of the WANIPConnection if applicable. Configurable only if MACAddressOverride is present and True.	-	1.0
MACAddressOverride	boolean	W	Whether the value of MACAddress parameter can be overridden. If False, the CPE's default value is used (or restored if it had previously been overridden).	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ConnectionTrigger	string	W	Trigger used to establish the IP connection. Enumeration of: "OnDemand" "AlwaysOn" "Manual" The above values are defined as follows: OnDemand: If this IP connection is disconnected for any reason, it is to remain disconnected until the CPE has one or more packets to communicate over this connection, at which time the CPE automatically attempts to reestablish the connection. AlwaysOn: If this IP connection is disconnected for any reason, the CPE automatically attempts to reestablish the connection (and continues to attempt to reestablish the connection as long it remains disconnected). Manual: If this IP connection is disconnected for any reason, it is to remain disconnected until the user of the CPE explicitly instructs the CPE to reestablish the connection. Note that the reason for an IP connection becoming disconnected to begin with might be either external to the CPE, such as non-renewal of a DHCP lease or momentary disconnection of the physical interface, or internal to the CPE, such as use of the IdleDisconnectTime and/or Auto-DisconnectTime parameters in this object. Note also that the means by which a CPE would keep an IP connection disconnected (while waiting for the designated trigger) if it is otherwise physically connected and has an IP address is a local matter specific to the implementation of the CPE.	"On-Demand"	1.0
RouteProtocolRx	string	W	Defines the Rx protocol to be used. Enumeration of: "Off" "RIPv1" (OPTIONAL) "RIPv2" (OPTIONAL) "OSPF" (OPTIONAL)	"Off"	1.0
ShapingRate	int[-1:]	W	Rate to shape this connection's egress traffic to. For leaky bucket (constant rate shaping), this is the constant rate. For token bucket (variable rate shaping), this is the average rate. If <= 100, in percent of the rate of the highest rate-constrained layer over which the packet will travel on egress. If > 100, in bits per second. A value of -1 indicates no shaping. For example, for packets destined for a WAN DSL interface, if the ATM layer is rate-constrained, then the rate is calculated relative to this rate. Otherwise, the rate is calculated relative to the physical-layer DSL rate.	-1	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ShapingBurstSize	unsignedInt	W	Burst size in bytes. For both leaky bucket (constant rate shaping) and token bucket (variable rate shaping) this is the bucket size and is therefore the maximum burst size.	-	1.1
PortMappingNumberOfEntries	unsignedInt	-	Total number of port mapping entries.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANIPConnection.{i}.-DHCPClient.	object	-	This object contains DHCP client configuration parameters.	-	1.4
SentDHCPOptionNumberOfEntries	unsignedInt	-	The number of entries in the SentDHCPOption table.	-	1.4
ReqDHCPOptionNumberOfEntries	unsignedInt	-	The number of entries in the ReqDHCPOption table.	-	1.4
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANIPConnection-.{i}.DHCPClient.SentDHCPOption.{i}.	object	W	Each instance of this object represents a DHCP option that MUST, if enabled, be sent in DHCP client requests. All sent DHCP options MUST be listed.	-	1.4
Enable	boolean	W	Enables or disables this SentDHCPOption table entry.	False	1.4
Tag	unsignedInt [1:254]	W	Option tag as defined in RFC 2132 [28].	-	1.4
Value	base64(340)	W	Base64 encoded option value.	<Empty>	1.4
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANIPConnection-.{i}.DHCPClient.ReqDHCPOption.{i}.	object	W	Each instance of this object represents a DHCP option that MUST, if enabled, be requested in DHCP client requests. All requested DHCP options MUST be listed.	-	1.4
Enable	boolean	W	Enables or disables this ReqDHCPOption table entry.	False	1.4
Order	unsignedInt [1:]	W	Position of the option in the DHCP client request. A value of 1 indicates the first entry. When this value is modified, if the value matches that of an existing entry, the Order value for the existing entry and all lower Order entries is incremented to ensure uniqueness of this value. A deletion causes Order values to be compacted. When a value is changed, incrementing occurs before compaction. The value on creation of a ReqDHCPOption table entry MUST be one greater than the largest current value.	-	1.4
Tag	unsignedInt [1:254]	W	Option tag as defined in RFC 2132 [28].	-	1.4
Value	base64(340)	-	Base64 encoded most recently received DHCP option value. If no option value has been received, then the value MUST represent an empty string. Received DHCP option values MAY, but need not, persist across CPE reboots.	<Empty>	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANIPConnection.{i}.-PortMapping.{i}.	object	W	Port mapping table. This table MUST contain all NAT port mappings associated with this connection, including static and dynamic port mappings programmatically created via local control protocol, such as UPnP. This table MUST NOT contain dynamic NAT binding entries associated with the normal operation of NAT. At most one entry in an instance of this table can exist with all of the same values for RemoteHost, ExternalPort, and PortMappingProtocol. If the ACS attempts to set the parameters of an existing entry such that this requirement would be violated, the CPE MUST reject the request. In this case, the SetParameterValues response MUST include a SetParameterValuesFault element for each parameter in the corresponding request whose modification would have resulted in such a violation. On creation of a new table entry, the CPE MUST choose default values for ExternalPort and PortMappingProtocol such that the new entry does not conflict with any existing entry.	-	1.0
PortMappingEnabled	boolean	W	Enables or disables the port mapping instance. On creation, an entry is disabled by default.	False	1.0
PortMappingLeaseDuration	unsignedInt	W	Determines the time to live, in seconds, of a port-mapping lease, where “time to live” means the number of seconds before the port mapping expires. A value of 0 means the port mapping is static. Support for dynamic (non-static) port mappings is OPTIONAL. That is, the only value for PortMappingLeaseDuration that MUST be supported is 0. For a dynamic (non-static) port mapping, when this parameter is read, the value represents the time remaining on the port-mapping lease. That is, for a dynamic port mapping, the value counts down toward 0. When a dynamic port-mapping lease expires, the CPE MUST automatically terminate that port mapping, and MUST automatically delete the corresponding PortMapping table entry.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
RemoteHost	string	W	This parameter is the IP address of the source of inbound packets. An empty string indicates a 'wildcard' (this will be a wildcard in most cases). CPE are REQUIRED only to support wildcards. When RemoteHost is a wildcard, all traffic sent to the ExternalPort on the WAN interface of the gateway is forwarded to the InternalClient on the InternalPort. When RemoteHost is specified as one external IP address, the NAT will only forward inbound packets from this RemoteHost to the InternalClient, all other packets will be dropped. If a CPE supports non-wildcard values for RemoteHost, it MAY additionally support the ability to have more than one port mapping with the same ExternalPort and PortMappingProtocol, but with differing values of RemoteHost. When wildcard values are used for RemoteHost and/or ExternalPort, the following precedence order applies (with the highest precedence listed first): 1. Explicit RemoteHost, explicit ExternalPort 2. Explicit RemoteHost, wildcard ExternalPort 3. Wildcard RemoteHost, explicit ExternalPort 4. Wildcard RemoteHost, wildcard ExternalPort If an incoming packet matches the criteria associated with more than one entry in this table, the CPE MUST apply the port mapping associated with the highest precedence entry. At most one entry in this table can exist with all of the same values for RemoteHost, ExternalPort, and PortMappingProtocol.	<Empty>	1.0
ExternalPort	unsignedInt	W	The external port (or the first port of a range of external ports) that the NAT gateway would listen on for connection requests to a corresponding InternalPort. Inbound packets to this external port on the WAN interface SHOULD be forwarded to InternalClient on the InternalPort. A value of zero (0) represents a 'wildcard.' If this value is a wildcard, connection requests on all external ports (that are not otherwise mapped) will be forwarded to InternalClient. In the wildcard case, the value(s) of InternalPort on InternalClient are ignored. When wildcard values are used for RemoteHost and/or ExternalPort, the following precedence order applies (with the highest precedence listed first): 1. Explicit RemoteHost, explicit ExternalPort 2. Explicit RemoteHost, wildcard ExternalPort 3. Wildcard RemoteHost, explicit ExternalPort 4. Wildcard RemoteHost, wildcard ExternalPort If an incoming packet matches the criteria associated with more than one entry in this table, the CPE MUST apply the port mapping associated with the highest precedence entry. At most one entry in this table can exist with all of the same values for RemoteHost, ExternalPort, and PortMappingProtocol.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ExternalPortEndRange	unsignedInt	W	Indicates the last port of the external port range that starts with ExternalPort. If an external port range is specified, then the behavior described for ExternalPort applies to all ports within the range. A value of zero (0) indicates that no external port range is specified, i.e. that the range consists only of ExternalPort. If ExternalPort is zero (wildcard), the value of this parameter MUST be ignored. If specified, the value of this parameter MUST be greater than or equal to the value of ExternalPort.	0	1.4
InternalPort	unsignedInt	W	The port on InternalClient that the gateway SHOULD forward connection requests to. A value of zero (0) is not allowed.	-	1.0
PortMappingProtocol	string	W	The protocol of the port mapping. Enumeration of: "TCP" "UDP" At most one entry in this table can exist with all of the same values for RemoteHost, ExternalPort, and PortMappingProtocol.	-	1.0
InternalClient	string(256)	W	The IP address or DNS host name of an internal client (on the LAN). Support for an IP address is mandatory. If InternalClient is specified as an IP address and the LAN device's IP address subsequently changes, the port mapping MUST remain associated with the original IP address. Support for DNS host names is OPTIONAL. If InternalClient is specified as a DNS host name and the LAN device's IP address subsequently changes, the port mapping MUST remain associated with this LAN device. In this case, it is the responsibility of the CPE to maintain the name-to-address mapping in the event of IP address changes. This can be accomplished, for example, by assigning the DNS host name via use of DHCP option 12 (Host Name) or option 81 (FQDN). Note that the ACS can learn the host name associated with a given LAN device via the Hosts table (InternetGatewayDevice.LANDevice.{i}.Hosts.). Read access to this parameter MUST always return the exact value that was last set by the ACS. For example, if the internal client is set to a DNS host name, it MUST read back as a DNS host name and not as an IP address. An empty string indicates an unconfigured InternalClient. If this parameter is unconfigured, this port mapping MUST NOT be operational. It MUST be possible to set the InternalClient to the broadcast IP address 255.255.255.255 for UDP mappings. This is to enable multiple NAT clients to use the same well-known port simultaneously.	<Empty>	1.0
PortMappingDescription	string(256)	W	User-readable description of this port mapping.	<Empty>	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.Stats.	object	-	This object contains statistics for all connections within the same WANConnectionDevice that share a common MAC address. The contents of this object SHOULD be identical for each such connection. This object is intended only for WANConnectionDevices that can support an Ethernet-layer on this interface (e.g., PPPoE, IPoE).	-	1.0
EthernetBytesSent	unsignedInt	-	The total number of bytes transmitted, including framing characters, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
EthernetBytesReceived	unsignedInt	-	The total number of bytes received, including framing characters, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
EthernetPacketsSent	unsignedInt	-	The total number of packets transmitted over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
EthernetPacketsReceived	unsignedInt	-	The total number of packets which were received over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
EthernetErrorsSent	unsignedInt	-	The total number of outbound packets that could not be transmitted because of errors, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetErrorsReceived	unsignedInt	-	The total number of inbound packets that contained errors preventing them from being deliverable, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetUnicastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were not addressed to a multicast or broadcast address, including those that were discarded or not sent, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetUnicastPacketsReceived	unsignedInt	-	The total number of received packets which were not addressed to a multicast or broadcast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
EthernetDiscardPacketsSent	unsignedInt	-	The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted, over all connections within the same WANConnectionDevice that share a common MAC address. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetDiscardPacketsReceived	unsignedInt	-	The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable, over all connections within the same WANConnectionDevice that share a common MAC address. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetMulticastPacketsSent	unsignedInt	-	The total number of packets requested for transmission, including those that were discarded or not sent, which were addressed to a multicast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetMulticastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a multicast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetBroadcastPacketsSent	unsignedInt	-	The total number of packets requested for transmission, including those that were discarded or not sent, which were addressed to a broadcast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetBroadcastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a broadcast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetUnknownProtoPacketsReceived	unsignedInt	-	The total number of packets which were discarded because of an unknown or unsupported protocol, received over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANPPPConnection.{i}.	object	W	This object enables configuration of PPP connections on the WAN interface of a CPE. If the Layer2Bridging object is implemented, the view that it provides of the CPE's underlying bridging configuration MUST be consistent with the view provided by any LANDevice and WAN**Conn-ection objects. The implications of this are explained in Annex A.6.	-	1.0
Enable	boolean	W	Enables or disables the connection instance. On creation of a WANPPPConnection instance, it is initially disabled.	False	1.0
Reset	boolean	W	When set to True, the device MUST tear down the existing PPP connection represented by this object and establish a new one. The device MUST initiate the reset after completion of the current CWMP session. The device MAY delay resetting the connection in order to avoid interruption of a user service such as an ongoing voice call. When read, this parameter always returns False.	False	1.4
ConnectionStatus	string	-	Current status of the connection. Enumeration of: "Unconfigured" "Connecting" "Authenticating" "Connected" "PendingDisconnect" "Disconnecting" "Disconnected"	-	1.0
PossibleConnectionTypes	string	-	A comma-separated list indicating the types of connections possible for this connection instance. Each element of the list is an enumeration of: "Unconfigured" "IP_Routed" "DHCP_Spoofed" "PPPoE_Bridged" "PPPoE_Relay" "PPTP_Relay" "L2TP_Relay"	-	1.0
ConnectionType	string	W	Specifies the connection type of the connection instance. Enumeration of: "Unconfigured" "IP_Routed" "DHCP_Spoofed" "PPPoE_Bridged" "PPPoE_Relay" "PPTP_Relay" "L2TP_Relay"	-	1.0
PPPoESessionID	unsignedInt [1:]	-	Represents the PPPoE Session ID.	-	1.4
DefaultGateway	string	-	Represents the IP Address of the remote end Default Gateway established through PPPoE.	-	1.4
Name	string(256)	W	User-readable name of this connection.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
Uptime	unsignedInt	-	The time in seconds that this connection has been up.	-	1.0
LastConnectionError	string	-	The cause of failure for the last connection setup attempt. Enumeration of: "ERROR_NONE" "ERROR_ISP_TIME_OUT" "ERROR_COMMAND_ABORTED" "ERROR_NOT_ENABLED_FOR_INTERNET" "ERROR_BAD_PHONE_NUMBER" "ERROR_USER_DISCONNECT" "ERROR_ISP_DISCONNECT" "ERROR_IDLE_DISCONNECT" "ERROR_FORCED_DISCONNECT" "ERROR_SERVER_OUT_OF_RESOURCES" "ERROR_RESTRICTED_LOGON_HOURS" "ERROR_ACCOUNT_DISABLED" "ERROR_ACCOUNT_EXPIRED" "ERROR_PASSWORD_EXPIRED" "ERROR_AUTHENTICATION_FAILURE" "ERROR_NO_DIALTONE" "ERROR_NO_CARRIER" "ERROR_NO_ANSWER" "ERROR_LINE_BUSY" "ERROR_UNSUPPORTED_BITSPERSECOND" "ERROR_TOO_MANY_LINE_ERRORS" "ERROR_IP_CONFIGURATION" "ERROR_UNKNOWN"	"ERROR-_NONE"	1.0
AutoDisconnectTime	unsignedInt	W	The time in seconds since the establishment of the connection after which connection termination is automatically initiated by the CPE. This occurs irrespective of whether the connection is being used or not. A value of 0 (zero) indicates that the connection is not to be shut down automatically.	-	1.0
IdleDisconnectTime	unsignedInt	W	The time in seconds that if the connection remains idle, the CPE automatically terminates the connection. A value of 0 (zero) indicates that the connection is not to be shut down automatically.	-	1.0
WarnDisconnectDelay	unsignedInt	W	Time in seconds the Status remains in the pending disconnect state before transitioning to disconnecting state to drop the connection.	-	1.0
RSIPAvailable	boolean	-	Indicates if Realm-specific IP (RSIP) is available as a feature on the CPE.	-	1.0
NATEnabled	boolean	W	Indicates if Network Address Translation (NAT) is enabled for this connection.	-	1.0
Username	string(64)	W	Username to be used for authentication.	<Empty>	1.0
Password	string(64)	W	Password to be usef for authentication. When read, this parameter returns an empty string, regardless of the actual value.	<Empty>	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
PPPEncryptionProtocol	string	-	Describes the PPP encryption protocol used between the WAN device and the ISP POP. Enumeration of: "None" "MPPE"	-	1.0
PPPCompressionProtocol	string	-	Describes the PPP compression protocol used between the WAN device and the ISP POP. Enumeration of: "None" "Van Jacobsen" "STAC LZS"	-	1.0
PPPAuthenticationProtocol	string	-	Describes the PPP authentication protocol used between the WAN device and the ISP POP. Enumeration of: "PAP" "CHAP" "MS-CHAP"	-	1.0
ExternalIPAddress	string	-	This is the external IP address used by NAT for this connection.	-	1.0
RemoteIPAddress	string	-	The remote IP address for this connection.	-	1.0
MaxMRUSize	unsignedInt [1:1540]	W	The maximum allowed size of frames sent from the remote peer.	-	1.0
CurrentMRUSize	unsignedInt [1:1540]	-	The current MRU in use over this connection.	-	1.0
DNSEnabled	boolean	W	Whether or not the device SHOULD attempt to query a DNS server across this connection.	True	1.0
DNSOverrideAllowed	boolean	W	Whether or not a manually set, non-empty DNS address can be overridden by a DNS entry received from the WAN.	False	1.0
DNSServers	string(64)	W	Comma-separated list of DNS server IP addresses for this connection. Support for more than three DNS Servers is OPTIONAL.	-	1.0
MACAddress	string	W	The physical address of the WANPPPConnection if applicable. Configurable only if MACAddressOverride is present and True. If TransportType is "PPPoA", the value of this parameter is irrelevant and MUST be an empty string.	-	1.0
MACAddressOverride	boolean	W	Whether the value of MACAddress parameter can be overridden. If False, the CPE's default value is used (or restored if it had previously been overridden). If TransportType is "PPPoA", the value of this parameter is irrelevant and MUST be False.	-	1.0
TransportType	string	-	PPP transport type of the connection. Enumeration of: "PPPoA" "PPPoE" "L2TP" (for future use) "PPTP" (for future use)	-	1.0
PPPoEACName	string(256)	W	PPPoE Access Concentrator.	<Empty>	1.0
PPPoEServiceName	string(256)	W	PPPoE Service Name.	<Empty>	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ConnectionTrigger	string	W	Trigger used to establish the PPP connection. Enumeration of: "OnDemand" "AlwaysOn" "Manual" The above values are defined as follows: OnDemand: If this PPP connection is disconnected for any reason, it is to remain disconnected until the CPE has one or more packets to communicate over this connection, at which time the CPE automatically attempts to reestablish the connection. AlwaysOn: If this PPP connection is disconnected for any reason, the CPE automatically attempts to reestablish the connection (and continues to attempt to reestablish the connection as long it remains disconnected). Manual: If this PPP connection is disconnected for any reason, it is to remain disconnected until the user of the CPE explicitly instructs the CPE to reestablish the connection. Note that the reason for a PPP connection becoming disconnected to begin with might be either external to the CPE, such as termination by the BRAS or momentary disconnection of the physical interface, or internal to the CPE, such as use of the IdleDisconnectTime and/or Auto-DisconnectTime parameters in this object.	"On-Demand"	1.0
RouteProtocolRx	string	W	Defines the Rx protocol to be used. Enumeration of: "Off" "RIPv1" (OPTIONAL) "RIPv2" (OPTIONAL) "OSPF" (OPTIONAL)	"Off"	1.0
PPPLCPEcho	unsignedInt	-	PPP LCP Echo period in seconds.	-	1.0
PPPLCPEchoRetry	unsignedInt	-	Number of PPP LCP Echo retries within an echo period.	-	1.0
ShapingRate	int[-1:]	W	Rate to shape this connection's egress traffic to. For leaky bucket (constant rate shaping), this is the constant rate. For token bucket (variable rate shaping), this is the average rate. If <= 100, in percent of the rate of the highest rate-constrained layer over which the packet will travel on egress. If > 100, in bits per second. A value of -1 indicates no shaping. For example, for packets destined for a WAN DSL interface, if the ATM layer is rate-constrained, then the rate is calculated relative to this rate. Otherwise, the rate is calculated relative to the physical-layer DSL rate.	-1	1.1
ShapingBurstSize	unsignedInt	W	Burst size in bytes. For both leaky bucket (constant rate shaping) and token bucket (variable rate shaping) this is the bucket size and is therefore the maximum burst size.	-	1.1

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
PortMappingNumberOfEntries	unsignedInt	-	Total number of port mapping entries.	-	1.0
InternetGatewayDevice.WANDevice.{i}.WAN-ConnectionDevice.{i}.WANPPPConnection.{i}.-PortMapping.{i}.	object	W	Port mapping table. This table MUST contain all NAT port mappings associated with this connection, including static and dynamic port mappings programmatically created via local control protocol, such as UPnP. This table MUST NOT contain dynamic NAT binding entries associated with the normal operation of NAT. At most one entry in an instance of this table can exist with all of the same values for RemoteHost, ExternalPort, and PortMappingProtocol. If the ACS attempts to set the parameters of an existing entry such that this requirement would be violated, the CPE MUST reject the request. In this case, the SetParameterValues response MUST include a SetParameterValuesFault element for each parameter in the corresponding request whose modification would have resulted in such a violation. On creation of a new table entry, the CPE MUST choose default values for ExternalPort and PortMappingProtocol such that the new entry does not conflict with any existing entry.	-	1.0
PortMappingEnabled	boolean	W	Enables or disables the port mapping instance. On creation, an entry is disabled by default.	False	1.0
PortMappingLeaseDuration	unsignedInt	W	Determines the time to live, in seconds, of a port-mapping lease, where “time to live” means the number of seconds before the port mapping expires. A value of 0 means the port mapping is static. Support for dynamic (non-static) port mappings is OPTIONAL. That is, the only value for PortMappingLeaseDuration that MUST be supported is 0. For a dynamic (non-static) port mapping, when this parameter is read, the value represents the time remaining on the port-mapping lease. That is, for a dynamic port mapping, the value counts down toward 0. When a dynamic port-mapping lease expires, the CPE MUST automatically terminate that port mapping, and MUST automatically delete the corresponding PortMapping table entry.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
RemoteHost	string	W	This parameter is the IP address of the source of inbound packets. An empty string indicates a 'wildcard' (this will be a wildcard in most cases). CPE are REQUIRED only to support wildcards. When RemoteHost is a wildcard, all traffic sent to the ExternalPort on the WAN interface of the gateway is forwarded to the InternalClient on the InternalPort. When RemoteHost is specified as one external IP address, the NAT will only forward inbound packets from this RemoteHost to the InternalClient, all other packets will be dropped. If a CPE supports non-wildcard values for RemoteHost, it MAY additionally support the ability to have more than one port mapping with the same ExternalPort and PortMappingProtocol, but with differing values of RemoteHost. When wildcard values are used for RemoteHost and/or ExternalPort, the following precedence order applies (with the highest precedence listed first): 1. Explicit RemoteHost, explicit ExternalPort 2. Explicit RemoteHost, wildcard ExternalPort 3. Wildcard RemoteHost, explicit ExternalPort 4. Wildcard RemoteHost, wildcard ExternalPort If an incoming packet matches the criteria associated with more than one entry in this table, the CPE MUST apply the port mapping associated with the highest precedence entry. At most one entry in this table can exist with all of the same values for RemoteHost, ExternalPort, and PortMappingProtocol.	<Empty>	1.0
ExternalPort	unsignedInt	W	The external port (or the first port of a range of external ports) that the NAT gateway would listen on for connection requests to a corresponding InternalPort. Inbound packets to this external port on the WAN interface SHOULD be forwarded to InternalClient on the InternalPort. A value of zero (0) represents a 'wildcard.' If this value is a wildcard, connection request on all external ports (that are not otherwise mapped) will be forwarded to InternalClient. In the wildcard case, the value(s) of InternalPort on InternalClient are ignored. When wildcard values are used for RemoteHost and/or ExternalPort, the following precedence order applies (with the highest precedence listed first): 1. Explicit RemoteHost, explicit ExternalPort 2. Explicit RemoteHost, wildcard ExternalPort 3. Wildcard RemoteHost, explicit ExternalPort 4. Wildcard RemoteHost, wildcard ExternalPort If an incoming packet matches the criteria associated with more than one entry in this table, the CPE MUST apply the port mapping associated with the highest precedence entry. At most one entry in this table can exist with all of the same values for RemoteHost, ExternalPort, and PortMappingProtocol.	-	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
ExternalPortEndRange	unsignedInt	W	Indicates the last port of the external port range that starts with ExternalPort. If an external port range is specified, then the behavior described for ExternalPort applies to all ports within the range. A value of zero (0) indicates that no external port range is specified, i.e. that the range consists only of ExternalPort. If ExternalPort is zero (wildcard), the value of this parameter MUST be ignored. If specified, the value of this parameter MUST be greater than or equal to the value of ExternalPort.	0	1.4
InternalPort	unsignedInt	W	The port on InternalClient that the gateway SHOULD forward connection requests to. A value of zero (0) is not allowed.	-	1.0
PortMappingProtocol	string	W	The protocol of the port mapping. Enumeration of: "TCP" "UDP" At most one entry in this table can exist with all of the same values for RemoteHost, ExternalPort, and PortMappingProtocol.	-	1.0
InternalClient	string(256)	W	The IP address or DNS host name of an internal client (on the LAN). Support for an IP address is mandatory. If InternalClient is specified as an IP address and the LAN device's IP address subsequently changes, the port mapping MUST remain associated with the original IP address. Support for DNS host names is OPTIONAL. If InternalClient is specified as a DNS host name and the LAN device's IP address subsequently changes, the port mapping MUST remain associated with this LAN device. In this case, it is the responsibility of the CPE to maintain the name-to-address mapping in the event of IP address changes. This can be accomplished, for example, by assigning the DNS host name via use of DHCP option 12 (Host Name) or option 81 (FQDN). Note that the ACS can learn the host name associated with a given LAN device via the Hosts table (InternetGatewayDevice.LANDevice.{i}.Hosts.). Read access to this parameter MUST always return the exact value that was last set by the ACS. For example, if the internal client is set to a DNS host name, it MUST read back as a DNS host name and not as an IP address. An empty string indicates an unconfigured InternalClient. If this parameter is unconfigured, this port mapping MUST NOT be operational. It MUST be possible to set the InternalClient to the broadcast IP address 255.255.255.255 for UDP mappings. This is to enable multiple NAT clients to use the same well-known port simultaneously.	<Empty>	1.0
PortMappingDescription	string(256)	W	User-readable description of this port mapping.	<Empty>	1.0

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.Stats.	object	-	This object contains statistics for all connections within the same WANConnectionDevice that share a common MAC address. The contents of this object SHOULD be identical for each such connection. This object is intended only for WANConnectionDevices that can support an Ethernet-layer on this interface (e.g., PPPoE, IPoE).	-	1.0
EthernetBytesSent	unsignedInt	-	The total number of bytes transmitted, including framing characters, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
EthernetBytesReceived	unsignedInt	-	The total number of bytes received, including framing characters, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
EthernetPacketsSent	unsignedInt	-	The total number of packets transmitted over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
EthernetPacketsReceived	unsignedInt	-	The total number of packets which were received over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.0
EthernetErrorsSent	unsignedInt	-	The total number of outbound packets that could not be transmitted because of errors, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetErrorsReceived	unsignedInt	-	The total number of inbound packets that contained errors preventing them from being deliverable, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetUnicastPacketsSent	unsignedInt	-	The total number of packets requested for transmission which were not addressed to a multicast or broadcast address, including those that were discarded or not sent, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetUnicastPacketsReceived	unsignedInt	-	The total number of received packets which were not addressed to a multicast or broadcast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

Name[1]	Type	Write[2]	Description	Object Default[3]	Version[4]
EthernetDiscardPacketsSent	unsignedInt	-	The total number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted, over all connections within the same WANConnectionDevice that share a common MAC address. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetDiscardPacketsReceived	unsignedInt	-	The total number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable, over all connections within the same WANConnectionDevice that share a common MAC address. One possible reason for discarding such a packet could be to free up buffer space. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetMulticastPacketsSent	unsignedInt	-	The total number of packets requested for transmission, including those that were discarded or not sent, which were addressed to a multicast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetMulticastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a multicast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetBroadcastPacketsSent	unsignedInt	-	The total number of packets requested for transmission, including those that were discarded or not sent, which were addressed to a broadcast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetBroadcastPacketsReceived	unsignedInt	-	The total number of received packets which were addressed to a broadcast address, over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4
EthernetUnknownProtoPacketsReceived	unsignedInt	-	The total number of packets which were discarded because of an unknown or unsupported protocol, received over all connections within the same WANConnectionDevice that share a common MAC address. The value of this counter MAY be reset to zero when the CPE is rebooted.	-	1.4

2.4.1 Inform and Notification Requirements

For an Internet Gateway Device, all of the parameters listed in Table 3 that are present in the data model implementation are REQUIRED on every Inform.

Table 3 – Forced Inform parameters for an Internet Gateway Device

Parameter
InternetGatewayDevice.DeviceSummary
InternetGatewayDevice.DeviceInfo.SpecVersion
InternetGatewayDevice.DeviceInfo.HardwareVersion
InternetGatewayDevice.DeviceInfo.SoftwareVersion
InternetGatewayDevice.DeviceInfo.ProvisioningCode
InternetGatewayDevice.ManagementServer.ConnectionRequestURL
InternetGatewayDevice.ManagementServer.ParameterKey
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{j}.WAN{***}Connection.{k}.ExternalIPAddress[5]

Active Notification MUST be enabled for all of the parameters listed in Table 4 that are present in the data model implementation, regardless of the value of the Notification Attribute for these parameters. As a result, any change in the value of these parameters due to an entity other than the ACS MUST result in the CPE initiating a connection to the ACS to issue the Inform method call.

Table 4 – Forced Active Notification parameters for an Internet Gateway Device

Parameter
InternetGatewayDevice.DeviceInfo.SoftwareVersion
InternetGatewayDevice.DeviceInfo.ProvisioningCode

[5] Where {i}, {j}, and {k} refer to the default WAN connection, and {***} is either "IP" or "PPP" depending on the type of connection.

Active Notification MUST be enabled by default for all of the parameters listed in Table 5 that are present in the data model implementation. The Notification attribute for each of these parameters MUST be reset to this default state whenever the CPE sends an Inform message indicating the "0 BOOTSTRAP" Event code.

Table 5 - Default Active Notification parameters for an Internet Gateway Device

Parameter
InternetGatewayDevice.ManagementServer.ConnectionRequestURL
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{j}.WAN{***}Connection.{k}.-ExternalIPAddress[6]

CPE MUST support Active Notification (see [2]) for all parameters defined in the InternetGatewayDevice data model with the exception of those parameters listed in Table 6. For only those parameters listed in Table 6, the CPE MAY reject a request by an ACS to enable Active Notification via the SetParameterAttributes RPC by responding with fault code 9009 as defined in [2] (Notification request rejected).

CPE MUST support Passive Notification (see [2]) for all parameters defined in the InternetGatewayDevice data model, with no exceptions.

Table 6 – Parameters for which Active Notification MAY be denied by the CPE

Parameter[7]
InternetGatewayDevice.DeviceInfo.
UpTime
DeviceLog
InternetGatewayDevice.ManagementServer.
ParameterKey
InternetGatewayDevice.ManagementServer.ManageableDevice.{i}.
ManufacturerOUI
SerialNumber
ProductClass
InternetGatewayDevice.Time.
CurrentLocalTime
InternetGatewayDevice.Layer2Bridging.
MaxBridgeEntries
MaxFilterEntries
MaxMarkingEntries
InternetGatewayDevice.Layer2Bridging.AvailableInterface.{i}.
AvailableInterfaceKey
InterfaceType
InterfaceReference

[6] The CPE MUST initiate an Inform whenever either the value of this parameter changes or the default WAN connection changes to a different connection.

[7] The name of a Parameter referenced in this table is the concatenation of the object name shown in the yellow header, and the individual Parameter name.

Parameter[7]
InternetGatewayDevice.QueueManagement.
MaxQueues
MaxClassificationEntries
MaxAppEntries
MaxFlowEntries
MaxPolicerEntries
MaxQueueEntries
InternetGatewayDevice.QueueManagement.Policer.{i}.
PossibleMeterTypes
CountedPackets
CountedBytes
TotalCountedPackets
TotalCountedBytes
ConformingCountedPackets
ConformingCountedBytes
PartiallyConformingCountedPackets
PartiallyConformingCountedBytes
NonConformingCountedPackets
NonConformingCountedBytes
InternetGatewayDevice.QueueManagement.QueueStats.{i}.
OutputPackets
OutputBytes
DroppedPackets
DroppedBytes
QueueOccupancyPackets
QueueOccupancyPercentage
InternetGatewayDevice.IPPingDiagnostics.
DiagnosticsState
SuccessCount
FailureCount
AverageResponseTime
MinimumResponseTime
MaximumResponseTime
InternetGatewayDevice.TraceRouteDiagnostics.
DiagnosticsState
ResponseTime
RouteHopsNumberOfEntries
InternetGatewayDevice.TraceRouteDiagnostics.RouteHops.{i}.
HopHost
HopHostAddress
HopErrorCode
HopRTTimes
InternetGatewayDevice.DownloadDiagnostics.
DiagnosticsState
ROMTime
BOMTime

Parameter[7]
EOMTime
TestBytesReceived
TotalBytesReceived
TCPOpenRequestTime
TCPOpenResponseTime
InternetGatewayDevice.UploadDiagnostics.
DiagnosticsState
ROMTime
BOMTime
EOMTime
TotalBytesSent
TCPOpenRequestTime
TCPOpenResponseTime
InternetGatewayDevice.UDPEchoConfig.
PacketsReceived
PacketsResponded
BytesReceived
BytesResponded
TimeFirstPacketReceived
TimeLastPacketReceived
InternetGatewayDevice.LANDevice.{i}.LANEthernetInterfaceConfig.{i}.Stats.
BytesSent
BytesReceived
PacketsSent
PacketsReceived
ErrorsSent
ErrorsReceived
UnicastPacketsSent
UnicastPacketsReceived
DiscardPacketsSent
DiscardPacketsReceived
MulticastPacketsSent
MulticastPacketsReceived
BroadcastPacketsSent
BroadcastPacketsReceived
UnknownProtoPacketsReceived
InternetGatewayDevice.LANDevice.{i}.LANUSBInterfaceConfig.{i}.Stats.
BytesSent
BytesReceived
CellsSent
CellsReceived
ErrorsSent
ErrorsReceived
UnicastPacketsSent
UnicastPacketsReceived
DiscardPacketsSent

Parameter[7]
DiscardPacketsReceived
MulticastPacketsSent
MulticastPacketsReceived
BroadcastPacketsSent
BroadcastPacketsReceived
UnknownProtoPacketsReceived
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.
TotalPSKFailures
TotalIntegrityFailures
ChannelsInUse
TotalBytesSent
TotalBytesReceived
TotalPacketsSent
TotalPacketsReceived
TotalAssociations
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.Stats.
ErrorsSent
ErrorsReceived
UnicastPacketsSent
UnicastPacketsReceived
DiscardPacketsSent
DiscardPacketsReceived
MulticastPacketsSent
MulticastPacketsReceived
BroadcastPacketsSent
BroadcastPacketsReceived
UnknownProtoPacketsReceived
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.AssociatedDevice.{i}.
AssociatedDeviceMACAddress
AssociatedDeviceIPAddress
AssociatedDeviceAuthenticationState
LastRequestedUnicastCipher
LastRequestedMulticastCipher
LastPMKId
LastDataTransmitRate
InternetGatewayDevice.LANDevice.{i}.Hosts.Host.{i}.
LeaseTimeRemaining
InternetGatewayDevice.WANDevice.{i}.WANCommonInterfaceConfig.
TotalBytesSent
TotalBytesReceived
TotalPacketsSent
TotalPacketsReceived
MaximumActiveConnections
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.
UpstreamCurrRate
DownstreamCurrRate

Parameter[7]
UpstreamMaxRate
DownstreamMaxRate
UpstreamNoiseMargin
DownstreamNoiseMargin
UpstreamAttenuation
DownstreamAttenuation
UpstreamPower
DownstreamPower
TotalStart
ShowtimeStart
LastShowtimeStart
CurrentDayStart
QuarterHourStart
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.Total.
ReceiveBlocks
TransmitBlocks
CellDelin
LinkRetrain
InitErrors
LInit
InitTimeouts
LossOfFraming
LOF
ErroredSecs
ATUCErroredSecs
SeverelyErroredSecs
ATUCSeverelyErroredSecs
FECErrors
ATUCFECErrors
HECErrors
ATUCHECErrors
CRCErrors
ATUCCRCErrors
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.Showtime.
ReceiveBlocks
TransmitBlocks
CellDelin
LinkRetrain
InitErrors
LInit
InitTimeouts
LossOfFraming
LOF
ErroredSecs
ATUCErroredSecs
SeverelyErroredSecs

Parameter[7]
ATUCSeverelyErroredSecs
FECErrors
ATUCFECErrors
HECErrors
ATUCHECErrors
CRCErrors
ATUCCRCErrors
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.LastShowtime.
ReceiveBlocks
TransmitBlocks
CellDelin
LinkRetrain
InitErrors
LInit
InitTimeouts
LossOfFraming
LOF
ErroredSecs
ATUCErroredSecs
SeverelyErroredSecs
ATUCSeverelyErroredSecs
FECErrors
ATUCFECErrors
HECErrors
ATUCHECErrors
CRCErrors
ATUCCRCErrors
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.CurrentDay.
ReceiveBlocks
TransmitBlocks
CellDelin
LinkRetrain
InitErrors
LInit
InitTimeouts
LossOfFraming
LOF
ErroredSecs
ATUCErroredSecs
SeverelyErroredSecs
ATUCSeverelyErroredSecs
FECErrors
ATUCFECErrors
HECErrors
ATUCHECErrors
CRCErrors

Parameter[7]
ATUCCRCErrors
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.QuarterHour.
ReceiveBlocks
TransmitBlocks
CellDelin
LinkRetrain
InitErrors
LInit
InitTimeouts
LossOfFraming
LOF
ErroredSecs
ATUCErroredSecs
SeverelyErroredSecs
ATUCSeverelyErroredSecs
FECErrors
ATUCFECErrors
HECErrors
ATUCHECErrors
CRCErrors
ATUCCRCErrors
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.TestParams.
HLOGGds
HLOGGus
HLOGpsds
HLOGpsus
HLOGMTds
HLOGMTus
QLNGds
QLNGus
QLNpsds
QLNpsus
QLNMTds
QLNMTus
SNRGds
SNRGus
SNRpsds
SNRpsus
SNRMTds
SNRMTus
LATNds
LATNus
SATNds
SATNus
InternetGatewayDevice.WANDevice.{i}.WANEthernetInterfaceConfig.Stats.
BytesSent

Parameter[7]
BytesReceived
PacketsSent
PacketsReceived
InternetGatewayDevice.WANDevice.{i}.WANDSLDiagnostics.
LoopDiagnosticsState
ACTPSDds
ACTPSDus
ACTATPds
ACTATPus
HLINSCds
HLINSCus
HLINGds
HLINGus
HLOGGds
HLOGGus
HLOGpsds
HLOGpsus
HLOGMTds
HLOGMTus
LATNpbds
LATNpbus
SATNds
SATNus
HLINpsds
HLINpsus
QLNGds
QLNGus
QLNpsds
QLNpsus
QLNMTds
QLNMTus
SNRGds
SNRGus
SNRpsds
SNRpsus
SNRMTds
SNRMTus
BITSpsds
BITSpsus
GAINSpsds
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANDSLLinkConfig.
ATMTransmittedBlocks
ATMReceivedBlocks
AAL5CRCErrors
ATMCRCErrors
ATMHECErrors

Parameter[7]
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANATMF5LoopbackDiagnostics.
DiagnosticsState
SuccessCount
FailureCount
AverageResponseTime
MinimumResponseTime
MaximumResponseTime
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPTMLinkConfig.Stats.
BytesSent
BytesReceived
FramesSent
FramesReceived
ErrorsSent
ErrorsReceived
UnicastPacketsSent
UnicastPacketsReceived
DiscardPacketsSent
DiscardPacketsReceived
MulticastPacketsSent
MulticastPacketsReceived
BroadcastPacketsSent
BroadcastPacketsReceived
UnknownProtoPacketsReceived
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.
Uptime
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.PortMapping.{i}.
PortMappingLeaseDuration
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.Stats.
EthernetBytesSent
EthernetBytesReceived
EthernetPacketsSent
EthernetPacketsReceived
EthernetErrorsSent
EthernetErrorsReceived
EthernetUnicastPacketsSent
EthernetUnicastPacketsReceived
EthernetDiscardPacketsSent
EthernetDiscardPacketsReceived
EthernetMulticastPacketsSent
EthernetMulticastPacketsReceived
EthernetBroadcastPacketsSent
EthernetBroadcastPacketsReceived
EthernetUnknownProtoPacketsReceived
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.
Uptime
CurrentMRUSize

Parameter[7]
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.PortMapping.{i}.
PortMappingLeaseDuration
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.Stats.
EthernetBytesSent
EthernetBytesReceived
EthernetPacketsSent
EthernetPacketsReceived
EthernetErrorsSent
EthernetErrorsReceived
EthernetUnicastPacketsSent
EthernetUnicastPacketsReceived
EthernetDiscardPacketsSent
EthernetDiscardPacketsReceived
EthernetMulticastPacketsSent
EthernetMulticastPacketsReceived
EthernetBroadcastPacketsSent
EthernetBroadcastPacketsReceived
EthernetUnknownProtoPacketsReceived

2.4.2 Version 1.0 Data Model Requirements

For version 1.0 of the Internet Gateway Device data model no profiles are defined because the profile mechanism was not supported by that version. However, the requirements for version 1.0 of the data model can easily be mapped to the profiles defined in section 3. Specifically:

- An implementation of version 1.0 of the InternetGatewayDevice data model MUST implement all of the objects and parameters in the Baseline:1 profile with the exception of the DeviceSummary parameter, which was not part of the version 1.0 data model.
- Each of the following profiles indicate objects and parameters that are conditionally REQUIRED in the version 1.0 data model:
  - EthernetLAN:1 (REQUIRED if CPE has a LAN-side Ethernet interface)
  - USBLAN:1 (REQUIRED if CPE has a LAN-side USB interface)
  - WiFiLAN:1 (REQUIRED if CPE has a LAN-side 802.11 interface)
  - ADSLWAN:1 (REQUIRED if CPE has a WAN-side ADSL interface)
  - EthernetWAN:1 (REQUIRED if CPE has a WAN-side Ethernet interface)
  - POTSWAN:1 (REQUIRED if CPE has a WAN-side POTS interface)
- Each of the following profiles indicate objects and parameters for which there are no specific requirements in the version 1.0 data model:
  - Time:1
  - IPPing:1
  - ATMLoopback:1
  - DSLDiagnostics:1
- All other objects and parameters associated with version 1.0 of the data model are considered OPTIONAL.

3 Profile Definitions

This section specifies the profiles defined for the Internet Gateway Device data model. The use of profiles for this data model follows the definition and usage conventions described in [3].

3.1 Notation

The following abbreviations are used to specify profile requirements:

Abbreviation	Description
R	Read support is REQUIRED.
W	Both Read and Write support is REQUIRED. This MUST NOT be specified for a parameter that is defined as read-only.
P	The object is REQUIRED to be present.
C	Creation and deletion of instances of the object via AddObject and DeleteObject is REQUIRED.
A	Creation of instances of the object via AddObject is REQUIRED, but deletion is not REQUIRED.
D	Deletion of instances of the object via DeleteObject is REQUIRED, but creation is not REQUIRED.

3.2 Baseline Profile

Table 7 defines the Baseline profile for the InternetGatewayDevice:1 object. The minimum REQUIRED versions for this profile are as follows:

- Baseline:1: InternetGatewayDevice:1.1
- Baseline:2: InternetGatewayDevice:1.4

Table 7 – Baseline profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.	P	P
DeviceSummary	R	R
LANDeviceNumberOfEntries	R	R
WANDeviceNumberOfEntries	R	R
InternetGatewayDevice.DeviceInfo.	P	P
Manufacturer	R	R
ManufacturerOUI	R	R
ModelName	R	R
Description	R	R
SerialNumber	R	R
HardwareVersion	R	R
SoftwareVersion	R	R
SpecVersion	R	R
ProvisioningCode	W	W
UpTime	R	R
DeviceLog	R	R
InternetGatewayDevice.ManagementServer.	P	P
URL	W	W
Username	W	W
Password	W	W
PeriodicInformEnable	W	W
PeriodicInformInterval	W	W

Name	Requirement (v1)	Requirement (v2)
PeriodicInformTime	W	W
ParameterKey	R	R
ConnectionRequestURL	R	R
ConnectionRequestUsername	W	W
ConnectionRequestPassword	W	W
UpgradesManaged	W	W
InternetGatewayDevice.Layer3Forwarding.	P	P
DefaultConnectionService	W	W
ForwardNumberOfEntries	R	R
InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}.	PC	PC
Enable	W	W
Status	R	R
StaticRoute	-	R
Type	W	W
DestIPAddress	W	W
DestSubnetMask	W	W
SourceIPAddress	W	W
SourceSubnetMask	W	W
GatewayIPAddress	W	W
Interface	W	W
ForwardingMetric	W	W
InternetGatewayDevice.LANConfigSecurity.	P	P
ConfigPassword	W	W
InternetGatewayDevice.LANDevice.{i}.	P	P
LANEthernetInterfaceNumberOfEntries	R	R
LANUSBInterfaceNumberOfEntries	R	R
LANWLANConfigurationNumberOfEntries	R	R
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.	P	P
MACAddress	-	R
DHCPServerConfigurable	W	W
DHCPServerEnable	W	W
DHCPRelay	R	R
MinAddress	W	W
MaxAddress	W	W
ReservedAddresses	W	W
SubnetMask	W	W
DNSServers	W	W
DomainName	W	W
IPRouters	W	W
IPInterfaceNumberOfEntries	R	R
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.IP-Interface.{i}.	P	P
Enable	W	W
IPInterfaceIPAddress	W	W
IPInterfaceSubnetMask	W	W
IPInterfaceAddressingType	W	W

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.LANDevice.{i}.Hosts.	P	P
HostNumberOfEntries	R	R
InternetGatewayDevice.LANDevice.{i}.Hosts.Host.{i}.	P	P
IPAddress	R	R
AddressSource	R	R
LeaseTimeRemaining	R	R
MACAddress	R	R
Layer2Interface	-	R
HostName	R	R
InterfaceType	R	R
Active	R	R
InternetGatewayDevice.LANInterfaces.	-	P
LANEthernetInterfaceNumberOfEntries	-	R
LANUSBInterfaceNumberOfEntries	-	R
LANWLANConfigurationNumberOfEntries	-	R
InternetGatewayDevice.WANDevice.{i}.	P	P
WANConnectionNumberOfEntries	R	R
InternetGatewayDevice.WANDevice.{i}.WANCommonInterfaceConfig.	P	P
EnabledForInternet	R	R
WANAccessType	R	R
Layer1UpstreamMaxBitRate	R	R
Layer1DownstreamMaxBitRate	R	R
PhysicalLinkStatus	R	R
TotalBytesSent	R	R
TotalBytesReceived	R	R
TotalPacketsSent	R	R
TotalPacketsReceived	R	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.	P	P
WANIPConnectionNumberOfEntries	R	R
WANPPPConnectionNumberOfEntries	R	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIP-Connection.{i}.	PC	PC
Enable	W	W
Reset	-	W
ConnectionStatus	R	R
PossibleConnectionTypes	R	R
ConnectionType	W[8]	W[8]
Name	W	W
Uptime	R	R
LastConnectionError	R	R
RSIPAvailable	R	R

[8] For writing, CPE are REQUIRED only to support the values that are listed in the corresponding PossibleConnectionTypes parameter.

Name	Requirement (v1)	Requirement (v2)
NATEnabled	W[9]	W[9]
AddressingType	R	R
ExternalIPAddress	R	R
SubnetMask	R	R
DefaultGateway	R	R
DNSEnabled	R	R
DNSOverrideAllowed	R	R
DNSServers	R	R
MACAddress	R	R
ConnectionTrigger	W	W
RouteProtocolRx	W	W
PortMappingNumberOfEntries	R	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIP-Connection.{i}.PortMapping.{i}.	PC	PC
PortMappingEnabled	W	W
PortMappingLeaseDuration	R	R
RemoteHost	W	W
ExternalPort	W	W
InternalPort	W	W
PortMappingProtocol	W	W
InternalClient	W	W
PortMappingDescription	W	W
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIP-Connection.{i}.Stats.	P[10]	P
EthernetBytesSent	R[10]	R[10]
EthernetBytesReceived	R[10]	R[10]
EthernetPacketsSent	R[10]	R[10]
EthernetPacketsReceived	R[10]	R[10]
EthernetErrorsSent	-	R[10]
EthernetErrorsReceived	-	R[10]
EthernetUnicastPacketsSent	-	R[10]
EthernetUnicastPacketsReceived	-	R[10]
EthernetDiscardPacketsSent	-	R[10]
EthernetDiscardPacketsReceived	-	R[10]
EthernetMulticastPacketsSent	-	R[10]
EthernetMulticastPacketsReceived	-	R[10]
EthernetBroadcastPacketsSent	-	R[10]
EthernetBroadcastPacketsReceived	-	R[10]
EthernetUnknownProtoPacketsReceived	-	R[10]
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPP-Connection.{i}.	PC	PC
Enable	W	W

[9] Write support for this parameter is REQUIRED only if NAT is supported by the CPE.
[10] Required only for WANConnectionDevice instances that are configured to support an Ethernet layer.

Name	Requirement (v1)	Requirement (v2)
Reset	-	W
ConnectionStatus	R	R
PossibleConnectionTypes	R	R
ConnectionType	W[8]	W[8]
PPPoESessionID	-	R
DefaultGateway	-	R
Name	W	W
Uptime	R	R
LastConnectionError	R	R
RSIPAvailable	R	R
NATEnabled	W[9]	W[9]
Username	W	W
Password	W	W
ExternalIPAddress	R	R
DNSEnabled	R	R
DNSOverrideAllowed	R	R
DNSServers	R	R
MACAddress	R	R
TransportType	R	R
PPPoEACName	W	W
PPPoEServiceName	W	W
ConnectionTrigger	W	W
RouteProtocolRx	W	W
PortMappingNumberOfEntries	R	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPP-Connection.{i}.PortMapping.{i}.	PC	PC
PortMappingEnabled	W	W
PortMappingLeaseDuration	R	R
RemoteHost	W	W
ExternalPort	W	W
InternalPort	W	W
PortMappingProtocol	W	W
InternalClient	W	W
PortMappingDescription	W	W
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPP-Connection.{i}.Stats.	P[10]	P[9]
EthernetBytesSent	R[10]	R[9]
EthernetBytesReceived	R[10]	R[9]
EthernetPacketsSent	R[10]	R[9]
EthernetPacketsReceived	R[10]	R[9]
EthernetErrorsSent	-	R[10]
EthernetErrorsReceived	-	R[10]
EthernetUnicastPacketsSent	-	R[10]
EthernetUnicastPacketsReceived	-	R[10]
EthernetDiscardPacketsSent	-	R[10]
EthernetDiscardPacketsReceived	-	R[10]

Name	Requirement (v1)	Requirement (v2)
EthernetMulticastPacketsSent	-	R[10]
EthernetMulticastPacketsReceived	-	R[10]
EthernetBroadcastPacketsSent	-	R[10]
EthernetBroadcastPacketsReceived	-	R[10]
EthernetUnknownProtoPacketsReceived	-	R[10]

3.3 EthernetLAN Profile

Table 8 defines the EthernetLAN profile for the InternetGatewayDevice:1 object. The minimum REQUIRED versions for this profile are as follows:

- EthernetLAN:1: InternetGatewayDevice:1.1
- EthernetLAN:2: InternetGatewayDevice:1.4

Table 8 – EthernetLAN profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.LANDevice.{i}.LANEthernetInterfaceConfig.{i}.	P	P
Enable	W	W
Status	R	R
Name	-	R
MACAddress	R	R
MACAddressControlEnabled	W[11]	W[11]
MaxBitRate	W	W
DuplexMode	W	W
InternetGatewayDevice.LANDevice.{i}.LANEthernetInterfaceConfig.{i}.Stats.	P	P
BytesSent	R	R
BytesReceived	R	R
PacketsSent	R	R
PacketsReceived	R	R
ErrorsSent	-	R
ErrorsReceived	-	R
UnicastPacketsSent	-	R
UnicastPacketsReceived	-	R
DiscardPacketsSent	-	R
DiscardPacketsReceived	-	R
MulticastPacketsSent	-	R
MulticastPacketsReceived	-	R
BroadcastPacketsSent	-	R
BroadcastPacketsReceived	-	R
UnknownProtoPacketsReceived	-	R

[11] Support for this parameter is REQUIRED only if the parameter InternetGatewayDevice.LANDevice.-{i}.LANHostConfigManagement.AllowedMACAddresses is present.

3.4 USBLAN Profile

Table 9 defines the USBLAN profile for the InternetGatewayDevice:1 object. The minimum REQUIRED versions for this profile are as follows:

- USBLAN:1: InternetGatewayDevice:1.1
- USBLAN:2: InternetGatewayDevice:1.4

Table 9 – USBLAN profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.LANDevice.{i}.LANUSBInterfaceConfig.{i}.	P	P
Enable	W	W
Status	R	R
Name	-	R
MACAddress	R	R
MACAddressControlEnabled	W[11]	W[11]
Standard	R	R
Type	R	R
Rate	R	R
Power	R	R
InternetGatewayDevice.LANDevice.{i}.LANUSBInterfaceConfig.{i}.Stats.	P	P
BytesSent	R	R
BytesReceived	R	R
CellsSent	R	R
CellsReceived	R	R
ErrorsSent	-	R
ErrorsReceived	-	R
UnicastPacketsSent	-	R
UnicastPacketsReceived	-	R
DiscardPacketsSent	-	R
DiscardPacketsReceived	-	R
MulticastPacketsSent	-	R
MulticastPacketsReceived	-	R
BroadcastPacketsSent	-	R
BroadcastPacketsReceived	-	R
UnknownProtoPacketsReceived	-	R

3.5 WiFiLAN Profile

Table 10 defines the WiFiLAN profile for the InternetGatewayDevice:1 object. The minimum REQUIRED versions for this profile are as follows:

- WiFiLAN:1: InternetGatewayDevice:1.1
- WiFiLAN:2: InternetGatewayDevice:1.4

Table 10 – WiFiLAN profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.	P	P
Enable	W	W
Status	R	R
Name	-	R
BSSID	R	R
MaxBitRate	W	W
Channel	W	W
AutoChannelEnable	-	W
SSID	W	W
BeaconType	W	W
MACAddressControlEnabled	W[11]	W[11]
Standard	R	R
WEPKeyIndex	W	W
KeyPassphrase	W	W
WEPEncryptionLevel	R	R
BasicEncryptionModes	W	W
BasicAuthenticationMode	W	W
WPAEncryptionModes	W	W
WPAAuthenticationMode	W	W
PossibleChannels	R	R
BasicDataTransmitRates	W	W
OperationalDataTransmitRates	W	W
PossibleDataTransmitRates	R	R
SSIDAdvertisementEnabled	-	W
RadioEnabled	W	W
TransmitPowerSupported	-	R
TransmitPower	-	W
AutoRateFallBackEnabled	W	W
TotalBytesSent	R	R
TotalBytesReceived	R	R
TotalPacketsSent	R	R
TotalPacketsReceived	R	R
TotalAssociations	R	R
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.Stats.	-	P
ErrorsSent	-	R
ErrorsReceived	-	R
UnicastPacketsSent	-	R
UnicastPacketsReceived	-	R

Name	Requirement (v1)	Requirement (v2)
DiscardPacketsSent	-	R
DiscardPacketsReceived	-	R
MulticastPacketsSent	-	R
MulticastPacketsReceived	-	R
BroadcastPacketsSent	-	R
BroadcastPacketsReceived	-	R
UnknownProtoPacketsReceived	-	R
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.-AssociatedDevice.{i}.	P	P
AssociatedDeviceMACAddress	R	R
AssociatedDeviceIPAddress	R	R
AssociatedDeviceAuthenticationState	R	R
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.WEPKey.{i}.	P	P
WEPKey	W	W
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.PreSharedKey.-{i}.	P	P
PreSharedKey	W	W
KeyPassphrase	W	W

3.6 WiFiWMM Profile

Table 11 defines the WiFiWMM:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 11 – WiFiWMM:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.	P
WMMSupported	R
UAPSDSupported	R
WMMEnable	W
UAPSDEnable	W
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.APWMMParameter.{i}.	P
AIFSN	W
ECWMin	W
ECWMax	W
TXOP	W
AckPolicy	W
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.STAWMMParameter.{i}.	P
AIFSN	W
ECWMin	W
ECWMax	W
TXOP	W
AckPolicy	W

3.7 WiFiWPS Profile

Table 12 defines the WiFiWPS:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 12 – WiFiWPS:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.WPS.	P
Enable	W
DeviceName	R
DevicePassword	W
UUID	R
Version	R
ConfigMethodsSupported	R
ConfigMethodsEnabled	W
SetupLockedState	R
SetupLock	W
ConfigurationState	R
LastConfigurationError	R
RegistrarNumberOfEntries	R
RegistrarEstablished	R
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.WPS.Registrar.{i}.	P
Enable	W
UUID	R
DeviceName	R

3.8 ADSLWAN Profile

Table 13 defines the ADSLWAN:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.1.

Note: This profile is valid for G.992.1 modems.

Table 13 – ADSLWAN:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.	P
Enable	W
Status	R
UpstreamCurrRate	R
DownstreamCurrRate	R
UpstreamMaxRate	R
DownstreamMaxRate	R
UpstreamNoiseMargin	R
DownstreamNoiseMargin	R
UpstreamAttenuation	R
DownstreamAttenuation	R
UpstreamPower	R
DownstreamPower	R
ATURVendor	R

Name	Requirement
ATURCountry	R
ATUCVendor	R
ATUCCountry	R
TotalStart	R
ShowtimeStart	R
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.	P
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.Total.	P
ReceiveBlocks	R
TransmitBlocks	R
CellDelin	R
LinkRetrain	R
InitErrors	R
InitTimeouts	R
LossOfFraming	R
ErroredSecs	R
SeverelyErroredSecs	R
FECErrors	R
ATUCFECErrors	R
HECErrors	R
ATUCHECErrors	R
CRCErrors	R
ATUCCRCErrors	R
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.Stats.Showtime.	P
ReceiveBlocks	R
TransmitBlocks	R
CellDelin	R
LinkRetrain	R
InitErrors	R
InitTimeouts	R
LossOfFraming	R
ErroredSecs	R
SeverelyErroredSecs	R
FECErrors	R
ATUCFECErrors	R
HECErrors	R
ATUCHECErrors	R
CRCErrors	R
ATUCCRCErrors	R
InternetGatewayDevice.WANDevice.{i}.WANDSLConnectionManagement.	P
ConnectionServiceNumberOfEntries	R
InternetGatewayDevice.WANDevice.{i}.WANDSLConnectionManagement.ConnectionService.{i}.	P
WANConnectionDevice	R
WANConnectionService	R
DestinationAddress	R
LinkType	R
ConnectionType	R

Name	Requirement
Name	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.	PC
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANDSLLinkConfig.	P
Enable	W
LinkStatus	R
LinkType	W[12]
AutoConfig	R
DestinationAddress	W
ATMTransmittedBlocks	R
ATMReceivedBlocks	R
AAL5CRCErrors	R
ATMCRCErrors	R

3.9 ADSL2WAN Profile

Table 14 defines the ADSL2WAN:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4

Note: This profile is valid for G.992.3 and G.992.5 modems.

Table 14 – ADSL2WAN:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.	P
Enable	W
Status	R
LinkEncapsulationSupported	R
LinkEncapsulationRequested	W
LinkEncapsulationUsed	R
StandardsSupported	R
StandardUsed	R
UpstreamMaxRate	R
DownstreamMaxRate	R
UpstreamNoiseMargin	R
DownstreamNoiseMargin	R
UpstreamPower	R
DownstreamPower	R
TotalStart	R
ShowtimeStart	R
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.TestParams.	P
HLOGpsds	R
HLOGpsus	R
HLOGMTds	R
HLOGMTus	R

[12] For writing, CPE need not to support values for this parameter that correspond to modes of operation that are not supported by the CPE.

Name	Requirement
QLNpsds	R
QLNpsus	R
QLNMTds	R
QLNMTus	R
SNRpsds	R
SNRpsus	R
SNRMTds	R
SNRMTus	R
LATNds	R
LATNus	R
SATNds	R
SATNus	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.	PC
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANDSLLinkConfig.	P
Enable	W
LinkStatus	R
LinkType	W[13]
AutoConfig	R
DestinationAddress	W
ATMTransmittedBlocks	R
ATMReceivedBlocks	R
ATMCRCErrors	R

3.10 VDSL2WAN Profile

Table 15 defines the VDSL2WAN:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4

Table 15 – VDSL2WAN:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.	P
Enable	W
Status	R
LinkEncapsulationSupported	R
LinkEncapsulationRequested	W
LinkEncapsulationUsed	R
StandardsSupported	R
StandardUsed	R
AllowedProfiles	R
CurrentProfile	R
UPBOKLE	R
UpstreamMaxRate	R

[13] For writing, CPE need not to support values for this parameter that correspond to modes of operation that are not supported by the CPE.

Name	Requirement
DownstreamMaxRate	R
UpstreamNoiseMargin	R
DownstreamNoiseMargin	R
UpstreamAttenuation	R
DownstreamAttenuation	R
UpstreamPower	R
DownstreamPower	R
TRELLISds	R
TRELLISus	R
ACTSNRMODEds	R
ACTSNRMODEus	R
ACTUALCE	R
SNRMpbds	R
SNRMpbus	R
TotalStart	R
ShowtimeStart	R
InternetGatewayDevice.WANDevice.{i}.WANDSLInterfaceConfig.TestParams.	P
HLOGGds	R
HLOGGus	R
HLOGpsds	R
HLOGpsus	R
HLOGMTds	R
HLOGMTus	R
QLNGds	R
QLNGus	R
QLNpsds	R
QLNpsus	R
QLNMTds	R
QLNMTus	R
SNRGds	R
SNRGus	R
SNRpsds	R
SNRpsus	R
SNRMTds	R
SNRMTus	R
LATNds	R
LATNus	R
SATNds	R
SATNus	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.	PC
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANDSLLinkConfig.	P
Enable	W

Name	Requirement
LinkStatus	R
LinkType	W[14]
AutoConfig	R
DestinationAddress	W
ATMTransmittedBlocks	R
ATMReceivedBlocks	R
ATMCRCErrors	R

3.11 PTMWAN Profile

Table 16 defines the PTMWAN:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 16 – PTMWAN:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPTMLinkConfig.	P
Enable	W
LinkStatus	R
MACAddress	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPTMLinkConfig.Stats.	P
BytesSent	R
BytesReceived	R
FramesSent	R
FramesReceived	R
OOSNearEnd	R
OOSFarEnd	R
ErrorsSent	R
ErrorsReceived	R
UnicastPacketsSent	R
UnicastPacketsReceived	R
DiscardPacketsSent	R
DiscardPacketsReceived	R
MulticastPacketsSent	R
MulticastPacketsReceived	R
BroadcastPacketsSent	R
BroadcastPacketsReceived	R
UnknownProtoPacketsReceived	R

[14] For writing, CPE need not to support values for this parameter that correspond to modes of operation that are not supported by the CPE.

3.12 EthernetWAN Profile

Table 17 defines the EthernetWAN:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.1.

Table 17 – EthernetWAN:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANEthernetInterfaceConfig.	P
Enable	W
Status	R
MACAddress	R
MaxBitRate	W
DuplexMode	W
InternetGatewayDevice.WANDevice.{i}.WANEthernetInterfaceConfig.Stats.	P
BytesSent	R
BytesReceived	R
PacketsSent	R
PacketsReceived	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANEthernetLinkConfig.	P
EthernetLinkStatus	R

3.13 POTSWAN Profile

Table 18 defines the POTSWAN:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.1.

Table 18 – POTSWAN:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPOTSLinkConfig.	P
Enable	W
LinkStatus	R
ISPPhoneNumber	R
ISPInfo	R
LinkType	R
NumberOfRetries	R
DelayBetweenRetries	R

3.14 QoS Profile

Table 19 defines the QoS profile for the InternetGatewayDevice:1 object. The minimum REQUIRED versions for this profile are:

- QoS:1: InternetGatewayDevice:1.1
- QoS:2: InternetGatewayDevice:1.4

Table 19 – QoS profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.QueueManagement.	P	P
Enable	W	W
MaxQueues	R	R
MaxClassificationEntries	R	R
ClassificationNumberOfEntries	R	R
MaxAppEntries	R	R
AppNumberOfEntries	R	R
MaxFlowEntries	R	R
FlowNumberOfEntries	R	R
MaxPolicerEntries	R	R
PolicerNumberOfEntries	R	R
MaxQueueEntries	R	R
QueueNumberOfEntries	R	R
DefaultForwardingPolicy	W	W
DefaultTrafficClass	-	W
DefaultPolicer	W	W
DefaultQueue	W	W
DefaultDSCPMark	W	W
DefaultEthernetPriorityMark	W	W
AvailableAppList	R	R
InternetGatewayDevice.QueueManagement.Classification.{i}.	PC	PC
ClassificationKey	R	-[15]
ClassificationEnable	W	W
ClassificationStatus	R	R
ClassificationOrder	W	W
ClassInterface	W	W
DestIP	W	W
DestMask	W	W
DestIPExclude	W	W
SourceIP	W	W
SourceMask	W	W
SourceIPExclude	W	W
Protocol	W	W
ProtocolExclude	W	W

[15] This parameter is OBSOLETED.

Name	Requirement (v1)	Requirement (v2)
DestPort	W	W
DestPortRangeMax	W	W
DestPortExclude	W	W
SourcePort	W	W
SourcePortRangeMax	W	W
SourcePortExclude	W	W
SourceMACAddress	W	W
SourceMACExclude	W	W
DestMACAddress	W	W
DestMACExclude	W	W
DSCPCheck	W	W
DSCPExclude	W	W
DSCPMark	W	W
EthernetPriorityCheck	W	W
EthernetPriorityExclude	W	W
EthernetPriorityMark	W	W
VLANIDCheck	W	W
VLANIDExclude	W	W
ForwardingPolicy	W	W
TrafficClass	-	W
ClassPolicer	W	W
ClassQueue	W	W
InternetGatewayDevice.QueueManagement.Policer.{i}.	PC	PC
PolicerKey	R	-[15]
PolicerEnable	W	W
PolicerStatus	R	R
CommittedRate	W	W
CommittedBurstSize	W	W
ExcessBurstSize	-	W
PeakRate	-	W
PeakBurstSize	-	W
MeterType	W	W
PossibleMeterTypes	R	R
ConformingAction	W	W
PartialConformingAction	-	W
NonConformingAction	W	W
CountedPackets	R	R
CountedBytes	R	R
InternetGatewayDevice.QueueManagement.Queue.{i}.	PC	PC
QueueKey	R	-[15]
QueueEnable	W	W
QueueStatus	R	R
TrafficClasses	-	W
QueueInterface	W	W
QueueBufferLength	R	R

Name	Requirement (v1)	Requirement (v2)
QueueWeight	W	W
QueuePrecedence	W	W
REDThreshold	W	W
REDPercentage	W	W
DropAlgorithm	W	W
SchedulerAlgorithm	W	W
ShapingRate	W	W
ShapingBurstSize	W	W
InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}.	-	-
ForwardingPolicy	W	W
InternetGatewayDevice.WANDevice.{i}.WANEthernetInterfaceConfig.	-	-
ShapingRate	-	W
ShapingBurstSize	-	W
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIP-Connection.{i}.	-	-
ShapingRate	W	W
ShapingBurstSize	W	W
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPP-Connection.{i}.	-	-
ShapingRate	W	W
ShapingBurstSize	W	W

3.15 QoSDynamicFlow Profile

Table 20 defines the QoSDynamicFlow:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED versions for this profile are:

- QoSDynamicFlow:1: InternetGatewayDevice:1.1
- QoSDynamicFlow:2: InternetGatewayDevice:1.4

Table 20 – QoSDynamicFlow profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.QueueManagement.App.{i}.	PC	PC
AppKey	R	-[15]
AppEnable	W	W
AppStatus	R	R
ProtocolIdentifier	W	W
AppName	W	W
AppDefaultForwardingPolicy	W	W
AppDefaultTrafficClass	-	W
AppDefaultPolicer	W	W
AppDefaultQueue	W	W
AppDefaultDSCPMark	W	W
AppDefaultEthernetPriorityMark	W	W
InternetGatewayDevice.QueueManagement.Flow.{i}.	PC	PC
FlowKey	R	-[15]

Name	Requirement (v1)	Requirement (v2)
FlowEnable	W	W
FlowStatus	R	R
FlowType	W	W
FlowTypeParameters	W	W
FlowName	W	W
AppIdentifier	W	W
FlowForwardingPolicy	W	W
FlowTrafficClass	-	W
FlowPolicer	W	W
FlowQueue	W	W
FlowDSCPMark	W	W
FlowEthernetPriorityMark	W	W
InternetGatewayDevice.QueueManagement.Classification.{i}.	-	-
ClassApp	W	W

3.16 QoSStats Profile

Table 21 defines the QoSStats:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 21 – QoSStats:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.QueueManagement.	P
QueueStatsNumberOfEntries	R
InternetGatewayDevice.QueueManagement.Policer.{i}.	-
TotalCountedPackets	R
TotalCountedBytes	R
ConformingCountedPackets	R
ConformingCountedBytes	R
NonConformingCountedPackets	R
NonConformingCountedBytes	R
InternetGatewayDevice.QueueManagement.QueueStats.{i}.	PC
Enable	W
Status	R
Queue	W
Interface	W
OutputPackets	R
OutputBytes	R
DroppedPackets	R
DroppedBytes	R
QueueOccupancyPackets	R
QueueOccupancyPercentage	R

3.17 Bridging Profile

Table 22 defines the Bridging profile for the InternetGatewayDevice:1 object. The minimum REQUIRED versions for this profile are:

- Bridging:1: InternetGatewayDevice:1.1
- Bridging:2: InternetGatewayDevice:1.4

Table 22 – Bridging profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.Layer2Bridging.	P	P
MaxBridgeEntries	R	R
MaxDBridgeEntries	-	R
MaxQBridgeEntries	-	R
MaxFilterEntries	R	R
MaxMarkingEntries	R	R
BridgeNumberOfEntries	R	R
FilterNumberOfEntries	R	R
MarkingNumberOfEntries	R	R
AvailableInterfaceNumberOfEntries	R	R
InternetGatewayDevice.Layer2Bridging.Bridge.{i}.	PC	PC
BridgeKey	R	R
BridgeStandard	-	W
BridgeEnable	W	W
BridgeStatus	R	R
BridgeName	W	W
VLANID	W	W
InternetGatewayDevice.Layer2Bridging.Filter.{i}.	PC	PC
FilterKey	R	R
FilterEnable	W	W
FilterStatus	R	R
FilterBridgeReference	W	W
ExclusivityOrder	W	W
FilterInterface	W	W
VLANIDFilter	W	W
AdmitOnlyVLANTagged	W	W
EthertypeFilterList	W	W
EthertypeFilterExclude	W	W
SourceMACAddressFilterList	W	W
SourceMACAddressFilterExclude	W	W
DestMACAddressFilterList	W	W
DestMACAddressFilterExclude	W	W
InternetGatewayDevice.Layer2Bridging.Marking.{i}.	PC	PC
MarkingKey	R	R
MarkingEnable	W	W
MarkingStatus	R	R
MarkingBridgeReference	W	W
MarkingInterface	W	W

Name	Requirement (v1)	Requirement (v2)
VLANIDUntag	W	W
VLANIDMark	W	W
VLANIDMarkOverride	-	W
EthernetPriorityMark	W	W
EthernetPriorityOverride	W	W
InternetGatewayDevice.Layer2Bridging.AvailableInterface.{i}.	P	P
AvailableInterfaceKey	R	R
InterfaceType	R	R
InterfaceReference	R	R

3.18 BridgingPortVLAN Profile

Table 23 defines the BridgingPortVLAN:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 23 – BridgingPortVLAN:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.Layer2Bridging.	P
MaxVLANEntries	R
InternetGatewayDevice.Layer2Bridging.Bridge.{i}.	P
PortNumberOfEntries	R
VLANNumberOfEntries	R
InternetGatewayDevice.Layer2Bridging.Bridge.{i}.Port.{i}.	PC
PortEnable	W
PortInterface	W
PortState	R
PVID	W
AcceptableFrameTypes	W
IngressFiltering	W
InternetGatewayDevice.Layer2Bridging.Bridge.{i}.VLAN.{i}.	PC
VLANEnable	W
VLANName	W
VLANID	W

3.19 Time Profile

Table 24 defines the Time profile for the InternetGatewayDevice:1 object. The minimum REQUIRED versions for this profile are:

- Time:1: InternetGatewayDevice:1.1
- Time:2: InternetGatewayDevice:1.4

Table 24 – Time profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.Time.	P	P
Enable	-	W

Name	Requirement (v1)	Requirement (v2)
Status	-	R
NTPServer1	W	W
NTPServer2	W	W
CurrentLocalTime	R	R
LocalTimeZone	W	-[16]
LocalTimeZoneName	W	W
DaylightSavingsUsed	W	-[16]
DaylightSavingsStart	W	-[16]
DaylightSavingsEnd	W	-[16]

3.20 CaptivePortal Profile

Table 25 defines the CaptivePortal:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 25 – CaptivePortal:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.CaptivePortal.	P
Enable	W
Status	R
AllowedList	W
CaptivePortalURL	W

3.21 IPPing Profile

Table 26 defines the IPPing:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.1.

Table 26 – IPPing:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.IPPingDiagnostics.	P
DiagnosticsState	W
Interface	W
Host	W
NumberOfRepetitions	W
Timeout	W
DataBlockSize	W
DSCP	W
SuccessCount	R
FailureCount	R
AverageResponseTime	R
MinimumResponseTime	R
MaximumResponseTime	R

[16] This parameter is OBSOLETED.

3.22 TraceRoute Profile

Table 27 defines the TraceRoute:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 27 – TraceRoute:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.TraceRouteDiagnostics.	P
DiagnosticsState	W
Interface	W
Host	W
NumberOfTries	W
Timeout	W
DataBlockSize	W
DSCP	W
MaxHopCount	W
ResponseTime	R
RouteHopsNumberOfEntries	R
InternetGatewayDevice.TraceRouteDiagnostics.RouteHops.{i}.	P
HopHost	R
HopHostAddress	R
HopErrorCode	R
HopRTTimes	R

3.23 Download Profile

Table 28 defines the Download:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.3.

Table 28 – Download:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.Capabilities. PerformanceDiagnostic.	P
DownloadTransports	R
InternetGatewayDevice.DownloadDiagnostics.	P
DiagnosticsState	W
Interface	W
DownloadURL	W
DSCP	W
EthernetPriority	W
ROMTime	R
BOMTime	R
EOMTime	R
TestBytesReceived	R
TotalBytesReceived	R

3.24 DownloadTCP Profile

Table 29 defines the DownloadTCP:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.3.

Table 29 – DownloadTCP:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.DownloadDiagnostics.	P
TCPOpenRequestTime	R
TCPOpenResponseTime	R

3.25 Upload Profile

Table 30 defines the Upload:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.3.

Table 30 – Upload:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.Capabilities. PerformanceDiagnostic.	P
UploadTransports	R
InternetGatewayDevice.UploadDiagnostics.	P
DiagnosticsState	W
Interface	W
UploadURL	W
DSCP	W
EthernetPriority	W
ROMTime	R
BOMTime	R
EOMTime	R
TestFileLength	R
TotalBytesSent	R

3.26 UploadTCP Profile

Table 31 defines the UploadTCP:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.3.

Table 31 – UploadTCP:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.UploadDiagnostics.	P
TCPOpenRequestTime	R
TCPOpenResponseTime	R

3.27 UDPEcho Profile

Table 32 defines the UDPEcho:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.3.

Table 32 – UDPEcho:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.UDPEchoConfig.	P
Enable	W
Interface	W
SourceIPAddress	W
UDPPort	W
PacketsReceived	R
PacketsResponded	R
BytesReceived	R
BytesResponded	R
TimeFirstPacketReceived	R
TimeLastPacketReceived	R
EchoPlusSupported	R

3.28 UDPEchoPlus Profile

Table 33 defines the UDPEchoPlus:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.3.

Table 33 – UDPEchoPlus:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.UDPEchoConfig.	P
EchoPlusEnabled	W

3.29 ATMLoopback Profile

Table 34 defines the ATMLoopback:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.1.

Table 34 – ATMLoopback:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANATMF5Loopback-Diagnostics.	P
DiagnosticsState	W
NumberOfRepetitions	W

Name	Requirement
Timeout	W
SuccessCount	R
FailureCount	R
AverageResponseTime	R
MinimumResponseTime	R
MaximumResponseTime	R

3.30 DSLDiagnostics Profile

Table 35 defines the DSLDiagnostics:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.1.

Note: This profile is valid for G.992.1 modems.

Table 35 – DSLDiagnostics:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANDSLDiagnostics.	P
LoopDiagnosticsState	W
ACTPSDds	R
ACTPSDus	R
ACTATPds	R
ACTATPus	R
HLINSCds	R
HLINpsds	R
QLNpsds	R
SNRpsds	R
BITSpsds	R
GAINSpsds	R

3.31 ADSL2DSLDiagnostics Profile

Table 36 defines the ADSL2DSLDiagnostics:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4

Note: This profile is valid for G.992.3 and G.992.5 modems.

Table 36– ADSL2DSLDiagnostics:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANDSLDiagnostics.	P
LoopDiagnosticsState	W
ACTPSDds	R
ACTPSDus	R
ACTATPds	R
ACTATPus	R
HLINSCds	R
HLINSCus	R
HLINpsds	R
HLINpsus	R

Name	Requirement
HLOGpsds	R
HLOGpsus	R
HLOGMTds	R
HLOGMTus	R
QLNpsds	R
QLNpsus	R
QLNMTds	R
QLNMTus	R
SNRpsds	R
SNRpsus	R
SNRMTds	R
SNRMTus	R
LATNpbds	R
LATNpbus	R
SATNds	R
SATNus	R

3.32 VDSL2DSLDiagnostics Profile

Table 37 defines the VDSL2DSLDiagnostics:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4

Note: This profile is valid for G.993.2 modems.

Table 37 – VDSL2DSLDiagnostics:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.WANDevice.{i}.WANDSLDiagnostics.	P
LoopDiagnosticsState	W
ACTPSDds	R
ACTPSDus	R
ACTATPds	R
ACTATPus	R
HLINSCds	R
HLINSCus	R
HLINGds	R
HLINGus	R
HLINpsds	R
HLINpsus	R
HLOGGds	R
HLOGGus	R
HLOGpsds	R
HLOGpsus	R
HLOGMTds	R
HLOGMTus	R
QLNGds	R
QLNGus	R
QLNpsds	R

Name	Requirement
QLNpsus	R
QLNMTds	R
QLNMTus	R
SNRGds	R
SNRGus	R
SNRpsds	R
SNRpsus	R
SNRMTds	R
SNRMTus	R
LATNpbds	R
LATNpbus	R
SATNds	R
SATNus	R

3.33 DeviceAssociation Profile

The DeviceAssociation profile implies support for all of the Gateway requirements defined in Annex F of [2], including the support for the data model parameters as shown in Table 38. The minimum REQUIRED versions for this profile are as follows:

- DeviceAssociation:1: InternetGatewayDevice:1.2
- DeviceAssociation:2: InternetGatewayDevice:1.4

Table 38 – DeviceAssociation Profile definition for InternetGatewayDevice:1

Name	Requirement (v1)	Requirement (v2)
InternetGatewayDevice.ManagementServer.	-	-
ManageableDeviceNumberOfEntries	R	R
InternetGatewayDevice.ManagementServer.ManageableDevice.{i}.	P	P
ManufacturerOUI	R	R
SerialNumber	R	R
ProductClass	R	R
Host	-	R

3.34 UDPConnReq Profile

The UDPConnReq:1 profile for an Internet Gateway Device implies support for all of the CPE requirements defined in Annex G of [2], including support for the data model parameters as shown in Table 39. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.2.

This profile only applies to Internet Gateway Devices that are acting as CPE behind a NAT gateway as described in Annex G of [2].

Table 39 – UDPConnReq:1 Profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.ManagementServer.	-
UDPConnectionRequestAddress	R
UDPConnectionRequestAddressNotificationLimit	W
STUNEnable	W

Name	Requirement
STUNServerAddress	W
STUNServerPort	W
STUNUsername	W
STUNPassword	W
STUNMaximumKeepAlivePeriod	W
STUNMinimumKeepAlivePeriod	W
NATDetected	R

3.35 DHCPCondServing Profile

Table 40 defines the DHCPCondServing:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 40 – DHCPCondServing:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.	P
DHCPConditionalPoolNumberOfEntries	R
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.DHCPConditionalServing-Pool.{i}.	PC
Enable	W
PoolOrder	W
SourceInterface	W
VendorClassID	W
ClientID	W
UserClassID	W
Chaddr	W
ChaddrMask	W
LocallyServed	W
MinAddress	W
MaxAddress	W
ReservedAddresses	W
SubnetMask	W
DNSServers	W
DomainName	W
IPRouters	W
DHCPLeaseTime	W
DHCPServerIPAddress	W

3.36 DHCPOption Profile

Table 41 defines the DHCPOption:1 profile for the InternetGatewayDevice:1 object. The minimum REQUIRED version for this profile is InternetGatewayDevice:1.4.

Table 41 – DHCPOption:1 profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.	P
DHCPOptionNumberOfEntries	R

Name	Requirement
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.DHCPOption.{i}.	PC
Enable	W
Tag	W
Value	W
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.DHCPConditionalServing-Pool.{i}.	PC
DHCPOptionNumberOfEntries	R
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.DHCPConditionalServing-Pool.{i}.DHCPOption.{i}.	PC
Enable	W
Tag	W
Value	W
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.DHCP-Client.	P
SentDHCPOptionNumberOfEntries	R
ReqDHCPOptionNumberOfEntries	R
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.DHCP-Client.SentDHCPOption.{i}.	PC[17]
Enable	W
Tag	W
Value	W
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.DHCP-Client.ReqDHCPOption.{i}.	PC[18]
Enable	W
Tag	W
Value	R

[17] This table is REQUIRED to support sending of option 60 (Vendor Class Identifier) and option 77 (User Class Identifier) values.

[18] This table is REQUIRED to support requesting of option 60 (Vendor Class Identifier), option 61 (Client Identifier) and option 77 (User Class Identifier) values.

Normative References

The following documents are referenced by this specification. A list of currently valid Broadband Forum Technical Reports is published at http://www.broadband-forum.org.

[1] RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt

[2] TR-069 Amendment 2, *CPE WAN Management Protocol*, DSL Forum Technical Report

[3] TR-106 Amendment 1, *Data Model Template for TR-069-Enabled Devices*, DSL Forum Technical Report

[4] *Simple Object Access Protocol (SOAP) 1.1*, http://www.w3.org/TR/2000/NOTE-SOAP-20000508

[5] *Organizationally Unique Identifiers (OUIs)*, http://standards.ieee.org/faqs/OUI.html

[6] RFC 2616, *Hypertext Transfer Protocol -- HTTP/1.1*, http://www.ietf.org/rfc/rfc2616.txt

[7] *HTML 4.01 Specification*, http://www.w3.org/TR/html4

[8] RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, http://www.ietf.org/rfc/rfc3986.txt

[9] RFC 3489, *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, http://www.ietf.org/rfc/rfc3489.txt

[10] *References on RED (Random Early Detection) Queue Management*, http://www.icir.org/floyd/red.html

[11] *Blue: A New Class of Active Queue Management Algorithms*, http://www.thefengs.com/wuchang/work/blue

[12] *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, http://www.w3.org/TR/REC-xml

[13] IEEE Std 802.1D-2004, *Media Access Control (MAC) Bridges*, June 9, 2004, available from http://standards.ieee.org/getieee802/download/802.1D-2004.pdf

[14] IEEE Std 802.1Q-2005, *Virtual Bridged Local Area Networks*, May 19, 2006, available from http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf.

[15] ITU-T Recommendation G.997.1, *Physical Layer Management for digital subscriber line (DSL) transceivers, revision 3*, to be published in 2006, earlier version available at http://www.itu.int/rec/T-REC-G.997.1-200305-S/E

[16] ITU-T Recommendation G.991.1, *High bit rate Digital Subscriber Line (DSL) transceivers*, Oct 1998, available at http://www.itu.int/rec/T-REC-G.991.1/en

[17] ITU-T Recommendation G.991.2, *Single-pair high-speed digital subscriber line (SHDSL) transceivers*, Dec 2003, available at http://www.itu.int/rec/T-REC-G.991.2/en

[18] ITU-T Recommendation G.992.1, *Asymmetric digital subscriber line (ADSL) transceivers*, Jul 1999, available at http://www.itu.int/rec/T-REC-G.992.1/en

[19] ITU-T Recommendation G.992.2, *Splitterless asymmetric digital subscriber line (ADSL) transceivers*, Jul 1999, available at http://www.itu.int/rec/T-REC-G.992.2/en

[20] ITU-T Recommendation G.992.3, *Asymmetric digital subscriber line transceivers 2 (ADSL2)*, Jan 2005, available at http://www.itu.int/rec/T-REC-G.992.3/en

[21] ITU-T Recommendation G.992.4, *Splitterless asymmetric digital subscriber line transceivers 2 (splitterless ADSL2)*, July 2002, available at http://www.itu.int/rec/T-REC-G.992.4/en

[22] ITU-T Recommendation G.992.5, *Asymmetric Digital Subscriber Line (ADSL) transceivers – Extended bandwidth ADSL2 (ADSL2plus)*, Jan 2005, available at http://www.itu.int/rec/T-REC-G.992.5/en

[23] ITU-T Recommendation G.993.1, *Very high speed digital subscriber line transceivers*, Jun 2004, available at http://www.itu.int/rec/T-REC-G.993.1/en

[24] ITU-T Recommendation G.993.2, *Very high speed digital subscriber line transceivers 2 (VDSL2)*, Feb 2006, available at http://www.itu.int/rec/T-REC-G.993.2/en

[25] RFC 862, *Echo Protocol*, http://www.ietf.org/rfc/rfc862.txt

[26] RFC 959, *File Transfer Protocol*, http://www.ietf.org/rfc/rfc959.txt

[27] RFC 2131, *Dynamic Host Configuration Protocol*, http://tools.ietf.org/rfc/rfc2131.txt

[28] RFC 2132, *DHCP Options and BOOTP Vendor Extensions*, http://tools.ietf.org/rfc/rfc2132.txt

[29] RFC 2225, *Classical IP and ARP over ATM*, http://tools.ietf.org/rfc/rfc2225.txt

[30] RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, http://tools.ietf.org/rfc/rfc2474.txt

[31] RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*, http://tools.ietf.org/rfc/rfc2516.txt

[32] RFC 2597, *Assured Forwarding PHB Group*, http://tools.ietf.org/rfc/rfc2597.txt

[33] RFC 2634, *Enhanced Security Services for S/MIME*, http://tools.ietf.org/rfc/rfc2634.txt

[34] RFC 2662, *Definitions of Managed Objects for the ADSL Lines*, http://tools.ietf.org/rfc/rfc2662.txt

[35] RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, http://tools.ietf.org/rfc/rfc2684.txt

[36] RFC 2697, *A Single Rate Three Color Marker*, http://tools.ietf.org/rfc/rfc2697.txt

[37] RFC 2698, *A Two Rate Three Color Marker*, http://tools.ietf.org/rfc/rfc2698.txt

[38] RFC 2898, *PKCS #5: Password-Based Cryptography Specification Version 2.0*, http://tools.ietf.org/rfc/rfc2898.txt

[39] RFC 3004, *The User Class Option for DHCP*, http://tools.ietf.org/rfc/rfc3004.txt

[40] RFC 3066, *Tags for the Identification of Languages*, http://www.ietf.org/rfc/rfc3066.txt

[41] RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, http://www.ietf.org/rfc/rfc3246.txt

[42] RFC 3261, *SIP: Session Initiation Protocol*, http://www.ietf.org/rfc/rfc3261.txt

[43] RFC 3376, *Internet Group Management Protocol, Version 3*, http://tools.ietf.org/rfc/rfc3376.txt

[44] RFC 3435, *Media Gateway Control Protocol (MGCP) Version 1.0*, http://tools.ietf.org/rfc/rfc3435.txt

[45] RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture*, http://www.ietf.org/rfc/rfc3513.txt

[46] RFC 3550, *RTP: A Transport Protocol for Real-Time Applications*, http://www.ietf.org/rfc/rfc3550.txt

[47] RFC 3925, *Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)*, http://tools.ietf.org/rfc/rfc3925.txt

[48] RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*, http://tools.ietf.org/rfc/rfc4122.txt

[49] RFC 4566, *SDP: Session Description Protocol*, http://tools.ietf.org/rfc/rfc4566.txt

[50] TR-143. *Enabling Network Throughput Performance Tests and Statistical Monitoring*, DSL Forum Technical Report, http://www.dslforum.org/techwork/tr/TR-143.pdf

[51] *Wi-Fi Protected Setup Specification, Version 1.0h*, Wi-Fi Alliance, December 2006

Annex A. Queuing and Bridging

A.1 Queuing and Bridging Model

Figure 2 shows the queuing and bridging model for an Internet Gateway Device. This model relates to the QueueManagement object as well as the Layer2Bridging and Layer3Forwarding objects. The elements of this model are described in the following sections.

Note – the queuing model described in this Annex is meant strictly as a model to clarify the intended behavior of the related data objects. There is no implication intended that an implementation has to be structured to conform to this model.

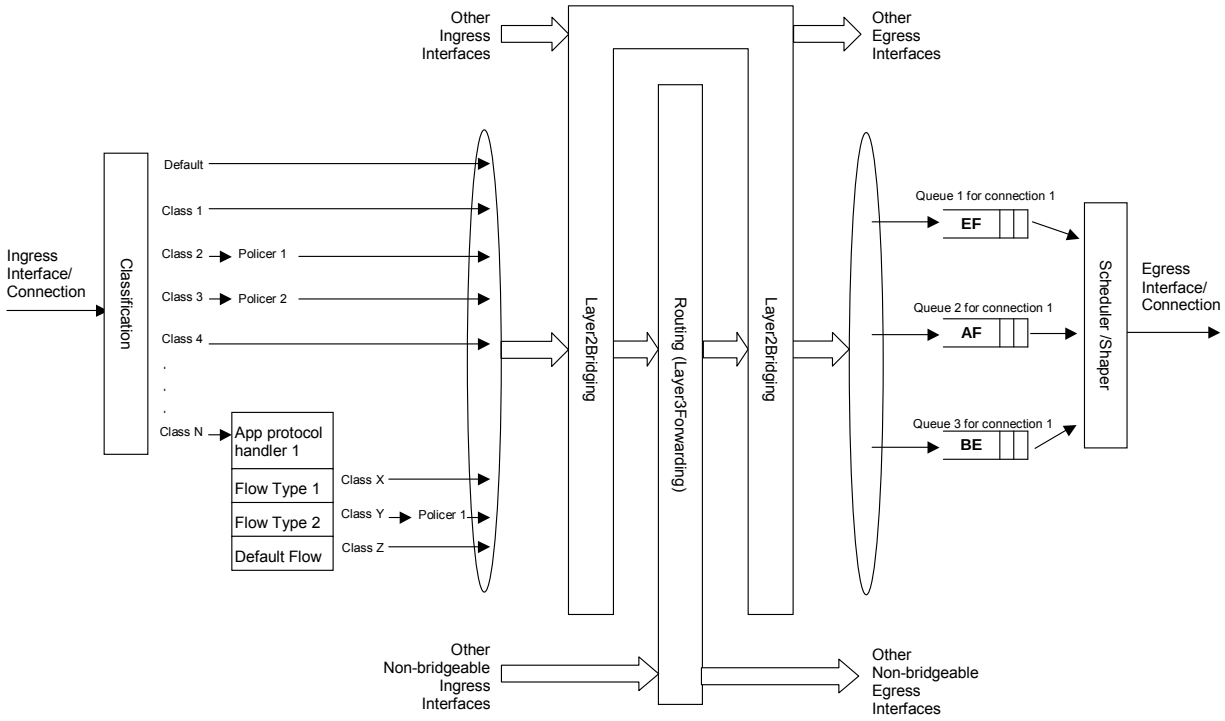

Figure 2 – Queuing model of an Internet Gateway Device

A.1.1 Packet Classification

The Classification table within the QueueManagement object specifies the assignment of each packet arriving at an ingress interface to a specific internal class. This classification can be based on a number of matching criteria, such as destination and source IP address, destination and source port, and protocol.

Each entry in the Classification table includes a series of elements, each indicated to be a Classification Criterion. Each classification criterion can be set to a specified value, or can be set to a value that indicates that criterion is not to be used. A packet is defined to match the classification criteria for that table entry

only if the packet matches all of the specified criteria. That is, a logical AND operation is applied across all classification criteria within a given Classification table entry.

> *Note – to apply a logical OR to sets of classification criteria, multiple entries in the Classification table can be created that specify the same resulting queuing behavior.*

For each classification criterion, the Classification table also includes a corresponding "exclude" flag. This flag can be used to invert the sense of the associated classification criterion. That is, if this flag is False for a given criterion, the classifier is to include only packets that meet the specified criterion (as well as all others). If this flag is True for a given criterion, the classifier is to include all packets except those that meet the associated criterion (in addition to meeting all other criteria).

For a given entry in the Classification table, the classification is to apply only to those interfaces specified by the ClassInterface element. This element can specify a particular ingress interface, all LAN-side interfaces, all WAN-side interfaces, a local IP-layer source within the Internet Gateway Device, or all sources. Depending on the particular interface, not all classification criteria will be applicable. For example, Ethernet layer classification criteria would not apply to packets arriving on a non-bridged ATM VC.

Packet classification is modeled to include all ingress packets regardless of whether they ultimately will be bridged or routed through the Internet Gateway Device. The packet classifier is not modeled to apply to packets that are embedded in a tunnelled connection (such as, PPPoE, L2TP, or tunnelled IPsec). In such cases, classification would apply only to the outer tunnel packets, but not the embedded packets contained within. An exception is for tunnels that terminate in the Internet Gateway Device itself. That is, for connections that terminate in the Internet Gateway Device, such as a PPP connection, the classification is applied to the IP packets contained within.

A.1.1.1 Classification Order

The class assigned to a given packet corresponds to the first entry in the Classification table (given the specified order of the entries in the table) whose matching criteria match the packet. If there is no entry that matches the packet, the packet is assigned to a default class.

Classification rules are sensitive to the order in which they are applied because certain traffic might meet the criteria of more than one Classification table entry. The ClassificationOrder parameter is responsible for identifying the order in which the Classification entries are to be applied.

The following rules apply to the use and setting of the ClassificationOrder parameter:

- ClassificationOrder goes in order from 1 to n, where n is equal to the number of entries in the Classification table. 1 is the highest precedence, and n the lowest. For example, if entries with ClassificationOrder of 4 and 7 both have rules that match some particular traffic, the traffic will be classified according to the entry with the 4.
- The CPE is responsible for ensuring that all ClassificationOrder values are unique and sequential.
    - If an entry is added (number of entries becomes n+1), and the value specified for ClassificationOrder is greater than n+1, then the CPE will set ClassificationOrder to n+1.
    - If an entry is added (number of entries becomes n+1), and the value specified for ClassificationOrder is less than n+1, then the CPE will create the entry with that specified value, and increment the ClassificationOrder value of all existing entries with ClassificationOrder equal to or greater than the specified value.
    - If an entry is deleted, the CPE will decrement the ClassificationOrder value of all remaining entries with ClassificationOrder greater than the value of the deleted entry.
    - If the ClassificationOrder value of an entry is changed, then the value will also be changed for other entries greater than or equal to the lower of the old and new values, and less than the larger of the old and new values. If the new value is less than the old, then these other entries will all have ClassificationOrder incremented. If the new value is greater than the old, then the other entries will have ClassificationOrder decremented and the changed entry will be given a

value of <new value>-1. For example, an entry is changed from 8 to 5. The existing 5 goes to 6, 6 to 7, and 7 to 8. If the entry goes from 5 to 8, then 6 goes to 5, 7 to 6, and the changed entry is 7. This is consistent with the behavior that would occur if the change were considered to be an Add of a new entry with the new value, followed by a Delete of the entry with the old value.

A.1.1.2 Dynamic Application Specific Classification

In some situations, traffic to be classified cannot be identified by a static set of classification criteria. Instead, identification of traffic flows might require explicit application awareness. The model accommodates such situations via the App and Flow tables in the QueueManagement object.

Each entry in the App table is associated with an application-specific protocol handler, identified by the ProtocolIdentifier, which contains a URN. For a particular CPE, the AvailableAppList parameter indicates which protocol handlers that CPE is capable of supporting, if any. A list of standard protocol handlers and their associated URNs is specified in section A.3, though a CPE can also support vendor-specific protocol handlers as well. Multiple App table entries can refer to the same ProtocolIdentifier.

The role of the protocol handler is to identify and classify flows based on application awareness. For example, a SIP protocol handler might identify a call-control flow, an audio flow, and a video flow. The App and Flow tables are used to specify the classification outcome associated with each such flow.

For each App table entry there can be one or more associated Flow table entries. Each flow table identifies a type of flow associated with the protocol handler. The FlowType element is used to identify the specific type of flow associated with each entry. For example, a Flow table entry for a SIP protocol handler might refer only to the audio flows associated with that protocol handler. A list of standard FlowType values is given in section A.3, though a CPE can also support vendor-specific flow types.

A protocol handler can be defined as being fed from the output of a Classification table entry. That is, a Classification entry can be used to single out control traffic to be passed to the protocol handler, which then subsequently identifies associated flows. Doing so allows more than one instance of a protocol handler associated with distinct traffic. For example, one could define two App table entries associated with SIP protocol handlers. If the classifier distinguished control traffic to feed into each handler based on the destination IP address of the SIP server, this could be used to separately classify traffic for different SIP service providers. In this case, each instance of the protocol handler would identify only those flows associated with a given service. Note that the Classification table entry that feeds each protocol handler wouldn't encompass all of the flows; only the traffic needed by the protocol handler to determine the flows—typically only the control traffic.

A.1.1.3 Classification Outcome

Each Classification entry specifies a tuple composed of either:

- A Queue and (optionally) a Policer, or
- An App table entry

Each entry also specifies:

- Outgoing DiffServ and Ethernet priority marking behavior
- A ForwardingPolicy tag that can be referenced in the Layer3Forwarding table to affect packet routing (note that the ForwardingPolicy tag affects only routed traffic)

Note that the information associated with the classification outcome is modeled as being carried along with each packet as it flows through the system.

If a packet does not match any Classification table entry, the DefaultQueue, DefaultPolicer, default markings, and default ForwardingPolicy are used.

If a Queue/Policer tuple is specified, classification is complete. If, however, an App is specified, the packet is passed to the protocol handler specified by the ProtocolIdentifier in the specified App table entry

for additional classification (see section A.1.1.2). If any of the identified flows match the FlowType specified in any Flow table entry corresponding to the given App table entry (this correspondence is indicated by the App identifier), the specified tuple and markings for that Flow table entry is used for packets in that flow. Other flows associated with the application, but not explicitly identified, use the default tuple and markings specified for that App table entry.

A.1.2 Policing

The Policer table defines the policing parameters for ingress packets identified by either a Classification table entry (or the default classification) or a dynamic flow identified by a protocol handler identified in the App table.

Each Policer table entry specifies the packet handling characteristics, including the rate requirements and behavior when these requirements are exceeded.

A.1.3 Queuing and Scheduling

The Queue table specifies the number and types of queues, queue parameters, shaping behavior, and scheduling algorithm to use. Each Queue table entry specifies a set of egress interfaces for which a queue with the corresponding characteristics needs to exist.

> *Note – If the CPE can determine that among the interfaces specified for a queue to exist, packets classified into that queue cannot egress to a subset of those interfaces (from knowledge of the current routing and bridging configuration), the CPE can choose not to instantiate the queue on those interfaces.*
>
> *Note – Packets classified into a queue that exit through an interface for which the queue is not specified to exist, will instead use the default queuing behavior. The default queue itself will exist on all egress interfaces.*

The model defined here is not intended to restrict where the queuing is implemented in an actual implementation. In particular, it is up to the particular implementation to determine at what protocol layer it is most appropriate to implement the queuing behavior (IP layer, Ethernet MAC layer, ATM layer, etc.). In some cases, however, the QueueManagement configuration would restrict the choice of layer where queueing can be implemented. For example, if a queue is specified to carry traffic that is bridged, then it could not be implemented as an IP-layer queue.

> *Note – care needs to be taken to avoid having multiple priority queues multiplexed onto a single connection that is rate shaped. In such cases, the possibility exists that high priority traffic can be held back due to rate limits of the overall connection exceeded by lower priority traffic. Where possible, each priority queue will be shaped independently using the shaping parameters in the Queue table.*

The scheduling parameters defined in the Queue table apply to the first level of what might be a more general scheduling hierarchy. This specification does not specify the rules that an implementation needs to apply to determine the most appropriate scheduling hierarchy given the scheduling parameters defined in the Queue table.

As an example, take a situation where the output of four distinct queues is to be multiplexed into a single connection, and two entries share one set of scheduling parameters while the other two entries share a different set of scheduling parameters. In this case, it might be appropriate to implement this as a scheduling hierarchy with the first two queues multiplexed with a scheduler defined by the first pair, and the second two queues being multiplexed with a scheduler defined by the second pair. The lower layers of this scheduling hiearchy cannot be directly determined from the content of the Queue table.

A.1.4 Bridging

For each interface, the output of the classifier is modeled to feed a set of layer 2 bridges as specified by the Layer2Bridging object. Each bridge specifies layer 2 connectivity between one or more layer 2 LAN and/or WAN interfaces, and optionally one or more layer 3 connections to the local router.

Each bridge corresponds to a single entry in the Bridge table of the Layer2Bridging object. Each entry contains (by reference) one or more Filter table entries. Each Filter table entry specifies an interface or set of interfaces to include in the bridge, and can also specify layer 2 filter criteria to selectively bridge traffic among the specified interfaces.

> *Note – each Bridge table entry can contain a Bridge Port table (as a sub-object). If this table is supported, it explicitly defines which interfaces are to be included in the bridge, and also defines various bridge port parameters.*

Each Filter table entry selects one or more interfaces among those listed in the AvailableInterface table. This table would normally include all layer 2 interfaces that include an Ethernet MAC layer. This would exclude, for example, a non-bridged ATM VC carrying IPoA or PPPoA. Each entry in the Filter table refers to a specific layer 2 interface. A Filter table entry can also include LAN-side or WAN-side layer 3 connections to the local router, such as PPP or IP connections. When using Layer2Bridging to include a layer 3 connection in a bridge, this overrides the default association of that connection with a layer 2 object as indicated by the IGD data model connection object hierarchy, and results in an update of the IGD data model hierarchy. The implications of this are explained in Annex A.6.

> *Note – from the point of view of a bridge, packets arriving into the bridge from the local router (either LAN-side or WAN-side) are treated as ingress packets, even though the same packets, which just left the router, are treated as egress from the point of view of the router. For example, a Filter table entry might admit packets on ingress to the bridge from a particular WANIPConnection, which means that it admits packets on their way out of the router over this layer 3 connection.*

A.1.4.1 Filtering

Traffic from a given interface (or set of interfaces) can be selectively admitted to a given Bridge, rather than bridging all traffic from that interface. Each entry in the Filter table includes a series of classification criteria. Each classification criterion can be set to a specified value, or can be set to a value that indicates that criterion is not to be used. A packet is admitted to the Bridge only if the packet matches all of the specified criteria. That is, a logical AND operation is applied across all classification criteria within a given Filter table entry.

> *Note – to apply a logical OR to sets of classification criteria, multiple entries in the Filter table can be created that refer to the same interfaces and the same Bridge table entry.*

> *Note – a consequence of the above rule is that, if a packet does not match the criteria of any of the enabled Filter table entries, then it will not be admitted to any bridges, i.e. it will be dropped. As a specific example of this, if none of the enabled Filter table entries reference a given interface, then all packets arriving on that interface will be dropped.*

For each classification criterion, the Filter table also includes a corresponding "exclude" flag. This flag can be used to invert the sense of the associated classification criterion. That is, if this flag is False for a given criterion, the Bridge will admit only packets that meet the specified criterion (as well as all other criteria). If this flag is True for a given criterion, the Bridge will admit all packets except those that meet the associated criterion (in addition to meeting all other criteria).

Note that because the classification criteria are based on layer 2 packet information, if the selected interface for a given Filter table entry is a layer 3 connection from the local router, the layer 2 classification criteria do not apply.

A.1.4.2 Exclusivity Order

Each Filter table entry is defined as either exclusive or non-exclusive. Any packet that matches the filter criteria of one or more exclusive filters is admitted to the Bridge associated with the first exclusive entry in the Filter table (relative to the specified ExclusivityOrder).

If there is no exclusive filter that matches a packet, then the packet is admitted to all Bridges associated with non-exclusive filters that match the packet.

The following rules apply to the use and setting of the ExclusivityOrder parameter:

- If the ExclusivityOrder is zero, the filter is defined to be non-exclusive.
- If the ExclusivityOrder is one or greater, the filter is defined to be exclusive.
- Among exclusive filters, the ExclusivityOrder goes in order from 1 to n, where n is equal to the number of exclusive filters. 1 is the highest precedence, and n the lowest.
- The CPE is responsible for ensuring that all ExclusivityOrder values among exclusive filters are unique and sequential.
    - If an exclusive filter is added (number of exclusive filters becomes n+1) or a non-exclusive filter is changed to be exclusive, and the value specified for ExclusivityOrder is greater than n+1, then the CPE will set ExclusivityOrder to n+1.
    - If an exclusive filter is added (number of entries becomes n+1) or a non-exclusive filter is changed to be exclusive, and the value specified for ExclusivityOrder is less than n+1, then the CPE will create the entry with that specified value, and increment the ExclusivityOrder value of all existing exclusive filters with ExclusivityOrder equal to or greater than the specified value.
    - If an exclusive filter is deleted or an exclusive filter is changed to non-exclusive, the CPE will decrement the ExclusivityOrder value of all remaining exclusive filter with ExclusivityOrder greater than the value of the deleted entry.
    - If the ExclusivityOrder value of an exclusive filter is changed, then the value will also be changed for other exclusive filters greater than or equal to the lower of the old and new values, and less than the larger of the old and new values. If the new value is less than the old, then these other entries will all have ExclusivityOrder incremented. If the new value is greater than the old, then the other entries will have ExclusivityOrder decremented and the changed entry will be given a value of <new value>-1. For example, an entry is changed from 8 to 5. The existing 5 goes to 6, 6 to 7, and 7 to 8. If the entry goes from 5 to 8, then 6 goes to 5, 7 to 6, and the changed entry is 7. This is consistent with the behavior that would occur if the change were considered to be an Add of a new exclusive filter with the new value, followed by a Delete of the exclusive filter with the old value.

A.1.4.3 Egress from a Bridge

Packets admitted to a bridge from any interface are bridged across all of the interfaces considered part of that bridge. An interface is considered part of a bridge if it is specified by any of the Filter table or Marking table entries that are associated with the bridge. That is, the union of all interfaces specified either for potential admission into the bridge or for special marking treatment on egress are considered part of the bridge. This can include both layer 2 interfaces as well as layer 3 connections to the local router.

> *Note – if the Bridge Port table is supported, it explicitly defines which interfaces are considered part of the bridge. This overrides the implicit definition that is provided by the Filter and Marking tables.*
>
> *Note – a consequence of the above rukes is that, if no layer 3 interfaces are part of a given bridge, then no packets that are admitted to that bridge can be passed to the IP layer.*

For a given bridge, packets on egress can optionally be marked distinctly for specific interfaces. The Marking table allows the CPE to be configured to selectively either remove all VLANID/priority marking

from a packet on egress, or modify the VLANID and/or Ethernet priority marking on egress. This can be done selectively per interface.

A.2 Default Layer 2/3 QoS Mapping

Table 42 presents a "default" mapping between layer 2 and layer 3 QoS. In practice, it is a guideline for automatic marking of DSCP (layer 3) based upon Ethernet Priority (layer 2) and the other way around. Please refer to the QueueManagement object DSCPMark and EthernetPriorityMark parameters (and related parameters) for configuration of a default automatic DSCP / Ethernet Priority mapping.

Automatic marking of DSCP or Ethernet Priority is likely only in the following cases:

- WAN → LAN: to map DSCP (layer 3) to Ethernet Priority (layer 2)
- LAN → WAN: to map Ethernet Priority (layer 2) to DSCP (layer 3)

Automatic marking in the LAN → LAN case is unlikely, since LAN QoS is likely to be supported only at layer 2, and LAN DSCP values, if used, will probably be a direct representation of Ethernet Priority, e.g. Ethernet Priority shifted left by three bits.

In the table, grayed and bolded items are added to allow two-way mapping between layer 2 and layer 3 QoS (where the mapping is ambiguous, the grayed values SHOULD be ignored and the bolded values SHOULD be used). If, when mapping from layer 3 to layer 2 QoS, the DSCP value is not present in the table, the mapping SHOULD be based only on the first three bits of the DSCP value, i.e. on DSCP & 111000.

Table 42 – Default Layer 2/3 QoS Mapping

Layer 2		Layer 3	
Ethernet Priority	Designation	DSCP	Per Hop Behavior
001 (1)	BK	000000 (0x00)	Default
010 (2)	spare	000000 (0x00)	
000 (0)	BE	000000 (0x00) **000000 (0x00)**	Default CS0
011 (3)	EE	001110 (0x0e) 001100 (0x0c) 001010 (0x0a) **001000 (0x08)**	AF13 AF12 AF11 CS1
100 (4)	CL	010110 (0x16) 010100 (0x14) 010010 (0x12) **010000 (0x10)**	AF23 AF22 AF21 CS2
101 (5)	VI	011110 (0x1e) 011100 (0x1c) 011010 (0x1a) **011000 (0x18)**	AF33 AF32 AF31 CS3
110 (6)	VO	100110 (0x26) 100100 (0x24) 100010 (0x22) **100000 (0x20)**	AF43 AF42 AF41 CS4
110 (6)	VO	101110 (0x2e) **101000 (0x28)**	EF CS5
111 (7)	NC	110000 (0x30) **111000 (0x38)**	CS6 CS7

A.3 URN Definitions for App and Flow Tables

A.3.1 ProtocolIdentifier

Table 43 lists the URNs defined for the ProtocolIdentifier parameter in the App table of the QueueManagement service. Additional standard or vendor-specific URNs can be defined following the standard synax for forming URNs.

Table 43 – ProtocolIdentifer URNs

URN	Description
urn:dslforum-org:sip	Session Initiaion Protocol (SIP) as defined by RFC 3261 [42]
urn:dslforum-org:h.323	ITU-T Recommendation H.323
urn:dslforum-org:h.248	ITU-T Recommendation H.248 (MEGACO)
urn:dslforum-org:mgcp	Media Gateway Control Protocol (MGCP) as defined by RFC 3435 [44]
urn:dslforum-org:pppoe	Bridged sessions of PPPoE

A.3.2 FlowType

A syntax for forming URNs for the FlowType parameter in the Flow table of the QueueManagement service are defined for the Session Description Protocol (SDP) as defined by RFC 4566 [49]. Additional standard or vendor-specific URNs can be defined following the standard synax for forming URNs.

A URN to specify an SDP flow is formed as follows:

```
urn:dslforum-org:sdp-[MediaType]-[Transport]
```

[MediaType] corresponds to the "media" sub-field of the "m" field of an SDP session description.

[Transport] corresponds to the "transport" sub-field of the "m" field of an SDP session description.

Non-alphanumeric characters in either field are removed (e.g., "rtp/avp" becomes "rtpavp").

For example, the following would be valid URNs referring to SDP flows:

```
urn:dslforum-org:sdp-audio-rtpavp
urn:dslforum-org:sdp-video-rtpavp
urn:dslforum-org:sdp-data-udp
```

For FlowType URNs following this convention, there is no defined use for FlowTypeParameters, which SHOULD be left empty.

For the ProtocolIdentifier urn:dslforum-org:pppoe, a single flow type is defined referring to the entire PPPoE session. The URL for this FlowType is:

```
urn:dslforum-org:pppoe
```


A.3.3 FlowTypeParameters

For the FlowType urn:dslforum-org:pppoe, Table 44 specifies the defined FlowTypeParameter values.

Table 44 – FlowTypeParameter values for FlowType urn:dslforum-org:pppoe

Name	Description of Value
ServiceName	The PPPoE service name. If specified, only bridged PPPoE sessions designated for the named service would be considered part of this flow. If this parameter is not specified, or is empty, bridged PPPoE associated with any service considered part of this flow.
ACName	The PPPoE access concentrator name. If specified, only bridged PPPoE sessions designated for the named access concentrator would be considered part of this flow. If this parameter is not specified, or is empty, bridged PPPoE associated with any access concentrator considered part of this flow.
PPPDomain	The domain part of the PPP username. If specified, only bridged PPPoE sessions in which the domain portion of the PPP username matches this value are considered part of this flow. If this parameter is not specified, or is empty, all bridged PPPoE sessions are considered part of this flow.

A.4 Example Queuing Architecture for RG (from TR-059)

The queuing and scheduling discipline envisioned upstream for the RG is shown in Figure 3.

There are multiple access sessions supported in this model, however, all traffic is classified and scheduled in a monolithic system. So, while it might appear at first that the Diffserv queuing and scheduling might apply only to IP-aware access – in fact all access, IP, Ethernet, or PPP is managed by the same system that adheres to the Diffserv model.

For example, at the bottom of the figure, BE treatment is given to the non-IP-aware access sessions (PPPoE started behind the RG or delivered to an L2TP tunnel delivery model). This queue might be repeated several times in order to support fairness among multiple PPPoE accesses – or it can be a monolithic queue with separate rate limiters applied to the various access sessions.

The PTA access is a single block of queues. This is done because NSP access typically works with a single default route to the NSP, and managing more than one simultaneously at the RG would be perilous. The ∑ rate limiter would limit the overall access traffic for a service provider.

Rate limiters are also shown within the EF and AF service classes because the definition of those Diffserv types is based on treating the traffic differently when it falls into various rates.

Finally, at the top of the diagram is the ASP access block of queues. In phase 1A, these queues are provisioned and provide aggregate treatment of traffic mapped to them. In phase 1B, it will become possible to assign AF queues to applications to give them specific treatment instead of aggregate treatment. The EF service class can also require a high degree of coordination among the applications that make use of it so that its maximum value is not exceeded.

Notable in this architecture is that all the outputs of the EF, AF, and BE queues are sent to a scheduler (**S**) that pulls traffic from them in a strict priority fashion. In this configuration EF traffic is, obviously, given highest precedence and BE is given the lowest. The AF service classes fall in-between.

Note that there is significant interest in being able to provide a service arrangement that would allow general Internet access to have priority over other (bulk rate) services.[19] Such an arrangement would be accomplished by assigning the bulk rate service class to BE and by assigning the default service class (Internet access) as AF with little or no committed information rate.

Given this arrangement, the precedence of traffic shown in the figure is arranged as:

1. EF – red dotted line
2. AF – blue dashed line (with various precedence among AF classes as described in RFC 2597 [32])
3. BE – black solid line

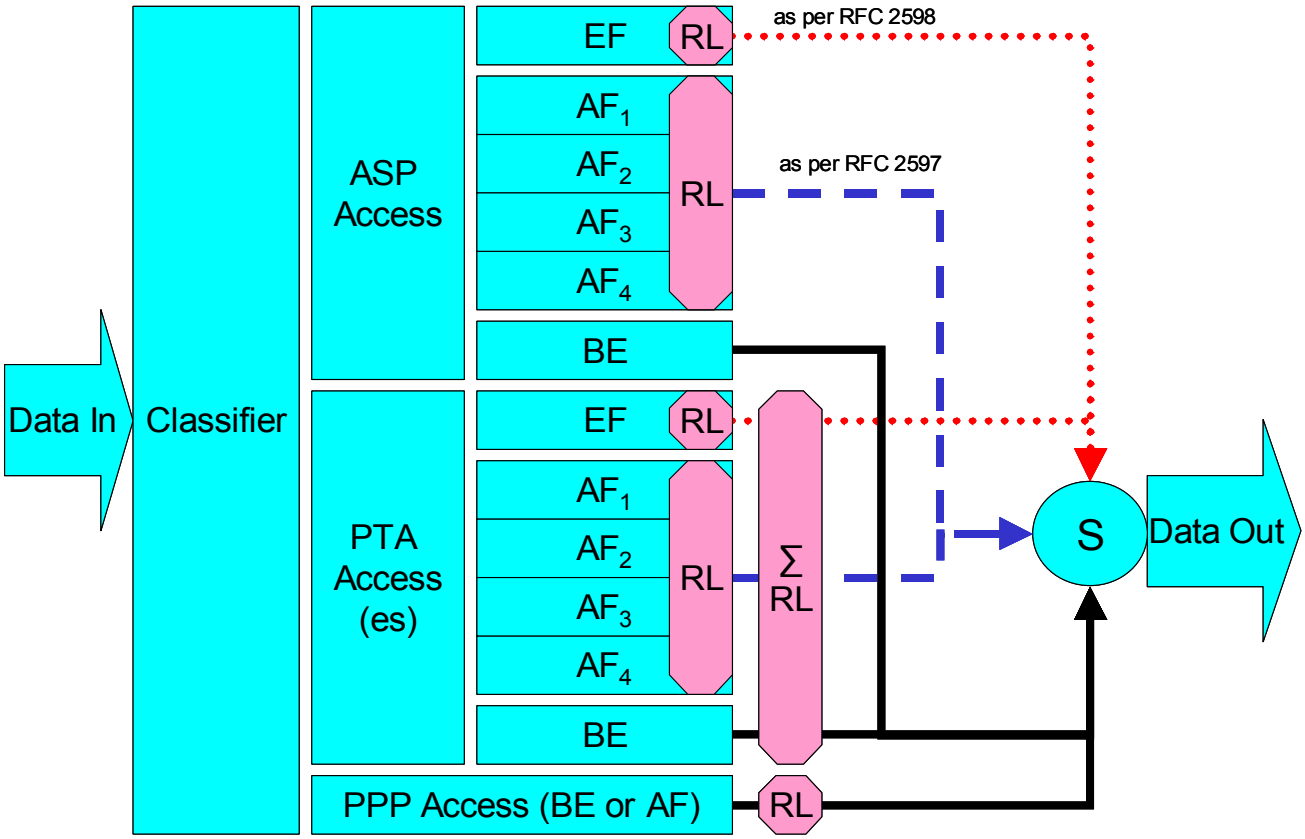

Figure 3 – Queuing and Scheduling Example for RG

In Figure 3 the following abbreviations apply:

ASP – Application Service Provider
PTA – PPP Terminated Aggregation
PPP – Point-to-Point Protocol
EF – Expedited Forwarding – as defined in RFC 3246 [41]
AF – Assured Forwarding – as defined in RFC 2597 [32]
BE – Best Effort forwarding

[19] This "bulk rate" service class would typically be used for background downloads and potentially for peer-to-peer applications as an alternative to blocking them entirely.

RL – Rate Limiter
∑RL – Summing Rate Limiter (limits multiple flows)
S – Scheduler

A.5 Layer2Bridging Use Case: Interface Based Bridging

In an ITU-H.610 architecture using multi-VC and multi-edges to offer multi-services (high speed Internet, TVoDSL, etc.), one VC or a group of VCs are associated with each service. Regarding the CPE, some services can be layer 2 based if the service provider needs to have a layer 2 view of the home devices (for example, set-top boxes). If the services are offered by different service providers, and shared Internet access is also provided via the Internet Gateway, conflict between the local DHCP server and remote DHCP servers can occur. If there is no QoS on the home network there might also be issues regarding the priority of different streams. One solution is to associate one or more physical ports of the Internet Gateway with a specific service associated with one or more VCs.

As an example, Ethernet port 1 might be dedicated to a TVoDSL service and this port would be included in the same bridge with the VCs supporting the TVoDSL service. In this case, the other home network ports would be associated with the shared Internet access service. To achieve this, an interface-based bridge would be created using the Layer2Bridging object. A Bridge table entry would be created along with associated Filter table entries for Ethernet port 1, and each VC associated with the TVoDSL service. In this case no filter criteria would be used in each Filter table entry. If the subscriber's services are modified, the Layer2Bridging configuration might need to be modified accordingly.

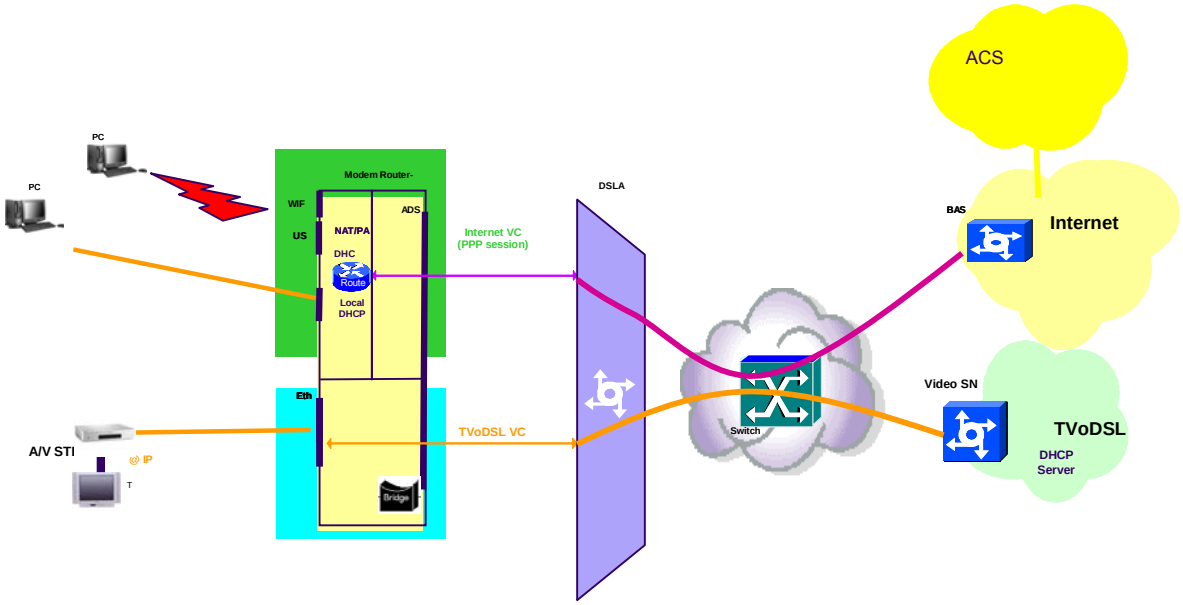

Figure 4 – Example of interface-based bridging

A.6 Relationship between Layer2Bridging and LANDevice / WAN**Connection

The Layer2Bridging, LANDevice and WAN**Connection objects are all relevant to the CPE's bridging configuration. Specifically:

- Layer2Bridging describes and configures all the bridges in the device.

- LANDevice describes an "implicit" bridge in which some or all traffic is bridged between the IP interface represented by LANDevice, and its child layer 2 interface objects (LAN**Interface-Config, WLANConfiguration).
- WANPPPConnection with ConnectionType = "PPPoE_Bridged" describes a bridge.
- WANIPConnection with ConnectionType = "IP_Bridged" describes a bridge.

Only Layer2Bridging provides a complete description of the device's bridging configuration. The definitions of the above-mentioned objects and parameters make it clear that they must all be consistent with each other.

This consistency requirement is perhaps best understood by realizing that, below the InternetGateway-Device data model, there is an underlying device and configuration. The TR-069 objects are just a way of representing and configuring items that are aspects of the device and its configuration, and which are nothing to do with TR-069 per se. Such items could also be configured independently of TR-069, e.g. via a vendor configuration file or a user interface.

A.6.1 Populating the Data Model on Reboot

Imagine what happens when the device reboots. The bridges are all present in the device configuration, so the question is how they show up the InternetGatewayDevice data model. The data model population logic will be similar to that shown in the following pseudocode:

```
# LANDevice and WANDevice

For each physical WAN interface (DSL, Ethernet etc)
    Add a WANDevice instance, and populate WANCommonInterfaceConfig and
                                                          WAN**InterfaceConfig

For each WAN-side layer 2 interface (ATM PVC, Ethernet link etc)
    Add a WANConnectionDevice instance within the correct WANDevice, and populate
                                                          WAN**LinkConfig

For each WAN-side layer 3 interface (IP, PPP)
    Add a WAN**Connection instance within the correct WANConnectionDevice
    If the layer 3 interface is attached to a WAN / LAN bridge
        Set ConnectionType to "IP_Bridged" / "PPPoE_Bridged"

For each LAN IP interface
    Add a LANDevice instance, and populate LANHostConfigManagement with
                                                          DHCP server settings etc
    For each IP address on the IP interface
        Add and populate an IPInterface child of the LANDevice

For each LAN-side layer 2 interface
    If interface traffic can be delivered to (or come from) a LAN IP interface
                                        (whether or not this involves bridging)
        Place the layer 2 interface under the relevant LANDevice instance(s)
    Else
        Place the layer 2 interface under the LANInterfaces object

# Layer2Bridging (if implemented)

For each valid bridge interface or router connection (as described in the
                        definition of AvailableInterface.{i}.InterfaceReference)
    Add and populate an AvailableInterface instance

For each bridge:
    Add and populate a Bridge instance
    For each bridge filter rule
        Add and populate a Filter instance
    For each bridge marking rule
        Add and populate a Marking instance
```


Please note the following:

- The criterion for setting the WAN**Connection ConnectionType to “IP_Bridged” or “PPPoE_-Bridged” is “layer 3 interface is attached to a WAN / LAN bridge”. This is the only way in which WAN**Connection can indicate the existence of such a bridge.
- The criterion for including a layer 2 interface under a LANDevice is “traffic can be delivered to (or come from)”. This just means that there is at least one (enabled) bridge filter that can allow traffic to flow between the LANDevice’s IP interface and the layer 2 interface. LANDevice is unable to represent the details of the filter rules.
- The pseudocode does not mention whether objects are enabled or disabled. Consider disabling a bridge (not the TR-069 object… an actual bridge). This would be expected to disable the corresponding Layer2Bridging Bridge object. The bridge is not explicitly modeled on the LANDevice side, but the LANDevice’s IP interface is layered on top of the bridge, and can be up only if the bridge is up.

A.6.2 Updating the Data Model on Configuration Changes

Now imagine what happens when the device configuration changes in a way that affects any of the objects mentioned in the pseudocode. Conceptually, all of the objects are deleted and then re-populated by the pseudocode logic. In practice, of course, the implementation would probably make only the minimal changes in moving from the old to the new state.

A.6.3 Bridging Behavior when Layer2Bridging is not Implemented

If Layer2Bridging is not implemented, then bridging cannot be configured using the InternetGateway-Device data model. The only possible bridge-related configuration parameter is WAN**Connection’s ConnectionType. This makes sense only if there is a single (or at least a default) LANDevice, because there is no way to select which LANDevice to attach to the bridge. Therefore, on devices that don’t implement Layer2Bridging, any non-trivial bridging configuration will have to use vendor-specific configuration files, and the remarks in the previous sections will still apply.

A.6.4 Case Studies

This section considers two case studies, each of which illustrates a different aspect of the relationship between Layer2Bridging and WAN**Connection. Both case studies refer to the example configurations of Figure 5and Figure 6.

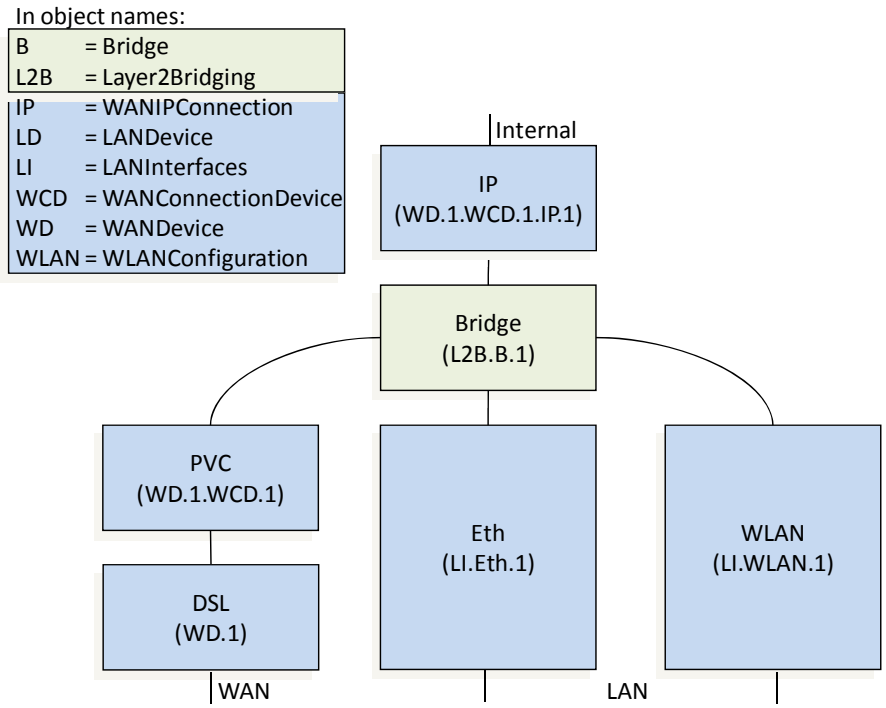

Figure 5 – WAN / LAN bridged example

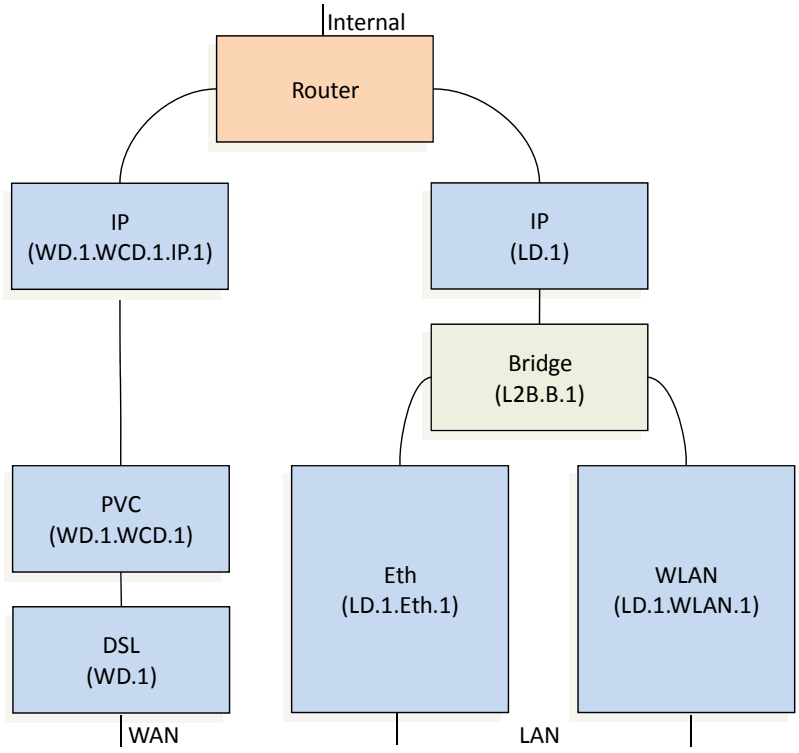

Figure 6 – WAN / LAN routed example

A.6.4.1 Creating a WANIPConnection Instance

In the bridged configuration of Figure 5, suppose that InternetGatewayDevice.WANDevice.1.WAN-ConnectionDevice.1.WANIPConnection.1 has just been created. There is a bridge "between" it and its parent WANConnectionDevice, but this is indicated in the WANDevice object hierarchy only via WANIP-Connection's ConnectionType value of "IP_Bridged". If Layer2Bridging is implemented, this bridge will of course be modeled there.

In the routed configuration of Figure 6, similarly suppose that InternetGatewayDevice.WANDevice.1.-WANConnectionDevice.1.WANIPConnection.1 has just been created. In this case, there is no WAN-side bridge, which will be indicated by WANIPConnection's ConnectionType value of "IP_Routed".

A.6.4.2 Attaching a WANConnectionDevice Instance to a Bridge

The routed configuration of of Figure 6 can be converted to the bridged configuration of Figure 5 by using Layer2Bridging to re-configure the bridge as follows:

- Detach the LAN IP interface InternetGatewayDevice.LANDevice.1
- Attach the PVC InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1

The WAN IP interface InternetGatewayDevice.WANConnectionDevice.1.WANIPConnection.1, which was previously attached to the PVC InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1 will automatically be attached to the bridge. As in the previous use case, the bridge is "between" it and its parent WANConnectionDevice.

The only visible change in the WANDevice object hierarchy will be that WANIPConnection's Connection-Type value will change from "IP_Routed" to "IP_Bridged". In fact the bridge has been inserted "between" the WANIPConnection and its parent WANConnectionDevice.

In the LANDevice object hierarchy, as indicated in the Figures, the LANEthernetInterfaceConfig and WLANConfiguration objects will move from LANDevice.1 to LANInterfaces.

Annex B. LinkType and ConnectionType Interdependencies

For DSL CPE, the parameters LinkType in the WANDSLLinkConfig object and ConnectionType in the WANPPPConnection and WANIPConnection objects are interdependent. The LinkType parameter describes the ATM-layer encapsulation to be used for the corresponding ATM VC (in conjunction with the ATMEncapsulation parameter). The value of LinkType determines the possible types of connections that can be carried over the corresponding VC. Specifically, the LinkType determines:

- Whether the associated WANConnectionDevice object can contain WANPPPConnection objects, WANIPConnection objects, or both.
- The allowed values for the ConnectionType parameter within a WANPPPConnection object or WANIPConnection contained within the corresponding WANConnectionDevice.

Table 45 summarizes these interdependencies for a WANPPPConnection. For each value of LinkType listed across the top of the table, the table indicates allowed values of the ConnectionType for a WANPPPConnection. Entries with a check mark are allowed values, while entries marked "Forbidden" are not allowed.

For the columns that are marked "WANPPPConnection Forbidden," it is invalid to create a WANPPPConnection object in a WANConnectionDevice for which the LinkType is so configured.

Table 45 – LinkType and ConnectionType Interdependencies for a WANPPPConnection

LinkType / ConnectionType	PPPoA	EoA	IPoA	CIP	PPPoE	Unconfigured
IP_Routed	✓	✓	WANPPP-Connection Forbidden	WANPPP-Connection Forbidden	WANPPP-Connection Forbidden	WANPPP-Connection Forbidden
DHCP_Spoofed	✓	✓				
PPPoE_Bridged	Forbidden	✓				
PPTP_Relay	✓	✓				
L2TP_Relay	✓	✓				
PPPoE_Relay	✓	Forbidden				
Unconfigured	✓	✓				

Table 46 summarizes these interdependencies for a WANIPConnection. For each value of LinkType listed across the top of the table, the table indicates allowed values of the ConnectionType for a WANIPConnection. Entries with a check mark are allowed values, while entries marked "Forbidden" are not allowed.

For the columns that are marked "WANIPConnection Forbidden," it is invalid to create a WANIPConnection object in a WANConnectionDevice for which the LinkType is so configured.

Table 46 – LinkType and ConnectionType Interdependencies for a WANIPConnection

LinkType / ConnectionType	PPPoA	EoA	IPoA	CIP	PPPoE	Unconfigured
IP_Routed	WANIP-Connection Forbidden	✓	✓	✓	WANIP-Connection Forbidden	WANIP-Connection Forbidden
IP_Bridged		✓	Forbidden	Forbidden		
Unconfigured		✓	✓	✓		

Note that the LinkType value of "PPPoE" is DEPRECATED since creation of either type of WAN connection object is forbidden when this value is set. This is due to the service-provider requirement to allow both PPPoE and IP simultaneously on the same ATM VC. To support PPPoE, the LinkType "EoA" MUST be used, since this LinkType also allows IP connections.

Note also that while the value "Unconfigured" is an allowed value for the LinkType and ConnectionType, a WAN connection can only be operational if both the corresponding LinkType and ConnectionType are set to values other than "Unconfigured".

Appendix I. Managed bridge configuration in a multi-PVC scenario

This Appendix describes issues to be addressed in configuring a managed bridge in a multi-PVC scenario, and gives an example configuration.

I.1 Description of scenario

I.1.1 Network Traffic Classes and Priorities

The IGD has to support a Triple Play service, Figure 7, which means that network traffic needs to be prioritized in order to meet the different service requirements.

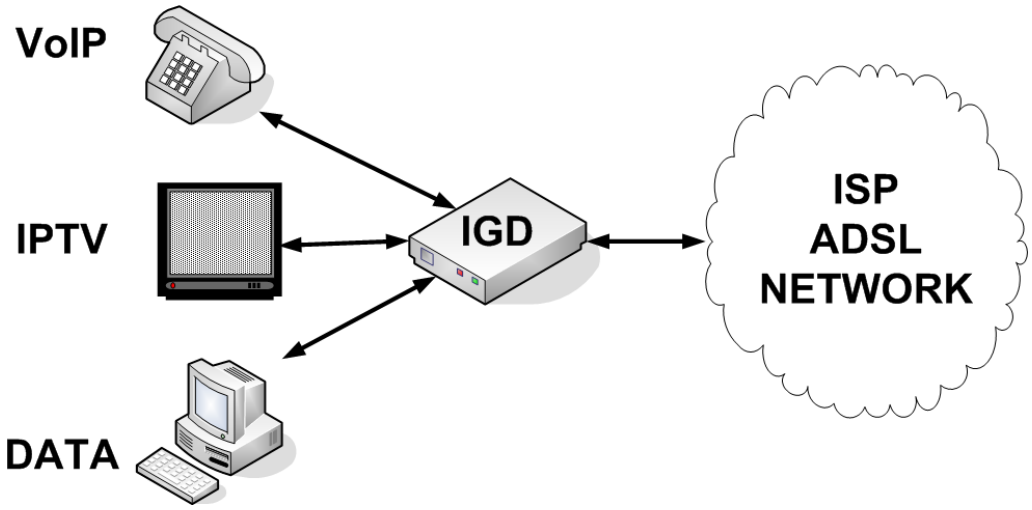

Figure 7 – Triple Play Service

Figure 8 illustrates the different upstream priorities. They are explained below.

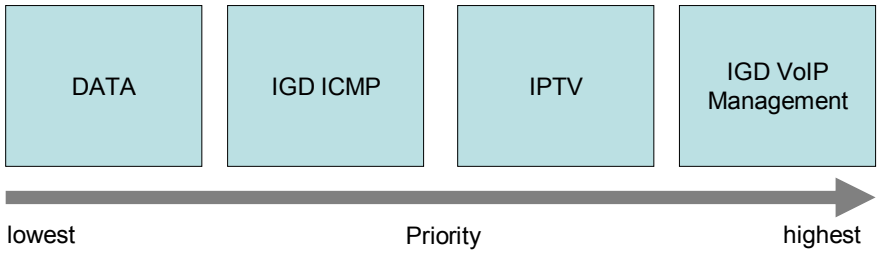

Figure 8 – Triple Play Upstream Priorities

Both the VoIP control and RTP protocols need to avoid, at any cost, congestion, delay, jitter, packet loss, etc. IGD Management traffic also needs to have a high priority. Otherwise a network intensive subscriber application could prevent IGD configuration, management and inventory activities. This network traffic has to be handled by a strict priority data queue.

Having given the highest priority to the VoIP and Management traffic, the second most critical traffic type is IPTV. This network traffic has to be handled by a premium data queue.

The next one, in requested priority order, is the IGD's ICMP traffic (e.g. ping and echo). This traffic is rather important for the first level of IP troubleshooting, but it cannot interfere with VoIP, management and IPTV traffic. This network traffic has to be handled by a high priority best effort data queue.

Finally, with the lowest priority, there is the default data traffic. This is usually the traffic generated by subscriber PCs. In the case of congestion, delay or packet loss, it's up to the TCP/UDP protocol endpoints to fix the problem. In such cases, retransmission is not likely to be an issue. This network traffic has to be handled by the lowest priority best effort data queue.

I.1.2 Mapping to PVCs

In Figure 9, the network traffic belonging to the IGD itself, i.e. VoIP and Management (TR-069, Telnet, SNMP, ICMP, etc.) is sent and received on PVC vpi1/vci1. Its ATM QoS is CBR.

IPTV network traffic uses PVC vpi2/vci2 (both upstream and downstream), with an ATM VBR-rt.

PVC vpi3/vci3 is used for all generic network traffic (both upstream and downstream).

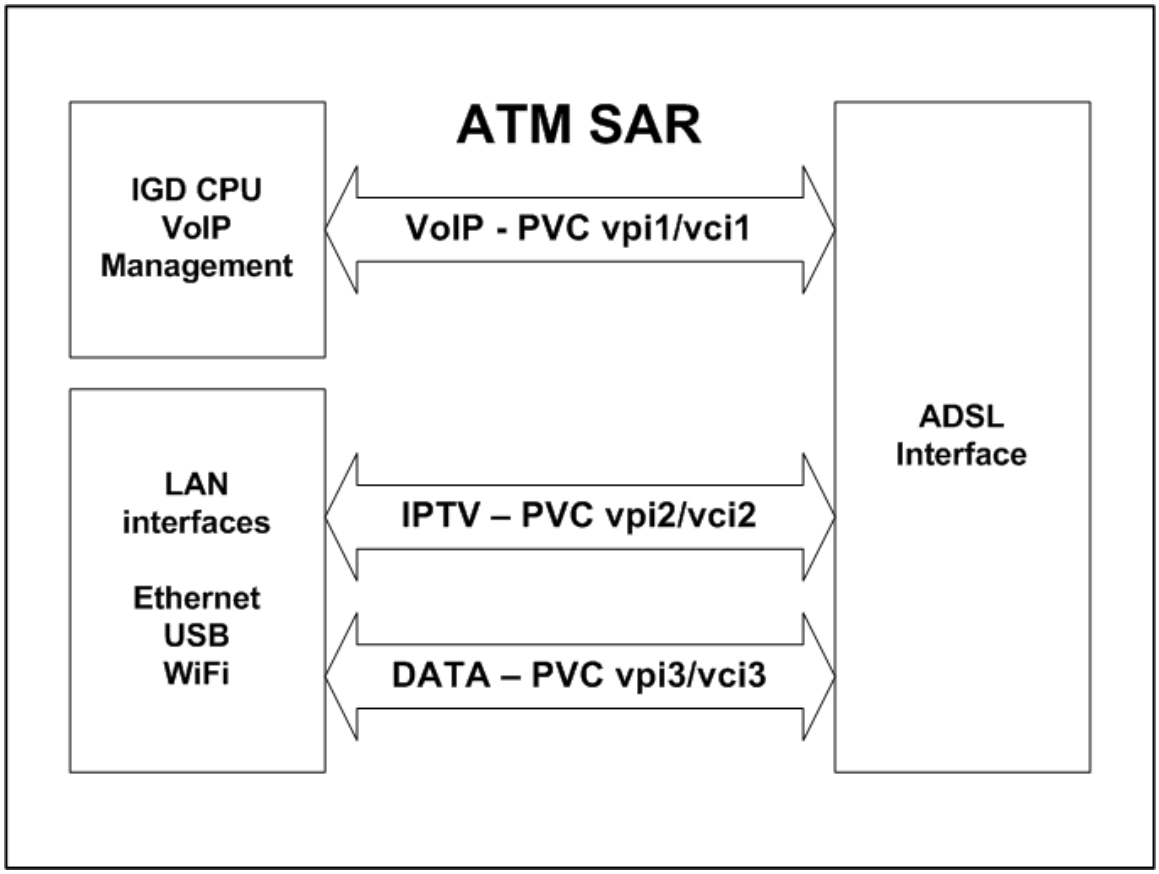

Figure 9 – IGD Physical Ingress/Egress Interfaces Block Diagram

I.2 Example Configuration

This section gives an example configuration for the scenario described in the previous section.

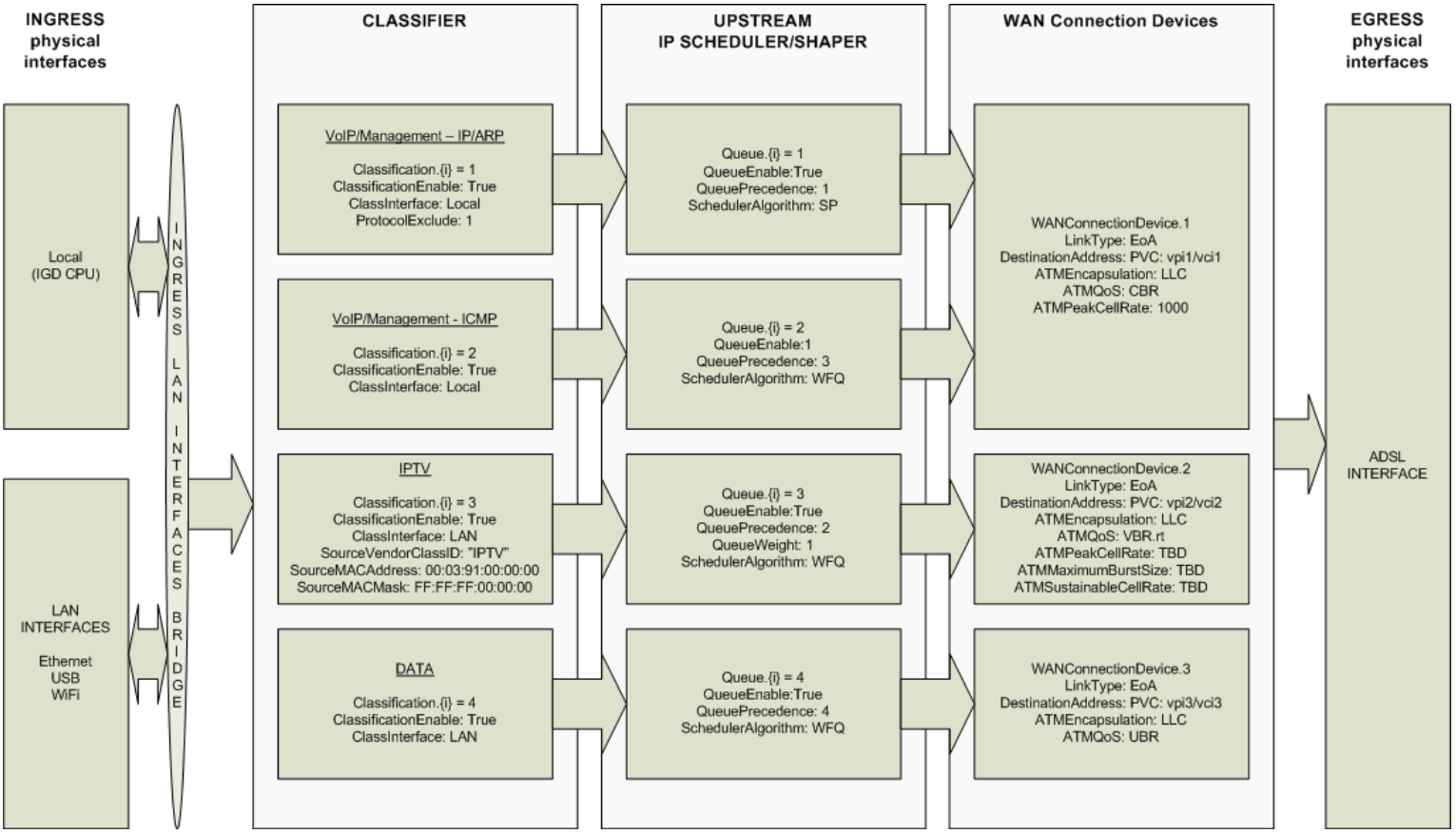

Figure 10 – IGD Upstream Data Model Diagram

Note on the IGD Local interface and the LAN interface bridge:

- Even though it is not explicitly defined in the IGD configuration, there needs to be, at least, a layer 2 bridge between the IGD local interface and the LAN interfaces, in order to perform the lP address lease negotiations between the IGD onboard DHCP server and the DHCP client connected to the IGD LAN interfaces.
- This bridge needs to have some smart features, since some of its actions are controlled by objects other than Layer2Bridging and QueueManagement. For example, transmission of DHCP messages to the WAN is controlled by InternetGatewayDevice.LANDevice.{i}.LANHostConfig-Management's parameters such as DHCPServerEnable, DHCPRelay and LocallyServed.

I.2.1 IGD WAN Connection Device Definitions

```
# WAN Connection Device definitions
#
InternetGatewayDevice.WANDevice.1. =

    # PVC 1 - VoIP and Management
    #
    WANConnectionDevice.1. =
        WANDSLLinkConfig. =
            Enable: True
            LinkType: EoA
            DestinationAddress: PVC: vpi1/vci1
```



```
            ATMEncapsulation: LLC
            ATMQoS: CBR
            ATMPeakCellRate: 1000


    # PVC 2 - IPTV
    #
    WANConnectionDevice.2. =
        WANDSLLinkConfig. =
            Enable: True
            LinkType: EoA
            DestinationAddress: PVC: vpi2/vci2
            ATMQoS: VBR.rt
            ATMPeakCellRate: TBD
            ATMMaximumBurstSize: TBD
            ATMSustainableCellRate: TBD


    # PVC 3 - DATA
    #
    WANConnectionDevice.3. =
        WANDSLLinkConfig. =
            Enable: True
            LinkType: EoA
            DestinationAddress: PVC: vpi3/vci3
            ATMQoS: UBR
```

I.2.2 IGD Default Queue Definitions

```
# Queue Management – Upstream Queue Definitions
#
InternetGatewayDevice.QueueManagement. =
    Enable: True
```

Note that, since all the queue definitions (see section I.2.4) have their own QueueInterface parameters set to a specific egress interface, which in turn identifies a PVC, it's impossible to have a single default value. Therefore the only meaningful default parameter is InternetGatewayDevice.QueueManagement.Enable; the remaining parameters are not applicable.

I.2.3 IGD Upstream Classification definitions

```
# Queue Management - Upstream Classification Definitions
#
InternetGatewayDevice.QueueManagement. =

    # Classification – IGD VoIP and Management (without ICMP protocol)
    #
    Classification.1. =
        ClassificationEnable: True
        ClassificationOrder: 1
        ClassInterface: Local
        ClassQueue: 1
        Protocol: 1
        ProtocolExclude: 1

    # Classification – IGD ICMP protocol only
    #
    Classification.2. =
        ClassificationEnable: True
        ClassificationOrder: 2
        ClassInterface: Local
```



```
        ClassQueue: 2

    # IPTV
    #
    Classification.3. =
        ClassificationEnable: True
        ClassificationOrder: 3
        ClassInterface: LAN
        ClassQueue: 3
        SourceVendorClassID: "TBD"
        SourceMACAddress: ""
        SourceMACMask: ""

    # IPTV – placeholder, to be used for quick implementation of future IPTV STB
    #
    Classification.4. =
        ClassificationEnable: False
        ClassificationOrder: 4
        ClassInterface: LAN
        ClassQueue: 3
        SourceVendorClassID: "TBD"
        SourceMACAddress: ""
        SourceMACMask: ""

    # DATA
    #
    Classification.5. =
        ClassificationEnable: True
        ClassificationOrder: 5
        ClassInterface: LAN
        ClassQueue: 4
```

Note on IPTV placeholder:

- As in the previous example, some structures in the configuration can be defined and kept disabled in order to ease the pre-configuration process.
- To add a new definition, in such cases, it is necessary only to set the placeholder parameter values and enable the object.
- This process is faster and does not require deleting all the objects and reinserting them in the new order.
- Such a process, although not difficult in itself, would require significant regression test time in order to cope with all the possible field configurations.

I.2.4 IGD Upstream Queue definitions

```
# Queue Management – Upstream Queue Definitions
#
InternetGatewayDevice.QueueManagement. =

    # Queue VoIP and Management (without ICMP protocol)
    #
    Queue.1. =
        QueueEnable: True
        QueueInterface: InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1
        QueuePrecedence: 1
        SchedulerAlgorithm: SP (Strict Priority)

    # Queue VoIP and Management (ICMP protocol only)
    #
    Queue.2. =
        QueueEnable: True
        QueueInterface: InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1
```



```
        QueuePrecedence: 3
        SchedulerAlgorithm: WFQ (Weighted Fair Queuing)

    # Queue IPTV
    #
    Queue.3. =
        QueueEnable: True
        QueueInterface: InternetGatewayDevice.WANDevice.1.WANConnectionDevice.2
        QueuePrecedence: 2
        QueueWeight: 2
        SchedulerAlgorithm: WFQ (Weighted Fair Queuing)

    # Queue Data Default
    #
    Queue.4. =
        QueueEnable: True
        QueueInterface: InternetGatewayDevice.WANDevice.1.WANConnectionDevice.3
        QueuePrecedence: 4
        SchedulerAlgorithm: WFQ (Weighted Fair Queuing)
```

I.2.5 IGD DHCP Server

```
# DHCP Server Pool – Generic for customer PCs
#
InternetGatewayDevice.LANDevice.1.LANHostConfigManagement. =
    DHCPServerEnable: True
    MinAddress: 0.0.0.0
    MaxAddress: 0.0.0.0
    ReservedAddresses: 0.0.0.0
    SubnetMask: 0.0.0.0
    DNSServers: 0.0.0.0, 0.0.0.0
    DomainName: "tbd.xx"
    IPRouters: 0.0.0.0
    DHCPLeaseTime: 1800
```

Note that IP addresses, as well as other parameter values, are just dummies for the example and would be replaced with appropriate values in a real implementation

I.2.6 IGD DHCP Conditional Serving Pool

```
# DHCP Server – Conditional Serving Pool
#
InternetGatewayDevice.LANDevice.1.LANHostConfigManagement. =

    # IPTV
    #
    DHCPConditionalServingPool.1. =
      Enable: True
      PoolOrder: 1
      VendorClass: "TBD"
      Chaddr: ""
      ChaddrMask: ""
      LocallyServed: 1
      MinAddress: 0.0.0.0
      MaxAddress: 0.0.0.0
      SubnetMask: 0.0.0.0
      DNSServers: 0.0.0.0, 0.0.0.0
      DomainName: "tbd.xx"
      IPRouters: 0.0.0.0
      DHCPLeaseTime: 1800

    # IPTV – placeholder, to be used for quick implementation of future IPTV STB
```



```
#
DHCPConditionalServingPool.2. =
  Enable: False
  PoolOrder: 2
  VendorClass: "TBD"
  Chaddr: ""
  ChaddrMask: ""
  LocallyServed: 1
  MinAddress: 0.0.0.0
  MaxAddress: 0.0.0.0
  SubnetMask: 0.0.0.0
  DNSServers: 0.0.0.0, 0.0.0.0
  DomainName: "tbd.xx"
  IPRouters: 0.0.0.0
  DHCPLeaseTime: 1800
```


Appendix II. Use of the Bridging Objects for VLAN Tagging

In the case of an Ethernet WAN Interface or a VDSL2 WAN Interface based on PTM-EFM, 802.1Q Tagging can be used to tag egress traffic on the WAN interface. This choice enables a multi-VLAN architecture in order to deploy a multi-service configuration (high speed Internet, VoIP, Video Phone, IPTV, etc.), where one VLAN or a group of VLANs are associated with each service.

If 802.1Q tagging on the WAN interface is used, it is necessary to have a way to associate LAN incoming 802.1Q tagged or untagged traffic or internally generated traffic (PPPoE, IPoE connections) to the egress (and vice-versa). The solution is to apply coherent bridging rules.

Regarding different traffic bridging rules, the possible cases characterized are the following:

- Tagged LAN to tagged WAN traffic (pure VLAN bridging), with VLAN ID translation as a special case
- Untagged LAN to tagged WAN traffic
- Internally generated to tagged WAN traffic

To better understand the different cases, refer to Figure 11 and to the following examples.

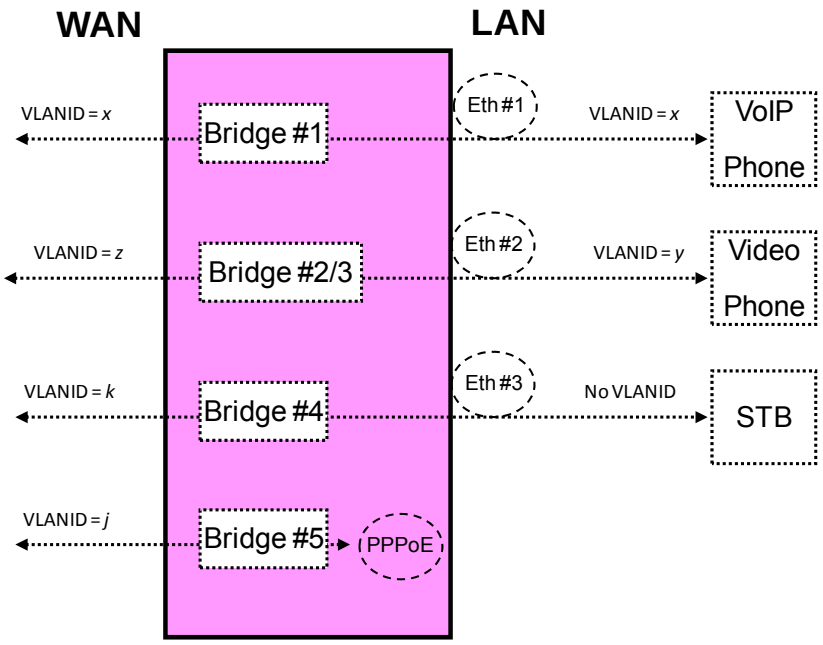

Figure 11 – Examples of VLAN configuration based on Layer2Bridging

II.1 Tagged LAN to tagged WAN traffic (VLAN bridging)

Ethernet port 1 (instance InternetGatewayDevice.LANDevice.1.LANEthernetInterfaceConfig.1) might be dedicated to VoIP service, receiving VLAN ID *x* tagged traffic from a VoIP phone, and this port would be included in the same bridge dedicated to VoIP service on the WAN interface (instance InternetGateway-Device.WANDevice.1.WANConnectionDevice.1), identified with the same VLAN ID *x*.

To achieve this, an interface-based bridge would be created using the Layer2Bridging object. A Bridge table entry would be created along with two associated Filter objects with entries for Ethernet port 1 and the WAN interface, for the VLAN ID *x* associated with VoIP.

The Layer2Bridging configuration rules for this situation are summarized in Table 47. Note that, although FilterInterface is shown as a full path name, it would in fact be the value of the corresponding Available-InterfaceKey parameter.

Table 47 – Tagged LAN to tagged WAN configuration

<table>
<tr><th>Description</th><th>Layer2Bridging TR-069 Configuration</th></tr>
<tr><td rowspan="2">Bridge between WAN and Eth-1 interfaces with VLANID=x</td><td>BRIDGE (VLANID=x)
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Bridge.{i}.</td><td>-</td></tr><tr><td>BridgeKey</td><td>1</td></tr><tr><td>BridgeEnable</td><td>True</td></tr><tr><td>BridgeName</td><td>Bridge_1</td></tr><tr><td>VLANID</td><td>x</td></tr></table></td></tr>
<tr><td>FILTER #1: with WAN interface
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterEnable</td><td>True</td></tr><tr><td>FilterBridgeReference</td><td>1</td></tr><tr><td>FilterInterface</td><td>InternetGatewayDevice.WANDevice.1.-WANConnectionDevice.1</td></tr><tr><td>VLANIDFilter</td><td>-1</td></tr><tr><td>AdmitOnlyVLANTagged</td><td>True</td></tr></table>FILTER#2 has the same parameters of FILTER#1 but is applied to Ethernet-1 interface.
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterInterface</td><td>InternetGatewayDevice.LANDevice.1.-LANEthernetInterfaceConfig.1</td></tr></table></td></tr>
</table>

II.2 Tagged LAN to tagged WAN traffic (special case with VLAN ID translation)

Ethernet port 2 (instance InternetGatewayDevice.LANDevice.1.LANEthernetInterfaceConfig.2) might be dedicated to Video Phone service, receiving VLAN ID *y* tagged traffic from a Video phone, and this port would be included in the same bridge dedicated to Video Phone service on the WAN interface (instance InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1), identified by a different VLAN ID (VLAN ID *z*). In this case a VLAN translation needs to be performed.

To achieve this, a pair of unidirectional interface-based bridges would be created using the Layer2Bridging object, one for LAN-to-WAN traffic and the other for WAN-to-LAN traffic. For each bridge, a Bridge table entry would be created along with two associated Filter object entries for {Ethernet port 2/VLAN ID *y*} and {WAN interface/VLAN ID *z*}, to identify ingress frames. After that, in order to re-mark the egress frames appropriately, a Marking object would also be created for each bridge, with Marking table entries for the egress interfaces: {Ethernet port 2/VLAN ID *y*} and {WAN interface/VLAN ID *z*}.

> *Note – if a single bi-directional bridge had been used, then in order to define the VLAN Member Sets correctly Filter entries for both VLAN ID y and VLAN ID z would be needed for each of the bridge interfaces. This would permit ingress of VLAN ID z packets to the LAN interface, and of VLAN ID y packets to the WAN interface, which would be incorrect behavior. With the two-bridge approach, the LAN-to-WAN bridge bridges only VLAN ID y packets (marked z on egress), and the WAN-to-LAN bridge bridges only VLAN ID z packets (marked y on egress).*

The Layer2Bridging configuration rules for this situation are summarized in Table 48 (LAN-to-WAN) and Table 49 (WAN-to-LAN). Note that, although FilterInterface and MarkingInterface are shown as full path names, they would in fact be the values of the corresponding AvailableInterfaceKey parameters.

Table 48 – Tagged LAN to tagged WAN configuration (VLAN ID translation; LAN-to-WAN)

<table>
<tr><th>Description</th><th>Layer2Bridging TR-069 Configuration</th></tr>
<tr><td rowspan="3">Unidirectional bridge with VLAN translation between Eth-2 (VLANID=y) and WAN (VLANID=z)</td><td colspan="2">BRIDGE (VLANID=y)
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Bridge.{i}.</td><td>-</td></tr><tr><td>BridgeKey</td><td>2</td></tr><tr><td>BridgeEnable</td><td>True</td></tr><tr><td>BridgeName</td><td>Bridge_2</td></tr><tr><td>VLANID</td><td>y</td></tr></table></td></tr>
<tr><td>FILTER#1: WAN interface (no ingress; excludes all Ethertypes)
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterEnable</td><td>True</td></tr><tr><td>FilterBridgeReference</td><td>2</td></tr><tr><td>FilterInterface</td><td>InternetGateway-Device.-WANDevice.1.-WANConnection-Device.1</td></tr><tr><td>VLANIDFilter</td><td>y</td></tr><tr><td>AdmitOnlyVLANTagged</td><td>False</td></tr><tr><td>EthertypeFilterList</td><td><Empty></td></tr><tr><td>EthertypeFilterExclude</td><td>False</td></tr></table></td><td>MARKING #1: WAN interface and VLANIDMark=z (Override=True)
<table><tr><td>InternetGatewayDevice.-Layer2Bridging.Marking.{i}.</td><td>-</td></tr><tr><td>MarkingEnable</td><td>True</td></tr><tr><td>MarkingBridgeReference</td><td>2</td></tr><tr><td>MarkingInterface</td><td>InternetGateway-Device.WANDevice.1.-WANConnection-Device.1</td></tr><tr><td>VLANIDUntag</td><td>False</td></tr><tr><td>VLANIDMark</td><td>z</td></tr><tr><td>VLANIDMarkOverride</td><td>True</td></tr></table></td></tr>
<tr><td>FILTER#2: Eth-2 interface and VLANIDFilter=y
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterEnable</td><td>True</td></tr><tr><td>FilterBridgeReference</td><td>2</td></tr><tr><td>FilterInterface</td><td>InternetGateway-Device.LANDevice.1.-LANEthernetInterface-Config.2.</td></tr><tr><td>VLANIDFilter</td><td>y</td></tr><tr><td>AdmitOnlyVLANTagged</td><td>True</td></tr><tr><td>EthertypeFilterList</td><td><Empty></td></tr><tr><td>EthertypeFilterExclude</td><td>True</td></tr></table></td><td>MARKING #2: not needed (no LAN egress for this bridge)</td></tr>
</table>

Table 49 – Tagged LAN to tagged WAN configuration (VLAN ID translation; WAN-to-LAN)

<table>
<tr><th>Description</th><th colspan="2">Layer2Bridging TR-069 Configuration</th></tr>
<tr><td rowspan="3">Unidirectional bridge with VLAN translation between WAN (VLANID=z) and Eth-2 (VLANID=y)</td><td colspan="2">BRIDGE (VLANID=z)
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Bridge.{i}.</td><td>-</td></tr><tr><td>BridgeKey</td><td>3</td></tr><tr><td>BridgeEnable</td><td>True</td></tr><tr><td>BridgeName</td><td>Bridge_3</td></tr><tr><td>VLANID</td><td>z</td></tr></table></td></tr>
<tr><td>FILTER#1: WAN interface and VLANIDFilter=z
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterEnable</td><td>True</td></tr><tr><td>FilterBridgeReference</td><td>3</td></tr><tr><td>FilterInterface</td><td>InternetGateway-Device.-WANDevice.1.-WANConnection-Device.1</td></tr><tr><td>VLANIDFilter</td><td>z</td></tr><tr><td>AdmitOnlyVLANTagged</td><td>True</td></tr><tr><td>EthertypeFilterList</td><td><Empty></td></tr><tr><td>EthertypeFilterExclude</td><td>True</td></tr></table></td><td>MARKING #1: not needed (no WAN egress for this bridge)</td></tr>
<tr><td>FILTER#2: Eth-2 interface (no ingress; excludes all Ethertypes)
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterEnable</td><td>True</td></tr><tr><td>FilterBridgeReference</td><td>3</td></tr><tr><td>FilterInterface</td><td>InternetGateway-Device.LANDevice.1.-LANEthernetInterface-Config.2.</td></tr><tr><td>VLANIDFilter</td><td>z</td></tr><tr><td>AdmitOnlyVLANTagged</td><td>False</td></tr><tr><td>EthertypeFilterList</td><td><Empty></td></tr><tr><td>EthertypeFilterExclude</td><td>False</td></tr></table></td><td>MARKING #2: Eth-2 interface and VLANIDMark=y (Override=True)
<table><tr><td>InternetGatewayDevice.-Layer2Bridging.-Marking.{i}.</td><td>-</td></tr><tr><td>MarkingEnable</td><td>True</td></tr><tr><td>MarkingBridgeReference</td><td>3</td></tr><tr><td>MarkingInterface</td><td>InternetGatewayDevice.-LANDevice.1.-LANEthernetInterface-Config.2.</td></tr><tr><td>VLANIDUntag</td><td>False</td></tr><tr><td>VLANIDMark</td><td>y</td></tr><tr><td>VLANIDMarkOverride</td><td>True</td></tr></table></td></tr>
</table>

II.3 Untagged LAN to tagged WAN traffic

Ethernet port 3 (instance InternetGatewayDevice.LANDevice.1.LANEthernetInterfaceConfig.3) might be dedicated to IPTV service, receiving untagged traffic from a STB, and this port would be included in the same bridge dedicated to IPTV service on the WAN interface (instance InternetGatewayDevice.WAN-Device.1.WANConnectionDevice.1), identified with the VLAN ID *k*.

To achieve this, an interface-based bridge would be created using the Layer2Bridging object. A Bridge table entry would be created along with two associated Filter objects entries for {Ethernet port 3/No VLAN ID Tag} and {WAN interface/VLAN ID *k*}, to identify ingress frames. After that, in order to re-mark the egress frames appropriately, two Marking objects would also be created, with Marking table entries for {Ethernet port 3/No VLAN ID Tag} and {WAN interface/VLAN ID *k*}.

> *Note – the second Marking object is not in fact necessary, because untagged frames arriving on Ethernet port 3will be associated with the port VLAN ID (PVID) k on ingress. However, it does no harm.*

The Layer2Bridging configuration rules for this situation are summarized in Table 50. Note that, although FilterInterface and MarkingInterface are shown as full path names, they would in fact be the values of the corresponding AvailableInterfaceKey parameters.

Table 50 – Untagged LAN to tagged WAN configuration

<table>
<tr><th>Description</th><th>Layer2Bridging TR-069 Configuration</th></tr>
<tr><td rowspan="3">Bridge between WAN (VLANID=k) and Eth-3 untagged</td><td>BRIDGE (VLANID=k)
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Bridge.{i}.</td><td>-</td></tr><tr><td>BridgeKey</td><td>4</td></tr><tr><td>BridgeEnable</td><td>True</td></tr><tr><td>BridgeName</td><td>Bridge_4</td></tr><tr><td>VLANID</td><td>k</td></tr></table></td></tr>
<tr><td>FILTER #1: WAN interface and VLANIDFilter=k
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterEnable</td><td>True</td></tr><tr><td>FilterBridgeReference</td><td>4</td></tr><tr><td>FilterInterface</td><td>InternetGateway-Device-.WANDevice.1.-WANConnection-Device.1</td></tr><tr><td>VLANIDFilter</td><td>-1</td></tr><tr><td>AdmitOnlyVLANTagged</td><td>True</td></tr></table></td><td>MARKING #1: WAN interface and VLANIDMark=k (Override=True)
<table><tr><td>InternetGatewayDevice.-Layer2Bridging.Marking.{i}.</td><td>-</td></tr><tr><td>MarkingEnable</td><td>True</td></tr><tr><td>MarkingBridgeReference</td><td>4</td></tr><tr><td>MarkingInterface</td><td>InternetGateway-Device-.WANDevice.1.-WANConnection-Device.1</td></tr><tr><td>VLANIDUntag</td><td>False</td></tr><tr><td>VLANIDMark</td><td>k</td></tr><tr><td>VLANIDMarkOverride</td><td>True</td></tr></table></td></tr>
<tr><td>FILTER #2: Eth-3 interface and VLANIDFilter=-1 (AdmitOnlyVLANTagged=False)</td><td>MARKING #2: Eth-3 interface and VLANIDUntag=True
<table><tr><td>InternetGatewayDevice.-Layer2Bridging.Marking.{i}.</td><td>-</td></tr><tr><td>MarkingEnable</td><td>True</td></tr><tr><td>MarkingBridgeReference</td><td>4</td></tr><tr><td>MarkingInterface</td><td>Internet-Gateway-Device.-LANDevice.1.-LANEthernet-Interface-Config.3.</td></tr></table></td></tr>
</table>

InternetGatewayDevice.Layer2-Bridging.Filter.{i}.	-
FilterEnable	True
FilterBridgeReference	4
FilterInterface	Internet-Gateway-Device.-LANDevice.1.-LANEthernet-Interface-Config.3.
VLANIDFilter	-1
AdmitOnlyVLANTagged	False

VLANIDUntag	True
VLANIDMark	-1
VLANIDMarkOverride	False

II.4 Internally generated to tagged WAN traffic

A CPE PPPoE internal session (instance InternetGatewayDevice.WANDevice.1.WANConnectionDevice.-1.WANPPPConnection.1) might be dedicated to Management service and this logical interface would be included in the same bridge with the VLAN ID *j* dedicated to Management service on the WAN interface (instance InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1).

To achieve this, an interface-based bridge would be created using the Layer2Bridging object. A Bridge table entry would be created along with the two associated Filter table entries for the PPP and WAN interfaces, to identify ingress frames. After that, in order appropriately to re-mark the egress frames, one Marking object would also be created for the WAN interface and VLAN ID *j*.

> *Note – the Marking object is not in fact necessary, because untagged frames arriving on the PPP interface will be associated with the port VLAN ID (PVID) j on ingress. However, it does no harm.*

The Layer2Bridging configuration rules for this situation are summarized in Table 51. Note that, although FilterInterface and MarkingInterface are shown as full path names, they would in fact be the values of the corresponding AvailableInterfaceKey parameters.

Table 51 – Internally generated to tagged WAN configuration

<table>
<tr><th>Description</th><th>Layer2Bridging TR-069 Configuration</th></tr>
<tr><td rowspan="2">Management traffic with PPP and WAN interface with VLAN ID=j</td><td>BRIDGE (VLANID=j)
<table><tr><td>InternetGatewayDevice.Layer2-Bridging.Bridge.{i}.</td><td>-</td></tr><tr><td>BridgeKey</td><td>5</td></tr><tr><td>BridgeEnable</td><td>True</td></tr><tr><td>BridgeName</td><td>Bridge_5</td></tr><tr><td>VLANID</td><td>j</td></tr></table></td></tr>
<tr><td>FILTER #1: with WAN interface
<table><tr><td>InternetGatewayDevice.-Layer2Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterEnable</td><td>True</td></tr><tr><td>FilterBridgeReference</td><td>5</td></tr><tr><td>FilterInterface</td><td>InternetGateway-Device.WANDevice.1.-WANConnection-Device.1</td></tr><tr><td>VLANIDFilter</td><td>-1</td></tr><tr><td>AdmitOnlyVLANTagged</td><td>False</td></tr></table>FILTER#2 has the same parameters of FILTER#1 but is applied to the PPP interface
<table><tr><td>InternetGatewayDevice.-Layer2Bridging.Filter.{i}.</td><td>-</td></tr><tr><td>FilterInterface</td><td>InternetGateway-Device.WANDevice.1.-WANConnection-Device.1.-WANPPPConnection.1</td></tr></table></td><td>MARKING #1: WAN interface and VLANIDMark=j (Override=True)
<table><tr><td>InternetGatewayDevice.-Layer2Bridging.Marking.{i}.</td><td>-</td></tr><tr><td>MarkingEnable</td><td>True</td></tr><tr><td>MarkingBridgeReference</td><td>5</td></tr><tr><td>MarkingInterface</td><td>Internet-Gateway-Device.-WANDevice.1.WANConnectionDevice.1</td></tr><tr><td>VLANIDUntag</td><td>False</td></tr><tr><td>VLANIDMark</td><td>j</td></tr><tr><td>VLANIDMarkOverride</td><td>True</td></tr></table></td></tr>
</table>

II.5 Other issues

The previous rules can be applied to allow all combinations of traffic. If the subscriber's services are modified, the Layer2Bridging configuration might need to be modified accordingly.

It can be interesting to detail the configuration of three special cases:

- More than one LAN interface in a bridge
- 802.1D (re-)marking
- More than one VLAN ID tag for the same LAN interface

II.5.1 More than one LAN interface in a bridge

Referring to the example in section II.2, Tagged LAN to tagged WAN traffic (special case with VLAN ID translation), consider adding another Ethernet interface (e.g. Ethernet port 4 = instance InternetGatewayDevice.LANDevice.1.LANEthernetInterfaceConfig.4) to the Video Phone service. The behaviour is the same as for the existing Ethernet port 2 (instance InternetGatewayDevice.LANDevice.1.LANEthernetInterfaceConfig.2).

To achieve this, new Filter and Marking entries #3 need to be added for interface Eth-4. The Layer2Bridging configuration rules for this situation are summarized in Table 52 and Table 53.

Table 52 – Changes to configuration from Table 48 (LAN-to-WAN)

Description	Layer2Bridging TR-069 Configuration	
Bridge with VLAN translation between Eth-2/Eth-4 (VLANID=*y*) and WAN (VLANID=z)	See Table 48 for detailed parameters BRIDGE (VLANID=*y*)	
	FILTER#1: WAN interface and VLANIDFilter=*y* (no ingress)	MARKING #1: WAN interface and VLANIDMark=z (Override=True)
	FILTER#2: Eth-2 interface and VLANIDFilter=*y*	
	FILTER#3: Eth-4 interface and VLANIDFilter=*y*	

Table 53 – Changes to configuration from Table 49 (WAN-to-LAN)

Description	Layer2Bridging TR-069 Configuration	
Bridge with VLAN translation between WAN (VLANID=z) and Eth-2/Eth-4 (VLANID=*y*)	See Table 49 for detailed parameters BRIDGE (VLANID=z)	
	FILTER#1: WAN interface and VLANIDFilter=z	
	FILTER#2: Eth-2 interface and VLANIDFilter=z (no ingress)	MARKING #2: Eth-2 interface and VLANIDMark=*y* (Override=True)
	FILTER#3: Eth-4 interface and VLANIDFilter=z (no ingress)	MARKING #3: Eth-4 interface and VLANIDMark=*y* (Override=True)

II.5.2 802.1D (re-)marking

The 802.1Q Tag includes the 802.1D user priority bits field. All the previous cases can also be extended to mark (or re-mark) this 802.1D field. To achieve this, in the Marking object defined (or added, if not already present), the EthernetPriorityMark and EthernetPriorityOverride parameters need to be configured with the desired values. The Layer2Bridging configuration rules for the case of management traffic are summarized in Table 54. Compare it with Table 51.

Table 54 – Changes to configuration from Table 51

Description	Layer2Bridging TR-069 Configuration
Management traffic with PPP and WAN interface with VLANID=*j*	See Table 51 for detailed parameters BRIDGE (VLANID=*j*)

	FILTER #1: with WAN interface FILTER #2: with PPP interface	MARKING #1: WAN interface and VLANIDMark=*j* (Override=True) InternetGatewayDevice.-Layer2Bridging.Marking.{i}. - EthernetPriorityMark *p* EthernetPriorityOverride True

II.5.3 More than one VLAN ID tag admitted on the same LAN interface

Another scenario that can be further detailed is the case of more than one VLAN ID tag admitted on the same LAN interface. A practical example would be a 2 box scenario, with a User Device generating traffic segregated in multiple VLANs (e.g. a router offering services to the customer), and an Internet Gateway Device, providing WAN connectivity to the Access Network, with the connection between the two pieces of equipment using an Ethernet interface.

In this case, we assume the User Device is able to tag the different traffic flows, segregating the different services (Voice, Video, …) into different VLANs. The IGD needs, on the same LAN interface, to be able to receive different VLAN ID and correctly forward or translate to the WAN interface (and vice versa). To achieve this, appropriate Layer2Bridging objects need to be configured.

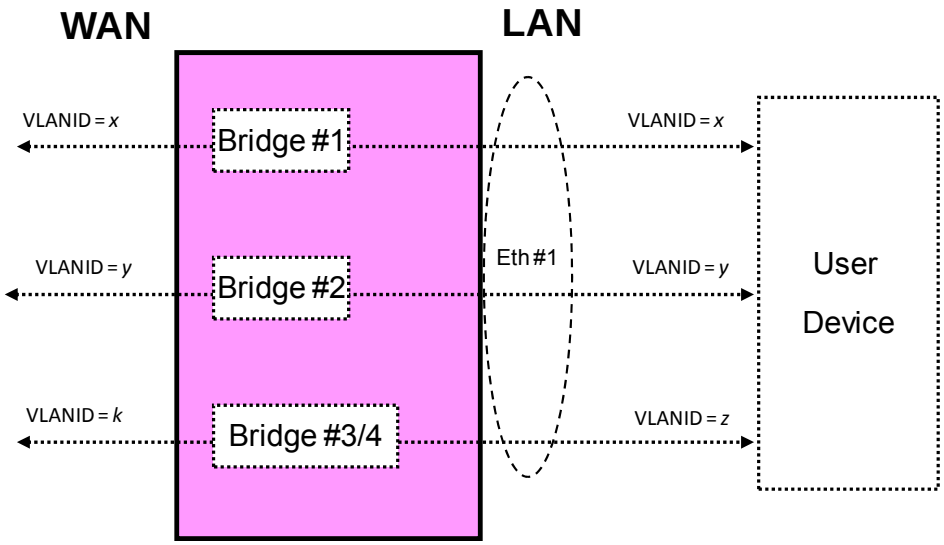

Figure 12 – Example of VLAN configuration in a 2 box scenario

Referring to Figure 12 as an example, assume the case of three VLANs (VLAN ID=x,y,z) offered by a User Device to the IGD on the same LAN interface (Eth-1). The IGD bridges two of them (VLAN ID=x,y) and translates the other one (VLAN ID=z) to the WAN interface (VLAN ID=*k*).

On the IGD, this can be achieved using a combination of the Layer2Bridging objects detailed in the preceding sections, with 3 bridge entries and their related Filter and Marking entries. Refer to Table 55 for the global configuration.

Table 55 – More than one VLAN ID tag admitted on the same LAN interface

Description	Layer2Bridging TR-069 Configuration
Bridge between WAN and Eth-1 interfaces with VLANID=x	See Table 47 for detailed parameters BRIDGE #1 (VLANID=*x*)

	FILTER #1: with WAN interface FILTER #2: with Eth-1 interface	
Bridge between WAN and Eth-1 interfaces with VLANID=y	See Table 47 for detailed parameters BRIDGE #2 (VLANID=*y*)	
	FILTER #1: with WAN interface FILTER #2: with Eth-1 interface	
Unidirectional bridge with VLAN translation between Eth-1 (VLANID=*z*) and WAN (VLANID=*k*)	See Table 48 for detailed parameters BRIDGE #3 (VLANID=*z*)	
	FILTER#1: WAN interface (no ingress)	MARKING #1: WAN interface and VLANIDMark=*k* (Override=True)
	FILTER#2: Eth-1 interface	MARKING #2: not needed (no egress)
Unidirectional bridge with VLAN translation between WAN (VLANID=*k*) and Eth-1 (VLANID=*z*)	See Table 49 for detailed parameters BRIDGE #4 (VLANID=*k*)	
	FILTER#1: WAN interface	MARKING #1: not needed (no egress)
	FILTER#2: Eth-1 interface (no ingress)	MARKING #2: Eth-1 interface and VLANIDMark=z (Override=True)

