TECHNICAL REPORT

DSL Forum
TR-064

LAN-Side DSL CPE Configuration

May 2004

Produced by:
DSLHome-Technical Working Group

Editor:
Barbara Stark, BellSouth

Working Group Co-Chairs:
Greg Bathrick, Texas Instruments
George Pitsoulakis, Westell

Abstract:

This Working Text will specify the method for configuring DSL CPE through software on PCs inside the LAN.

Notice:

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

Table of Contents

1 Introduction

This document describes a specific implementation to be used for DSL CPE LAN-side configuration. The goal is for this to become the standard interface for this sort of configuration.

A summary of the interface described in this document is shown in Table 1. This includes authentication for security, use of standard protocols to facilitate rapid adoption, CPE address discovery, common access for all CPE types, and an OS agnostic management application.

Table 1: Summary of LAN-Side Configuration Interface

Aspect	Solution
Management Protocol	Standards based XML over SOAP protocol.
Parameter Model	Parameters defined using UPnP model as base, disallowing values and parameters that are inconsistent with DSL model, and adding objects as needed for DSL CPE.
Security	HTTP Digest Authentication; optional SSL 3.0 or TLS 1.0 encryption.
CPE Type	Supports Bridge/Router/PPPoE on-board IP pass-thru CPEs.
Management Usage	CPE turn-up, status determination, monitoring, diagnostics.
CPE discovery	Standards-based DHCP and SSDP device discovery.
OS Support	CPE management app with integrated XML over SOAP stack to operate on any OS, native XML/SOAP OS support is not required or desired.
Proprietary/Open	Standards-based with open interface published.

2 Scope

This document describes a specific implementation to be used for DSL CPE LAN-side configuration. It describes the protocol stack, discovery, security, and the XML structure to be used. It does not attempt to address WAN-side configuration, although many of the XML parameters and structures can be used for that, as well. Figure 1 shows the architecture considered in this document. Use cases that show the sequence of communication are included in Appendix A. This Device Control Protocol (DCP) is compliant with UPnP Device Architecture 1.0 (UDA). If there are any ambiguities or conflicts in the text, it MUST be assumed that UDA compliance supercedes any other requirements.

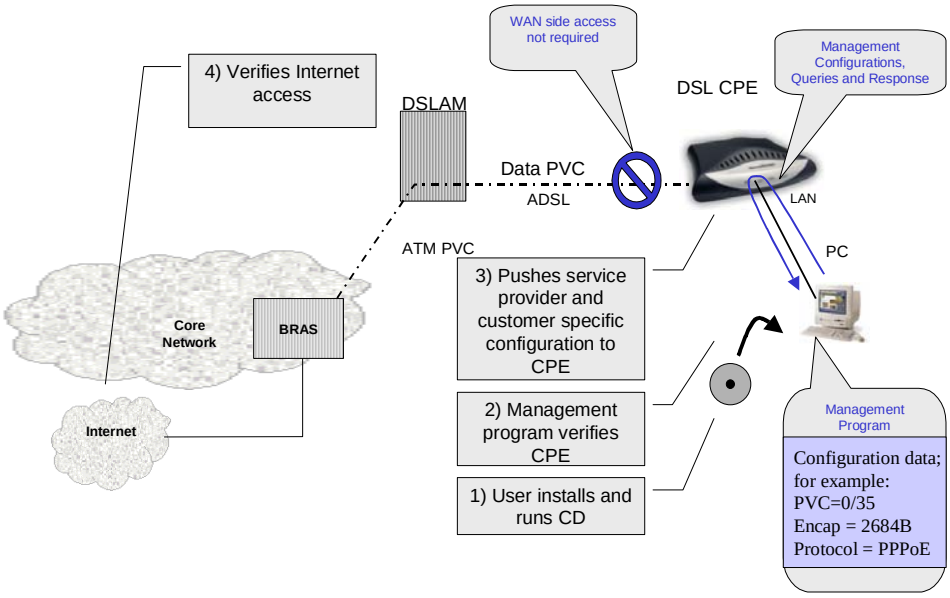


Figure 1: LAN-side DSL CPE Management Example

3 Discovery

3.1 CPE LAN Address Assignment

The CPE may be a router or a bridge.

If the CPE is a router with a DHCP server then it will have a default LAN IP address for management visible as the default gateway address to DHCP clients. However, if the router is misconfigured or the CPE is a bridge, its management address may not be well known. Also, if the CPE is not running a DHCP server then there is no way to assure that it has a unique IP address on the LAN related to other LAN devices when it has a static management IP address assignment. On most PC operating systems, if no DHCP server is present the PC will auto-IP on a private 169.254.x.x subnet. In this case, the PC cannot communicate with the CPE because it will not emit packets on another subnet. The PC will only emit packets if it has a default route (which it does not have when it auto-IPs) or the destination is on the same subnet that it auto-IPed to. Therefore, if no DHCP server is enabled on the CPE, the CPE MUST perform auto-IP as defined in <draft-ietf-zeroconf-ipv4-linklocal-07.txt> so that it chooses a unique, unused IP address.

3.2 Simple Service Discovery Protocol (SSDP) Implementation

The question then becomes how to find the CPE IP address. Fortunately, a protocol called Simple Service Discovery Protocol (SSDP) [2] already exists to discover LAN devices. In order to be consistent across all CPE types, SSDP MUST be implemented to notify the DSL CPE management application of its CPE IP address. SSDP is an existing, well known protocol that supports device discovery and identification. SSDP is defined in <draft-cai-ssdp-v1-03.txt>. Some common operating systems including Window ME and Windows XP have native support for SSDP. With native SSDP support the operating system discovers all SSDP clients on the LAN and populates the network neighborhood with icons that allow users to go straight to the devices. SSDP is gaining wider support but it must be acknowledged that non-Windows OSes do not natively support SSDP and 1st generation network device implementations of SSDP are chatty.

Therefore, SSDP MUST be implemented in the CPE management application to ensure that the CPE management application is OS independent.

3.2.1 SSDP Advertisement on Start-up of CPE Device

When started the CPE MUST broadcast SSDP discovery advertisement (NOTIFY) messages as specified in the UDA. In summary, the requirements are as follows:

1. Three announcements for the root device, with the following NT header values:
    a. "NT: upnp:rootdevice" to announce itself as a UPnP root device
    b. "NT: uuid:*device-UUID*" to announce itself via its UUID (UUIDs are described in the next section)
    c. "NT: urn:dslforum-org:device:InternetGatewayDevice:1" to announce itself as a TR-064-compliant CPE device
2. Two announcements for each embedded device, with the following NT header values:
    a. "NT: urn:dslforum-org:device:*deviceType*:1"
    b. "NT: uuid:*device-UUID*" to announce the embedded device via its UUID
3. One announcement for each service type in each device, with the following NT header value:
    a. "NT: urn:dslforum-org:service:*serviceType*:1"
4. Each announcement specifies a lease time; each of the above announcements must be re-sent prior to the expiry of the lease. The UDA recommends that they be re-sent at randomly-distributed intervals of less than one half of the lease time. The UDA specifies that this lease time MUST be no less than 30 minutes; devices compliant with this DCP have a further restriction that the minimum lease value MUST be no less than 24 hours (86400 seconds) and SHOULD be no less than 2 weeks (1209600 seconds).

Following is an example series of messages corresponding to the required announcements specified above:

Announcement as UPnP Root Device

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age=1209600
LOCATION: http://192.168.1.254:51004/devicedesc.xml
NT: upnp:rootdevice
NTS: ssdp:alive
SERVER: pX/2.0 UPnP/1.0 Server/1.0
USN: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e::upnp:rootdevice
```

The other discovery messages are the same format with the following differences in NT and USN:

Announcement via Device UUID

```
NT: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e
USN: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e
```

Announcement as a TR-064 compliant device

```
NT: urn:dslforum-org:device:InternetGatewayDevice:1
USN: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e::urn:dslforum-
org:device:InternetGatewayDevice:1
```

Announcement of embedded devices (LANDevice in this example)

```
NT: urn:dslforum-org:device:LANDevice:1
USN: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e::urn:dslforum-
org:device:LANDevice:1
```

Announcement via UUID for the embedded devices:

```
NT: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e
USN: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e
```

Announcement of embedded service within each device (Layer3Forwarding in this example):

```
NT: urn:dslforum-org:service:Layer3Forwarding:1
USN: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e::urn:dslforum-
org:service:Layer3Forwarding:1
```

3.2.2 SSDP Search on Start-up of CPE Management Application

When started the CPE management application MUST broadcast a SSDP discovery request message (M-SEARCH) with a specific embedded device Search Type of “ST: urn:dslforum-org:device:InternetGatewayDevice:1” so that only the DSL CPE will respond. An example discover request is shown below:

```
M-SEARCH * HTTP/1.1
MX: 10
ST: urn:dslforum-org:device:InternetGatewayDevice:1
HOST: 239.255.255.250:1900
MAN: "ssdp:discover”
```

When the CPE replies, it MUST reply to the SSDP discovery request and identify itself by unique name in the form of “USN: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e”. The 128 bit Hex string “739f…243e” is a UUID [see http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-00.txt] of which the last 48 bits are the CPE’s LAN MAC address (if the CPE has more than one LAN interface, it should consistently use the MAC address of its primary LAN interface). Note that the CPE must generate a UUID not only for its root device but also for each of its embedded devices. Typically this is done by concatenating various strings that constitute a unique device signature, calculating their MD5 hash, and concatenating the first 80 bits of the hash with the 48 bit MAC address to generate the UUID. The MAC address is required so that the management app is aware of the MAC address in case the management app is required to manually add an ARP and route entry. The Location URL reported MUST be unique from those used for UPnP access. Therefore, when the CPE returns the SSDP discover response message, it MUST indicate the IP address and port number in the Location field. For example, this will take the form, “Location: http://192.168.0.1:<port>/devicedesc.xml”. An example response is shown below:

```
HTTP/1.1 200  OK
ST: urn:dslforum-org:device:InternetGatewayDevice:1
EXT:
```

```
SERVER: pX/2.0 UPnP/1.0 Server/1.0
USN: uuid:739f75f0-a90c-4e42-ac13-2cc42d3c243e::urn:dslforum-
org:device:InternetGatewayDevice:1
CACHE-CONTROL: max-age=1209600
LOCATION: http://192.168.0.1:51004/devicedesc.xml

Content-Length: 0
```

The basic SSDP message flow between the CPE management app and CPE is shown below in Figure 2.

The DDD MUST be supported as specified in the UPnP device architecture [2]; this DDD MUST contain service descriptor document (SCPD) URL (‘s). These SCPD URLs MAY be used by the control point to retrieve the service descriptor documents to learn which (optional) actions (and associated parameters) are implemented.

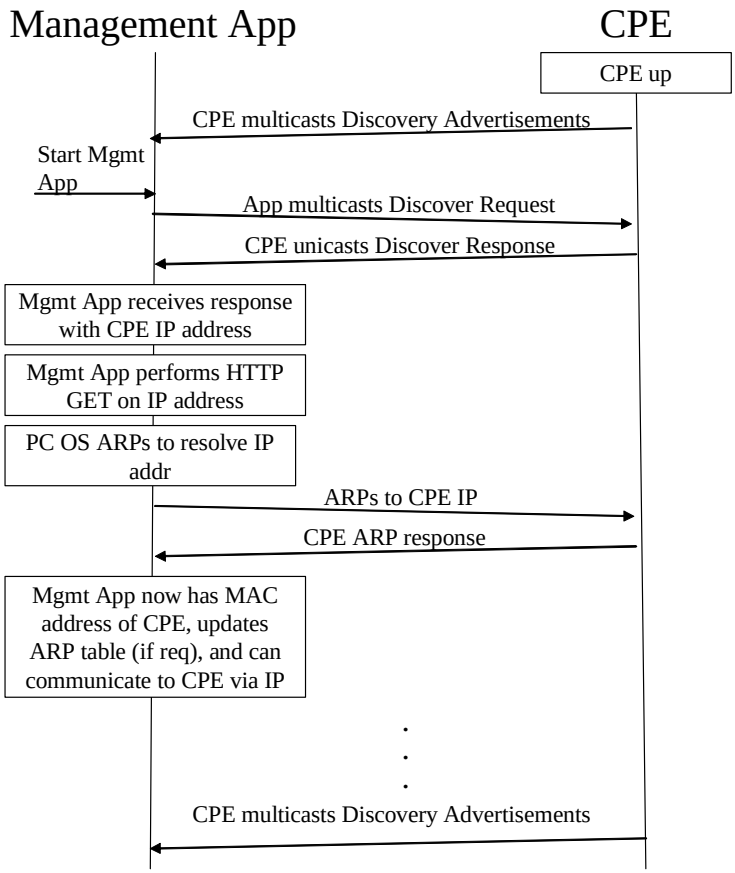


Figure 2: CPE Discovery and Communications

4 Security

Security is important in order to protect CPE from intentional or unintentional misconfiguration. The main objective is to provide authentication in order to prevent unauthorized configuration.

4.1 Access Restriction

Access to any action that allows configuration changes to the CPE MUST be password protected.

Access to actions limited to read-only CPE information such as status or statistics SHOULD NOT be password protected.

> *Note – allowing unprotected access to read-only information should avoid the need to store the password in the client's computer since it would only be needed to make configuration changes. This should improve overall security since a stored password could itself be vulnerable.*

Sensitive information, such as passwords, MUST NOT be readable at all. Attempts to read such information MUST NOT return the actual state value, but rather a well-known value such as an empty string.

4.2 Authentication

Access to any password-protected action MUST require HTTP digest authentication. This is achieved by using both the POST path of the control URL (which is unique for each service instance) and the SOAP action "urn:dslforum-org:service:serviceType:v#actionName" to uniquely identify actions that require authentication.

Two distinct username-password pairs are defined to allow access to the CPE, each with different access privileges. Since this specification does not define distinct access privileges associated with different users, a fixed username value shall be defined for each. Support for both pairs is REQUIRED.

The following table describes the use of each username-password pair.

Username	**Password**	**Access**
"dslf-config"	ConfigPassword state variable as set via the LANConfigSecurity service.	Allows access to all password-protected actions.
"dslf-reset"	ResetPassword as printed on the CPE packaging. The ResetPassword cannot be modified.	Allows access to only the SetConfigPassword action so that the ConfigPassword may be set to a new value.

> *Note – while the ConfigPassword is settable by the client software, this specification makes no assumptions about the source of this password—whether it is selected by the user, selected by the client software, or obtained by some other means.*
>
> *Note – since the username values are fixed, there is no need for these values to be visible in the user interface of the client software.*

After establishment of a session to a password-protected action, the CPE MUST automatically close that session after 30 minutes if not closed by the client. Subsequent access to password-protected actions MUST be authenticated again.

4.3 Password Initialization

Two distinct states are defined for the CPE with respect to authentication: Factory state and Normal state. The Factory state is defined to allow the ConfigPassword to be set for the first time without requiring the user to have knowledge of some other password to do so. The required CPE behavior for these two states is defined in the following table:

State	Condition	Access Restrictions
Factory	Initial state of the CPE from the factory.	**Actions** **Access** SetConfigPassword Unrestricted Password-protected actions No access Unprotected actions Unrestricted
Normal	After the action SetConfigPassword has been successfully used at least once to set the ConfigPassword.	**Actions** **Access** SetConfigPassword Authentication required: dslf-config OR dslf-reset Password-protected actions Authentication required: dslf-config Unprotected actions Unrestricted

Note – the client software need not know a priori which state the CPE is in when attempting to set the ConfigPassword. Instead it may rely on whether or not it receives an HTTP 401 response code on an attempt to access the SetConfigPassword action.

4.4 Encryption

Access to the LANConfigSecurity service SHOULD occur over an encrypted https link using either SSL 3.0 or TLS 1.0. For a CPE that supports https access to the LANConfigSecurity service, the CPE MAY also support unencrypted http access, but is not required to do so. To avoid the need for the CPE to incorporate a public-key certificate, the minimum level of security required for an https connection is Anonymous Key Exchange as defined in the SSL/TLS specifications.

A client or CPE MAY also support the use of https for services other than LANConfigSecurity. If so, it MUST also support unencrypted http access to these services as well.

4.5 Co-existence with UPnP IGD

The CPE management method outlined here and UPnP IGD may be implemented independently. If the CPE implements both UPnP IGD and DSL CPE Management, it may share protocol stacks and information models but the DSL CPE management method MUST be secured as described in the above sections. This approach provides the flexibility to allow UPnP IGD to operate as-is while also providing the security required for CPE configuration management.

The use of the URN prefix "dslforum-org" for this DCP indicates the following differences from the standard UPnP gateway IGD DCP:

- Actions marked as secure will be authenticated using HTTP digest authentication.
- Variables defined within this document have no defined events.
- The services in this DCP exist in the DSL Forum schema namespace so that non-IGD variables and services do not require the X_ prefix.

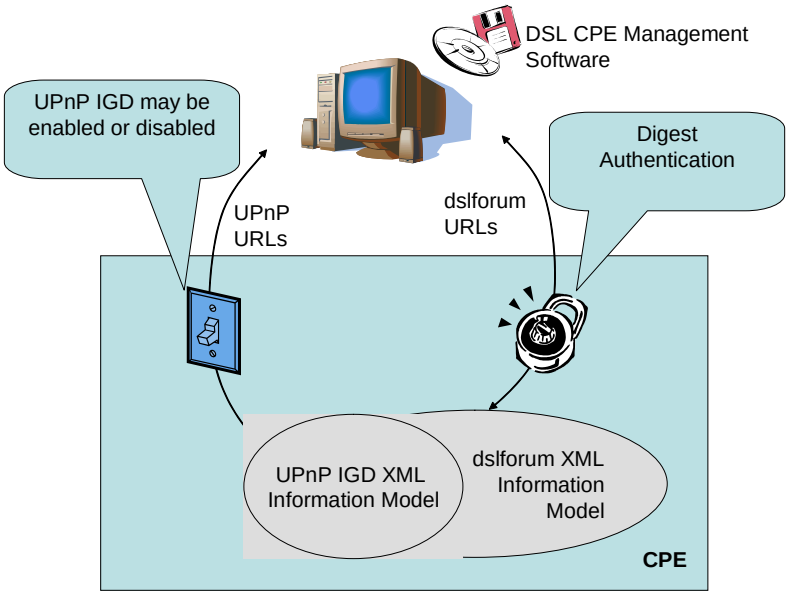


Figure 3: Co-existence with UPnP IGD

4.6 Other Means of Access Restriction

A CPE may provide additional means by which LAN-side CPE configuration may be restricted. For example, the CPE may allow a user, service provider, or other entity to indicate specific state variables that

cannot be modified via the LAN-side CPE configuration interface. For example, the CPE WAN Management Protocol allows such control of LAN-side access via the WAN management interface.

If an attempt is made to modify a state variable (generally via Set, Add, or Delete actions) for which LAN-side access is not granted, the CPE MUST respond with the "WriteAccessDisabled" (898) error code.

5 Protocols

The XML-based DSL CPE Management Application and CPE software protocol stack is shown below in Figure 4. Note that the DSL CPE Management Application is OS agnostic and runs on top of the native OS and TCP/UDP/IP protocol stacks. The XML/SOAP/HTTP and SSDP/HTTPU protocol stacks are part of the CPE and Management Application software, so no native OS support is implied or required.

All protocols indicated on the CPE side are required. First, TCP and UDP over IP over layer 2 (Ethernet for example) is used to communicate to the CPE. TCP/UDP over IP protocols are already necessary for data traffic as well as local management of the CPE. The SSL/TLS protocol may be optionally used as mentioned in Section 4.4 to provide data privacy. The HTTP protocol is also required to support an HTML based CPE management GUI. So, every protocol mentioned up to this point is "standard equipment" for a managed bridge or router CPE. In order to manage the CPE using the protocols and methods outlined in this document, the following additional protocols are used. Management and monitoring of CPE parameters and status is conveyed using XML. XML is a flexible text based format to describe the services supported on a CPE and the parameters and values which can be read or written. However, XML is almost free-form and therefore a standard schema must be defined for its use. For this purpose, the Simple Object Action Protocol (SOAP) [3] protocol is used so that messages can be constructed to define actions to perform along with associated parameters and responses which contain status and return parameters. So, SOAP is like a message based remote procedure call.

The other protocols shown include SSDP, HTTP over UDP, HTTP Multicast over UDP, and SSL. SSDP is used for device discovery if the CPE IP address is unknown or misconfigured. SSDP provides the means to discover and identify the CPE and services. The CPE Management Application uses multicast HTTP over UDP to send a SSDP discovery request message to the LAN to discover the CPE. The CPE then uses a HTTP over UDP message to send the SSDP discovery request response back to the Management Application.

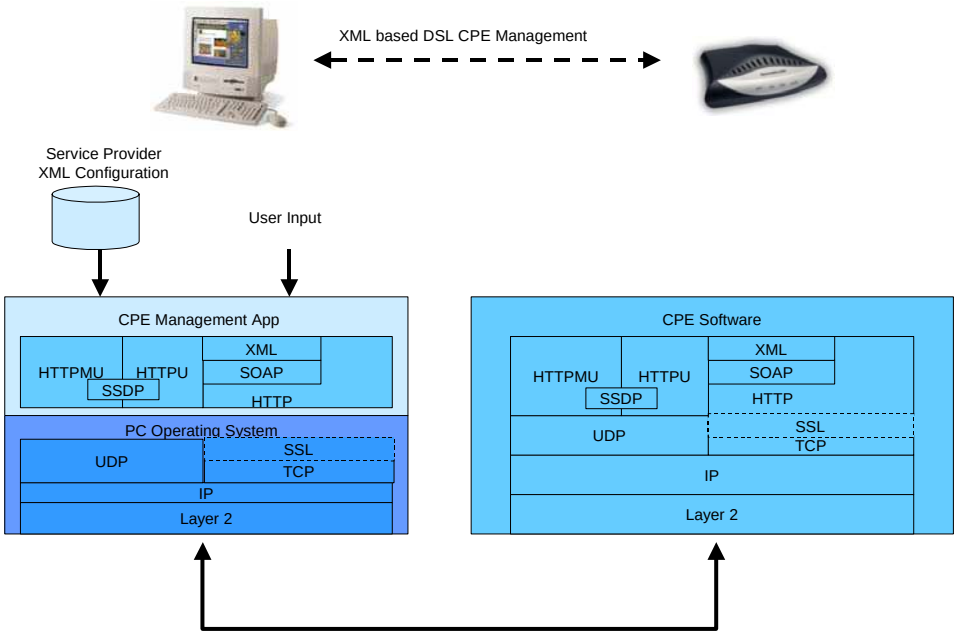


Figure 4: XML-based DSL CPE Management Protocol Stack

6 XML Parameters

The UPnP IGD 1.0 specification[1] is used as the basis for the XML schema and parameter definition for LAN-side DSL CPE configuration.

6.1 The Model

The model shown in Figure 5 uses the UPnP IGD model as its base. UPnP devices and services are shown in black (solid line). New services that are defined in this document are shown in blue (dashed line).

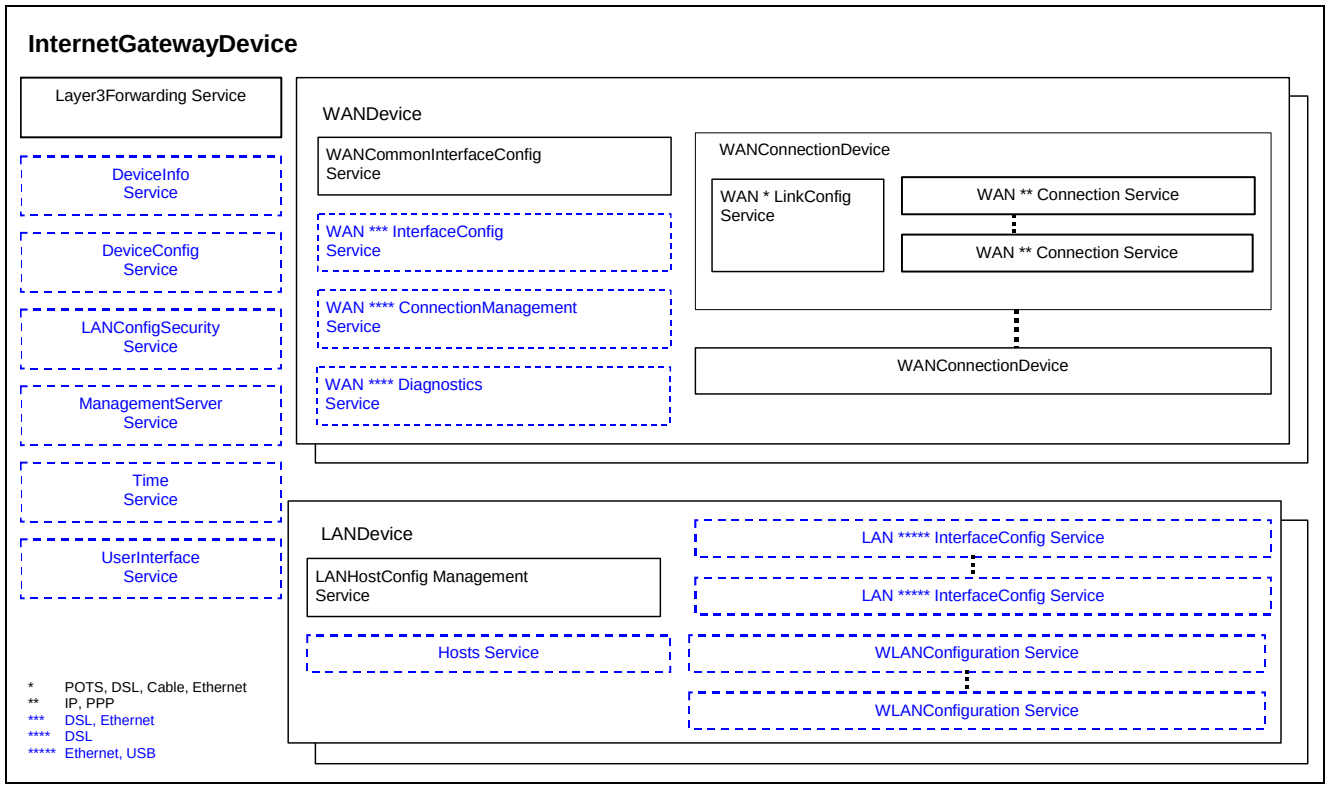


Figure 5 – Extended Internet Gateway Device overview

The following is a tree representation of the additional service templates and their place within the overall UPnP device hierarchy. Devices are indicated in **bold black**; UPnP services are indicated in *black italics*; additional services (Refer to UPnP IGD v1.0 specification for more details on those services) are indicated in blue underline.

InternetGatewayDevice

Layer3Forwarding

DeviceInfo

DeviceConfig

LANConfigSecurity

ManagementServer

Time

UserInterface

LANDevice

LANHostConfigMgmt

LANEthernetInterfaceConfig

WLANConfiguration

LANUSBInterfaceConfig

Hosts

WANDevice

WANCommonInterfaceConfig

WANDSLInterfaceConfig

WANEthernetInterfaceConfig

WANDSLConnectionManagement

WANDSLDiagnostics

WANConnectionDevice

WANPOTSLinkConfig

WANDSLLinkConfig

WANCableLinkConfig

WANEthernetLinkConfig

WANIPConnection

WANPPPConnection

The following table lists all specified devices and services and the implementation requirements.

Device or Service Name	Requirement Status
InternetGatewayDevice	Required for all devices.
Layer3Forwarding	Required for all devices.
DeviceInfo	Required for all devices.
DeviceConfig	Required for all devices.
LANConfigSecurity	Required for all devices.
ManagementServer	Required for CPE that support management via the CPE WAN Management Protocol.
Time	Optional.
UserInterface	Optional.
LANDevice	Required for all devices.

LANHostConfigMgmt	Required for all devices.
LANEthernetInterfaceConfig	Required for devices that implement a LAN-side Ethernet interface. A managed switch is modelled with 1 LANDevice containing 1 LANEthernetInterfaceConfig for each port of the switch.
WLANConfiguration	Required for devices that implement a LAN-side 802.11abg Wireless Access Point.
LANUSBInterfaceConfig	Required for devices that implement a LAN-side remote NDIS USB Interface.
Hosts	Deferred.
WANDevice	Required for all devices.
WANCommonInterfaceConfig	Required for all devices.
WANDSLInterfaceConfig	Required for devices that implement DSL modem WAN interface.
WANEthernetInterfaceConfig	Required for devices that implement Ethernet WAN interface.
WANDSLConnectionManagement	Required for devices that implement DSL modem WAN interface.
WANDSLDiagnostics	Required for devices with ADSL2 WAN interface, otherwise Optional.
WANConnectionDevice	Required for all devices.
WANPOTSLinkConfig	Required for devices with internal POTS modem or support for external POTS modem through USB or serial interface.
WANDSLLinkConfig	Required for devices the implement DSL modem WAN interface.
WANCableLinkConfig	Required for devices the implement cable modem WAN interface.
WANEthernetLinkConfig	Required for devices the implement WAN-side Ethernet interface.
WANIPConnection	Required for all currently configured IP or bridged connections.
WANPPPConnection	Required for all currently configured PPP connections.

The following abbreviations are used in the tables below:

- R: Required
- O: Optional
- D: Deferred (not required by this version of the specification; may be required in future version)
- S: Secured (only applicable to actions), requires authentication; combined with R/O indication
- N/A: Not Applicable

Note: the R/O column of the State Variable and Action tables for each service are conditional on whether the service is implemented at all. For an optional service, for example, an “R” variable or action is required only if the service itself is implemented.

Where an action includes optional state variables, the following rules MUST be followed in the case that one or more optional state variable is not implemented:

- Where a state variable is an IN argument and is not implemented by the CPE, the argument MUST be present in the action call and its value MAY be ignored by the CPE.

- Where a state variable is an OUT argument and is not implemented by the CPE, the argument MUST be present in the action response and MUST be set to one of the following type-specific default values:

Type	Default Value
Integer (of any size)	0
Unsigned integer (of any size)	0
String	<empty string>
Boolean	0
DateTime	00010101T00:00:00

Note that a control point SHOULD use the service description document to determine if an optional value is implemented by the device or not so that it can correctly determine how to interpret default values.

Vendors may extend the model by supporting additional allowed values for state variables, or by adding new state variables, actions, embedded devices, or services. Vendors can also add new XML elements or attributes to device and service descriptions. The rules for vendor additions are described in the UPnP device architecture documentation. Generally, extension names have an "X_" prefix; TR-64 extensions MUST adopt the stronger convention that the prefix be of the form "X_VENDOR_," where VENDOR is an upper-case string that uniquely identifies the vendor (e.g. domain name owned by the vendor). If vendors extend allowed value lists, the Maximum length of these values MUST be 31 characters as specified in the UPnP device architecture documentation.

6.2 Variable Formats

Unless otherwise noted, variable format and syntax is as indicated in the UPnP device architecture documentation. There are two additional formatting requirements not covered in UPnP; they are as follows:

- IP Address: an IPv4 non-padded address in dotted-decimal format.

- IP Subnet Mask: an IPv4 subnet mask (same format as IP Address).

- MAC Address: 12 hexadecimal digits (digits 0..9 letters A..F, a-f) displayed as six pairs of digits separated by colons. Leading zeros in pairs of digits might be suppressed.

Note that since all IP address and subnet formats are IPv4, there is no IPv6 support in this specification at this time.

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 size or is not an enumeration, the default maximum is 16 characters. Action arguments containing strings longer than the specified maximum MAY result in an "Invalid Args" (402) error response.

The UPnP Device Architecture query for the value or a state variable (using the QueryStateVariable action) MUST NOT be supported. An attempt to query a state variable MUST result in an “Invalid Action” (401) error.

6.3 Transactional Approach

Single actions (whether gets or sets) MAY occur independent of any explicit locking mechanism. There is an implied lock active while the device is executing the action; i.e. a CPE MUST process requests coming from different interfaces serially. If necessary, the device MAY reboot or perform other activities in order to effect any changes required by the action. After reboot, the device MUST have committed any state changes to non-volatile memory, and MUST be capable of responding to discovery requests. Any device configuration changes MUST be reflected in the device description document at this time.

Complex action sequences (e.g read-modify-write sequences) SHOULD make use of a locking mechanism. Before the first action in a logical series, the Control Point SHOULD issue the ConfigurationStarted action, which initiates a lock. After the sequence of actions is complete, the Control Point MUST issue the ConfigurationFinished action, which frees the lock. The CPE MAY apply changes associated with each action in a sequence as they are called or it MAY wait for the ConfigurationFinished action to apply them. If necessary, the CPE MAY reboot or perform other activities to commit any changes into non-volatile memory upon completion of the ConfigurationFinished action. After reboot the device MUST have committed any state changes to non-volatile memory and MUST be capable of responding to discovery requests, and any device configuration changes must be reflected in the device description document. The CPE MUST respond with the correct Status state variable (see the state variable table below) when the ConfigurationFinished method is called.

The CPE MUST NOT process set actions from other Control Points or interfaces at this time and SHOULD NOT process read actions; it SHOULD queue requests for processing after the ConfigurationFinished has been called. If the queued requests exceed the device capacity, or the device reboots so that it loses this information, it MUST respond with the correct standard “Out of Memory” (603) UPnP error code, or other appropriate interface-specific error messages.

Since the Control Point might never issue a ConfigurationFinished command because of network or other errors, the CPE MUST implement a device-specified timeout in order to release the device lock. The timeout SHOULD NOT be less than thirty (30) nor greater than forty-five (45) seconds from the time the CPE receives the last action from the control point that issued the ConfigurationStarted. The device MUST commit any changes received to date at this time and MAY reboot at this time if necessary to apply the changes. It MUST, however, be capable of responding to new discovery requests and accurately reflecting its current configuration in the device description document. No support for any rollback mechanism is required.

The CPE MUST NOT voluntarily reboot during a transaction sequence started by a ConfigurationStarted but before a ConfigurationFinished is called, but if this does occur, the Control Point loses its lock on the device, and the CPE MAY respond to actions from other interfaces.

The CPE MUST be capable of determining that the Control Point that issued the ConfigurationStarted is the same Control Point issuing the other actions during the lock. The control point MUST generate a session ID generated by the Control Point to identify itself uniquely. This session ID is only required when the control point is using the explicit locking mechanism. This session ID MUST be unique per control point and MAY be unique per session. This session ID MUST be of the argument type UUID.

The Control Point MUST generate a unique session ID, which it passes as an argument to the CPE in the ConfigurationStarted action. All subsequent actions within the scope of that transaction series MUST contain this session ID in the SOAP header. Any actions without this session ID or with a different session ID must be queued as described above.

Should the device experience a reset or the timeout occur during the transaction sequence started by a ConfigurationStarted but before a ConfigurationFinished is called, the control point loses its lock, and its session ID is no longer valid. The CPE MUST respond to the Control Point with a "SessionIDExpired" (899) error code should it receive any further actions from the control point with this session ID.

6.4 Tables

A uniform structure is utilized in defining new Tables for access via the LAN CPE configuration Management. Each table is REQUIRED to identify the KEY, which is a variable or set of variables that uniquely identify the row in the table. The variable or set of variables constituting the KEY will be used to identify the target row in all Delete, Set, and GetSpecific operations.

In the event that a table may allow duplicate entries, or cannot otherwise identify a variable or set of variables that can uniquely identify a row in the table, the table MUST include an explicit Key variable to serve in the role of the KEY. It is RECOMMENDED that such a Key variable be defined as a positive integer.

The standard operations to manipulate data in a Table are as follows:

- *Add – The add command will input all table variables and create a new row in the table to hold them. The add operation will return a failure result if the specified KEY already exists (701 ValueAlreadySpecified). Note that the port mapping table has its own specific failures for the add operation (718 ConflictInMappingEntry).*

  Note: In the case that an explicit Key variable was used in a table due to the inability of other variables to serve as a unique KEY, the Add operation SHOULD be defined to return the Key variable for the created row rather than having its value passed as an input. This avoids the need for the management application to be responsible for uniqueness of the Key. The CPE SHOULD NOT assign a value to the Key variable that has been used for a previously deleted table row. The CPE SHOULD exhaust the full space of integer values for a given table before re-using Keys.

- *Delete – Requires the variable(s) that constitute the KEY for that row as input argument(s) to indicate which row to delete. Returns an error if the specified KEY is not present in the table (702 ValueSpecifiedIsInvalid).*

- *Get Specific – Retrieves all information in the specific row using variable(s) that constitute the KEY as the only input argument(s), returning the remaining non-KEY variables as output arguments. Returns an error if the specified KEY is not present in the table (714 NoSuchEntryInArray).*

- *Get Generic – Allows the caller to view each table entry one at a time by utilizing an 'Index' argument (non-negative Integer). Successive calls to this action are only meaningful within a locked management session (see section 6.3). If used outside of a locked session, the results cannot be guaranteed accurate. Returns an error if the Index is greater than or equal to the number of entries in the table (713 SpecifiedArrayIndexInvalid).*

- *Set – The variable(s) that constitute the KEY is used to identify a particular row in the table. The remaining input arguments correspond to the variables in the row that are being modified by the Set operation. Returns an error if the specified KEY is not present in the table (702 ValueSpecifiedIsInvalid).*

6.5 InternetGatewayDevice

6.5.1 DeviceInfo

6.5.1.1 Overview

This service allows the reporting of basic device information, including the current hardware and software versions. This service is required.

6.5.1.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:DeviceInfo:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
ManufacturerName	-	String	String, nonzero length. Max length = 64 characters	The manufacturer of the CPE (human readable string).	N/A	R
ManufacturerOUI	-	String	String, 6 characters.	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 http://standards.ieee.org/faqs/OUI.html.	N/A	R
ModelName	-	String	String, nonzero length. Max length = 64 characters	Model name of the CPE (human readable string).	N/A	R
Description	-	String	String, nonzero length. Max length = 256 characters	A full description of the CPE device (human readable string).	N/A	R
ProductClass	-	String	String. Max length = 64 characters	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.	N/A	O
SerialNumber	-	String	String, nonzero length. Max length = 64 characters	Serial number of the CPE.	N/A	R
SoftwareVersion	-	String	String, nonzero length. Max length = 64 characters	Represents the current firmware version. It should be in the syntax used by the individual vendor to identify software versions.	N/A	R
AdditionalSoftwareVersions	-	String	A comma separated list of any additional versions. Max length = 64 characters	Represents any additional software version information the vendor may wish to supply.	N/A	O
ModemFirmwareVersion	-	String	String. Max length = 64 characters	Represents 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.	N/A	O
EnabledOptions	-	String	A comma separate list of option names. Max length = 1024 characters	Lists the Name of each option that is currently enabled in the CPE.	N/A	O

HardwareVersion	-	String	String, nonzero length. Max length = 64 characters	Represents the current hardware revision of the IGD.	N/A	R
AdditionalHardwareVersions	-	String	A comma separated list of any additional versions. Max length = 64 characters	Represents any additional hardware version information the vendor may wish to supply.	N/A	O
SpecVersion	-	String	Not specified Max length = 16 characters	Represents the version of the specification implemented by the device. Currently 1.0 is the only available version.	N/A	R
ProvisioningCode	-	String	String, nonzero length. Max length = 64 characters	Identifier of the primary service provider, which MAY be used 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”.	N/A	O
UpTime	-	ui4	Non-negative integer.	Time in seconds since the CPE was last restarted.	N/A	O
FirstUseDate	-	DateTime	-	Date and time that the CPE first successfully established a network connection.	N/A	O
DeviceLog	-	String	Max length = 32 Kbytes	Vendor specific content containing the contents of all logs.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
GetInfo	NewManufacturerName NewManufacturerOUI NewModelName NewDescription NewProductClass NewSerialNumber NewSoftwareVersion NewAdditionalSoftwareVersions NewModemFirmwareVersion NewEnabledOptions NewHardwareVersion NewAdditionalHardwareVersions NewSpecVersion NewProvisioningCode NewUpTime NewFirstUseDate	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	ManufacturerName ManufacturerOUI ModelName Description ProductClass SerialNumber SoftwareVersion AdditionalSoftwareVersions ModemFirmwareVersion EnabledOptions HardwareVersion AdditionalHardwareVersions SpecVersion ProvisioningCode UpTime FirstUseDate	Retrieves the value of the state variables in this service.	402, 501	R
SetProvisioningCode	NewProvisioningCode	IN	ProvisioningCode	Sets the value of the ProvisioningCode.	402, 501	O S
GetDeviceLog	NewDeviceLog	OUT	DeviceLog	Retrieves the current CPE log file.	402, 501	R S

6.5.2 DeviceConfig

6.5.2.1 Overview

This service allows the handling of the configuration of the device. This service is required.

6.5.2.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:DeviceConfig:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
PersistentData	-	String	String, nonzero length. Max length = 256 bytes	Arbitrary user data that must be persisted across IGD reboots.	N/A	R
ConfigFile	-	String	String, nonzero length. Max length = 32 Kbytes	A dump of the currently running configuration on the IGD. This is a vendor proprietary configuration file.	N/A	O
A_ARG_TYPE_Status	-	String	RebootRequired ChangesApplied	Indicate whether a CPE has already applied changes or requires a reboot before committing changes to CPE state.	N/A	R
A_ARG_TYPE_UUID	-	uuid	N/A	Universally Unique ID. For more information on generating UUIDs, please refer to the IETF draft: http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-00.txt Used as Session ID to uniquely identify the control point issuing commands to a CPE.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
GetPersistentData	NewPersistentData	OUT	PersistentData	Retrieves the value of the PersistentData state variable.	402, 501	R
SetPersistentData	NewPersistentData	IN	PersistentData	Sets the value of the PersistentData state variable.	402, 501	R S
GetConfiguration	NewConfigFile	OUT	ConfigFile	Retrieves the value of the ConfigFile state variable.	402, 501	O
SetConfiguration	NewConfigFile	IN	ConfigFile	Sets the value of the ConfigFile state variable.	402, 501	O S
ConfigurationStarted	NewSessionID	IN	A_ARG_TYPE_UUID	Indicates to the CPE that a transaction the CPE is beginning and that it should block access to CPE configuration from other interfaces.	402,501	R S
ConfigurationFinished	NewStatus	OUT	A_ARG_TYPE_Status	Indicates to the CPE that a transaction is completed and that it may release the device lock.	402, 501	R S

FactoryReset	This action does not have any arguments.	-	This action does not have any related state variables.	This action restores the factory default settings on the IGD.	402, 501	R S
Reboot	This action does not have any arguments.	-	This action does not have any related state variables.	Shuts the IGD down and then restarts it with the latest state variable values in place.	402, 501	R S

6.5.2.3 Theory of Operation

The ConfigFile variable enables the ability to backup and restore the last known good state of the IGD. It returns a vendor-specific document, which defines the state of the IGD at the moment the GetConfiguration action is invoked and which can be used to restore the state should the device get into an unknown or non-operational state. The document obtained by the GetConfiguration must be capable of restoring the IGD's state when passed in with the SetConfiguration action. This capability would be used by an ongoing desktop support tool and is not intended to replace initial configuration of the device by a control point.

6.5.3 Layer3Forwarding

6.5.3.1 Overview

This service allows the handling of the routing and forwarding configuration of the device. This service is required.

6.5.3.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:Layer3Forwarding:1

StateVariables

Variable Name	From IGD[1]	Table	Data Type	Allowed Value	Description	Default Value	R/O
DefaultConnectionService	✓	-	String	String, Max length = 256 characters	Specifies a connection service instance in a WANConnectionDevice. See UPnP Layer3Forwarding:1 v1.01 for more details.	N/A	R
ForwardNumberOfEntries	-	Forwarding (index)	ui2	>=0	Number of forwarding entries.	N/A	R
Enable	-	Forwarding	Boolean	1, 0	Enables or disables the table entry.	N/A	R
Status	-	Forwarding	String	Disabled Enabled Error	Indicates the status of the table entry.	N/A	R
Type	-	Forwarding	String	Default Network Host	Describes the type of route.	N/A	R
DestIPAddress	-	Forwarding KEY	String	IP Address	Destination network address.	N/A	R

[1] Indicates whether or not the variable was defined in UPnP IGD 1.0. If not, the variable is specific to this specification.

DestSubnetMask	-	Forwarding KEY	String	IP Subnet Mask	Destination network subnet mask.	N/A	R
SourceIPAddress	-	Forwarding KEY	String	IP Address	Source network address.	N/A	R
SourceSubnetMask	-	Forwarding KEY	String	IP Subnet Mask	Source network subnet mask.	N/A	R
GatewayIPAddress	-	Forwarding	String	IP Address	IP address of the gateway.	N/A	R
Interface	-	Forwarding	String	String, nonzero length. Max length = 256 characters	Outgoing interface. 2-tuple referring to WAN**Connection service instance or LANHostConfigManagement service.	N/A	R
ForwardingMetric	-	Forwarding	i4	>=-1	Forwarding metric.	-1	R
MTU	-	Forwarding	ui2	Between 1 and 1540, inclusive	The maximum allowed size of an Ethernet frame for this route.	N/A	O

Actions, Arguments & Errors

Name	From IGD[2]	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetDefaultConnection Service	✓	NewDefaultConnectionService	IN	DefaultConnectionService	Sets the value of the DefaultConnectionService.	402, 501, 720, 721, 723	R S
GetDefaultConnection Service	✓	NewDefaultConnectionService	OUT	DefaultConnectionService	Retrieves the value of the DefaultConnectionService.	402, 501	R
Forwarding Table Actions							
GetForwardNumberOf Entries	-	NewForwardNumberOfEntries	OUT	ForwardNumberOfEntries	Retrieves the value of the ForwardNumberOfEntries state variable.	402, 501	R
AddForwardingEntry	-	NewType NewDestIPAddress NewDestSubnetMask NewSourceIPAddress| NewSourceSubnetMask NewGatewayIPAddress NewInterface NewForwardingMetric NewMTU	IN IN IN IN IN IN IN IN IN	Type DestIPAddress DestSubnetMask SourceIPAddress| SourceSubnetMask GatewayIPAddress Interface ForwardingMetric MTU	Inserts an entry in the forwarding table.	402, 501, 701	R S
DeleteForwardingEntry	-	NewDestIPAddress NewDestSubnetMask NewSourceIPAddress NewSourceSubnetMask	IN IN IN IN	DestIPAddress DestSubnetMask SourceIPAddress SourceSubnetMask	Deletes an entry in the forwarding table.	402, 501, 702	R S
GetSpecific ForwardingEntry	-	NewDestIPAddress NewDestSubnetMask NewSourceIPAddress NewSourceSubnetMask NewGatewayIPAddress NewEnable NewStatus NewType NewInterface NewForwardingMetric NewMTU	IN IN IN IN OUT OUT OUT OUT OUT OUT OUT	DestIPAddress DestSubnetMask SourceIPAddress SourceSubnetMask GatewayIPAddress Enable Status Type Interface ForwardingMetric MTU	Retrieve an entry in the forwarding table.	402, 501, 714	R

[2] Indicates whether or not the action was defined in UPnP IGD 1.0. If not, the action is specific to this specification.

GetGeneric ForwardingEntry	-	NewForwardingIndex NewEnable NewStatus NewType NewDestIPAddress NewDestSubnetMask NewSourceIPAddress NewSourceSubnetMask NewGatewayIPAddress NewInterface NewForwardingMetric NewMTU	IN OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	ForwardNumberOfEntries Enable Status Type DestIPAddress DestSubnetMask SourceIPAddress SourceSubnetMask GatewayIPAddress Interface ForwardingMetric MTU	Retrieves Forwarding table one entry at a time. Control points can call this action with an incrementing array index until no more entries are found on the gateway. Index is ranging from 0 to ForwardingNumberOfEntries-1.	402, 501, 713	R
SetForwardingEntryEnable	-	NewDestIPAddress NewDestSubnetMask NewSourceIPAddress NewSourceSubnetMask NewEnable	IN IN IN IN IN	DestIPAddress DestSubnetMask SourceIPAddress SourceSubnetMask Enable	Sets the value of the Enable state variable to enable or disable a particular forwarding entry.	402, 501, 702	R S

6.5.3.3 Theory of Operation

This service models the IP forwarding table of the IGD. Support of the parameters of a "standard" forwarding table entry is required. There are two supported types of forwarding table entries:

- *Destination IP Address Route*
  In this case an entry includes: <DestIPAddress, DestSubnetMask, GatewayIPAddress, Interface, ForwardingMetric>.

  SourceIPAddress and SourceSubnetMask are "0.0.0.0"

- *Source IP Address Route*
  In this case an entry includes: <DestIPAddress, DestSubnetMask, SourceIPAddress, SourceSubnetMask, GatewayIPAddress, Interface, ForwardingMetric>.

  When making the forwarding decision, first the source IP address is matched (longest prefix first) followed by destination IP address (longest prefix first) for entries with matching source IP address.

Within the ordered ForwardingTable source IP address entries (longest prefix first) are placed before destination IP address (longest prefix first) entries.

Setting the DefaultConnectionService state variable automatically creates / updates the default interface route to point to the DefaultConnectionService interface.

6.5.4 LANConfigSecurity Service

6.5.4.1 Overview

This service enables the ability to set the password that will allow the device to be configured via the xml interface.

The LANConfigSecurity service is REQUIRED for all CPE adhering to this specification. There is one instance of this service for a given CPE device.

6.5.4.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:LANConfigSecurity:1

State Variables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
ConfigPassword	-	String	Nonzero length; Max length = 64	A password to allow access to any protected auto-configuration service.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetConfigPassword	NewPassword	IN	ConfigPassword	Replaces the existing ConfigPassword with the value specified in NewPassword.	402, 501	R S

6.5.5 ManagementServer

6.5.5.1 Overview

This service allows configuration of the communication between a CPE and a management server using the CPE WAN Management Protocol (CWMP) as currently defined in TR-069. This management server is also known as an Auto-Configuration Server (ACS).

6.5.5.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:ManagementServer:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
URL	-	String	Max length = 256 characters	URL for the CPE to connect to the ACS (auto-configuration server) using the CPE WAN Management Protocol. In one of the following forms: • x-dslf_cwmp://host:port/ • x-dslf_cwmps://host:port/ The “s” suffix of the scheme component (x-cwmps) indicates the use of SSL. The “host” portion of the URL MAY be either a domain name or IP address.	N/A	R
Password	-	String	Max length = 256 characters	Password used to authenticate the CPE when making a connection to the ACS using the CPE WAN Management Protocol. This password is used only if the CPE does not use client-side certificate-based authentication.	N/A	R

PeriodicInformEnable	-	Boolean	1, 0	Whether or not the CPE must periodically send CPE information to ACS using the Inform method call.	N/A	R
PeriodicInformInterval	-	ui4	>=0	The duration of the interval in seconds for which the CPE MUST attempt to connect with the ACS and call the Inform method if PeriodicInformEnable is true.	N/A	R
PeriodicInformTime	-	DateTime	-	An absolute time reference to determine when the CPE should initiate the Inform method calls. Each Inform call must occur at this reference time plus or minus an integer multiple of the PeriodicInformInterval.	N/A	R
ParameterKey	-	String	Max length = 32 characters	The value of the ParameterKey argument from the most recent SetParameterValues method call from the ACS. If there have been no SetParameterValues calls, this value is empty.	N/A	R
ParameterHash	-	String	24 characters	Base64 encoded MD5 hash of all parameter values in the entire parameter database (except this one). Change in any parameter value should result in a change to this value.	N/A	R
ConnectionRequestURL	-	String	Max length = 256 characters	HTTP URL for an ACS to make a Connection Request notification to the CPE. The "host" portion of the URL MAY be the IP address for the management interface of the CPE in lieu of a domain name.	N/A	R
ConnectionRequestUsername	-	String	Max length = 256 characters	Username used to authenticate an ACS making a Connection Request to the CPE.	N/A	R
ConnectionRequestPassword	-	String	Max length = 256 characters	Password used to authenticate an ACS making a Connection Request to the CPE.	N/A	R
UpgradesManaged	-	Boolean	1, 0	Indicates whether or not the ACS will manage upgrades for the CPE. If true (1), the CPE SHOULD not use other means other than the ACS to seek out available upgrades. If false (0), the CPE MAY use other means for this purpose.	N/A	R
KickURL	-	String	Max length = 256 characters	LAN-accessible URL from which the CPE can be "kicked" to initiate the Kicked RPC method call. MUST be an absolute URL including a domain or IP address as would be used on the LAN side of the CPE. Present only for a CPE that supports the Kicked RPC method as defined in the CPE WAN Management Protocol.	N/A	O
DownloadProgressURL	-	String	Max length = 256 characters	LAN-accessible URL to which a web-server associated with the ACS may redirect a user's browser on initiation of a file download to observe the status of the download. Present only for a CPE that provides a LAN-side web page to show progress during a file download.	N/A	O

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
GetInfo	NewURL NewPeriodicInformEnable NewPeriodicInformInterval NewPeriodicInformTime NewParameterKey	OUT OUT OUT OUT OUT	URL PeriodicInformEnable PeriodicInformInterval PeriodicInformTime ParameterKey	Retrieves the value of the required state variables in this service (except for passwords, which are not readable).	402, 501	R

	NewParameterHash NewConnectionRequestURL NewConnectionRequestUsername NewUpgradesManaged	OUT OUT OUT OUT	ParameterHash ConnectionRequestURL ConnectionRequestUsername UpgradesManaged	readable).		
SetManagementServerURL	NewURL	IN	URL	Sets the value of the URL state variable indicating the URL of the management server.	402, 501	R S
SetManagementServerPassword	NewPassword	IN	Password	Sets the value of the Password state variable.	402, 501	R S
SetPeriodicInform	NewPeriodicInformEnable NewPeriodicInformInterval NewPeriodicInformTime	IN IN IN	PeriodicInformEnable PeriodicInformInterval PeriodicInformTime	Sets the value of the periodic-Inform-related state variables.	402, 501	R S
SetConnectionRequestAuthentication	NewConnectionRequestUsername NewConnectionRequestPassword	IN IN	ConnectionRequestUsername ConnectionRequestPassword	Sets the value of the username/password used to authenticate an ACS connection request.	402, 501	R S
SetUpgradeManagement	NewUpgradesManaged	IN	UpgradesManaged	Sets the value of the UpgradesManaged state variable.	402, 501	R S
GetKickURL	NewKickURL	OUT	KickURL	Retrieves the value of the KickURL state variable.	402, 501	O
GetDownloadProgressURL	NewDownloadProgressURL	OUT	DownloadProgressURL	Retrieves the value of the DownloadProgressURL state variable.	402, 501	O

6.5.6 Time

6.5.6.1 Overview

This service allows configuration of a NTP / SNTP time client in the CPE.

6.5.6.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:Time:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
NTPServer1	-	String	Max length = 64 characters	First NTP/SNTP timeserver. Either a domain name or IP address.	N/A	R
NTPServer2	-	String	Max length = 64 characters	Second NTP/SNTP timeserver. Either a domain name or IP address.	N/A	R
NTPServer3	-	String	Max length = 64 characters	Third NTP/SNTP timeserver. Either a domain name or IP address.	N/A	O
NTPServer4	-	String	Max length = 64 characters	Fourth NTP/SNTP timeserver. Either a domain name or IP address.	N/A	O
NTPServer5	-	String	Max length = 64 characters	Fifth NTP/SNTP timeserver. Either a domain name or IP address.	N/A	O
CurrentLocalTime	-	DateTime	-	The current date and time in the CPE's local time zone.	N/A	R

LocalTimeZone	-	String	6 characters	The local time offset from UTC in one of the following forms: • +hh:mm • -hh:mm	N/A	R
LocalTimeZoneName	-	String	Max length = 64 characters	Name of the local time zone (human readable string).	N/A	R
DaylightSavingsUsed	-	Boolean	1, 0	Whether or not daylight savings time is in use in the CPE's local time zone.	N/A	R
DaylightSavingsStart	-	DateTime	-	Date and time daylight savings time begins each year if used (year is ignored). If daylight savings time is not used, this value is ignored.	N/A	R
DaylightSavingsEnd	-	DateTime	-	Date and time daylight savings time ends each year if used (year is ignored). If daylight savings time is not used, this value is ignored.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
GetInfo	NewNTPServer1	OUT	NTPServer1	Retrieves the value of the state variables in this service.	402, 501	R
	NewNTPServer2	OUT	NTPServer2			
	NewNTPServer3	OUT	NTPServer3			
	NewNTPServer4	OUT	NTPServer4			
	NewNTPServer5	OUT	NTPServer5			
	NewCurrentLocalTime	OUT	CurrentLocalTime			
	NewLocalTimeZone	OUT	LocalTimeZone			
	NewLocalTimeZoneName	OUT	LocalTimeZoneName			
	NewDaylightSavingsUsed	OUT	DaylightSavingsUsed			
	NewDaylightSavingsStart	OUT	DaylightSavingsStart			
	NewDaylightSavingsEnd	OUT	DaylightSavingsEnd			
SetNTPServers	NewNTPServer1	IN	NTPServer1	Sets the value of all NTP/SNTP servers used by the CPE.	402, 501	R S
	NewNTPServer2	IN	NTPServer2			
	NewNTPServer3	IN	NTPServer3			
	NewNTPServer4	IN	NTPServer4			
	NewNTPServer5	IN	NTPServer5			
SetLocalTimeZone	NewLocalTimeZone	IN	LocalTimeZone	Sets the state variables that specify the time-zone and associated daylight-savings time behavior.	402, 501	R S
	NewLocalTimeZoneName	IN	LocalTimeZoneName			
	NewDaylightSavingsUsed	IN	DaylightSavingsUsed			
	NewDaylightSavingsStart	IN	DaylightSavingsStart			
	NewDaylightSavingsEnd	IN	DaylightSavingsEnd			

6.5.7 UserInterface

6.5.7.1 Overview

This service allows configuration of the behavior of the user interface of the CPE.

6.5.7.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:UserInterface:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O

PasswordRequired	-	Boolean	1, 0	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. Present only if the CPE provides a password-protected LAN-side user interface.	N/A	O
PasswordUserSelectable	-	Boolean	1, 0	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. Present only if the CPE provides a password-protected LAN-side user interface and supports LAN-side Auto-Configuration.	N/A	O
UpgradeAvailable	-	Boolean	1, 0	Indicates that a CPE upgrade is available, allowing the CPE to display this information to the user.	N/A	O
WarrantyDate	-	DateTime	-	Indicates the date and time that the warranty associated with the CPE is to expire.	N/A	O
ISPName	-	String	Max length = 64 characters	The name of the customer's ISP.	"your ISP"	O
ISPHelpDesk	-	String	Max length = 32 characters	The help desk phone number of the ISP.	empty string	O
ISPHomePage	-	String	Max length = 256 characters	The URL of the ISP's home page.	empty string	O
ISPHelpPage	-	String	Max length = 256 characters	The URL of the ISP's on-line support page.	empty string	O
ISPLogo	-	Base64 String	Max length = 5460 characters	Base64 encoded ASCII string of a GIF or JPEG image. The binary image is constrained to 4095 bytes or less.	empty string	O
ISPLogoSize	-	ui2	< 4096	Unencoded binary image size in bytes. If ISPLogoSize input value is 0 then the logo is cleared. ISPLogoSize can also be used as a check to verify correct transfer and conversion of Base64 string to image size.	N/A	O
ISPMailServer	-	String	Max length = 256 characters	The URL of the ISP's mail server.	empty string	O
ISPNewsServer	-	String	Max length = 256 characters	The URL of the ISP's news server.	empty string	O
TextColor	-	String	Size = 6 characters	The color of text on the GUI screens in RGB hexadecimal notation (e.g., FF0088).	empty string	O
BackgroundColor	-	String	Size = 6 characters	The color of the GUI screen backgrounds in RGB hexadecimal notation (e.g., FF0088).	empty string	O
ButtonColor	-	String	Size = 6 characters	The color of buttons on the GUI screens in RGB hexadecimal notation (e.g., FF0088).	empty string	O
ButtonTextColor	-	String	Size = 6 characters	The color of text on buttons on the GUI screens in RGB hexadecimal notation (e.g., FF0088).	empty string	O
AutoUpdateServer	-	String	Max length = 256 characters	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	empty string	O

				the UpgradesManaged variable in the ManagementServer service is true (1).		
UserUpdateServer	-	String	Max length = 256 characters	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 UpgradesManaged variable in the ManagementServer service is true (1).	vendor-defined	O
ExampleLogin	-	String	Max length = 40 characters	An example of a correct login, according to ISP-specific rules.	empty string	O
ExamplePassword	-	String	Max length = 30 characters	An example of a correct password, according to ISP-specific rules.	empty string	O

Actions, Arguments & Errors

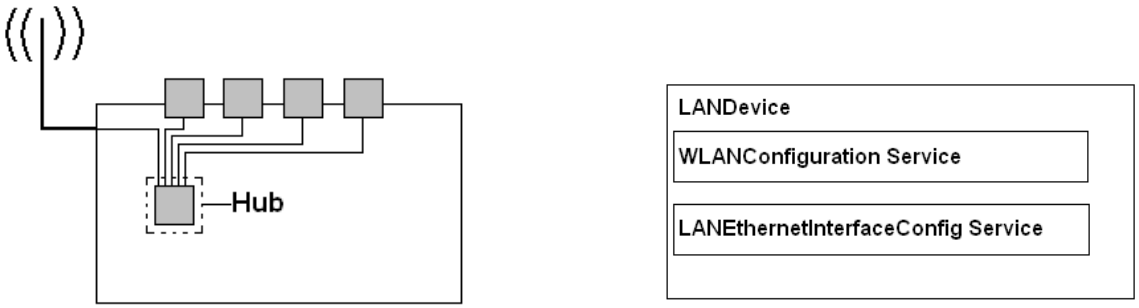
Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
GetInfo	NewPasswordRequired NewPasswordUserSelectable NewUpgradeAvailable NewWarrantyDate	OUT OUT OUT OUT	PasswordRequired PasswordUserSelectable UpgradeAvailable WarrantyDate	Retrieves the value of the state variables in this service.	402, 501	R
SetUserInterfacePasswordModes	NewPasswordRequired NewPasswordUserSelectable	IN IN	PasswordRequired PasswordUserSelectable	Sets the value of the password-related state variables.	402, 501	O S
SetUpgradeAvailable	NewUpgradeAvailable	IN	UpgradeAvailable	Sets the value of the UpgradeAvailable state variable.	402, 501	O S
GetInterfaceVariables	NewISPName NewISPHelpDesk NewISPHomePage NewISPHelpPage NewISPMailServer NewISPNewsServer NewTextColor NewBackgroundColor NewButtonColor NewButtonTextColor NewAutoUpdateServer NewUserUpdateServer NewExampleLogin NewExamplePassword	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	ISPName ISPHelpDesk ISPHomePage ISPHelpPage ISPMailServer ISPNewsServer TextColor BackgroundColor ButtonColor ButtonTextColor AutoUpdateServer UserUpdateServer ExampleLogin ExamplePassword	Retrieves the value of all these GUI variables.	402, 501	O
SetInterfaceVariables	NewISPName NewISPHelpDesk NewISPHomePage NewISPHelpPage NewISPMailServer NewISPNewsServer NewTextColor NewBackgroundColor NewButtonColor NewButtonTextColor NewAutoUpdateServer NewUserUpdateServer NewExampleLogin NewExamplePassword	IN IN IN IN IN IN IN IN IN IN IN IN IN IN	ISPName ISPHelpDesk ISPHomePage ISPHelpPage ISPMailServer ISPNewsServer TextColor BackgroundColor ButtonColor ButtonTextColor AutoUpdateServer UserUpdateServer ExampleLogin ExamplePassword	Sets the value of all these GUI variables.	402, 501	O S
SetISPLogo	NewISPLogoSize NewISPLogo	IN IN	ISPLogoSize ISPLogo	Set the values of the Logo state variables.	402, 501	O S
GetISPLogo	NewISPLogoSize NewISPLogo	OUT OUT	ISPLogoSize ISPLogo	Retrieves the values of the Logo state variables.	402, 501	O

6.6 LANDevice

Theory of Operation:

A LANDevice corresponds to a physically attached network to the DSL CPE; each LANDevice has at most one MAC address. **ConfigurationServices within a LANDevice correspond to interfaces to that single physical network. In order to illustrate this concept, four different implementation scenarios are presented below.

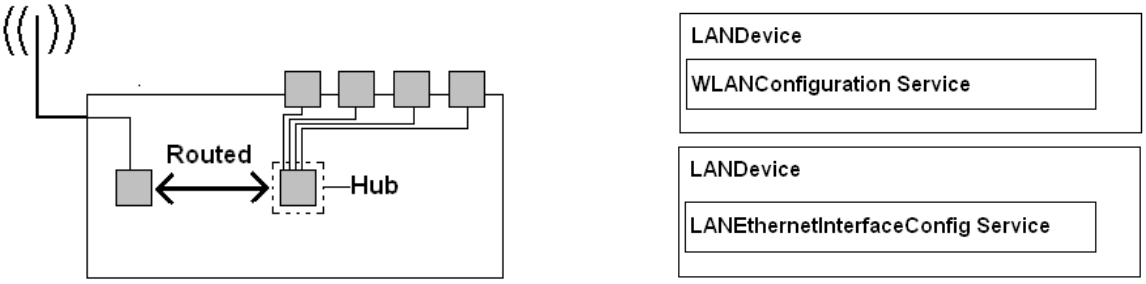
Figure 6: Internally bridged device



In this example, 4 Ethernet ports and 1 wireless interface are bridged together on one physical network. All attached clients would receive IP addresses in the same subnet, and no routing is required in order for these ports to pass traffic to one another. This device would have 1 LANDevice that contains 1 WLANConfiguration service and 1 LANEthernetInterfaceConfig service. The IP interfaces table in LANHostConfigManagmenet would have exactly 1 entry to indicate the IP address of the LAN interface of the gateway.

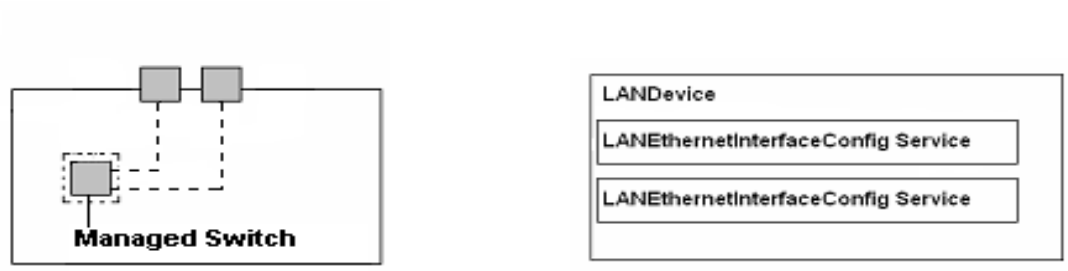
Note that there may be more than one LANEthernetInterfaceConfig service within a given LANDevice if required. For example, 2 LANInterfaces with 2 different bit rates would require 2 services.

Figure 7: Internally Routed Device



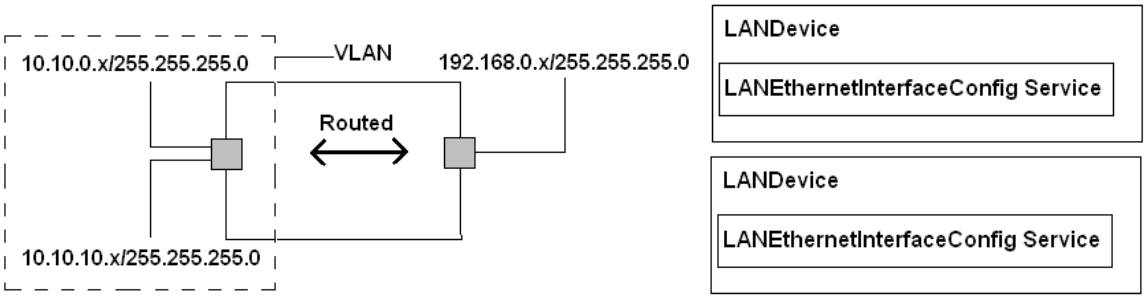
This device supports one physically attached network through the Ethernet hub and another physically attached network through the wireless interface. Traffic from one network must be routed through the gateway to communicate with devices on the other. Modeling this CPE requires 2 LANDevices: one with an LANEthernetInterfaceConfig service and one with a WLANConfiguration service. The LANHostConfigManagement service on each LANDevice has exactly 1 entry in the IPInterfaces table to indicate the IP address of the LAN interface of the gateway.

Figure 8: Managed Switch



This device has 2 Ethernet ports that comprise a managed switch. In this configuration, each port is modelled by a separate LANEthernetInterfaceConfig service within a single LAN device. This configuration allows for management of each individual Ethernet port.

Figure 9: Virtual LAN (VLAN)



This device has 2 Ethernet ports supporting 2 separate attached physical networks with 2 separate IP pools. As above, these would be modelled with 2 LANDevices, each with a LANEthernetInterfaceConfig service.

In addition, the network on the left is setup as a VLAN, combining two different address ranges together into a single physical LAN. In the LANHostConfigManagement service of that LANDevice, the IP interfaces table contains two entries corresponding to the 2 IP addresses of the VLAN.

6.6.1 LANHostConfigManagement

6.6.1.1 Overview

This service enables reporting of LAN-related device information and setting and configuring LAN IP addressing.

6.6.1.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:LANHostConfigManagement:1

StateVariables

Variable Name	From IGD	Table	Data Type	Allowed Value	Description	Default Value	R/O
DHCPServerConfigurable	✓	-	Boolean	1, 0	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.	N/A	R
DHCPRelay	✓	-	Boolean	1, 0	Indicates if the DHCP server performs the role of a server or a relay on the LAN interface.	N/A	R
SubnetMask	✓	-	String	IP Subnet Mask	Specifies the client's network subnet mask.	N/A	R
DNSServers	✓	-	String	String	Comma separated list of DNS servers offered to DHCP clients. Support for more than three DNS Servers is Optional.	N/A	R
DomainName	✓	-	String	String, Max length = 64 characters	Sets the domain name to provide to clients on the LAN interface.	N/A	R
MinAddress	✓	-	String	IP Address	Specifies first address in the pool to be assigned by the DHCP server on the LAN interface.	N/A	R
MaxAddress	✓	-	String	IP Address	Specifies last address in the pool to be assigned by the DHCP server on the LAN interface.	N/A	R
IPRouters	✓	-	String	IP Address(es), comma separated list	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.	N/A	R
ReservedAddresses	✓	-	String	IP Address(es), comma separated list	Comma separated list of addresses marked reserved from the address allocation pool.	N/A	R
DHCPLeaseTime	-	-	i4	>=-1 -1 for infinite lease	Specifies the lease time of client assigned addresses.	7200 seconds	O
DHCPServerEnable	-	-	Boolean	0, 1	State of the DHCP server on the LAN interface.	N/A	R
UseAllocatedWAN	-	-	String	Normal, UseAllocatedSubnet, Passthrough	If Normal, then DHCP addresses are to be allocated out of a private address pool. If UseAllocatedSubnet, instructs the CPE to allocate DHCP addresses from the WAN subnet block for the WAN connection identified in AssociatedConnection. If Passthrough, then the specified LAN Host identified by the Passthrough MAC address is given the WAN IP address.	N/A	O

AssociatedConnection	-	-	String	Max length = 256 characters	Specifies the connection service instance for the connection to be used for address allocation if UseAllocatedWAN is set to UseAllocatedSubnet or Passthrough.	N/A	O
PassthroughLease	-	-	ui4	0-65535	DHCP lease time in seconds given to the specified LAN Host when the WAN IP address is passed-through. Note: A temporary private IP address with short lease (for example, 1 min) may be given to the passthrough LAN Host before the WAN IP address is acquired.	N/A	O
PassthroughMACAddress	-	-	String	MAC Address	Represents the hardware address of the LAN Host that is used to passthrough the WAN IP address if UseAllocatedWAN is "Passthrough.".	N/A	O
AllowedMACAddresses	-	-	String	MAC Address	Represents a comma-separated list of hardware addresses that are allowed to connect to this connection if MACAddress-ControlEnabled is 1 for a given interface.	N/A	O
Enable	-	IPInterface	Boolean		Enables or disables this entry.	N/A	R
IPInterfaceIPAddress	-	IPInterface KEY	String	IP Address	IP address of the LAN-side interface of the IGD.	N/A	R
IPInterfaceSubnetMask	-	IPInterface KEY	String	IP Address	Subnet mask of the LAN-side interface of the IGD.	255.255.255.0	R
IPInterfaceAddressingType	-	IPInterface	String	DHCP, Static, AutoIP	Represents the addressing method used to assign the LAN IP address of the CPE on this interface.	Static	R
IPInterfaceNumberOfEntries	-	IPInterface (index)	ui2	>=0	Number of IP interface at LAN side of the IGD. 1 is a typical value for IGDs not supporting Multihomed interfaces. Support for more than one interface instance is Optional.	N/A	R

Actions, Arguments & Errors

Name	From IGD	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
GetInfo	-	NewDHCPServerConfigurable	OUT	DHCPServerConfigurable	Retrieves all of the state variables not associated with a table.	402, 501	R
		NewDHCPRelay	OUT	DHCPRelay			
		NewSubnetMask	OUT	SubnetMask			
		NewDNSServers	OUT	DNSServers			
		NewDomainName	OUT	DomainName			
		NewMinAddress	OUT	MinAddress			
		NewMaxAddress	OUT	MaxAddress			
		NewIPRouters	OUT	IPRouters			
		NewReservedAddresses	OUT	ReservedAddresses			
		NewDHCPLeaseTime	OUT	DHCPLeaseTime			
		NewDHCPServerEnable	OUT	DHCPServerEnable			
		NewUseAllocatedWAN	OUT	UseAllocatedWAN			
		NewAssociatedConnection	OUT	AssociatedConnection			
		NewPassthroughLease	OUT	PassthroughLease			
		NewPassthroughMACAddress	OUT	PassthroughMACAddress			
		NewAllowedMACAddresses	OUT	AllowedMACAddresses			

SetDHCPServerConfigurable	✓	NewDHCPServerConfigurable	IN	DHCPServerConfigurable	Sets DHCP server on the device as configurable or not.	402, 501	R S
GetDHCPServerConfigurable	✓	NewDHCPServerConfigurable	OUT	DHCPServerConfigurable	Retrieves the value of the DHCPServerConfigurable state variable.	402, 501	R
GetDHCPRelay	✓	NewDHCPRelay	OUT	DHCPRelay	Retrieves the value of the DHCPRelay state variable.	402, 501	R
SetSubnetMask	✓	NewSubnetMask	IN	SubnetMask	Sets the subnet mask of the subnet managed by this DHCP server in the context of the LANDevice.	402, 501, 701	R S
GetSubnetMask	✓	NewSubnetMask	OUT	SubnetMask	Retrieves the value of the SubnetMask state variable.	402, 501	R
SetIPRouter	✓	NewIPRouters	IN	IPRouters	Set the list of routers in IPRouters. Support for more than one Router address is Optional. The CPE MAY ignore any addresses after the first.	402, 501, 701	R S
DeleteIPRouter	✓	NewIPRouters	IN	IPRouters	Removes a router from the IPRouters list.	402, 501, 702	O S
GetIPRoutersList	✓	NewIPRouters	OUT	IPRouters	Retrieves the value of the IPRouters state variable.	402, 501	R
SetDomainName	✓	NewDomainName	IN	DomainName	Changes the domain for hosts served by the DHCP server on the LAN interface.	402, 501	R S
GetDomainName	✓	NewDomainName	OUT	DomainName	Retrieves the value of the IPRouters state variable.	402, 501	R
SetAddressRange	✓	NewMinAddress NewMaxAddress	IN IN	MinAddress MaxAddress	Sets the range of addresses assigned by the DHCP server.	402, 501	R S
GetAddressRange	✓	NewMinAddress NewMaxAddress	OUT OUT	MinAddress MaxAddress	Retrieves the values of the MinAddress and MaxAddress state variables.	402, 501	R
SetReservedAddress	✓	NewReservedAddresses	IN	ReservedAddresses	Sets the list of reserved addresses in ReservedAddresses.	402, 501, 701	R S
DeleteReservedAddress	✓	NewReservedAddresses	IN	ReservedAddresses	Removes an IP address from the ReservedAddresses list.	402, 501, 702	R S
GetReservedAddresses	✓	NewReservedAddresses	OUT	ReservedAddresses	Retrieves the value of the ReservedAddresses state variable.	402, 501	R
SetDNSServer	✓	NewDNSServers	IN	DNSServers	Sets the list of DNS Servers in DNSServers. Support for more than three DNS Servers is Optional. The CPE MAY ignore any addresses after the third.	402, 501, 701	R S
DeleteDNSServer	✓	NewDNSServers	IN	DNSServers	Removes an IP address from the DNSServers list.	402, 501, 702	R S
GetDNSServers	✓	NewDNSServers	OUT	DNSServers	Retrieves the value of the DNSServers state variable.	402, 501	R
SetDHCPLeaseTime	-	NewDHCPLeaseTime	IN	DHCPLeaseTime	Sets the DHCP server lease time.	402, 501	O S
SetDHCPServerEnable	-	NewDHCPServerEnable	IN	DHCPServerEnable	Enables or Disables the DHCP server on LAN side of the IGD.	402, 501	R S

SetAddressAllocation	-	NewUseAllocatedWAN NewAssociatedConnection NewPassthroughLease NewPassthroughMACAddress	IN IN IN IN	UseAllocatedWAN WANAssociatedConnection PassthroughLease PassthroughMACAddress	Sets the value of the Use-AllocatedWAN, AssociatedConnection, PassthroughLease, and PassthroughMACAddress state variables.	402, 501	O S
SetAllowedMACAddresses	-	NewAllowedMACAddresses	IN	AllowedMACAddresses	Sets the list of allowed MAC addresses specified by AllowedMACAddresses. An empty string argument will clear the list of allowed MAC addresses. To modify or append to an existing list, GetAllowedMACAddresses should be invoked first to obtain the current list, changed locally and updated using this set action.	402, 501	O S
DeleteAllowedMacAddress	-	NewAllowedMACAddress	IN	AllowedMACAddresses	Removes an allowed MAC address from the list specified by AllowedMACAddresses. This action accepts one address at a time, NOT a comma-separated list of addresses. This action must be invoked once for every address that needs to be deleted from the list.	402, 501	O S
IPInterface Table Actions							
GetIPInterfacetNumberOfEntries	-	NewIPInterfaceNumberOfEntries	OUT	IPInterfaceNumberOfEntries	Retrieves the value of the IPInterfaceNumberOfEntries state variable.	402, 501	R
SetIPInterface	-	NewEnable NewIPAddress NewSubnetMask NewIPAddressingType	IN IN IN IN	Enable IPInterfaceIPAddress IPInterfaceSubnetMask IPInterfaceAddressingType	Sets the state variable values of the IP interface.	402, 501, 702	R S
AddIPInterface	-	NewEnable NewIPAddress NewSubnetMask NewIPAddressingType	IN IN IN IN	Enable IPInterfaceIPAddress IPInterfaceSubnetMask IPInterfaceAddressingType	Add a new LAN side IP interface.	402, 501, 701	O S
DeleteIPInterface	-	NewIPAddress NewISubnetMask	IN IN	IPInterfaceIPAddress IPInterfaceSubnetMask	Removes an IP interface from the IP interfaces table.	402, 501, 702	O S
GetIPInterface SpecificEntry	-	NewIPAddress NewSubnetMask NewEnable NewIPAddressingType	IN IN OUT OUT	IPInterfaceIPAddress IPInterfaceSubnetMask Enable IPInterfaceAddressingType	Retrieves Specific LAN side IP Interface table entry.	402, 501, 714	R
GetIPInterface GenericEntry	-	NewIPInterfaceIndex NewEnable NewIPAddress NewSubnetMask NewIPAddressingType NewIPInterfaceNumberOfEntries	IN OUT OUT OUT OUT OUT	IPInterfaceNumberOfEntries Enable IPInterfaceIPAddress IPInterfaceSubnetMask IPInterfaceAddressingType IPInterfaceNumberOfEntries	Retrieves LAN side IP Interface table entry at a time. Control points can call this action with an incrementing array index until no more entries are found on the gateway. Index is ranging from 0 to IPInterfaceNumberOfEntries-1.	402, 501, 713	R

6.6.2 LANEthernetInterfaceConfig

6.6.2.1 Overview

This service models an Ethernet LAN connection on an IGD device. This service allows for the determination and modification of basic configuration information. This service must be implemented for IGDs that contain an Ethernet interface on the LAN side.

6.6.2.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:LANEthernetInterfaceConfig:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
Enable	-	Boolean	1, 0	Enables or disables this interface.	N/A	R
Status	-	String	Up NoLink Error Disabled	Indicates the status of this interface.	N/A	R
MACAddress	-	String	MAC Address	Represents the physical address of the interface.	N/A	R
MACAddressControlEnabled	-	Boolean	1,0	Enables MAC address control on this interface.	0	O
MaxBitRate	-	String	10 100 1000 Auto	Represents the maximum upstream and downstream bit rate available to this connection.	N/A (Mbps)	R
DuplexMode	-	String	Half Full Auto	Represents the duplex mode available to this connection.	N/A	R
Stats.BytesSent	-	ui4	>=0	Total number of bytes sent over the interface since the CPE was last reset.	N/A	R
Stats.BytesReceived	-	ui4	>=0	Total number of bytes received over the interface since the CPE was last reset.	N/A	R
Stats.PacketsSent	-	ui4	>=0	Total number of packets sent over the interface since the CPE was last reset.	N/A	R
Stats.PacketsReceived	-	ui4	>=0	Total number of packets received over the interface since the CPE was last reset.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetEnable	NewEnable	IN	Enable	Sets the value of the Enable state variable to enable or disable this interface.	402, 501	R S
SetMaxBitRate	NewMaxBitRate	IN	MaxBitRate	Sets the value of the related state variable.	402, 501, 702	O S
SetDuplexMode	NewDuplexMode	IN	DuplexMode	Sets the value of the related state variable.	402, 501, 702	O S
GetInfo	NewEnable NewStatus NewMACAddress NewMACAddressControlEnabled NewMaxBitRate NewDuplexMode	OUT OUT OUTOUT OUT OUT	Enable Status MACAddress MACAddressControlEnabled MaxBitRate DuplexMode	Retrieves all the state variables other than the statistics.	402, 501	R
GetStatistics	NewBytesSent NewBytesReceived NewPacketsSent NewPacketsReceived	OUT OUT OUT OUT	Stats.BytesSent Stats.BytesReceived Stats.PacketsSent Stats.PacketsReceived	Retrieves all the statistics for this interface.	402, 501	R
SetMACAddressControlEnabled	NewMACAddressControlEnabled	IN	MACAddressControlEnabled	Enables/disables MACAddress control for this interface.	402, 501	O S

6.6.3 LANUSBInterfaceConfig

6.6.3.1 Overview

This service models a USB connection on an IGD device. This service allows for the determination and modification of basic configuration information. This service must be implemented for IGDs that contain an USB interface on the LAN side.

6.6.3.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:LANUSBInterfaceConfig:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
Enable	-	Boolean	1, 0	Enables or disables this interface.	N/A	R
Status	-	String	Up NoLink Error Disabled	Status of the USB interface.	N/A	R
Type	-	String	Host Hub Device	Type of the USB interface.	N/A	R
Rate	-	String	Low Full High	One of Low (speed), Full (speed) or High (speed) (USB 2.0).	N/A	R
Power	-	String	Unknown Self Bus	One of Unkown, Bus (powered) or Self (powered).	N/A	R

Standard	-	String	String Max length= 6 characters	USB version supported by the device.	N/A	R
MACAddress	-	String	MAC Address	Represents the physical address of the interface.	N/A	R
MACAddressControlEnabled	-	Boolean	1,0	Enables MAC address control on this interface.	0	O
Stats.BytesSent	-	ui4	>=0	Total number of payload bytes sent over this interface since the CPE was last reset.	N/A	R
Stats.BytesReceived	-	ui4	>=0	Total number of payload bytes received over this interfacesince the CPE was last reset.	N/A	R
Stats.CellsSent	-	ui4	>=0	Total number of cells sent over this interface since the CPE was last reset.	N/A	R
Stats.CellsReceived	-	ui4	>=0	Total number of cells received over this interface since the CPE was last reset.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetEnable	NewEnable	IN	Enable	Sets the value of the Enable state variable to enable or disable this interface.	402, 501	R S
SetMACAddressControlEnabled	NewMACAddressControlEnabled	IN	MACAddressControlEnabled	Enables/disables MACAddress control for this interface.	402, 501	O S
GetInfo	NewEnable NewStatus NewMACAddress NewMACAddressControlEnabled NewType NewRate NewPower NewStandard	OUT OUT OUT OUT OUT OUT OUT OUT	Enable Status MACAddress MACAddressControlEnabled Type Rate Power Standard	Retrieves the values of related state variables other than the statistics.	402, 501	R
GetStatistics	NewBytesSent NewBytesReceived NewCellsSent NewCellsReceived	OUT OUT OUT OUT	Stats.BytesSent Stats.BytesReceived Stats.CellsSent Stats.CellsReceived	Retrieves the statistics for this interface.	402, 501	R

6.6.4 WLANConfiguration

6.6.4.1 Overview

This service models an 802.11x LAN connection[4] on a LAN device. This service allows for the determination and modification of basic configuration information. This service must be implemented for IGDs that contain an IEEE 802.11 interface on the LAN side.

6.6.4.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WLANConfiguration:1

StateVariables

Variable Name	From IGD	Table	Data Type	Allowed Value	Description	Default Value	R/O
Enable	-	-	Boolean	1, 0	Enables or disables this interface.	N/A	R
Status	-	-	String	Up Error Disabled	Status of the 802.11 interface.	N/A	R
MaxBitRate	-	-	String	Auto, or the largest of the OperationalData-TransmitRates values Max length = 4 characters	Represents the maximum upstream and downstream bit rate available to this connection.	(Mbps)	R
Channel	✓	-	ui1	Depends on PossibleChannels (which depends on Standard)	Describes the current radio channel used by the connection. Note: There is currently no way of requesting automatic selection of a channel.	N/A	R
SSID	✓	-	String	String, Max length = 32 characters	Describes 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.	N/A	R
BeaconType	✓	-	String	R: None, Basic, WPA O: 11i, BasicandWPA, Basicand11i, WPAand11i, BasicandWPAand11i	Indicates beacon types to be used with this connection.	N/A	R
WEPKeyIndex	✓	-	ui1	0-3	Index of the default WEP key.	N/A	R
WEPKey	✓	WEPKey	String	Hex String of appropriate length, Max length = 128 characters	A WEP key. The WEP encryption level is inferred from the key length, e.g. 10 characters for 40-bit encryption, or 26 characters for 104-bit encryption. The WEPKey table has a fixed size of 4 entries. Because of this, there is no need for a variable to indicate its current size.	N/A	R
WEPEncryptionLevel	✓	-	String	Disabled, 40-bit, 104-bit Max length = 31 characters	The allowed values let the control point determine all the supported key lengths (any additional vendor-supported values MUST start with the key length in bits).	N/A	R
PreSharedKeyIndex	✓	PreSharedKeyKEY	ui1	0-9	A PSK index. Note: An index of 0 indicates the default PSK, which is accessible via both the {Set,Get}SecurityKeys and the {Set,Get}PreSharedKey actions. Note: Support for the default PSK is REQUIRED. TR-064 implementations MAY in addition support 9 further PSKs. Therefore, the PreSharedKey table has either 1 or 10 entries.	N/A	O

AssociatedDeviceMAC-Address	✓	PreSharedKey	String	MAC Address	The MAC address associated with a preshared key, or an empty string if no MAC address is associated with the key. Note: The default PSK (index 0) is never associated with a specific device. Each of the 9 further optional PSKs MAY be associated with a specific device.	N/A	R
PreSharedKey	✓	PreSharedKey	String	64 byte Hex String, case-independent	A literal WPA PSK. Note: The default PSK has index 0. A further 9 PSKs MAY be supported.	N/A	R
KeyPassphrase	✓	PreSharedKey	String	String, Max length = 63 characters (WPA specifies 63 as the maximum length of a passphrase, to permit distinction between passphrases and PSKs, which are 64 characters long: 256 bits)	A passphrase from which WEP and PSK keys can be generated *by the control point* (the access point just stores the passphrase). For WEP keys it 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. For WPA PSK, the key is generated as specified by WPA, which uses PBKDF2 from PKCS #5: Password-based Cryptography Specification Version 2.0 (RFC2898). Note: The above algorithm for generating WEP keys from the passphrase implies that, if a passphrase is used, all four WEP keys will be the same. Note: The passphrase for the default PSK (index 0) is used to generate both the default PSK and the four WEP keys. There is a separate passphrase for each of the 9 further optional PSKs.	N/A	R
MACAddressControlEnabled	-	-	Boolean	0, 1	Indicates whether MAC Address Control is enabled or not on the connection. MAC Address Control limits the clients that connect to those which match a list of allowed MAC addresses specified in AllowedMACAddresses in LAN-HostConfigManagement.	N/A	R
Standard	-	-	String	a, b, g	Indicates which IEEE 802.11 mode the device is currently operating in.	N/A	R
BSSID	✓	-	String	MAC Address	Represents the physical address of the interface.	N/A	R
TotalBytesSent	✓	-	ui4	>=0	Total number of payload bytes sent over this interface since the CPE was last reset.	N/A	O
TotalBytesReceived	✓	-	ui4	>=0	Total number of payload bytes received over this interfacesince the CPE was last reset.	N/A	O
TotalPacketsSent	✓	-	ui4	>=0	Total number of packets sent over this interface since the CPE was last reset.	N/A	R
TotalPacketsReceived	✓	-	ui4	>=0	Total number of packets received over this interface since the CPE was last reset.	N/A	R

BasicEncryptionModes	✓	-	String	None, WEPEncryption Max length = 31 characters	Encryption modes that are available when basic 802.11 is enabled. “WEPEncryption” implies that all wireless clients can use WEP for data encryption.	N/A	R
BasicAuthenticationMode	✓	-	String	R: None O: EAPAuthenticat-ion Max length = 31 characters	Authentication modes that are available when basic 802.11 is enabled.	N/A	R
WPAEncryptionModes	✓	-	String	R: WEPEncryption, TKIPEncryption, WEPandTKIPEncryp-tion O: AESEncryption, WEPandAESEncryp-tion, TKIPandAES-Encryption, WEPand-TKIPandAESEncryp-tion Max length = 31 characters	Encryption modes that are available when WPA is enabled. The allowed values listed in the XML service description define the combinations of modes that are permitted (the value is not a comma-separated list).	N/A	R
WPAAuthenticationMode	✓	-	String	R: PSKAuthentica-tion O: EAPAuthentica-tion Max length = 31 characters	Authentication modes that are available when WPA is enabled.	N/A	R
The following variables are needed only for advanced use.							
PossibleChannels	✓	-	String	Comma-separated list; ranges “n-m” permitted Depends on Standard Max length = 1024 characters	The possible radio channels for the wireless standard (a, b or g) and the regulatory domain. For example, for 802.11b and North America, would be “1-11.”	N/A	R
BasicDataTransmit-Rates	✓	-	String	Comma-separated list Depends on Poss-ibleDataTransmit-Rates Max length = 256 characters	The maximum access point data transmit rates for unicast, multicast and broadcast frames. For example, BasicDataTransmit-Rates might be “1,2”, indicating that unicast, multicast and broadcast frames can be transmitted at 1 Mbps and 2 Mbps.	N/A (Mbps)	R
OperationalDataTransmit Rates	✓	-	String	Comma-separated list Depends on Poss-ibleDataTransmit-Rates Max length = 256 characters	The maximum access point data transmit rates for unicast frames (a superset of BasicDataTransmitRates). Given the value of BasicDataTransmitRates from the example above, OperationalData-TransmitRates might be “1,2,5.5,11”, indicating that unicast frames can in addition be transmitted at 5.5 Mbps and 11 Mbps.	N/A (Mbps)	R
PossibleDataTransmit-Rates	✓	-	String	Comma-separated list Max length = 256 characters	The data transmit rates for unicast frames at which the access point will permit a station to connect (a subset of OperationalDataTransmitRates).	N/A (Mbps)	R

					OperationalDataTransmitRates). Given the values of BasicDataTransmitRates and OperationalDataTransmit Rates 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.		
IEEE11iEncryptionModes	✓	-	String	R: WEPEncryption, TKIPEncryption, WEPandTKIPEncryption O: AESEncryption, WEPandAESEncryption, TKIPandAESEncryption, WEPandTKIPandAESEncryption Max length = 31 characters	Encryption modes that are available when 11i is enabled. The allowed values listed in the XML service description define the combinations of modes that are permitted (the value is not a comma-separated list). Note: The UPnP standard indicates that this is required, but all the actions that reference it are optional, so this is probably a mistake.	N/A	O
IEEE11iAuthenticationMode	✓	-	String	R: PSKAuthentication O: EAPAuthentication, EAPandPSKAuthentication Max length = 31 characters	Authentication modes that are available when 11i is enabled. Note: The UPnP standard indicates that this is required, but all the actions that reference it are optional, so this is probably a mistake.	N/A	O
TotalAssociations	✓	Associated Device (index)	ui2	>=0	The number of devices currently associated with the access point.	0	R
AssociatedDeviceMACAddress	✓	AssociatedDeviceKEY	String	MAC Address	The MAC address of an associated device. Note: This variable occurs twice in this service because it is associated with both the PreSharedKey and AssociatedDevice tables.	N/A	R
AssociatedDeviceIPAddress	✓	AssociatedDevice	String	IP Address Max length = 64 characters	The IP address or DNS name of an associated device.	N/A	R
AssociatedDeviceAuthenticationState	✓	AssociatedDevice	Boolean	0,1	Whether an associated device has authenticated (1) or not (0).	N/A	R
LastRequestedUnicastCipher	✓	AssociatedDevice	String	Max length = 256 characters	The unicast cipher that was most recently used for a station with a specified MAC address (11i only).	N/A	O
LastRequestedMulticastCipher	✓	AssociatedDevice	String	Max length = 256 characters	The multicast cipher that was most recently used for a station with a specified MAC address (11i only).	N/A	O
LastPMKId	✓	AssociatedDevice	String	Max length = 256 characters	The pairwise master key (PMK) that was most recently used for a station with a specified MAC address (11i only).	N/A	O
RadioEnabled	✓	-	Boolean	0,1	Indicates whether or not the access point radio is enabled.	1	R

AutoRateFallBackEnabled	✓	-	Boolean	0,1	Indicates whether the access point can automatically reduce the data rate in the event of undue noise or contention.	1	R
The following variables are needed only to provide optional extra functionality.							
InsecureOOBAccess-Enabled	✓	-	Boolean	0,1	Indicates whether insecure access via non-TR-064 mechanisms is permitted to data that would be only accessible via secure actions when using TR-064.	0	O
BeaconAdvertisement-Enabled	✓	-	Boolean	0,1	Indicates whether or not the access point is sending out beacons.	1	O
LocationDescription	✓	-	String	Escaped XML String (can be empty), Max length = 4096 characters	Opaque (to the access point) string that identifies it by name and location. If used, contains an XML string containing information about the AP's name and physical location.	N/A	O
RegulatoryDomain	✓	-	String	802.11d Regulatory Domain String, Size = 3 characters	First two octets are ISO/IEC 3166-1 2 char country code; third octet is " " (all environments), "O" (outside) or "I" (inside).	N/A	O
TotalPSKFailures	✓	-	ui4	>=0	The number of times pre-shared key (PSK) authentication has failed (relevant only to WPA and 11i).	0	O
TotalIntegrityFailures	✓	-	uil4	>=0	The number of times the MICHAEL integrity check has failed (relevant only to WPA and 11i).	0	O
ChannelsInUse	✓	-	String	Comma-separated list; ranges "n-m" permitted Max length = 1024 characters	The channels that the access point determines to be currently in use (including any that it is using itself).	N/A	O
DeviceOperationMode	✓	-	String	R: InfrastructureAcc-essPoint O: WirelessBridge, WirelessRepeater, WirelessStation Max length = 31 characters	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.	"Infrastructure-AccessPoint"	O
DistanceFromRoot	✓	-	ui1	>=0	The number of hops from the root access point to the wireless repeater or bridge.	0	O
PeerBSSID	✓	-	String	MAC Address	The physical address of the peer in wireless repeater or bridge mode.	N/A	O
AuthenticationServiceMode	✓	-	String	R: None O: LinkAuthenticat-ion, RadiusClient Max length = 31 characters	Indicates whether another service is involved in client authentication (LinkAuthentication for a co-located authentication server; RadiusClient for an external RADIUS server). Note: The UPnP standard indicates that this is still required, but the request to make it optional was accepted, so this is probably an editing error.	N/A	O

Actions, Arguments & Errors

Name	From IGD	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetEnable	-	NewEnable	IN	Enable	Sets the value of the Enable state variable to enable or disable this interface.	402, 501	R S
GetInfo	-	NewEnable NewStatus NewMaxBitRate NewChannel NewSSID NewBeaconType NewMACAddressControlEnabled NewStandard NewBSSID NewBasicEncryptionModes NewBasicAuthenticationMode	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	Enable Status MaxBitRate Channel SSID BeaconType MACAddressControlEnabled Standard NewBSSID BasicEncryptionModes BasicAuthenticationMode	Retrieves all of the state variables other than the statistics and the security keys.	402, 501	R
SetConfig	-	NewMaxBitRate NewChannel NewSSID NewBeaconType NewMACAddressControlEnabled NewBasicEncryptionModes NewBasicAuthenticationMode	IN IN IN IN IN IN IN	MaxBitRate Channel SSID BeaconType MACAddressControlEnabled BasicEncryptionModes BasicAuthenticationMode	Sets the configuration values for this connection.	402, 501, 605	R S
WEPKey Table Actions							
SetSecurityKeys	✓	NewWEPKey0 NewWEPKey1 NewWEPKey2 NewWEPKey3 NewPreSharedKey NewKeyPassphrase	IN IN IN IN IN IN	WEPKey WEPKey WEPKey WEPKey PreSharedKey KeyPassphrase	Sets the access point's four WEP keys, its default pre-shared key (key 0), and the passphrase from which they were generated (the passphrase is stored, but not used, by the access point). Note: Access to this action SHOULD occur over an encrypted https link.	402, 731	R S
GetSecurityKeys	✓	NewWEPKey0 NewWEPKey1 NewWEPKey2 NewWEPKey3 NewPreSharedKey NewKeyPassphrase	OUT OUT OUT OUT OUT OUT	WEPKey WEPKey WEPKey WEPKey PreSharedKey KeyPassphrase	Gets the access point's four WEP keys, its default pre-shared key (key 0), and the passphrase from which they were generated. Note: Access to this action SHOULD occur over an encrypted https link. If https encryption is not used, empty strings MUST be returned for all arguments.	402	R S
SetDefaultWEPKeyIndex	✓	NewDefaultWEPKeyIndex	IN	WEPKeyIndex	Sets the index of the default WEP key.	402	R S
GetDefaultWEPKeyIndex	✓	NewDefaultWEPKeyIndex	OUT	WEPKeyIndex	Gets the index of the default WEP key.	402	R S
SetBasBeaconSecurity-Properties	✓	NewBasicEncryptionModes NewBasicAuthenticationMode	IN IN	BasicEncryptionModes BasicAuthenticationMode	Sets encryption and authentication modes for basic 802.11 beacons.	402, 732, 734	R S
GetBasBeaconSecurity-Properties	✓	NewBasicEncryptionModes NewBasicAuthenticationMode	OUT OUT	BasicEncryptionModes BasicAuthenticationMode	Gets encryption and authentication modes for basic 802.11 beacons.	402	R
SetWPABeaconSecurity-Properties	✓	NewWPAEncryptionModes NewWPAAuthenticationMode	IN IN	WPAEncryptionModes WPAAuthenticationMode	Sets encryption and authentication modes for WPA beacons.	402, 732, 733, 734	R S
GetWPABeaconSecurity-Properties	✓	NewWPAEncryptionModes NewWPAAuthenticationMode	OUT OUT	WPAEncryptionModes WPAAuthenticationMode	Gets encryption and authentication modes for WPA beacons.	402	R

GetStatistics	-	NewTotalBytesSent NewTotalBytesReceived NewTotalPacketsSent NewTotalPacketsReceived	OUT OUT OUT OUT	TotalBytesSent TotalBytesReceived TotalPacketsSent TotalPacketsReceived	Retrieves the statistics for this interface.	402, 501	R
The following actions are needed only for advanced use.							
SetSSID	✓	NewSSID	IN	SSID	Sets the SSID. If the control point is a wireless client, it will lose connectivity until it changes to use the new SSID.	402, 501, 605	R S
GetSSID	✓	NewSSID	OUT	SSID	Gets the current SSID.	402	R
GetBSSID	✓	NewBSSID	OUT	BSSID	Gets the BSSID (MAC address) of the access point.	402	R
SetBeaconType	✓	NewBeaconType	IN	BeaconType	Sets the beacon type. The access point will immediately send out beacons according to the new setting, provided BeaconAdvertisementEnabled is TRUE.	402, 501	R S
GetBeaconType	✓	NewBeaconType	OUT	BeaconType	Gets the current beacon type.	402	R
SetChannel	✓	NewChannel	IN	Channel	Sets the wireless channel, which must be one of PossibleChannels.	402, 501, 728	R S
GetChannelInfo	✓	NewChannel NewPossibleChannels	OUT OUT	Channel PossibleChannels	Gets the current wireless channel and possible channels.	402	R
SetDataTransmitRates	✓	NewBasicDataTransmitRates NewOperationalDataTransmitRates	IN IN	BasicDataTransmitRates OperationalDataTransmitRates	Sets the access point's basic and operational data rates, both of which must be subsets ot the PossibleDataTransmitRates list.	402, 730	R S
GetDataTransmitRateInfo	✓	NewBasicDataTransmitRates NewOperationalDataTransmitRates NewPossibleDataTransmitRates	OUT OUT OUT	BasicDataTransmitRates OperationalDataTransmitRates PossibleDataTransmitRates	Gets details of data rates currently in use and supported by the access point.	402	R
GetByteStatistics	✓	NewTotalBytesSent NewTotalBytesReceived	OUT OUT	TotalBytesSent TotalBytesReceived	Gets the number of bytes sent and received since the last reset.	402	O
GetPacketStatistics	✓	NewTotalPacketsSent NewTotalPacketsReceived	OUT OUT	TotalPacketsSent TotalPacketsReceived	Gets the number of packets sent and received since the last reset.	402	R
Set11iBeaconSecurityProperties	✓	NewIEEE11iEncryptionModes NewIEEE11iAuthenticationMode	IN IN	IEEE11iEncryptionModes IEEE11iAuthenticationMode	Sets encryption and authentication modes for 11i beacons.	402, 733, 734	O S
Get11iBeaconSecurityProperties	✓	NewIEEE11iEncryptionModes NewIEEE11iAuthenticationMode	OUT OUT	IEEE11iEncryptionModes IEEE11iAuthenticationMode	Gets encryption and authentication modes for 11i beacons.	402	O
AssociatedDevice Table Actions							
GetTotalAssociations	✓	NewTotalAssociations	OUT	TotalAssociations	Gets the number of associated devices (wireless clients) currently associated with the access point.	402	R S
GetGenericAssociatedDeviceInfo	✓	NewAssociatedDeviceIndex NewAssociatedDeviceMACAddress NewAssociatedDeviceIPAddress NewAssociatedDeviceAuthState	IN OUT OUT OUT	TotalAssociations AssociatedDeviceMACAddress AssociatedDeviceIPAddress AssociatedDeviceAuthState	Gets details of the wireless client with the specified (zero-based) index.	402, 713	R S

GetSpecificAssociated-DeviceInfo	✓	NewAssociatedDeviceMAC-Address NewAssociatedDeviceIPAddress NewAssociatedDeviceAuthState	IN OUT OUT	AssociatedDeviceMAC-Address AssociatedDeviceIPAddress AssociatedDeviceAuthState	Gets details of the wireless client with the specified MAC address.	402, 714	R S
GetSpecificAssociated-Dev11iInfo	✓	NewAssociatedDeviceMAC-Address NewLastRequestedUnicastCipher NewLastRequestedMulticast-Cipher NewIEEE11iAuthenticationMode NewLastPMKId	IN OUT OUT OUT OUT	AssociatedDeviceMAC-Address LastRequestedUnicastCipher LastRequestedMulticast-Cipher IEEE11iAuthenticationMode LastPMKId	Gets 11i-related information for the wireless client with the specified MAC address. Note: LastPMKId can be returned as "" if the AP does not have access to it or does not use a RADIUS server.	402, 714	O S
GetByteStatsFor-AssociatedDev	✓	NewAssociatedDeviceMAC-Address NewTotalBytesSent NewTotalBytesReceived	IN OUT OUT	AssociatedDeviceMAC-Address TotalBytesSent TotalBytesReceived	Gets the number of bytes sent and received by the specified station since the last reset.	402	O
GetPacketStatsFor-AssociatedDev	✓	NewAssociatedDeviceMAC-Address NewTotalPacketsSent NewTotalPacketsReceived	IN OUT OUT	AssociatedDeviceMAC-Address TotalPacketsSent TotalPacketsReceived	Gets the number of packets sent and received by the specified station since the last reset.	402	O
SetRadioMode	✓	NewRadioEnabled	IN	RadioEnabled	Enables / disables the access point's radio.	402, 501	R S
GetRadioMode	✓	NewRadioEnabled	OUT	RadioEnabled	Gets the current radio mode.	402	R
SetAutoRateFallBack-Mode	✓	NewAutoRateFallBackEnabled	IN	AutoRateFallBackEnabled	Enables / disables auto rate fallback.	402	R S
GetAutoRateFallBack-Mode	✓	NewAutoRateFallBackEnabled	OUT	AutoRateFallBackEnabled	Gets the current auto rate fallback mode.	402	R
The following actions are needed only to provide optional extra functionality.							
SetInsecureOutOfBand-AccessMode	✓	NewInsecureOOBAccessEnabled	IN	InsecureOOBAccess-Enabled	Enables / disables insecure access by non-TR-064 mechanisms to data that can only be accessed, when using TR-064, via secure actions.	402, 501	O S
GetInsecureOutOfBand-AccessMode	✓	NewInsecureOOBAccessEnabled	OUT	InsecureOOBAccess-Enabled	Gets the current insecure out-of-band access mode.	402	O S
SetBeaconAdvertisement	✓	NewBeaconAdvertisementEnabled	IN	BeaconAdvertisementEnab-led	Enables / disables sending of beacons.	402, 501	O S
GetBeaconAdvertisement	✓	NewBeaconAdvertisementEnabled	OUT	BeaconAdvertisementEnab-led	Gets the current beacon advertisement mode.	402	O
SetLocationDescription	✓	NewLocationDescription	IN	LocationDescription	Sets the location description.	402, 605	O S
GetLocationDescription	✓	NewLocationDescription	OUT	LocationDescription	Gets the current location description.	402	O
SetRegulatoryDomain	✓	NewRegulatoryDomain	IN	RegulatoryDomain	Sets the regulatory domain.	402, 605	O S
GetRegulatoryDomain	✓	NewRegulatoryDomain	OUT	RegulatoryDomain	Gets the current regulatory domain.	402	O
GetFailureStatusInfo	✓	NewTotalIntegrityFailures NewTotalPSKFailures	OUT OUT	TotalIntegrityFailures TotalPSKFailures	Gets diagnostic parameters that may assist in detecting breakin attempts.	402	O
GetChannelsInUse	✓	NewChannelsInUse	OUT	ChannelsInUse	Gets the channels that are currently in use.	402, 501	O

SetDeviceOperationMode	✓	NewDeviceOperationMode NewSSID NewPeerBSSID NewChannel NewBasicDataTransmitRates NewOperationalDataTransmit-Rates NewDistanceFromRoot	IN IN IN IN IN IN IN	DeviceOperationMode SSID PeerBSSID Channel BasicDataTransmitRates OperationalDataTransmit-Rates DistanceFromRoot	Sets the access point operation mode. See the theory of operation section for more details.	402, 728, 729	O S
GetDeviceOperationMode	✓	NewDeviceOperationMode NewSSID NewPeerBSSID NewChannel NewBasicDataTransmitRates NewOperationalDataTransmit-Rates NewDistanceFromRoot	OUT OUT OUT OUT OUT OUT OUT	DeviceOperationMode SSID PeerBSSID Channel BasicDataTransmitRates OperationalDataTransmit-Rates DistanceFromRoot	Gets the current access point operation mode and related information.	402	O
SetAuthenticationService-Mode	✓	NewAuthenticationServiceMode	IN	AuthenticationServiceMode	Sets the access point authentication service mode (None, LinkAuthentication or RadiusClient). Note: The UPnP standard indicates that this is still required, but the request to make it optional was accepted, so this is probably an editing error.	402	O S
GetAuthenticationService-Mode	✓	NewAuthenticationServiceMode	OUT	AuthenticationServiceMode	Gets the access point authentication service mode. Note: The UPnP standard indicates that this is still required, but the request to make it optional was accepted, so this is probably an editing error.		O S
PreSharedKey Table Actions							
SetPreSharedKey	✓	NewPreSharedKeyIndex NewAssociatedDeviceMAC-Address NewPreSharedKey NewKeyPassphrase	IN IN IN IN	PreSharedKeyIndex AssociatedDeviceMAC-Address PreSharedKey KeyPassphrase	Sets one of the access point's pre-shared keys. If no MAC address is to be associated with the PSK, an empty string can be passed. Note: The UPnP standard does not support access to the default PSK (key 0) via SetPreSharedKey. TR-064 implementations MUST support the full range (0-9).	402	O S
GetPreSharedKey	✓	NewPreSharedKeyIndex NewAssociatedDeviceMAC-Address NewPreSharedKey NewKeyPassphrase	IN OUT OUT OUT	PreSharedKeyIndex AssociatedDeviceMAC-Address PreSharedKey KeyPassphrase	Gets one of the access point's pre-shared keys. If this PSK is not associated with a device, an empty string will be returned for the MAC address. Note: The UPnP standard does not support access to the default PSK (key 0) via GetPreSharedKey. TR-064 implementations MUST support the full range (0-9).	402, 714	O S
FactoryDefaultReset	✓	*None*	-	-	Resets all state variables to their factory default settings, disassociates all wireless clients (so the associated device list becomes empty), and resets all wireless sessions that were authenticated via WEP, PSK or EAP.	402	O S

					Note: FactoryDefaultReset affects only this WLANConfiguration service instance, not the entire device.		
ResetAuthentication	✓	*None*	-	-	Resets all wireless sessions that were authenticated via WEP, PSK or EAP, and removes all WEP and WPA keys (forcing new keys to be generated or reintroduced to the access point).	402	O S

6.6.4.3 Theory of Operation

Refer to the UPnP Forum WLANConfiguration specification[4] for detailed theory of operation.

6.6.5 Hosts

6.6.5.1 Overview

This service provides information about each of the hosts on this LAN, including those whose IP address was allocated by the CPE using DHCP as well as hosts with statically allocated IP addresses. It also allows the IP address of a particular host to be re-allocated by removing it from this table.

6.6.5.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:Hosts:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
HostNumberOfEntries	Host (index)	ui2	>=0	Number of entries in the Host table.	N/A	R
IPAddress	Host	String	IP Address	Current IP Address of the host.	N/A	R
AddressSource	Host	String	DHCP Static AutoIP	Indicates whether the IP address of the host was allocated by the CPE using DHCP or was assigned to the host statically.	N/A	R
LeaseTimeRemaining	Host	i4	≥ -1 -1 = Infinite lease	DHCP lease time remaining in seconds. The value must be 0 (zero) if the AddressSource is not DHCP.	N/A	R
MACAddress	Host KEY	String	MAC Address	MAC address of the host.	N/A	R
HostName	Host	String	Max length = 64 characters	The device's host name or empty string if unknown.	N/A	R
InterfaceType	Host	String	Ethernet USB 802.11 HomePNA HomePlug Other	Type of physical interface through which this host is connected to the CPE.	N/A	R
Active	Host	Boolean	1, 0	Whether or not the host is currently present on the LAN. The method of presence detection is a local matter to the CPE.	N/A	R

					The ability to list inactive hosts is Optional. If the CPE includes inactive hosts in this table, this variable MUST be set to zero for each inactive host. The length of time an inactive host remains listed in this table is a local matter to the CPE.		

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
HostTableActions						
GetHostNumberOfEntries	NewHostNumberOfEntries	OUT	HostNumberOfEntries	Retrieves the value of the HostNumberOfEntries state variable.	402, 501	R
GetSpecificHostEntry	NewMACAddress NewIPAddress NewAddressSource NewLeaseTimeRemaining NewInterfaceType NewActive NewHostName	IN OUT OUT OUT OUT OUT OUT	MACAddress IPAddress AddressSource LeaseTimeRemaining InterfaceType Active HostName	Retrieves all of the state variables associated with a particular host.	402, 501, 714	R
GetGenericHostEntry	NewIndex NewIPAddress NewAddressSource NewLeaseTimeRemaining NewMACAddress NewInterfaceType NewActive NewHostName	IN OUT OUT OUT OUT OUT OUT OUT	HostNumberOfEntries IPAddress AddressSource LeaseTimeRemaining MACAddress InterfaceType Active HostName	Retreives all of the state variables associated with a particular host.	402, 501, 713	R
DeleteHostEntry	NewMacAddress	IN	MacAddress	Removes the entry from the table..Note that the CPE MUST also remove the entry from its ARP table and returns the address to the DHCP pool if applicable.	402, 501, 702	O S

6.7 WANDevice

6.7.1 WANCommonInterfaceConfig

6.7.1.1 Overview

WANCommonInterfaceConfig is a standard UPnP service that models WAN interface properties common across all connection service instances.

6.7.1.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANCommonInterfaceConfig:1

StateVariables

Variable Name	From IGD	Table	Data Type	Allowed Value	Description	Default Value	R/O
WANAccessType	✓	-	String	DSL POTS Cable Ethernet	Specifies the WAN access (modem) between the residential network and the ISP.	N/A	R
Layer1UpstreamMaxBitRate	✓	-	ui4	>=0	Specifies the maximum upstream theoretical bit rate for the WAN device.	(Bits / second)	R

Layer1DownstreamMaxBitRate	✓	-	ui4	>=0	Specifies the maximum downstream theoretical bit rate for the WAN device.	(Bits / second)	R
Physical LinkStatus	✓	-	String	Up Down Initializing Unavailable	Indicates the state of the physical connection (link) from WANDevice to a connected entity.	N/A	R
WANAccessProvider	✓	-	String	String	Name of the Service Provider providing link connectivity on the WAN.	N/A	O
MaximumActiveConnections	✓	-	ui2	>=0	Indicates the maximum number of active connections the IGD can simultaneously support.	N/A	O
NumberOfActiveConnections	✓	Connection (Index)	ui2	>=0	Number of WAN**Connection service instances currently active on a WAN interface.	N/A	O
ActiveConnectionDeviceContainer	✓	Connection	String	Undefined	Refer to UPnP WANCommonInterfaceConfig v1.01 service definition for more details.	N/A	O
ActiveConnectionServiceID	✓	Connection KEY	String	Undefined	Refer to UPnP WANCommonInterfaceConfig v1.01 service definition for more details.	N/A	O
TotalBytesSent	✓	-	ui4	>=0	Represents the cumulative counter for total number of bytes sent upstream across all connection service instances on the WANDevice.	N/A	R
TotalBytesReceived	✓	-	ui4	>=0	Represents the cumulative counter for total number of bytes received downstream across all connection service instances on the WANDevice.	N/A	R
TotalPacketsSent	✓	-	ui4	>=0	Represent the cumulative counter for total number of packets (IP or PPP) sent upstream across all connection service instances on the WANDevice.	N/A	R
TotalPacketsReceived	✓	-	ui4	>=0	Represent the cumulative counter for total number of packets (IP or PPP) received downstream across all connection service instances on the WANDevice.	N/A	R
EnabledForInternet	✓	-	Boolean	0, 1	Refer to UPnP WANCommonInterfaceConfig v1.01 service definition for more details.	N/A	O

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetEnabledForInternet	NewEnabledForInternet	IN	EnabledForInternet	Refer to UPnP WANCommonInterfaceConfig v1.01 service definition for more details.	401, 402, 501	O S
GetEnabledForInternet	NewEnabledForInternet	OUT	EnabledForInternet	Refer to UPnP WANCommonInterfaceConfig v1.01 service definition for more details.	401, 402, 501	O

GetCommonLinkProperties	NewWANAccessType NewLayer1UpstreamMaxBitRate NewLayer1DownstreamMaxBitRate NewPhysicalLinkStatus	OUT OUT OUT OUT	WANAccessType Layer1UpstreamMaxBitRate Layer1DownstreamMaxBitRate PhysicalLinkStatus	Returns the values of the related state variables.	402	R
GetWANAccessProvider	NewAccessProvider	OUT	WANAccessProvider	Returns the value of the WANAccessProvider state variable.	402	O
GetMaximumActiveConnections	NewMaximumActiveConnections	OUT	MaximumActiveConnections	Returns the value of the MaximumActiveConnections state variable.	402	O
GetTotalBytesSent	NewTotalBytesSent	OUT	TotalBytesSent	Returns the value of the TotalBytesSent state variable.	402	R
GetTotalBytesReceived	NewTotalBytesReceived	OUT	TotalBytesReceived	Returns the value of the TotalBytesReceived state variable.	402	R
GetTotalPacketsSent	NewTotalPacketsSent	OUT	TotalPacketsSent	Returns the value of the TotalPacketsSent state variable.	402	R
GetTotalPacketsReceived	NewTotalPacketsReceived	OUT	TotalPacketsReceived	Returns the value of the TotalPacketsReceived state variable.	402	R
ConnectionTableActions						
GetNumberOfActiveConnections	NewNumberOfActiveConnections	OUT	NumberOfActiveConnections	Retrieves the value of the NumberOfActiveConnections state variable.	402, 501	O
GetSpecificActiveConnection	NewActiveConnectionServiceID NewActiveConnectionDeviceContainer	IN OUT	ActiveConnectionServiceID ActiveConnectionDeviceContainer	Returns the value of the ActiveConnection state variable.	402, 501, 714	O
GetActiveConnection	NewActiveConnectionIndex NewActiveConnectionDeviceContainer NewActiveConnectionServiceID	IN OUT OUT	NumberOfActiveConnections ActiveConnectionDeviceContainer ActiveConnectionServiceID	Returns the value of the ActiveConneciton state variable.	402, 501, 712	O

6.7.2 WANDSLInterfaceConfig

6.7.2.1 Overview

The WANDSLInterfaceConfig service models physical layer properties specific to a single physical connection of a DSL modem used for Internet access on an IGD. This service is required for an IGD with a DSL modem WAN interface, and is exclusive of any other WAN***InterfaceConfig service within a given WANDevice. The allowed values of the ModulationType state variable must reflect the supported modulation types of the device.

6.7.2.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANDSLInterfaceConfig:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
Enable	-	Boolean	1, 0	Enables or disables the link.	N/A	R

Status	-	String	Up Initializing EstablishingLink NoSignal Error Disabled	Status of the DSL physical link.	N/A	R
ModulationType	-	String	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	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	N/A	O
LineEncoding	-	String	DMT CAP 2B1Q 43BT PAM QAM	Describes 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.	N/A	O
LineNumber	-	ui1	≥ 1	Signifies the line pair that the modem is using to connection. LineNumber = 1 is the innermost pair.	N/A	O
UpstreamCurrRate	-	ui4	>=0	Provides a snapshot of the current payload bandwidth (expressed in Kbps) of the upstream DSL channel.	N/A	R
DownstreamCurrRate	-	ui4	>=0	Provides a snapshot of the current payload bandwidth (expressed in Kbps) of the downstream DSL channel.	N/A	R
UpstreamMaxRate	-	ui4	>=0	Describes the current attainable rate (expressed in Kbps) of the upstream DSL channel.	N/A	R
DownstreamMaxRate	-	ui4	>=0	Describes the current attainable rate (expressed in Kbps) of the downstream DSL channel.	N/A	R
UpstreamNoiseMargin	-	i4	-	Provides a snapshot of the current signal-to-noise ratio (expressed in 0.1 db) of the upstream DSL connection.	N/A	R
DownstreamNoiseMargin	-	i4	-	Provides a snapshot of the current signal-to-noise ratio (expressed in 0.1 dB) of the downstream DSL connection.	N/A	R
UpstreamAttenuation	-	i4	-	Provides a measurement of the upstream signal loss (expressed in 0.1 dB).	N/A	R
DownstreamAttenuation	-	i4	-	Provides a measurement of the downstream signal loss (expressed in 0.1 dB).	N/A	R
UpstreamPower	-	i4	-	Provides a snapshot of the current output power at the CPE's DSL interface (expressed in 0.1 dBmV).	N/A	R
DownstreamPower	-	i4	-	Provides a snapshot of the current received power signal level at the CPE's DSL interface (expressed in 0.1 dBmV).	N/A	R
DataPath	-	String	Interleaved Fast	Indicates whether the data path is fast (lower latency) or interleaved (lower error rate).	N/A	O
InterleavedDepth	-	ui2	>=0	ADSL Interleaved depth. This variable is only applicable if DataPath = Interleaved.	N/A	O

ATURVendor	-	String	Max length = 8 characters	ATU-R vendor identifier as defined in G.994.1 and T1.413.	N/A	R
ATURCountry	-	ui4	>=0	T.35 country code of the ATU-R vendor as defined in G.994.1.	N/A	R
ATURANSIStd	-	ui4	>=0	ATU-R T1.413 Revision Number as defined in T1.413 Issue 2.	N/A	O
ATURANSIRev	-	ui4	>=0	ATU-R Vendor Revision Number as defined in T1.413 Issue 2.	N/A	O
ATUCVendor	-	String	Max length = 8 characters	ATU-C vendor identifier as defined in G.992.1 and T1.413.	N/A	R
ATUCCountry	-	ui4	>=0	T.35 country code of the ATU-C vendor as defined in G.994.1.	N/A	R
ATUCANSIStd	-	ui4	>=0	ATU-C T1.413 Revision Number as defined in T1.413 Issue 2.	N/A	O
ATUCANSIRev	-	ui4	>=0	ATU-C Vendor Revision Number as defined in T1.413 Issue 2.	N/A	O
TotalStart	-	ui4	>=0	Number of seconds since the beginning of the period used for collection of Total statistics.	N/A	D
ShowtimeStart	-	ui4	>=0	Number of seconds since the most recent DSL Showtime—the beginning of the period used for collection of Showtime[3] statistics.	N/A	D
LastShowtimeStart	-	ui4	>=0	Number of seconds since the second most recent DSL Showtime—the beginning of the period used for collection of LastShowtime statistics.	N/A	D
CurrentDayStart	-	ui4	>=0	Number of seconds since the beginning of the period used for collection of CurrentDay statistics.	N/A	D
QuarterHourStart	-	ui4	>=0	Number of seconds since the beginning of the period used for collection of QuarterHour statistics.	N/A	D
Stats.Total.ReceiveBlocks	-	ui4	>=0	Total number of successfully received superframes.	N/A	R
Stats.Total.TransmitBlocks	-	ui4	>=0	Total number of successfully transmitted superframes.	N/A	R
Stats.Total.CellDelin	-	ui4	>=0	Total number of cell-delineation errors.	N/A	D
Stats.Total.LinkRetrain	-	ui4	>=0	Total number of link-retrain errors.	N/A	D
Stats.Total.InitErrors	-	ui4	>=0	Total number of initialization errors.	N/A	D
Stats.Total.InitTimeouts	-	ui4	>=0	Total number of initialization timeout errors.	N/A	D
Stats.Total.LossOfFraming	-	ui4	>=0	Total number of loss-of-framing errors.	N/A	D
Stats.Total.ErroredSecs	-	ui4	>=0	Total number of errored seconds.	N/A	D

[3] 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.

Stats.Total.SeverelyErroredSecs	-	ui4	>=0	Total number of severely errored seconds.	N/A	D
Stats.Total.FECErrors	-	ui4	>=0	Total number of FEC errors detected.	N/A	D
Stats.Total.ATUCFECErrors	-	ui4	>=0	Total number of FEC errors detected by the ATU-C.	N/A	D
Stats.Total.HECErrors	-	ui4	>=0	Total number of HEC errors detected.	N/A	D
Stats.Total.ATUCHECErrors	-	ui4	>=0	Total number of HEC errors detected by the ATU-C.	N/A	D
Stats.Total.CRCErrors	-	ui4	>=0	Total number of CRC errors detected.	N/A	D
Stats.Total.ATUCCRCErrors	-	ui4	>=0	Total number of CRC errors detected by the ATU-C.	N/A	D
Stats.Showtime.ReceiveBlocks	-	ui4	>=0	Number of successfully received superframes since the most recent DSL Showtime.	N/A	D
Stats.Showtime.TransmitBlocks	-	ui4	>=0	Number of successfully transmitted superframes since the most recent DSL Showtime.	N/A	D
Stats.Showtime.CellDelin	-	ui4	>=0	Number of cell-delineation errors since the most recent DSL Showtime.	N/A	D
Stats.Showtime.LinkRetrain	-	ui4	>=0	Number of link-retrain errors since the most recent DSL Showtime.	N/A	D
Stats.Showtime.InitErrors	-	ui4	>=0	Number of initialization errors since the most recent DSL Showtime.	N/A	D
Stats.Showtime.InitTimeouts	-	ui4	>=0	Number of initialization timeout errors since the most recent DSL Showtime.	N/A	D
Stats.Showtime.LossOfFraming	-	ui4	>=0	Number of loss-of-framing errors since the most recent DSL Showtime.	N/A	D
Stats.Showtime.ErroredSecs	-	ui4	>=0	Number of errored seconds since the most recent DSL Showtime.	N/A	D
Stats.Showtime.SeverelyErroredSecs	-	ui4	>=0	Number of severely errored seconds since the most recent DSL Showtime.	N/A	D
Stats.Showtime.FECErrors	-	ui4	>=0	Number of FEC errors detected since the most recent DSL Showtime.	N/A	D
Stats.Showtime.ATUCFECErrors	-	ui4	>=0	Number of FEC errors detected by the ATU-C since the most recent DSL Showtime.	N/A	D
Stats.Showtime.HECErrors	-	ui4	>=0	Number of HEC errors detected since the most recent DSL Showtime.	N/A	D
Stats.Showtime.ATUCHECErrors	-	ui4	>=0	Number of HEC errors detected by the ATU-C since the most recent DSL Showtime.	N/A	D
Stats.Showtime.CRCErrors	-	ui4	>=0	Number of CRC errors detected since the most recent DSL Showtime.	N/A	D
Stats.Showtime.ATUCCRCErrors	-	ui4	>=0	Number of CRC errors detected by the ATU-C since the most recent DSL Showtime.	N/A	D
Stats.LastShowtime.ReceiveBlocks	-	ui4	>=0	Number of successfully received superframes during the second most recent Showtime period.	N/A	D

Stats.LastShowtime.TransmitBlocks	-	ui4	>=0	Number of successfully transmitted superframes during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.CellDelin	-	ui4	>=0	Number of cell-delineation errors during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.LinkRetrain	-	ui4	>=0	Number of link-retrain errors during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.InitErrors	-	ui4	>=0	Number of initialization errors during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.InitTimeouts	-	ui4	>=0	Number of initialization timeout errors during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.LossOfFraming	-	ui4	>=0	Number of loss-of-framing errors during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.ErroredSecs	-	ui4	>=0	Number of errored seconds during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.SeverelyErroredSecs	-	ui4	>=0	Number of severely errored seconds during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.FECErrors	-	ui4	>=0	Number of FEC errors detected during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.ATUCFECErrors	-	ui4	>=0	Number of FEC errors detected by the ATU-C during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.HECErrors	-	ui4	>=0	Number of HEC errors detected during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.ATUCHECErrors	-	ui4	>=0	Number of HEC errors detected by the ATU-C during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.CRCErrors	-	ui4	>=0	Number of CRC errors detected during the second most recent Showtime period.	N/A	D
Stats.LastShowtime.ATUCCRCErrors	-	ui4	>=0	Number of CRC errors detected by the ATU-C during the second most recent Showtime period.	N/A	D
Stats.CurrentDay.ReceiveBlocks	-	ui4	>=0	Number of successfully received superframes during the current day.	N/A	D
Stats.CurrentDay.TransmitBlocks	-	ui4	>=0	Number of successfully transmitted superframes during the current day.	N/A	D
Stats.CurrentDay.CellDelin	-	ui4	>=0	Number of cell-delineation errors during the current day.	N/A	D
Stats.CurrentDay.LinkRetrain	-	ui4	>=0	Number of link-retrain errors during the current day.	N/A	D
Stats.CurrentDay.InitErrors	-	ui4	>=0	Number of initialization errors during the current day.	N/A	D
Stats.CurrentDay.InitTimeouts	-	ui4	>=0	Number of initialization timeout errors during the current day.	N/A	D
Stats.CurrentDay.LossOfFraming	-	ui4	>=0	Number of loss-of-framing errors during the current day.	N/A	D
Stats.CurrentDay.ErroredSecs	-	ui4	>=0	Number of errored seconds during the current day.	N/A	D

Stats.CurrentDay.SeverelyErroredSecs	-	ui4	>=0	Number of severely errored seconds during the current day.	N/A	D
Stats.CurrentDay.FECErrors	-	ui4	>=0	Number of FEC errors detected during the current day.	N/A	D
Stats.CurrentDay.ATUCFECErrors	-	ui4	>=0	Number of FEC errors detected by the ATU-C during the current day.	N/A	D
Stats.CurrentDay.HECErrors	-	ui4	>=0	Number of HEC errors detected during the current day.	N/A	D
Stats.CurrentDay.ATUCHECErrors	-	ui4	>=0	Number of HEC errors detected by the ATU-C during the current day.	N/A	D
Stats.CurrentDay.CRCErrors	-	ui4	>=0	Number of CRC errors detected during the current day.	N/A	D
Stats.CurrentDay.ATUCCRCErrors	-	ui4	>=0	Number of CRC errors detected by the ATU-C during the current day.	N/A	D
Stats.QuarterHour.ReceiveBlocks	-	ui4	>=0	Number of successfully received superframes durring the current quarter hour.	N/A	D
Stats.QuarterHour.TransmitBlocks	-	ui4	>=0	Number of successfully transmitted superframes durring the current quarter hour.	N/A	D
Stats.QuarterHour.CellDelin	-	ui4	>=0	Number of cell-delineation errors durring the current quarter hour.	N/A	D
Stats.QuarterHour.LinkRetrain	-	ui4	>=0	Number of link-retrain errors durring the current quarter hour.	N/A	D
Stats.QuarterHour.InitErrors	-	ui4	>=0	Number of initialization errors durring the current quarter hour.	N/A	D
Stats.QuarterHour.InitTimeouts	-	ui4	>=0	Number of initialization timeout errors durring the current quarter hour.	N/A	D
Stats.QuarterHour.LossOfFraming	-	ui4	>=0	Number of loss-of-framing errors durring the current quarter hour.	N/A	D
Stats.QuarterHour.ErroredSecs	-	ui4	>=0	Number of errored seconds durring the current quarter hour.	N/A	D
Stats.QuarterHour.SeverelyErroredSecs	-	ui4	>=0	Number of severely errored seconds durring the current quarter hour.	N/A	D
Stats.QuarterHour.FECErrors	-	ui4	>=0	Number of FEC errors detected durring the current quarter hour.	N/A	D
Stats.QuarterHour.ATUCFECErrors	-	ui4	>=0	Number of FEC errors detected by the ATU-C durring the current quarter hour.	N/A	D
Stats.QuarterHour.HECErrors	-	ui4	>=0	Number of HEC errors detected durring the current quarter hour.	N/A	D
Stats.QuarterHour.ATUCHECErrors	-	ui4	>=0	Number of HEC errors detected by the ATU-C durring the current quarter hour.	N/A	D
Stats.QuarterHour.CRCErrors	-	ui4	>=0	Number of CRC errors detected durring the current quarter hour.	N/A	D
Stats.QuarterHour.ATUCCRCErrors	-	ui4	>=0	Number of CRC errors detected by the ATU-C durring the current quarter hour.	N/A	D

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetEnable	NewEnable	IN	Enable	Sets the value of the Enable state variable to enable or disable the link..	402, 501	R S
GetInfo	NewEnable NewStatus NewModulationType NewLineEncoding NewLineNumber NewUpstreamCurrRate NewDownstreamCurrRate NewUpstreamMaxRate NewDownstreamMaxRate NewUpstreamNoiseMargin NewDownstreamNoiseMargin NewUpstreamAttenuation NewDownstreamAttenuation NewUpstreamPower NewDownstreamPower NewDataPath NewInterleavedDepth NewATURVendor NewATURCountry NewATURANSIStd NewATURANSIRev NewATUCVendor NewATUCCountry NewATUCANSIStd NewATUCANSIRev NewTotalStart NewShowtimeStart NewLastShowtimeStart NewCurrentDayStart NewQuarterHourStart	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	Enable Status ModulationType LineEncoding LineNumber UpstreamCurrRate DownstreamCurrRate UpstreamMaxRate DownstreamMaxRate UpstreamNoiseMargin DownstreamNoiseMargin UpstreamAttenuation DownstreamAttenuation UpstreamPower DownstreamPower DataPath InterleavedDepth ATURVendor ATURCountry ATURANSIStd ATURANSIRev ATUCVendor ATUCCountry ATUCANSIStd ATUCANSIRev TotalStart ShowtimeStart LastShowtimeStart CurrentDayStart QuarterHourStart	Retrieves all of the state variables not associated with a table.	402, 501	R
GetStatisticsTotal	NewReceiveBlocks NewTransmitBlocks NewCellDelin NewLinkRetrain NewInitErrors NewInitTimeouts NewLossOfFraming NewErroredSecs NewSeverelyErroredSecs NewFECErrors NewATUCFECErrors NewHECErrors NewATUCHECErrors NewCRCErrors NewATUCCRCErrors	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	Stats.Total.ReceiveBlocks Stats.Total.TransmitBlocks Stats.Total.CellDelin Stats.Total.LinkRetrain Stats.Total.InitErrors Stats.Total.InitTimeouts Stats.Total.LossOfFraming Stats.Total.ErroredSecs Stats.Total.SeverelyErroredSecs Stats.Total.FECErrors Stats.Total.ATUCFECErrors Stats.Total.HECErrors Stats.Total.ATUCHECErrors Stats.Total.CRCErrors Stats.Total.ATUCCRCErrors	Retreives all of the DSL statistics for the Total time period.	402, 501	R
GetStatisticsShowtime	NewReceiveBlocks NewTransmitBlocks NewCellDelin NewLinkRetrain NewInitErrors NewInitTimeouts NewLossOfFraming NewErroredSecs NewSeverelyErroredSecs NewFECErrors NewATUCFECErrors NewHECErrors NewATUCHECErrors NewCRCErrors NewATUCCRCErrors	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	Stats.Showtime.ReceiveBlocks Stats.Showtime.TransmitBlocks Stats.Showtime.CellDelin Stats.Showtime.LinkRetrain Stats.Showtime.InitErrors Stats.Showtime.InitTimeouts Stats.Showtime.LossOfFraming Stats.Showtime.ErroredSecs Stats.Showtime.SeverelyErroredSecs Stats.Showtime.FECErrors Stats.Showtime.ATUCFECErrors Stats.Showtime.HECErrors Stats.Showtime.ATUCHECErrors Stats.Showtime.CRCErrors Stats.Showtime.ATUCCRCErrors	Retreives all of the DSL statistics for the Showtime time period.	402, 501	D
GetStatisticsLastShowtime	NewReceiveBlocks NewTransmitBlocks NewCellDelin NewLinkRetrain NewInitErrors NewInitTimeouts NewLossOfFraming NewErroredSecs NewSeverelyErroredSecs NewFECErrors NewATUCFECErrors NewHECErrors	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	Stats.LastShowtime.ReceiveBlocks Stats.LastShowtime.TransmitBlocks Stats.LastShowtime.CellDelin Stats.LastShowtime.LinkRetrain Stats.LastShowtime.InitErrors Stats.LastShowtime.InitTimeouts Stats.LastShowtime.LossOfFraming Stats.LastShowtime.ErroredSecs Stats.LastShowtime.SeverelyErroredSecs Stats.LastShowtime.FECErrors Stats.LastShowtime.ATUCFECErrors Stats.LastShowtime.HECErrors	Retreives all of the DSL statistics for the LastShowtime time period.	402, 501	D

	NewATUCHECErrors	OUT	Stats.LastShowtime.ATUCHECErrors			
	NewCRCErrors	OUT	Stats.LastShowtime.CRCErrors			
	NewATUCCRCErrors	OUT	Stats.LastShowtime.ATUCCRCErrors			
GetStatisticsCurrentDay	NewReceiveBlocks	OUT	Stats.CurrentDay.ReceiveBlocks	Retreives all of the DSL statistics for the CurrentDay time period.	402, 501	D
	NewTransmitBlocks	OUT	Stats.CurrentDay.TransmitBlocks			
	NewCellDelin	OUT	Stats.CurrentDay.CellDelin			
	NewLinkRetrain	OUT	Stats.CurrentDay.LinkRetrain			
	NewInitErrors	OUT	Stats.CurrentDay.InitErrors			
	NewInitTimeouts	OUT	Stats.CurrentDay.InitTimeouts			
	NewLossOfFraming	OUT	Stats.CurrentDay.LossOfFraming			
	NewErroredSecs	OUT	Stats.CurrentDay.ErroredSecs			
	NewSeverelyErroredSecs	OUT	Stats.CurrentDay.SeverelyErroredSecs			
	NewFECErrors	OUT	Stats.CurrentDay.FECErrors			
	NewATUCFECErrors	OUT	Stats.CurrentDay.ATUCFECErrors			
	NewHECErrors	OUT	Stats.CurrentDay.HECErrors			
	NewATUCHECErrors	OUT	Stats.CurrentDay.ATUCHECErrors			
	NewCRCErrors	OUT	Stats.CurrentDay.CRCErrors			
	NewATUCCRCErrors	OUT	Stats.CurrentDay.ATUCCRCErrors			
GetStatisticsQuarterHour	NewReceiveBlocks	OUT	Stats.QuarterHour.ReceiveBlocks	Retreives all of the DSL statistics for the QuarterHour time period.	402, 501	D
	NewTransmitBlocks	OUT	Stats.QuarterHour.TransmitBlocks			
	NewCellDelin	OUT	Stats.QuarterHour.CellDelin			
	NewLinkRetrain	OUT	Stats.QuarterHour.LinkRetrain			
	NewInitErrors	OUT	Stats.QuarterHour.InitErrors			
	NewInitTimeouts	OUT	Stats.QuarterHour.InitTimeouts			
	NewLossOfFraming	OUT	Stats.QuarterHour.LossOfFraming			
	NewErroredSecs	OUT	Stats.QuarterHour.ErroredSecs			
	NewSeverelyErroredSecs	OUT	Stats.QuarterHour.SeverelyErroredSecs			
	NewFECErrors	OUT	Stats.QuarterHour.FECErrors			
	NewATUCFECErrors	OUT	Stats.QuarterHour.ATUCFECErrors			
	NewHECErrors	OUT	Stats.QuarterHour.HECErrors			
	NewATUCHECErrors	OUT	Stats.QuarterHour.ATUCHECErrors			
	NewCRCErrors	OUT	Stats.QuarterHour.CRCErrors			
	NewATUCCRCErrors	OUT	Stats.QuarterHour.ATUCCRCErrors			

6.7.3 WANEthernetInterfaceConfig

6.7.3.1 Overview

The WANEthernetInterfaceConfig service models physical layer properties specific to a single physical Ethernet connection of a RGW used for Internet access on an IGD. This service is required for an IGD with an Ethernet WAN interface, and is exclusive of any other WAN***InterfaceConfig service within a given WANDevice. Note that this service is not related to the Ethernet protocol layer sometimes associated with a DSL connection.

Definition of this service is the same as definition of the LANEthernetInterfaceConfig service.

6.7.3.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANEthernetInterfaceConfig:1

StateVariables

Refer to LANEthernetInterfaceConfig

Actions, Arguments & Errors

Refer to LANEthernetInterfaceConfig

6.7.4 WANDSLConnectionManagement

6.7.4.1 Overview

The WANDSLConnectionManagement is a service that enables a control point to manage the WAN device of a DSL enabled IGD. This service is required for an IGD with a DSL modem WAN interface.

6.7.4.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANDSLConnectionManagement:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
WANConnectionDevice	ConnectionService	String	String Max length = 256 characters	2-tuple (URN, uuid) referring to WANConnectionDevice the service belongs to.	N/A	R
WANConnectionService	ConnectionService KEY	String	String Max length = 256 characters	2-tuple (service URN, embedding device uuid) identifying the WAN**Connection service.	N/A	R
DestinationAddress	ConnectionService	String	PVC: VPI/VCI SVC: ATM connection name SVC: ATM address	Destination address of the WANConnectionDevice entry.	N/A	R
LinkType	ConnectionService	String	One of Link Types as described in WANDSLLinkConfig service	Link Type of the WANConnectionDevice entry.	N/A	R
ConnectionType	ConnectionService	String	One of PossibleConnectionTypes as described in WAN**Connection service	Connection Type of the WAN**Connection service entry.	N/A	R
Name	ConnectionService	String	String Max length = 32 characters	Friendly Name of the WAN**Connection service.	N/A	R
WANConnectionService-NumberOfEntries	ConnectionService (index)	ui2	>=0	Number of WAN**Connection service entries.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
ConnectionService Table Actions						
GetWANConnection-ServiceNumberOfEntries	NewWANConnectionService-NumberOfEntries	OUT	WANConnectionServiceNumberOfEntries	Retrieves the value of the WANConnection-ServiceNumberOfEntries state variable.	402, 501	R
GetGenericConnection ServiceEntry	NewConnectionServiceIndex NewWANConnectionDevice NewWANConnectionService NewName	IN OUT OUT OUT	WANConnectionServiceNumberOfEntries WANConnectionDevice WANConnectionService Name	Retrieves WAN Connection Services one entry at a time. Control points can call this action with an incrementing array index until no more entries are found on the gateway. Index is ranging from 0 to WANConnection-ServiceNumberOfEntries – 1.	402, 501, 713	R
GetSpecificConnection ServiceEntry	NewWANConnectionService NewWANConnectionDevice NewName	IN OUT OUT	WANConnectionService WANConnectionDevice Name	Retrieves Specific WAN Connection Services.	402, 501, 714	R
AddConnection DeviceAndService	NewDestinationAddress NewLinkType NewConnectionType NewName NewWANConnectionDevice	IN IN IN IN OUT	DestinationAddress LinkType ConnectionType Name WANConnectionDevice	Create a new WanConnectionDevice and insert a new WAN**Connection service in the created	402, 501, 701	R S

	NewWANConnectionService	OUT	WANConnectionService	device.		
AddConnectionService	NewWANConnectionDevice NewConnectionType NewName NewWANConnectionService	IN IN IN OUT	WANConnectionDevice ConnectionType Name WANConnectionService	Create a new WAN**Connection service in the specified WANConnectionDevice.	402, 501, 701	R S
DeleteConnectionService	NewWANConnectionService	IN	WANConnectionService	Deletes the specified WAN**Connection service.	402, 501, 702, 703.	R S

6.7.4.3 Theory of Operation

The WANDSLConnectionManagement provides a number of actions to manage the WAN**Connection services via adding or deleting services. The following actions are supported:

- The AddConnectionService action allows adding a new WAN**Connection service instance to an existing instance of a WANConnectionDevice.
- The AddConnectionDeviceAndService action allows adding a WAN**Connection instance to a newly created WANConnectionDevice by specifying the DestinationAddres, LinkType (WANDSLLinkConfig variables), ConnectionType and Name (WAN**Connection service variables).
- The DeleteConnectionService action allows deletion of a WAN**Connection service instance if the service ConnectionStatus is not “Connected”. If the Service is the only service in one WANConnectionDevice, the service AND WANConnectionDevice are deleted.

*Notice that UPnP requires that a WANDevice contains at least one instance of a WANConnectionDevice with at least one WAN**Connection service. So removing all WAN**Connection services will lead to UPnP no longer being able to advertise the IGD.*

In case this interface is co-existing with UPnP and both share the same data-model, adding or deleting WANConnectionDevices or WAN**Connection services will lead to updating of the device description document and re-advertising of the IGD.

***: IP or PPP*

6.7.5 WANDSLDiagnostics

6.7.5.1 Overview

The WANDSLDiagnostics is a service which allows for the operation of an explicit DSL diagnostic test, and the ability to examine the results of the last such test.

6.7.5.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANDSLDiagnostics:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O

LoopDiagnosticsState	-	String	None Requested Complete	Indicates availability of diagnostic data. Value may be set to Requested to initiate the diagnostic test, which brings down the DSL connection while the test is operating.	None	**R**
ACTPSDds	-	i4	-	Downstream actual power spectral density. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
ACTPSDus	-	i4	-	Upstream actual power spectral density. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
ACTATPds	-	i4	-	Downstream actual aggregate transmitter power. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
ACTATPus	-	i4	-	Upstream actual aggregate transmitter power. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
HLINSCds	-	i4	-	Downstream linear representation scale. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
HLINpsds	-	String	Comma-separated list of integers Max number of complex pairs = 256 for ADSL and ADSL2, 512 for ADSL2+	Downstream linear channel characteristics per subcarrier. Each successive pair of integers represents the real and imaginary parts of each complex value. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
QLNpsds	-	String	Comma-separated list of integers Max number of elements = 256 for ADSL and ADSL2, 512 for ADSL2+	Downstream quiet line noise per subcarrier. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
SNRpsds	-	String	Comma-separated list of integers Max number of elements = 256 for ADSL and ADSL2, 512 for ADSL2+	Downstream SNR per subcarrier. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
BITSpsds	-	String	Comma-separated list of integers Max number of elements = 256 for ADSL and ADSL2, 512 for ADSL2+	Downstream bit allocation per subcarrier. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R
GAINSpsds	-	String	Comma-separated list of integers Max number of elements = 256 for ADSL and ADSL2, 512 for ADSL2+	Downstream gain allocation per subcarrier. Interpretation of the value is as defined in ITU-T Rec. G.997.1.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetLoopDiagnosticsState	NewLoopDiagnosticsState	IN	LoopDiagnosticsState	Setting LoopDiagnostic-State to Requested Initiates Loop Diagnostics Data collection, which brings down the DSL connection while the test is operating. Values other than Requested are not allowed in this action.	402, 501	R
GetLoopDiagnosticsState	NewLoopDiagnosticsState	OUT	LoopDiagnosticsState	Returns the current value of LoopDiagnosticsState.	402, 501	R

GetLoopDiagnosticsResults	NewACTPSDds	OUT	ACTPSDds	Results of the last diagnostic test.	402, 501	R
	NewACTPSDus	OUT	ACTPSDus			S
	NewACTATPds	OUT	ACTATPds			
	NewACTATPus	OUT	ACTATPus			
	NewHLINSCds	OUT	HLINSCds			
	NewHLINpsds	OUT	HLINpsds			
	NewQLNpsds	OUT	QLNpsds			
	NewSNRpsds	OUT	SNRpsds			
	NewBITSpsds	OUT	BITSpsds			
	NewGAINSpsds	OUT	GAINSpsds			

6.8 WANConnectionDevice

An instance of a WANDevice includes one or more WANConnectionDevices with a corresponding set of services, depending on the connection method of the WAN interface

See the UPnP WANDevice and WANConnectionDevice specifications as well as the UPnP Theory of Operation section above for more details on these devices and related services.

6.8.1 WANDSLLinkConfig

6.8.1.1 Overview

WANDSLLinkConfig is a standard UPnP service that models the ATM link layer properties specific to a single physical connection of a DSL modem used for Internet access on an IGD. This service is required for an IGD with a DSL modem WAN interface, and is exclusive of any other WAN***LinkConfig service within a given WANConnectionDevice. The allowed values of the LinkType state variable must reflect the supported LinkTypes of the device.

6.8.1.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANDSLLinkConfig:1

StateVariables

Variable Name	From IGD	Table	Data Type	Allowed Value	Description	Default Value	R/O
Enable	-	-	Boolean	1, 0	Enables or disables the link.	N/A	R
LinkType	✓	-	String	EoA (RFC2684B) IPoA (RFC2684R) PPPoA CIP PPPoE Unconfigured	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	N/A	R
LinkStatus	✓	-	String	Up Down Initializing Unavailable	***This parameter has been duplicated to WANDSLInterfaceConfig which models the DSL connection outside the WANConnectionDevice (this is required for multiple VP/VC support over one DSL link).***	N/A	R
AutoConfig	✓	-	Boolean	0, 1	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	N/A	R
ModulationType	✓	-	String	ADSL_G.dmt ADSL_G.lite ADSL_G.dmt.bis ADSL_re-adsl ADSL_2plus ADLS_four ADSL_ANSI_T1.413 G.shdsl IDSL	***This parameter has been duplicated to WANDSLInterfaceConfig which models the DSL connection outside the WANConnectionDevice (this is required for multiple VP/VC support over one DSL link).***	N/A	O

				HDSL SDSL VDSL			
DestinationAddress	✓	-	String	PVC: VPI/VCI SVC: ATM connection name SVC: ATM address	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	N/A	R
ATMEncapsulation	✓	-	String	LLC VCMUX	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	N/A	O
FCSPreserved	✓	-	Boolean	-	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	N/A	O
VCSearchList	-	-	String	-	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, …	N/A	O
ATMAAL	-	-	String	AAL1 AAL2 AAL3 AAL4 AAL5	Describes the ATM Adaptation Layer (AAL) currently in use on the PVC.	N/A	O
ATMTransmittedBlocks	-	-	ui4	>=0	The current count of successfully transmitted blocks.	N/A	R
ATMReceivedBlocks	-	-	ui4	>=0	The current count of successfully received blocks.	N/A	R
ATMQoS	-	-	String	UBR CBR GFR VBR-nrt VBR-rt UBR+ ABR	Describes the ATM Quality Of Service (QoS) being used on the PVC.	N/A	O
ATMPeakCellRate	-	-	ui4	>=0	Specifies the upstream peak cell rate for the WANConnectionDevice instance.	(cells / second)	O
ATMMaximumBurstSize	-	-	ui4	>=0	Specifies the upstream maximum burst size for the WANConnectionDevice instance.	(cells)	O
ATMSustainableCellRate	-	-	ui4	>=0	Specifies the upstream sustainable cell rate for the WANConnectionDevice instance used for traffic shaping.	(cells / second)	O
AAL5CRCErrors	-	-	ui4	>=0	Provides a count of the AAL5 layer cyclic redundancy check errors.	N/A	R
ATMCRCErrors	-	-	ui4	>=0	Provides a count of the ATM layer cyclic redundancy check errors.	N/A	R
ATMHECErrors	-	-	ui4	>=0	Provides a count of the number of Header Error Check related errors at the ATM layer.	N/A	O

Actions, Arguments & Errors

Name	From IGD	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetEnable	-	NewEnable	IN	Enable	Sets the value of the Enable state variable to enable or disable the link.	402, 501	R S

GetInfo	-	NewEnable NewLinkType NewLinkStatus NewAutoConfig NewModulationType NewDestinationAddress NewATMEncapsulation NewFCSPreserved NewVCSearchList NewATMAAL NewATMQoS NewATMPeakCellRate NewATMMaximumBurstSize NewATMSustainableCellRate	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	Enable LinkType LinkStatus AutoConfig ModulationType DestinationAddress ATMEncapsulation FCSPreserved VCSearchList ATMAAL ATMQoS ATMPeakCellRate ATMMaximumBurstSize ATMSustainableCellRate	Retrieves all of the state variables in this service other than statistics.	402, 501	R
SetDSLLinkType	✓	NewLinkType	IN	LinkType	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402, 501, 719	R S
GetDSLLinkInfo	✓	NewLinkType	OUT	LinkType	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402, 501	R
GetAutoConfig	✓	NewAutoConfig	OUT	AutoConfig	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402	R
GetModulationType	✓	NewModulationType	OUT	ModulationType	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402	O
SetDestinationAddress	✓	NewDestinationAddress	IN	DestinationAddress	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402, 501, 719	O S
GetDestinationAddress	✓	NewDestinationAddress	OUT	DestinationAddress	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402, 501	R
SetATMEncapsulation	✓	NewATMEncapsulation	IN	ATMEncapsulation	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402, 501, 719	O S
GetATMEncapsulation	✓	NewATMEncapsulation	OUT	ATMEncapsulation	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402, 501	O
SetFCSPreserved	✓	NewFCSPreserved	IN	FCSPreserved	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402, 501, 719	O S
GetFCSPreserved	✓	NewFCSPreserved	OUT	FCSPreserved	Refer to UPnP WANDSLLinkConfig v1.01 service definition for more details.	402, 501	O
SetVCSearchList	-	NewVCSearchList	IN	VCSearchList	Sets the value of the VCSearchList state variable.	402, 501	O S
SetATMQos	-	NewATMQoS NewATMPeakCellRate NewATMSustainableCellRate NewATMMaximumBurstSize	IN IN IN IN	ATMQoS ATMPeakCellRate ATMSustainableCellRate ATMMaximumBurstSize	Configures the ATM QoS parameters used for traffic shaping per WANConnectionDevice (corresponds to an ATM VP/VC).	402, 501	O S
GetStatistics	-	NewATMTransmittedBlocks NewATMReceivedBlocks NewAAL5CRCErrors NewATMCRCErrors	OUT OUT OUT OUT	ATMTransmittedBlocks ATMReceivedBlocks AAL5CRCErrors ATMCRCErrors	Retrieves all of the statistics-related state variables.	402, 501	R

	NewATMHECErrors	OUT	ATMHECErrors

6.8.2 WANEthernetLinkConfig

6.8.2.1 Overview

WANEthernetLinkConfig is a standard UPnP service that models the Ethernet link layer properties specific to a single physical Ethernet connection of a GW used for Internet access on an IGD. This service is required for an IGD with an Ethernet WAN interface, and is exclusive of any other WAN***LinkConfig service within a given WANConnectionDevice. Note that this service is ***not*** related to the Ethernet protocol layer sometimes used in associated with a DSL connection.

6.8.2.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANEthernetLinkConfig:1

StateVariables

Variable Name	Table	Data Type	Allowed Value	Description	Default Value	R/O
EthernetLinkStatus	-	string	Up Down Unavailable	Refer to UPnP WANEthernetLinkConfig v1.01 service definition for more details.	N/A	R

Actions, Arguments & Errors

Name	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
GetEthernetLinkStatus	NewEthernetLinkStatus	OUT	EthernetLinkStatus	Retrieves the value of related state variable.	402, 501	R

6.8.3 WANPOTSLinkConfig

6.8.3.1 Overview

This service models a dial-up POTS connection within the IGD. This service is required for devices that support a POTS interface.

6.8.3.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANPOTSLinkConfig:1

StateVariables

Variable Name	From IGD	Data Type	Allowed Value	Description	Default Value	R/O
Enable	-	boolean	1, 0	Enables or disables the link.	N/A	R
LinkStatus	-	string	Up Down Dialing	Indicates the status of the link.	N/A	R

			Connecting Unavailable			
ISPPhoneNumber	✓	string Max length = 64 characters	N/A	This variable 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 E.164 specification. Delimiters such as brackets or hyphens between the digits of a phone number are to be ignored by the gateway.	Empty string	R
ISPInfo	✓	string Max length = 64 characters	N/A	This variable provides information identifying the Internet Service Provider. The format of the string is vendor specific.	Empty string	R
LinkType	✓	string	PPP_Dialup	This variable indicates the type of POTS link used for the dialup connection.	PPP_Dialup	R
NumberOfRetries	✓	ui4	>=0	This variable specifies the number of times the gateway should attempt an Internet connection setup before returning error.	Not specified	R
DelayBetweenRetries	✓	ui4	>=0	This variable specifies the number of seconds the gateway should wait between attempts to setup an Internet connection.	Not specified	R
Fclass	✓	string	Comma separated sub-strings indicating numbers 0, 1, 2, 2.0, 8, 80	This variable specifies capabilities of the POTS modem – i.e., if it handles data (0), fax (1,2,2.0), voice (8), DSVD (80).	Empty string	O
DataModulationSupported	✓	string	V92 V90 V34 V32bis V32	This variable exposes the modulation standard used for data.	Empty string	O
DataProtocol	✓	string	V42_LAPM V42_MNP4 V14 V80	This variable exposes the protocol standard for data transfers.	Depends on specific modem	O
DataCompression	✓	string	V42bis MNP5	This variable exposes the compression technology implemented on the modem.	Depends on specific modem	O
PlusVTRCommandSupported	✓	boolean	-	This variable is used for full duplex operation with data and voice.	Not specified	O

Actions, Arguments & Errors

Name	From IGD	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetEnable	-	NewEnable	IN	Enable	Sets the value of the Enable state variable to enable or disable the link.	402, 501	R S
GetInfo	-	NewEnable NewLinkStatus NewISPPhoneNumber NewISPInfo NewLinkType NewNumberOfRetries NewDelayBetweenRetries NewFclass NewDataModulationSupported	OUT OUT OUT OUT OUT OUT OUT OUT OUT	Enable LinkStatus ISPPhoneNumber ISPInfo LinkType NumberOfRetries DelayBetweenRetries Fclass DataModulationSupported	Retrieves all of the state variables in this service.	402, 501	R

		NewDataProtocol NewDataCompression NewPlusVTRCommandSupported	OUT OUT OUT	DataProtocol DataCompression PlusVTRCommandSupported			
SetISPInfo	✓	NewISPPhoneNumber NewISPInfo NewLinkType	IN IN IN	ISPPhoneNumber ISPInfo LinkType	This action changes the variables used to dial up the ISP.	402, 501	O
SetCallRetryInfo	✓	NewNumberofRetries NewDelayBetweenRetries	IN IN	NumberofRetries DelayBetweenRetries	This action changes the number of retry attempts if connection errors are encountered.	402,501	O
GetISPInfo	✓	NewISPPhoneNumber NewISPInfo NewLinkType	OUT OUT OUT	ISPPhoneNumber ISPInfo LinkType	This action retrieves the variables used to dial up the ISP.	402	R
GetCallRetryInfo	✓	NewNumberofRetries NewDelayBetweenRetries	OUT OUT	NumberofRetries DelayBetweenRetries	This action retrieves the number of retry attempts if connection errors are encountered.	402	R
GetFclass	✓	NewFclass	OUT	Fclass	This action queries modem capabilities.	402	O
GetDataModulationSupported	✓	NewDataModulationSupported	OUT	DataModulationSupported	This action queries the modulation standard used for data.	402	O
GetDataProtocol	✓	NewDataProtocol	OUT	DataProtocol	This action queries the protocol standard used for data transfers.	402	O
GetDataCompression	✓	NewDataCompression	OUT	DataCompression	This action queries the compression technology implemented on the modem.	402	O
GetPlusVTRCommandSupported	✓	NewPlusVTRCommandSupported	OUT	PlusVTRCommandSupported	This action queries for full duplex operation with data and voice.	402	O

6.8.3.3 Theory of Operation

When the gateway receives an action request to initiate a connection, it will try each of the ISP phone numbers sequentially starting from the first one to the last until a successful connection is made, or all the numbers are exhausted. For each phone number, upon a failed connection attempt the gateway will retry to connect as specified by the state variables NumberOfRetries and DelayBetweenRetries. This sequence will be interrupted if a connection termination action is received by the gateway (such as ForceTermination or RequestTermination). In this case the gateway will set the LastConnectionError state variable to ERROR_COMMAND_ABORTED and return ConnectionSetupFailed (704). If not interrupted, the gateway will return this error only if all the connection attempts fail. In this case the LastConnectionError state variable will be set to the appropriate RAS error received during the last failed connection attempt.

6.8.4 WANIPConnection

6.8.4.1 Overview

WANIPConnection is a standard UPnP service that enables a UPnP control point to configure and control IP connections on the WAN interface of a UPnP compliant IGD. This service is required for all WANConnection Devices not employing PPP addressing. This service must not be active for WANConnection Devices that do employ PPP addressing. The allowed values of the

PossibleConnectionType and ConnectionType state variables must reflect the supported ConnectionTypes of the device.

6.8.4.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANIPConnection:1

StateVariables

Variable Name	From IGD	Table	Data Type	Allowed Value	Description	Default Value	R/O
Enable	-	-	Boolean	1, 0	Enables or disables the connection.	N/A	R
ConnectionType	✓	-	String	One of PossibleConnectionType	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
PossibleConnectionTypes	✓	-	String	Unconfigured IP_Routed IP_Bridged	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
ConnectionStatus	✓	-	String	Unconfigured Connecting Connected PendingDisconnect Disconnetiong Disconnected	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
Name	-	-	String	String	Friendly Name of the connection. This name must be unique within the scope of the IGD.	N/A	R
Uptime	✓	-	ui2	>=0	Refer to UPnP WANIPConnection v1.01 service definition for more details.	(Seconds)	R
LastConnectionError	✓	-	String	One of the following: 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	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R R O O O O O O O O R R
AutoDisconnectTime	✓	-	ui4	>=0	Refer to UPnP WANIPConnection v1.01 service definition for more details.	(Seconds)	O
IdleDisconnectTime	✓	-	ui4	>=0	Refer to UPnP WANIPConnection v1.01 service definition for more details.	(Seconds)	O
WarnDisconnectDelay	✓	-	ui4	>=0	Refer to UPnP WANIPConnection v1.01 service definition for more details.	(Seconds)	O

RSIPAvailable	✓	-	Boolean	0, 1	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
NATEnabled	✓	-	Boolean	0, 1	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
ExternalIPAddress	✓	-	String	IP Address	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
SubnetMask	-	-	String	IP Subnet Mask	Subnet mask of the WAN interface. This variable will only be configurable from the LAN-side if the AddressingType is Static.	N/A	R
PortMappingNumberOfEntries	✓	PortMapping (index)	ui2	>=0	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
PortMappingEnabled	✓	PortMapping	Boolean	0, 1	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
PortMappingLeaseDuration	✓	PortMapping	ui4	>=0	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
RemoteHost	✓	PortMapping KEY	String	IP Address or Empty String	Refer to UPnP WANIPConnection v1.01 service definition for more details.	Empty String	R
ExternalPort	✓	PortMapping KEY	ui2	Between 0 & 65535 inclusive	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
InternalPort	✓	PortMapping	ui2	Between 1 & 65535 inclusive	Refer to UPnP WANIPConnection v1.01 service definition for more details.	N/A	R
PortMappingProtocol	✓	PortMapping KEY	String	TCP UDP	Refer to UPnP WANIPConnection v1.01 service definition for more details.	Empty String	R
InternalClient	✓	PortMapping	String	IP Address	Refer to UPnP WANIPConnection v1.01 service definition for more details.	Empty String	R
PortMappingDescription	✓	PortMapping	String	String	Refer to UPnP WANIPConnection v1.01 service definition for more details.	Empty String	R

AddressingType	-	-	String	DHCP, Static	Represents the method used to assign an address to the WAN side interface of the IGD.	N/A	R
DefaultGateway	-	-	String	IP Address	The default gateway of the WAN interface. This variable will only be configurable from the LAN-side if the AddressingType is Static.	N/A	R
MACAddress	-	-	String	MAC Address	The physical address of the WANIPConnection if applicable.	N/A	R
MACAddressOverride	-	-	Boolean	1, 0	Whether the value of MACAddress state variable can be overridden. If true (1), the action SetMAC-Address may be used to set the value of MAC-Address. If false (0), the CPE's default value is used (or restored if it had previously been overridden).	N/A	O
MaxMTUSize	-	-	ui2	Between 1 and 1540, inclusive	The maximum allowed size of an Ethernet frame from LAN-side devices.	N/A	O
DNSEnabled	-	-	Boolean	0, 1	Defines whether or not the device should attempt to query a DNS server across this connection.	1	R
DNSOverrideAllowed	-	-	Boolean	0, 1	Defines whether or not a manually set, "non-zero" DNS address can be overridden by a DNS entry received from the WAN.	0	R
DNSServers	-	-	String	String	Comma separated list of DNS servers configured on the WANIPConnection.	N/A	R
ConnectionTrigger	-	-	String	OnDemand AlwaysOn Manual	Defines the trigger used to establish the IP connection.	OnDemand	R
RouteProtocolRx	-	-	String	Off, (R) RIPv1 (O) RIPv2, (O) OSPF (O)	Defines the Rx protocol to be used.	Off	R

Actions, Arguments & Errors

Name	From IGD	Argument	Dir	Related State Variable(s)	Description	Errors	R/O

SetEnable	-	NewEnable	IN	Enable	Sets the value of the Enable state variable to enable or disable the connection.	402, 501	R S
GetInfo	-	NewEnable NewConnectionType NewPossibleConnectionTypes NewConnectionStatus NewName NewUptime NewLastConnectionError NewAutoDisconnectTime NewIdleDisconnectTime NewWarnDisconnectDelay NewRSIPAvailable NewNATEnabled NewExternalIPAddress NewSubnetMask NewAddressingType NewDefaultGateway NewMACAddress NewMACAddressOverride NewMaxMTUSize NewDNSEnabled NewDNSOverrideAllowed NewDNSServers NewConnectionTrigger NewRouetProtocolRx	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT	Enable ConnectionType PossibleConnectionTypes ConnectionStatus Name Uptime LastConnectionError AutoDisconnectTime IdleDisconnectTime WarnDisconnectDelay RSIPAvailable NATEnabled ExternalIPAddress SubnetMask AddressingType DefaultGateway MACAddress MACAddressOverride MaxMTUSize DNSEnabled DNSOverrideAllowed DNSServers ConnectionTrigger RouteProtocolRx	Retrieves all of the state variables not associated with a table.	402, 501	R
SetConnectionType	✓	NewConnectionType	IN	ConnectionType	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501, 703	R S
GetConnectionTypeInfo	✓	NewConnectionType NewPossibleConnectionType	OUT OUT	ConnectionType PossibleConnectionTypes	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501	R
RequestConnection	✓	*None*	-	-	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 704, 705, 706, 707, 708, 709, 710	R S
RequestTermination	✓	*None*	-	-	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501, 707, 710, 711	O S
ForceTermination	✓	*None*	-	-	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501, 707, 710, 711	R S
SetAutoDisconnectTime	✓	NewAutoDisconnectTime	IN	AutoDisconnectTime	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501	O S
SetIdleDisconnectTime	✓	NewIdleDisconnectTime	IN	IdleDisconnectTime	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501	O S
SetWarnDisconnectDelay	✓	NewWarnDisconnectDelay	IN	WarnDisconnectDelay	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501	O S
GetStatusInfo	✓	NewConnectionStatus NewLastConnectionError NewUpTime	OUT OUT OUT	ConnectionStatus LastConnectionError UpTime	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402,	R
GetAutoDisconnectTime	✓	NewAutoDisconnectTime	OUT	AutoDisconnectTime	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402	O

GetIdleDisconnectTime	✓	NewIdleDisconnectTime	OUT	IdleDisconnectTime	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402	O
GetWarnDisconnectDelay	✓	NewWarnDisconnectDelay	OUT	WarnDisconnectDelay	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402	O
GetNATRSIPStatus	✓	NewRSIPAvailavle NewNATEnabled	OUT OUT	RSIPAvailavle NATEnabled	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402	R
PortMapping Table Actions							
GetPortMappingNumberOfEntries	-	NewPortMappingNumberOfEntries	OUT	PortMappingNumberOfEntries	Retrieves the value of the PortMappingNumberOfEntries state variable.	402, 501	R
GetGeneric-PortMappingEntry	-	NewPortMappingIndex NewRemoteHost NewExternalPort NewProtocol NewInternalPort NewInternalClient NewEnabled NewPortMappingDescription NewLeaseDuration	IN OUT OUT OUT OUT OUT OUT OUT OUT	PortMappingNumberOfEntries RemoteHost ExternalPort PortMappingProtocol InternalPort InternalClient PortMappingEnabled PortMappingDescription PortMappingLeaseDuration	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 713	R
GetSpecificPortMappingEntry	✓	NewRemoteHost NewExternalPort NewProtocol NewInternalPort NewInternalClient NewEnabled NewPortMappingDescription NewLeaseDuration	IN IN IN OUT OUT OUT OUT OUT	RemoteHost ExternalPort PortMappingProtocol InternalPort InternalClient PortMappingEnabled PortMappingDescription PortMappingLeaseDuration	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 714	R
AddPortMapping	✓	NewRemoteHost NewExternalPort NewProtocol NewInternalPort NewInternalClient NewEnabled NewPortMappingDescription NewLeaseDuration	IN IN IN IN IN IN IN IN	RemoteHost ExternalPort PortMappingProtocol InternalPort InternalClient PortMappingEnabled PortMappingDescription PortMappingLeaseDuration	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501, 715, 716, 718, 724, 725, 726, 727	R S
DeletePortMapping	✓	NewRemoteHost NewExternalPort NewProtocol	IN IN IN	RemoteHost ExternalPort PortMappingProtocol	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 714	R S
GetExternalIPAddress	✓	NewExternalIPAddress	OUT	ExternalIPAddress	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501	R
SetIPInterfaceInfo	-	NewAddressingType NewExternalIPAddress NewSubnetMask NewDefaultGateway	IN IN IN IN	AddressingType ExternalIPAddress SubnetMask DefaultGateway	Set the WAN side IP interface. NewExternalIPAddress, NewSubnetMask, NewDefaultGateway are ignored if NewAddressingType is not “Static.”	402, 501, 702	O S
SetMACAddress	-	NewMACAddress	IN	MACAddress	Sets the value of MAC-Address. Valid only if MAC-AddressOverride is present and true (1).	402, 501, 728	O S
SetMaxMTUSize	-	NewMaxMTUSize	IN	MaxMTUSize	Set the WAN side Max MTU size.	402, 501, 702	O S
SetConnectionTrigger	-	NewConnectionTrigger	IN	ConnectionTrigger	Sets the value of the Connection Trigger variable.	402, 501	R S

SetRouteProtocolRx	-	NewRouteProtocolRx	IN	RouteProtocolRx	Sets the value of the Rx route protocol.	402, 501	R S

6.8.5 WANPPPConnection

6.8.5.1 Overview

WANPPPConnection is a standard UPnP service that enables a UPnP control point to configure and control PPP connections on the WAN interface of a UPnP compliant IGD. This service is required if the WANConnection Device employs PPP connections. This service must NOT be active if the WANConnection Device does not support PPP connections. The allowed values of the PossibleConnectionType and ConnectionType state variables must reflect the supported ConnectionTypes of the device.

6.8.5.2 Service Modelling Definitions

ServiceType

urn:dslforum-org:service:WANPPPConnection:1

StateVariables

Variable Name	From IGD	Table	Data Type	Allowed Value	Description	Default Value	R/O
Enable	-	-	Boolean	1, 0	Enables or disables the connection.	N/A	R
ConnectionType	✓	-	String	One of PossibleConnectionTypes	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
PossibleConnectionTypes	✓	-	String	Unconfigured IP_Routed DHCP_Spoofed PPPoE_Bridged PPTP_Relay L2TP_Relay PPPoE_Relay	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
ConnectionStatus	✓	-	String	Unconfigured Connecting Authenticating Connected PendingDisconnect Disconnecting Disconnected	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
Name	-	-	String	String	Friendly Name of the connection. This name must be unique within the scope of the IGD.	N/A	R
Uptime	✓	-	ui4	>=0	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	(Seconds)	R
UpstreamMaxBitRate	✓	-	ui4	>=0	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	(bits/second)	R
DownstreamMaxBitRate	✓	-	ui4	>=0	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	(bits/second)	R
LastConnectionError	✓	-	String	One of the following: ERROR_NONE ERROR_ISP_TIME_OUT ERROR_COMMAND_ABORTED	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R R O O

				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	for more details.		O O O O O O O O O O O O R O O O O O O O R R
AutoDisconnectTime	✓	-	ui4	>=0	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	(Seconds)	O
IdleDisconnectTime	✓	-	ui4	>=0	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	(Seconds)	O
WarnDisconnectDelay	✓	-	ui4	>=0	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	(Seconds)	O
ConnectionTrigger	-	-	String	OnDemand AlwaysOn Manual	Defines the trigger used to establish the PPP connection.	OnDemand	R
RSIPAvailable	✓	-	Boolean	0, 1	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
NATEnabled	✓	-	Boolean	0, 1	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
UserName	✓	-	String	Alphanumeric Text	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	Empty String	R
Password	✓	-	String	Alphanumeric Text	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	Empty String	R
PPPEncryptionProtocol	✓	-	String	MPPE, ...	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	O
PPPCompressionProtocol	✓	-	String	STAC LZS Van Jacobsen	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	O
PPPAuthenticationProtocol	✓	-	String	PAP CHAP MS-CHAP	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	O
ExternalIPAddress	✓	-	String	IP Address	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R

RemoteIPAddress	-	-	String	IP Address	The remote IP address of the WANPPPConneciton.	N/A	O
MaxMRUSize	-	-	ui2	Between 1 and 1540, inclusive	The maximum allowed size of frames sent from the remote peer.	N/A	O
CurrentMRUSize	-	-	ui2	Between 1 and 1540, inclusive	The current MRU in use over this connection.	N/A	O
PortMappingNumberOfEntries	✓	PortMapping (index)	ui2	>=0	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
PortMappingEnabled	✓	PortMapping	Boolean	0,1	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
PortMappingLeaseDuration	✓	PortMapping	ui4	0 to maximum value of ui4 (in seconds)	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
RemoteHost	✓	PortMapping KEY	String	IP Address or empty string	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
ExternalPort	✓	PortMapping KEY	ui2	Between 1 and 65535 inclusive	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
InternalPort	✓	PortMapping	ui2	Between 1 and 65535 inclusive	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
PortMappingProtocol	✓	PortMapping KEY	String	TCP UDP	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
InternalClient	✓	PortMapping	String	IP Address	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
PortMappingDescription	✓	PortMapping	String	String	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	N/A	R
MACAddress	-	-	String	MAC Address	The physical address of the WANPPP-Connection if applicable.	N/A	R
MACAddressOverride	-	-	Boolean	1, 0	Whether the value of MACAddress state variable can be overridden. If true (1), the action SetMACAddress may be used to set the value of MACAddress. If false (0), the CPE's default value is used (or restored if it had previously been overridden).	N/A	O
DNSEnabled	-	-	Boolean	0, 1	Defines whether or not the device should attempt to query a DNS	1	R

					server across this connection.		
DNSOverrideAllowed	-	-	Boolean	0, 1	Defines whether or not a manually set, "non-zero" DNS address can be overridden by a DNS entry received from the WAN.	0	R
DNSServers	-	-	String	String	Comma separated list of DNS servers configured on the WANIPConnection.	N/A	R
TransportType	-	-	String	PPPoA PPPoE L2TP (for future use) PPTP (for future use)	PPP Transport type of the WANPPPConnection.	N/A	R
PPPoEACName	-	-	String	String	PPPoE Access Concentrator.	N/A	R
PPPoEServiceName	-	-	String	String	PPPoE Service Name.	N/A	R
RouteProtocolRx	-	-	String	Off, (R) RIPv1 (O) RIPv2, (O) OSPF (O)	Defines the Rx protocol to be used.	Off	R
PPPLCPEcho	-	-	ui2	-	PPP LCP Echo period in seconds.	N/A	O
PPPLCPEchoRetry	-	-	ui2	-	Number of PPP LCP Echo retries within an echo period.	N/A	O

Actions, Arguments & Errors

Name	From IGD	Argument	Dir	Related State Variable(s)	Description	Errors	R/O
SetEnable	-	NewEnable	IN	Enable	Sets the value of the Enable state variable to enable or disable the connection.	402, 501	R S
GetInfo	-	NewEnable	OUT	Enable	Retrieves all of the state variables not associated with a table. Also excludes Password, which is not readable.	402, 501	R
		NewConnectionType	OUT	ConnectionType			
		NewPossibleConnectionTypes	OUT	PossibleConnectionTypes			
		NewConnectionStatus	OUT	ConnectionStatus			
		NewName	OUT	Name			
		NewUptime	OUT	Uptime			
		NewUpstreamMaxBitRate	OUT	UpstreamMaxBitRate			
		NewDownstreamMaxBitRate	OUT	DownstreamMaxBitRate			
		NewLastConnectionError	OUT	LastConnectionError			
		NewAutoDisconnectTime	OUT	AutoDisconnectTime			
		NewIdleDisconnectTime	OUT	IdleDisconnectTime			
		NewWarnDisconnectDelay	OUT	WarnDisconnectDelay			
		NewConnectionTrigger	OUT	ConnectionTrigger			
		NewRSIPAvailable	OUT	RSIPAvailable			
		NewNATEnabled	OUT	NATEnabled			
		NewUserName	OUT	UserName			
		NewPPPEncryptionProtocol	OUT	PPPEncryptionProtocol			
		NewPPPCompressionProtocol	OUT	PPPCompressionProtocol			
		NewPPPAuthenticationProtocol	OUT	PPPAuthenticationProtocol			
		NewExternalIPAddress	OUT	ExternalIPAddress			
		NewRemoteIPAddress	OUT	RemoteIPAddress			
		NewMACAddress	OUT	MACAddress			
		NewMACAddressOverride	OUT	MACAddressOverride			
		NewMaxMRUSize	OUT	MaxMRUSize			
		NewCurrentMRUSize	OUT	CurrentMRUSize			
		NewDNSEnabled	OUT	DNSEnabled			
		NewDNSOverrideAllowed	OUT	DNSOverrideAllowed			
		NewDNSServers	OUT	DNSServers			
		NewTransportType	OUT	TransportType			
		NewPPPoEACName	OUT	PPPoEACName			
		NewPPPoEServiceName	OUT	PPPoEServiceName			
		NewRouteProtocolRx	OUT	RouteProtocolRx			
		NewPPPLCPEcho	OUT	PPPLCPEcho			
		NewPPPLCPEchoRetry	OUT	PPPLCPEchoRetry			

SetConnectionType	✓	NewConnectionType	IN	ConnectionType	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402, 501, 703	R S
GetConnectionTypeInfo	✓	NewConnectionType NewPossibleConnectionType	OUT OUT	ConnectionType PossibleConnectionTypes	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402, 501	R
RequestConnection	✓	*None*	-	-	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402, 704, 705 706, 707 708, 709 710	R S
RequestTermination	✓	*None*	-	-	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402, 501, 707, 710, 711	O S
ForceTermination	✓	*None*	-	-	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402, 501, 707, 710, 711	R S
SetAutoDisconnectTime	✓	NewAutoDisconnectTime	IN	AutoDisconnectTime	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402, 501	O S
SetIdleDisconnectTime	✓	NewIdleDisconnectTime	IN	IdleDisconnectTime	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402, 501	O S
SetWarnDisconnectDelay	✓	NewWarnDisconnectDelay	IN	WarnDisconnectDelay	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402, 501	O S
GetStatusInfo	✓	NewConnectionStatus NewLastConnectionError NewUpTime	OUT OUT OUT	ConnectionStatus LastConnectionError UpTime	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402,	R
GetLinkLayerMaxBitRates	✓	NewUpstreamMaxBitRate NewDownstreamMaxBitRate	OUT OUT	UpstreamMaxBitRate DownstreamMaxBitRate	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402,	R
GetPPPEncryptionProtocol	✓	NewPPPEncryptionProtocol	OUT	PPPEncryptionProtocol	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402,	O
GetPPPCompressionProtocol	✓	NewPPPCompressionProtocol	OUT	PPPCompressionProtocol	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402,	O
GetPPPAuthenticationProtocol	✓	NewPPPAuthenticationProtocol`	OUT	PPPAuthenticationProtocol	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402,	O
GetUserName	✓	NewUserName	OUT	UserName	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402,	R S
SetUserName	-	NewUserName	IN	UserName	Sets the value of the related state variable.	402, 501, 702, 703	R S
SetPassword	-	NewPassword	IN	Password	Sets the value of the related state variable.	402, 501, 702, 703	R S

GetAutoDisconnectTime	✓	NewAutoDisconnectTime	OUT	AutoDisconnectTime	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402	O
GetIdleDisconnectTime	✓	NewIdleDisconnectTime	OUT	IdleDisconnectTime	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402	O
GetWarnDisconnectDelay	✓	NewWarnDisconnectDelay	OUT	WarnDisconnectDelay	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402	O
GetNATRSIPStatus	✓	NewRSIPAvailavle NewNATEnabled	OUT OUT	RSIPAvailavle NATEnabled	Refer to UPnP WANPPPConnection v1.01 service definition for more details.	402	R
SetPPPoEService	-	NewPPPoEACName NewPPPoEServiceName	IN IN	PPPoEACName PPPoEServiceName	Sets the values of the PPPoEACName and PPPoEServiceName state variables.	402, 501	O S
SetConnectionTrigger	-	NewConnectionTrigger	IN	ConnectionTrigger	Sets the value of the Connection Trigger variable.	402, 501	R S
PortMappingTable Actions							
GetPortMappingNumberOfEntries	-	NewPortMappingNumberOfEntries	OUT	PortMappingNumberOfEntries	Retrieves the value of the PortMappingNumberOfEntries state variable.	402, 501	R
GetGeneric-PortMappingEntry	-	NewPortMappingIndex NewRemoteHost NewExternalPort NewProtocol NewInternalPort NewInternalClient NewEnabled NewPortMappingDescription NewLeaseDuration	IN OUT OUT OUT OUT OUT OUT OUT OUT	PortMappingNumberOfEntries RemoteHost ExternalPort PortMappingProtocol InternalPort InternalClient PortMappingEnabled PortMappingDescription PortMappingLeaseDuration	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 713	R
GetSpecificPortMappingEntry	✓	NewRemoteHost NewExternalPort NewProtocol NewInternalPort NewInternalClient NewEnabled NewPortMappingDescription NewLeaseDuration	IN IN IN OUT OUT OUT OUT OUT	RemoteHost ExternalPort PortMappingProtocol InternalPort InternalClient PortMappingEnabled PortMappingDescription PortMappingLeaseDuration	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 714	R
AddPortMapping	✓	NewRemoteHost NewExternalPort NewProtocol NewInternalPort NewInternalClient NewEnabled NewPortMappingDescription NewLeaseDuration	IN IN IN IN IN IN IN IN	RemoteHost ExternalPort PortMappingProtocol InternalPort InternalClient PortMappingEnabled PortMappingDescription PortMappingLeaseDuration	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501, 715, 716, 718, 724, 725, 726, 727	R S
DeletePortMapping	✓	NewRemoteHost NewExternalPort NewProtocol	IN IN IN	RemoteHost ExternalPort PortMappingProtocol	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 714	R S
GetExternalIPAddress	✓	NewExternalIPAddress	OUT	ExternalIPAddress	Refer to UPnP WANIPConnection v1.01 service definition for more details.	402, 501	R
SetMACAddress	-	NewMACAddress	IN	MACAddress	Sets the value of MAC-Address. Valid only if MAC-AddressOverride is present and true (1).	402, 501, 728	O S
SetMaxMRUSize	-	NewMaxMRUSize	IN	MaxMRUSize	Set the WAN side Max MRU size.	402, 501, 702	O S

SetRouteProtocolRx	-	NewRouteProtocolRx	IN	RouteProtocolRx	Sets the value of the Rx route protocol.	402, 501	R S

6.9 Error Codes

errorCode	errorDescription	Description
401	Invalid Action	No action by that name at this service.
402	Invalid Args	Could be any of the following: not enough in args, too many in args, no in arg by that name, one or more in args are of the wrong data type
403	Out of sync	Out of synchronization (deprecated).
501	Action Failed	May be returned if current state of service prevents invoking that action.
600	Argument Value Invalid	The argument value is invalid.
601	Argument Value Out of Range	An argument value is less than the minimum or more than the maximum value of the allowedValueRange, or is not in the allowedValueList.
602	Optional Action Not Implemented	The requested action is optional and is not implemented by the device.
603	Out of Memory	The device does not have sufficient memory available to complete the action. This may be a temporary condition; the control point may choose to retry the unmodified request again later and it may succeed if memory is available.
604	Human Intervention Required	The device has encountered an error condition which it cannot resolve itself and required human intervention such as a reset or power cycle. See the device display or documentation for further guidance.
605	String Argument Too Long	A string argument is too long for the device to handle properly.
606	Action Not Authorized	The action requested requires authorization and the sender was not authorized.
701	ValueAlreadySpecified	The value specified in the action is already available in the list and is consequently ignored.
702	ValueSpecifiedIsInvalid	The specified value is not present in the list.
703	InactiveConnectionStateRequired	Current value of ConnectionStatus should be either Disconnected or Unconfigured to permit this action.
704	ConnectionSetupFailed	There was a failure in setting up the IP or PPP connection with the service provider.
705	ConnectionSetupInProgress	The connection is already in the process of being setup.
706	ConnectionNotConfigured	Current ConnectionStatus is Unconfigured.
707	DiscconectInProgress	The connection is in the process of being torn down.
708	InvalidLayer2Address	Corresponding Link Config service has an invalid VPI/VCI or phone number.
709	InternetAccessDisabled	The EnabledForInternet flag is set to 0.
710	InvalidConnectionType	This action is not permitted for the specified ConnectionType.
711	ConnectionAlreadyTerminated	An attempt was made to terminate a connection that is no longer active.
712	NullValueAtSpecifiedArrayIndex	The specified array index holds a NULL value.
713	SpecifiedArrayIndexInvalid	The specified array index is out of bounds.

errorCode	errorDescription	Description
714	NoSuchEntryInArray	The specified value does not exist in the array.
715	WildCardNotPermittedInSrcIP	The source IP address cannot be wild-carded.
716	WildCardNotPermittedInExtPort	The external port cannot be wild-carded.
718	ConflictInMappingEntry	The port mapping entry specified conflicts with a mapping assigned previously to another client.
719	ActionDisallowedWhenAutoConfigEnabled	The specified action is not permitted when auto configuration is enabled on the modem.
720	InvalidDeviceUUID	The UUID of a device specified in the action arguments is invalid.
721	InvalidServiceID	The Service ID of a service specified in the action arguments is invalid.
723	InvalidConnServiceSelection	The selected connection service instance cannot be set as a default connection.
724	SamePortValuesRequired	Internal and External port values must be the same.
725	OnlyPermanentLeasesSupported	The NAT implementation only supports permanent lease times on port mappings.
726	RemoteHostOnlySupportsWildcard	RemoteHost must be a wildcard and cannot be a specific IP address or DNS name.
727	ExternalPortOnlySupportsWildcard	ExternalPort must be a wildcard and cannot be a specific port value.
728	InvalidChannel	The requested Channel is not specified in the PossibleChannels.
729	InvalidMACAddress	BSSID is an invalid MAC address.
730	InvalidDataTransmissionRates	The requested DataTransmissionRates are not specified in the PossibleDataTransmissionRates.
731	InvalidWEPKey	WEPKey contains invalid characters or its length does not correspond to a supported encryption level.
732	NoWEPKeyIsSet	The complete WEP key set is empty (and EAP WEP re-keying is disabled).
733	NoPSKKeyIsSet	The complete PSK key set is empty.
734	NoEAPServer	No EAP server exists for EAP Authentication.
800-896	TBD	Specified by vendor.
897	SetMACAddressNotPermitted	The SetMACAddress action is not permitted because SetMACAddressOverride is not set.
898	WriteAccessDisabled	LAN-side write-access to one or more variables the action attempted to modify has been disabled (such as via the CPE WAN Management Protocol).
899	SessionIDExpired	Indicates that the session ID being used by the control point is no longer valid, either because of device reboot or because expiration of the device timer.

7 Events

No events are required or supported since use of this DSL CPE management method does not require constant always-on monitoring.

8 Glossary

ACS	Auto Configuration Server (TR-069 term)
AP	Access Point (UPnP term for wireless access point)
CP	Control Point (UPnP term)

CPE	Customer Premises Equipment
CWMP	CPE WAN Management Protocol
DCP	Device Control Protocol
DNS	Domain Name Server
DHCP	Dynamic Host Control Protocol
DSL	Digital Subscriber Loop
DSLAM	Digital Subscriber Loop Access Multiplexer
GUI	Graphical User Interface
HTTP	Hyper Text Transmission Protocol
IGD	Internet Gateway Device
IP	Internet Protocol
LAN	Local Area Network
MAC	Medium Access Control
NAT	Network Address Translation
OS	Operating System
PPP	Point-to-Point Protocol
RGW	Residential Gateway
SOAP	Simple Object Access Protocol
SSDP	Simple Service Discovery Protocol
TBD	To Be Determined
UDA	UPnP Device Architecture
UPnP	Universal Plug ‘n’ Play
VCI	Virtual Circuit Identifier
VPI	Virtual Path Identifier
XML	Extensible Mark-up Language

9 Normative References

[1] UPnP InternetGatewayDevice, Template Versions 1.01, For Universal Plug and Play Version 1.0, November 12, 2001, http://www.upnp.org/download/UPnP_IGD_DCP_v1.zip.

[2] UPnP Device Architecture, Version 1.0.1, May 6, 2003, http://www.upnp.org/download/Clean%20UPnPDA101-20030506.doc.

[3] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, H. Frystyk Nielsen “Simple Object Access Protocol (SOAP) 1.1”, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/, W3C Note, May 8, 2000.

[4] UPnP WLAN configuration service http://www.upnp.org/download/WLANConfigurationService%201.0.pdf

Appendix A Use Cases

In most deployments the CPE may require little or no user configuration. First, let's assume that the CPE is a basic layer 2 bridge. PPPoE client software may also be used on the PC. Initial installation would consist of the following sequence - the CPE powers-up, runs POST/diagnostics, initializes interfaces, and trains to the DSL line. The user then places a CD supplied by their service provider in their PC. The CPE management app on CD auto-runs, discovers the CPE and queries the CP for model and version. After the CPE is verified as supported, the CPE management app prompts the user for a username and password. After the user enters a username and password, the management app pushes the new username and password to the CPE. Once the username and password are set, the management app reads the CPE configuration from a service provider specific XML file on the installation CD and pushes the CPE configuration down to the CPE and sets a configuration flag. When this flag is set it identifies that the CPE has been configured. If the CPE is a router, the management app may present the user with a button to press if further CPE configuration via the user GUI is desirable. This button launches the web browser and displays the router main GUI screen. The following use case, Figure 10, graphically shows the process for CPE installation and turn-up. After initial configuration, the CPE management app may be used to query configuration or check CPE status. This is shown in Figure 11.

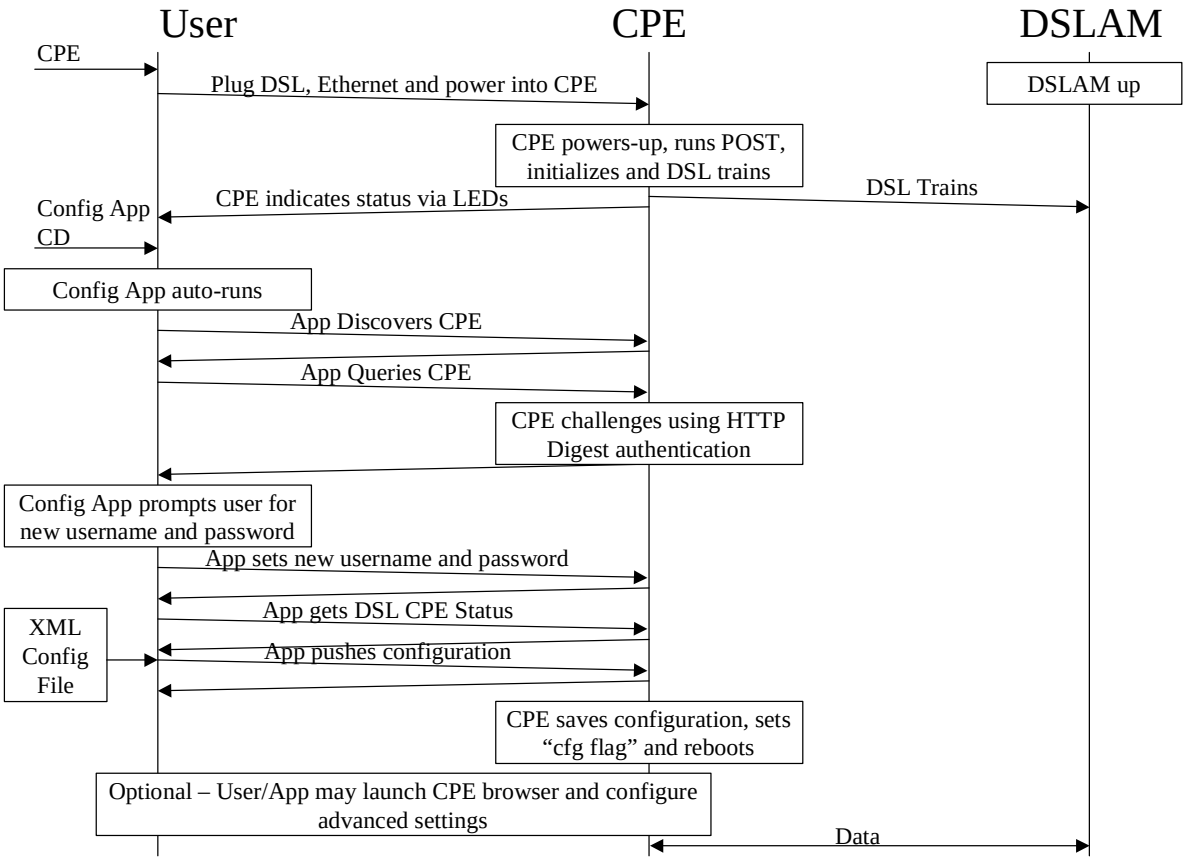


Figure 10: CPE Turn-up Use Case

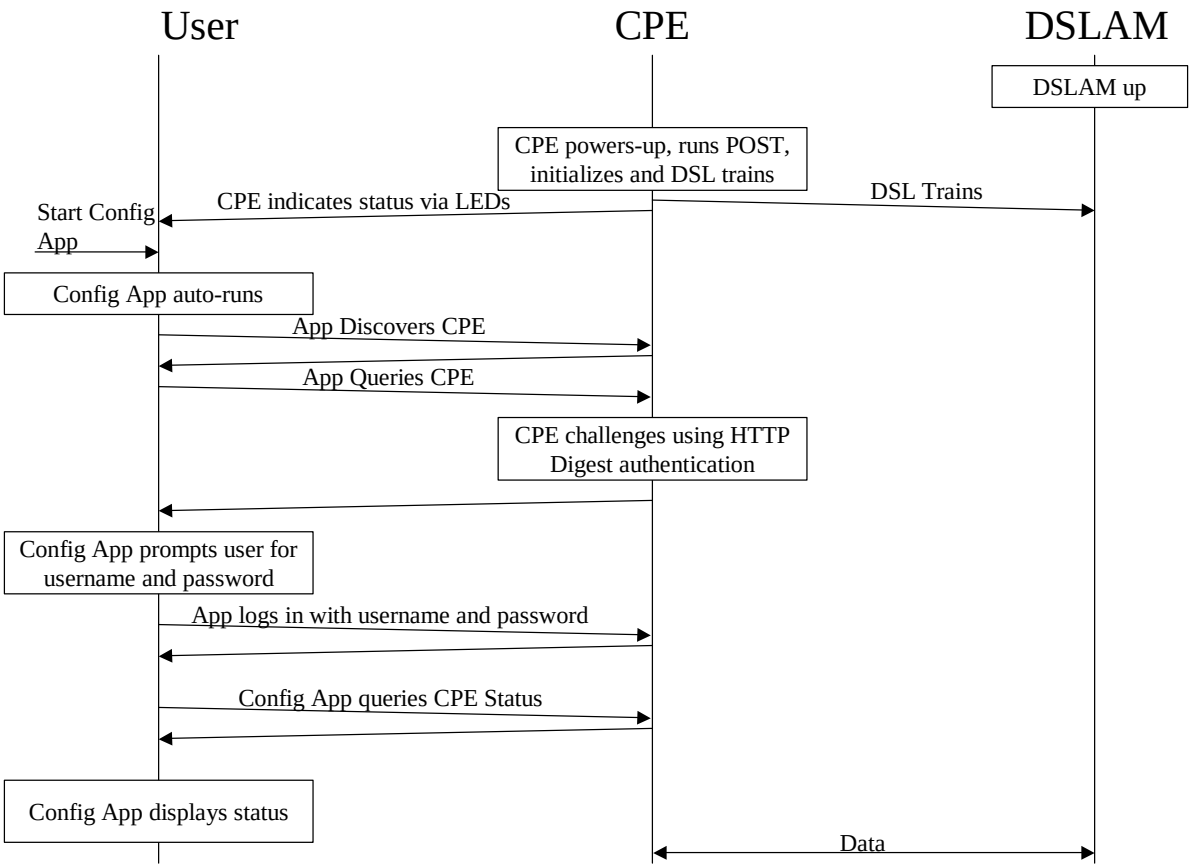


Figure 11: CPE Status Use Case

Appendix B WANDSLConnectionManagement Explained

Depending on whether WAN**Connection service is created in an existing WANConnectionDevice or a new instance of WANConnectionDevice, different actions are used. Respectively AddConnectionService and AddConnectionDeviceAndService.

Deletion of a WAN**Connection service is done using the DeleteConnectionService action.

Using any of those actions modifies the structure of the DDD; a Control Point would need to rediscover the device to learn of the new structure in the DDD.

AddConnectionService:

This action creates a new WAN**Connection service inside an existing instance of WANConnectionDevice, meaning using the same DestinationAddress (PVC, SVC…) and LinkType (EoA, IPoA, PPPoA, CIP, PPPoE…).

AddConnectionDeviceAndService:

This action creates first a new instance of WANConnectionDevice using DestinationAddress (PVC, SVC…), and LinkType (EoA, IPoA, PPPoA, CIP, PPPoE…) to fill in the WANDSLLinkConfig service of the new device.

After creating the WANConnectionDevice instance (including the WANDSLLinkConfig service), a WAN**Connection is created according to the provided ConnectionType (IP_Routed, IP_Bridged, DHCP_Spoofed, PPPoE_Bridged, PPTP_Relay, L2TP_Relay, PPPoE_Relay):

The following matrix shows how to choose between creating of a WANIPConnection or a WANPPPConnection service (this applies only for DSL modems). This differs from the UPnP model where a LinkType of PPPoE is allowed. Due to the requirement of some service providers to be able to support both an IP and PPPoE connection simultaneously on the same ATM PVC, the PPPoE option has been removed. PPPoE is clearly implied by having a WANPPPConnection with an EoA LinkType.

LinkType / ConnectionType	EoA	IPoA	PPPoA	CIP	PPPoE	Unconfigured
IP_Routed	**IP or PPP**	**IP**	**PPP**	**IP**	*-Forbidden-*	*-Forbidden-*
IP_Bridged	**IP**	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*
DHCP_Spoofed	**PPP**	*-Forbidden-*	**PPP**	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*
PPPoE_Bridged	**PPP**	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*
PPTP_Relay	**PPP**	*-Forbidden-*	**PPP**	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*
L2TP_Relay	**PPP**	*-Forbidden-*	**PPP**	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*
PPPoE_Relay	*-Forbidden-*	*-Forbidden-*	**PPP**	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*
Unconfigured	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*	*-Forbidden-*

If the provided pair (ConnectionType, LinkType) is not allowed (see –*Forbidden*- in previous table), error code 702 (ValueSpecifiedIsInvalid) is returned.

DeleteConnectionService:

This action deletes the specified instance of WAN**Connection service. The service can be deleted if it is not connected.

When trying to delete a service that is connected, the IGD returns error code 703 (InactiveConnectionStateRequired).

If the deleted service is the last WAN**Connection service instance in the last WAN**Connection service of the WANConnectionDevice instance, the WANConnectionDevice instance is automatically deleted.

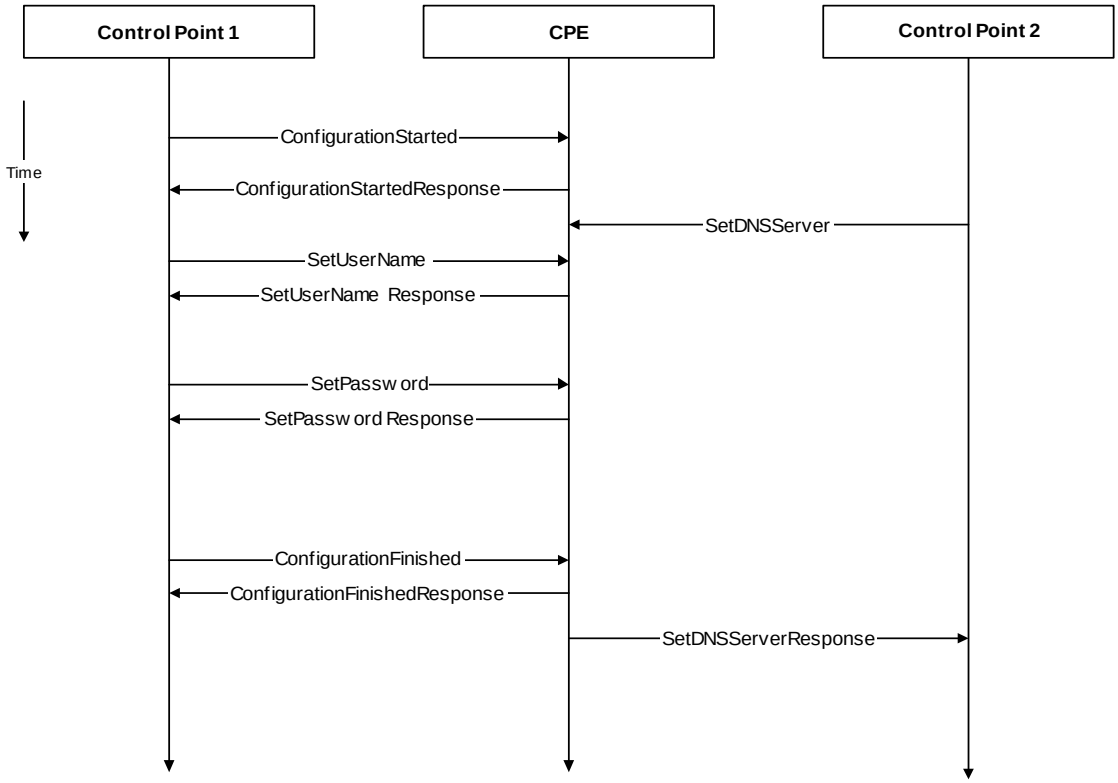
WAN**Connection services having AutoConfig state variable set to 1 cannot be deleted using this command.

Addition and Deletion of WAN Connection Services Due to non-XML Activities**

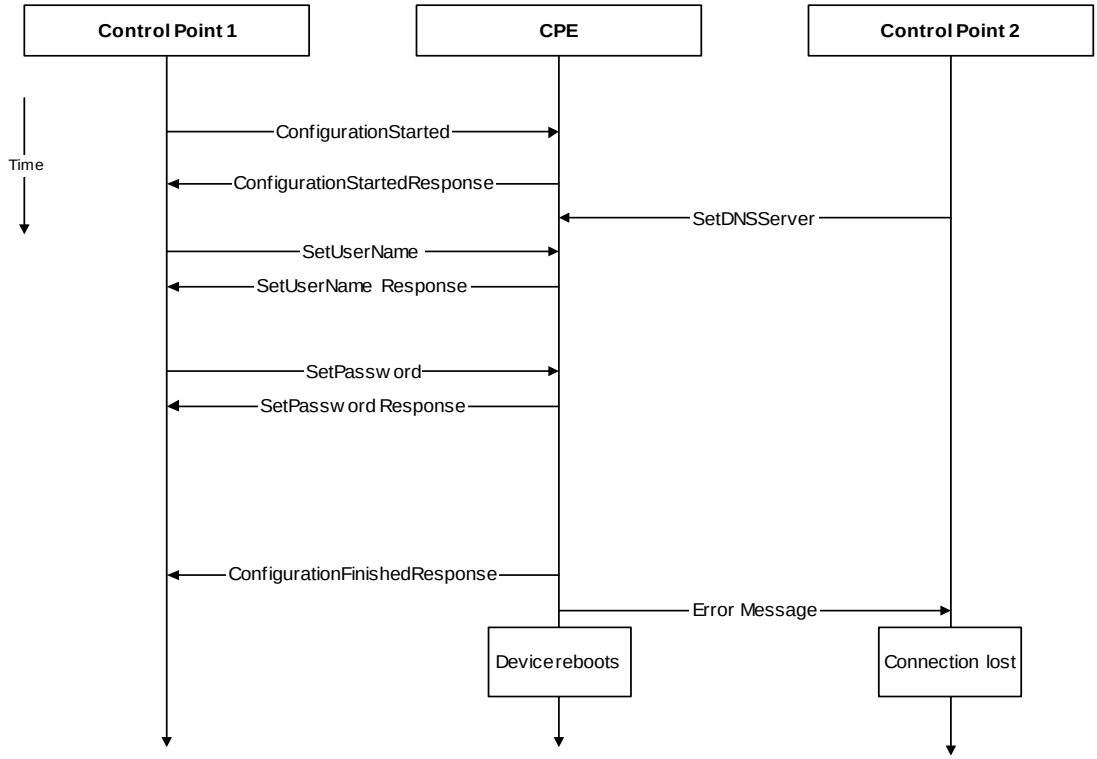
It should be noted that service instances can be automatically generated by the device. For example, if a user configures a new connection through an HTML interface, the device should automatically create a service instance. When the user enables bridging of PPPoE, this should create a “PPPoE_Bridged” WANPPPConnection service.

Appendix C Concurrency Diagrams and Sample Code

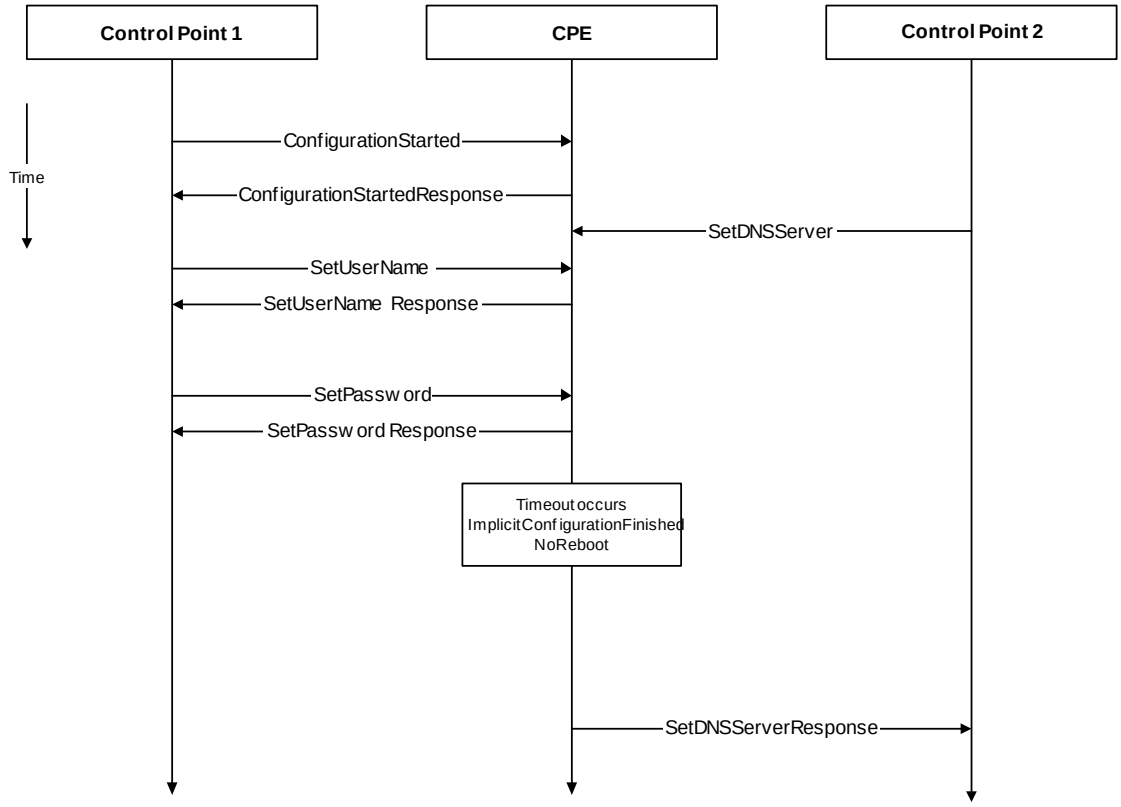
The first diagram represents the scenario in which Control Point one is making changes to the CPE and does not need a reboot to commit the changes.



The next diagram represents the scenario in which the CPE must reboot to apply the changes.



The next diagram illustrates the timeout for a CPE that requires no reboot:



Following is example code implementing the session ID.

Example ConfigurationStarted command with SessionID as an argument

```
POST path of control URL HTTP/1.1
HOST: host of control URL:port of control URL
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
SOAPACTION: "urn:dslforum-org:service:DeviceConfig:1#ConfigurationStarted"
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
<u:ConfigurationStarted xmlns:u="urn:dslforum-org:service:DeviceConfig:1">
<NewSessionID>2C0A2110-30BA-444e-8B83-566BC3F19C80</NewSessionID>
</u: ConfigurationStarted >
</s:Body>
</s:Envelope>
```

Example ConfigurationStarted Response

```
HTTP/1.1 200 OK
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
DATE: when response was generated
EXT:
SERVER: OS/version UPnP/1.0 product/version
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
<u:ConfigurationStartedResponse xmlns:u="dslforum-org:service:DeviceConfig:1">
</u:ConfigurationStartedResponse>
</s:Body>
</s:Envelope>
```

Example SOAP call using SessionID

```
POST path of control URL HTTP/1.1
HOST: host of control URL:port of control URL
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
SOAPACTION: "urn:dslforum-org: WANPPPConnection:1#SetUserName "
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Header>
<dslf:SessionID xmlns:dslf="dslforum-org"
s:mustUnderstand="1">2C0A2110-30BA-444e-8B83-566BC3F19C80</dslf:SessionID>
</s:Header>
<s:Body>
<u:SetUserName  xmlns:u="urn:dslforum-org:service:WANPPPConnection:1">
<NewUserName>fastconnection</NewUserName>
</u:SetUserName >
</s:Body>
</s:Envelope>
```

Response

```
HTTP/1.1 200 OK
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
DATE: when response was generated
EXT:
SERVER: OS/version UPnP/1.0 product/version
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
<u:SetUserNameResponse xmlns:u="urn:dslforum-org:service:WANPPPConnection:1">
</u:SetUserNameResponse>
</s:Body>
</s:Envelope>
```

Example ConfigurationFinsished method

POST path of control URL HTTP/1.1
HOST: host of control URL:port of control URL
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
SOAPACTION: "urn:dslforum-org:service: DeviceConfig:1#ConfigurationFinished"
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Header>
<dslf:SessionID xmlns:dslf="http://dslforum-org" s:mustUnderstand="1">2C0A2110-30BA-444e-8B83-566BC3F19C80</dslf:SessionID>
</s:Header>
<s:Body>
</s:Body>
</s:Envelope>

Response

HTTP/1.1 200 OK
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
DATE: when response was generated
EXT:
SERVER: OS/version UPnP/1.0 product/version
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body>
<u:ConfigurationFinishedResponse xmlns:u="urn:dslforum-org:service:DeviceConfig:1">
<NewStatus>RebootRequired</NewStatus>
</u:ConfigurationFinishedResponse>
</s:Body>
</s:Envelope>