IEEE1905 User Guide

For BCM96x DSL/PON Linux

Version 1.0

5300 California Avenue • Irvine, CA 92617-7013 USA • Phone: (949) 926-5000 • Fax: (949) 926-5203

Table of Contents

IEEE1905 User Guide

REVISION HISTORY

Revision Number	Date	Change Description
V1.0	06/27/2013	Initial Release.
V1.1	06/03/2014	Added details regarding releases and compile instructions (sections 4 and 5)

1.0 INTRODUCTION

The IEEE 1905.1 standard introduces a number of features such as Topology Discovery, Security Setup, AP Auto-Configuration and Link Metrics that facilitate management of the Home network. These features offer a set of standardized tools on top of which non-standard features such as link aggregation, best path selection and failover can be implemented.

This document addresses the implementation of the IEEE 1905.1 standard features. It provides a description of the architecture in a Linux environment, the features implemented, compile instructions, command line utility examples, a description of the internal data-model and its mapping to the IEEE 1905.1 standard data-model and finally, a description of the API interface.

Non-standard features that are built on top of IEEE 1905.1, such as link aggregation, best path selection and failover will be addressed in a separate document.

2.0 FEATURE DESCRIPTIONS

The BRCM IEEE1905.1 product offers the following features: Topology Discovery, Security Setup, AP Auto-Configuration and Link Metrics.

2.1 Topology Discovery

The Topology Discovery feature allows identifying all 1905 enabled devices present on the LAN network, including those that are multiple hop away, using a device unique 1905 MAC address. This discovery is performed at layer 2, allowing to determine what technology is used to interconnect each node (Wi-Fi, Ethernet, Homeplug, MoCA) and how exactly each nodes are connected. The diagram below is an example of a LAN network containing 7 devices supporting IEEE 1905.1. The blue circles correspond to the 1905 enabled devices found on the network, and the yellow circles correspond to the interfaces for each device. When two yellow circles are connected to each other, it means that the two 1905 devices can see each other through the connected interfaces and therefore, are connected to each other. The interfaces seen in the middle that form a pentagram are the PLC interfaces which can all see each other. This diagram is drawn from the Topology data model gathered at one of the 1905 devices in the network. All devices build the same data model independently.

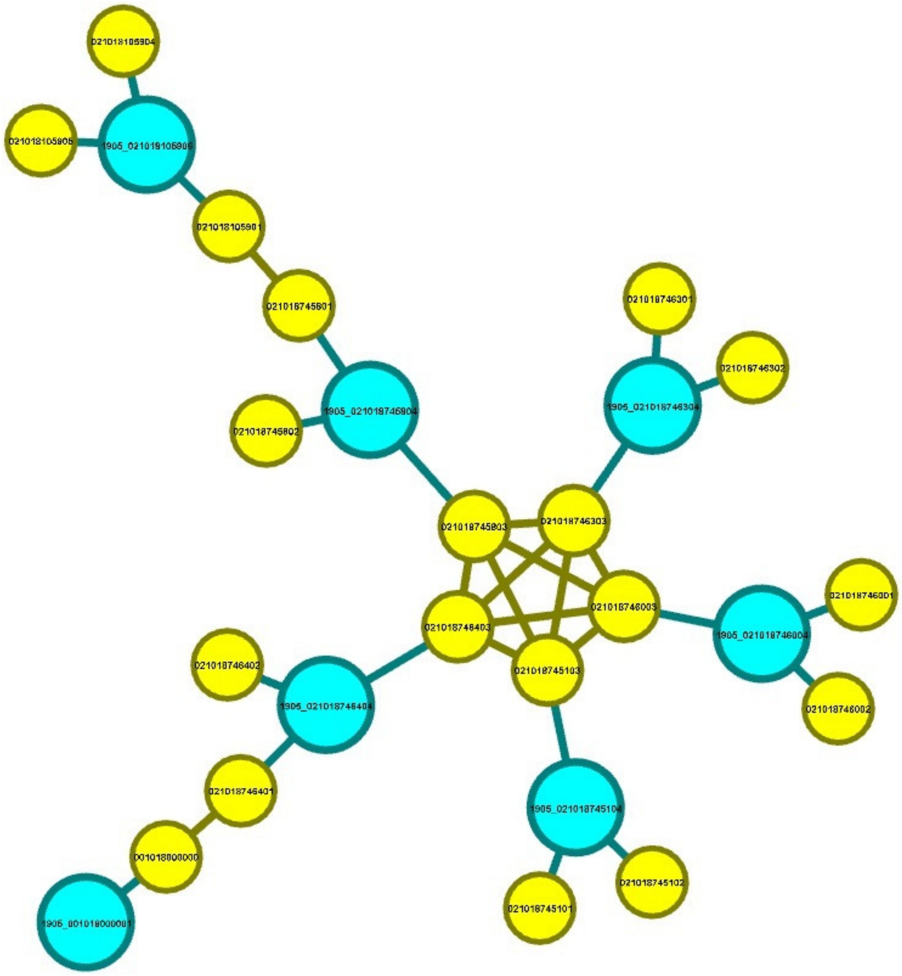


Figure 1. Topology Discovery Diagram for seven IEEE 1905.1 nodes

2.2 Security Setup

IEEE 1905.1 defines 3 methods for configuring security. One method is based on the Security Push button, one is based on an IEEE 1905.1 passphrase, and one is based on Near Field Communication (NFC), the last one being optional.

2.2.1 Security Push-Button

To understand how the security push button works, one needs to be familiar with the Wi-Fi WPS push button feature, or the HomePlug UKE push button features. In these technologies, to admit a new device in a secure network, the WPS algorithm needs to be triggered on two devices, one that is already present in the secure network, and one for the device being added to the network. To trigger the algorithm, a push button can be pressed on a device that is already part of the secure network, and another push button can be pressed on the device being added. Some devices which have a rich graphical user interface replace the physical push button with a menu item or a wizard that triggers the algorithm locally and inform the user to press the appropriate push button, generally on the gateway device.

The 1905 Security Push Button feature works similarly to the WPS push button in Wireless or to the UKE Security Push-Button in HomePlugAV, but offers additional benefits. In 1905, the WPS and UKE push buttons on a device are combined into a single Security Push Button. When pressed, the security protocol of each supported technology is triggered (WPS and UKE). There is no need to determine which of many push buttons need to be pressed on an existing or new device to start admission.

Since the 1905 Security push button simply triggers the underlying technology security mechanism, the 1905 Security Push Button mechanism remains compatible with existing devices. It is then possible to admit non-1905 Wi-Fi or HomePlug devices into the secure network using the push button or existing wizard GUI on these non-1905 devices.

Another benefit of the IEEE 1905.1 Security Push Button mechanism is that one can admit a new device supporting Wi-FI or HomePlug (or other future technology with a standardized push-button mechanism) by pressing the 1905 Security push button of any already admitted 1905 device in the network, irrespective of which technology that device supports. The device on which the push button is being pressed will trigger the admission protocol for the technologies it supports locally, but will inform other devices already admitted in the 1905 network that its push button has been pressed and to initiate the admission protocol on other technologies where it has not been initiated yet. This allows for example to start the Wi-Fi WPS algorithm from a device that only supports HomePlug and Ethernet and that is already part of the 1905 network. This device will inform other 1905 devices that the security push button has been pressed and a device receiving this message and supporting Wi-Fi WPS will trigger the algorithm.

2.2.2 1905 Passphrase

The IEEE 1905.1 standard also supports the ability to program a security passphrase directly into each IEEE 1905.1 device. However, this method has multiple issues so it is not currently implemented in Broadcom's 1905 solution. The main issue is that the IEEE 1905.1 passphrase is not compatible with the existing Wi-Fi passphrase security mechanism. If a passphrase is entered in a 1905 device, the same passphrase cannot be used in client such as a laptop wanting to join the 1905 secure network. This client would need to support special software to support the 1905 passphrase AND the "legacy" Wi-Fi security passphrase. This can quickly become unmanageable. If a solution to this problem can be found and explained, supporting this feature will be possible.

2.2.3 Near Field Communication (NFC)

The NFC feature allows approaching two devices supporting NFC in order to pair them. This feature is optional in IEEE 1905.1 and is currently not supported in the Broadcom 1905 software.

2.3 AP Auto-Configuration

The IEEE 1905.1 AP Auto-Configuration feature allows transferring the Wi-Fi Access Point Security configuration, typically from the Gateway, to a remote AP that being added to the 1905 secure network. After transfer, the new AP shares SSID, security setup, passphrase and other additional settings with the original AP, thus enables seamless roaming of Wi-Fi devices from one AP to the other.

The AP Auto-Configuration feature is triggered immediately after a new un-configured AP joins the secure 1905 network and finds a configured device with Wi-Fi AP support. There is no need to press the security push button to trigger the AP Auto-Configuration feature. The push-button is used to join a new device to the network and immediately after joining, the new un-configured device starts the AP Auto-Configuration protocol. In the case of two devices connected over Ethernet, there is no need to press a security push-button to join a new device to the 1905 network, so AP Auto-Configuration will trigger immediately after connecting Ethernet without pressing the Security push button.

AP Auto-Configuration requires one un-configured device, and at least one configured device. If more than one configured device exists with different settings, the un-configured device will randomly adopt the settings of either configured device. If multiple configured devices already share the same settings, a new un-configured device will get the same settings from either configured devices.

2.4 Link Metrics

IEEE 1905.1 Link metrics allow understanding the state of the network for management purposes, and can also be used to make bridging decisions by higher level software. There are really two types of link metrics. One consists of local link metrics, which are computed for each of the local interfaces of a 1905 device, and there are remote link metrics that can be gathered by requesting the information from neighbor 1905 devices. The local link metrics do not require transferring packets between 1905 nodes to be obtained and therefore can be collected frequently. The remote link metrics require sending requests and waiting for responses from neighbor nodes and therefore are obtained in a less timely manner.

The IEEE 1905.1 standard defined multiple link metrics. For the moment, the Broadcom 1905 software implements two of them (others will be added as the software development progresses):

macThroughputCapacity: The maximum MAC throughput of the Link estimated at the transmitter and expressed in Mb/s

linkAvailability: The estimated average percentage of time that the link is available for data transmissions.

Inside the Broadcom 1905 software, the linkAvailability is actually saved as macThroughputLinkAvailability in Mb/s instead of a percentage. Thus,

linkAvailability = macThroughputLinkAvailability / macThroughputCapacity.

3.0 ARCHITECTURE

The Broadcom IEEE 1905.1 software is implemented mostly as a Linux userspace application daemon which is called ieee1905. It generates and transmits packets on each of the device LAN interfaces and processes these packets when received. For certain features, such as Topology Discovery, the ieee1905 daemon can operate completely autonomously without interacting with other components in the system beside standard Linux sockets. For other features, such as Security Setup, AP Auto-Configuration and Link Metrics, the ieee1905 daemon must interact with each LAN interface drivers to provide the required functionality.

Non-standard features that are built on top of IEEE 1905.1, such as link aggregation, best path selection and failover have more impact on the architecture and are not discussed here.

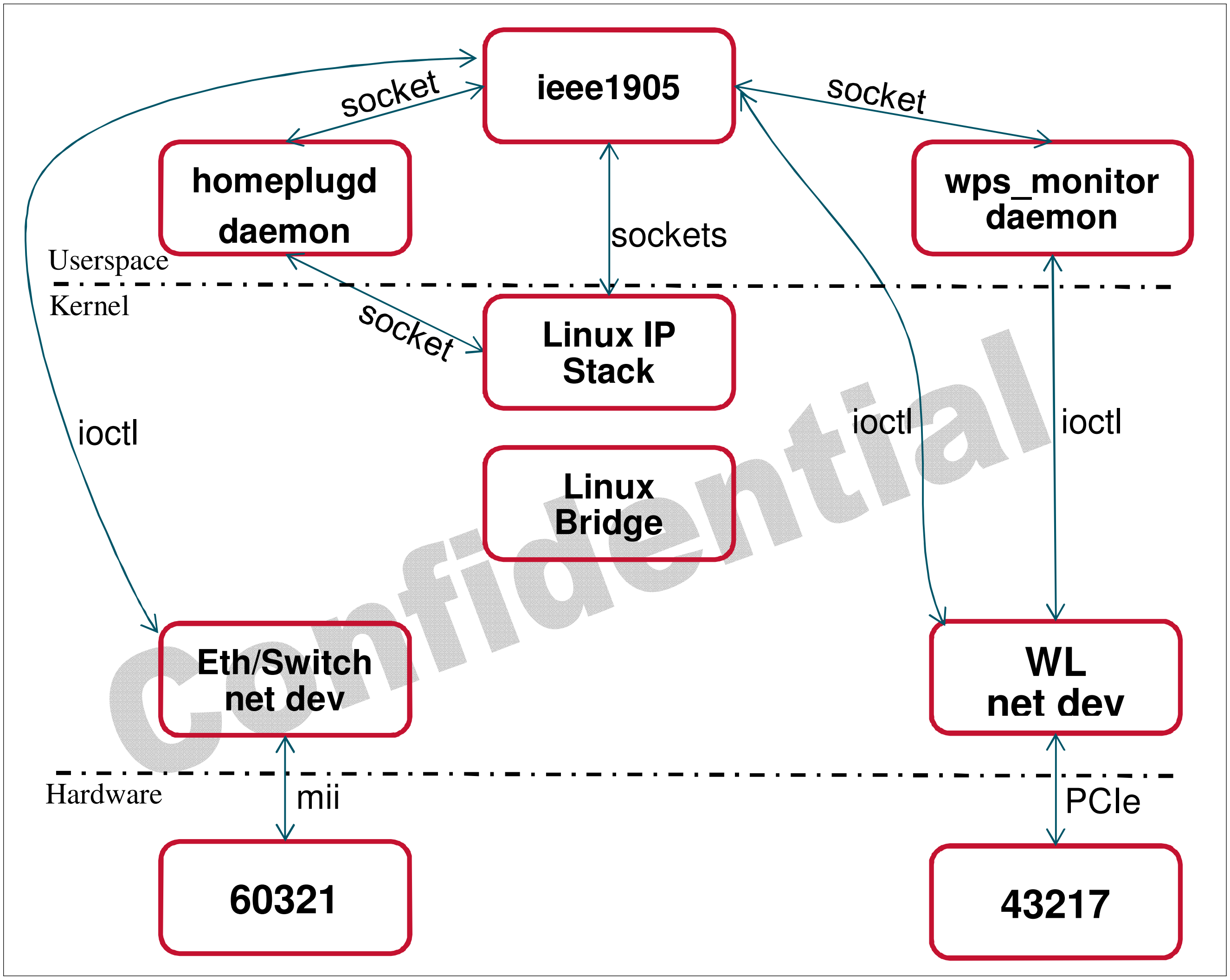


Figure 2. IEEE 1905 Control Path Block Diagram

The data path for IEEE 1905 protocol packets is shown in the next figure. Packets are sent to each interface independently in order to allow the gathering of a clear picture of where each packet is sent and received from. By sending the packets directly to each interface through Linux, it becomes possible to create a complete topology map of the network.

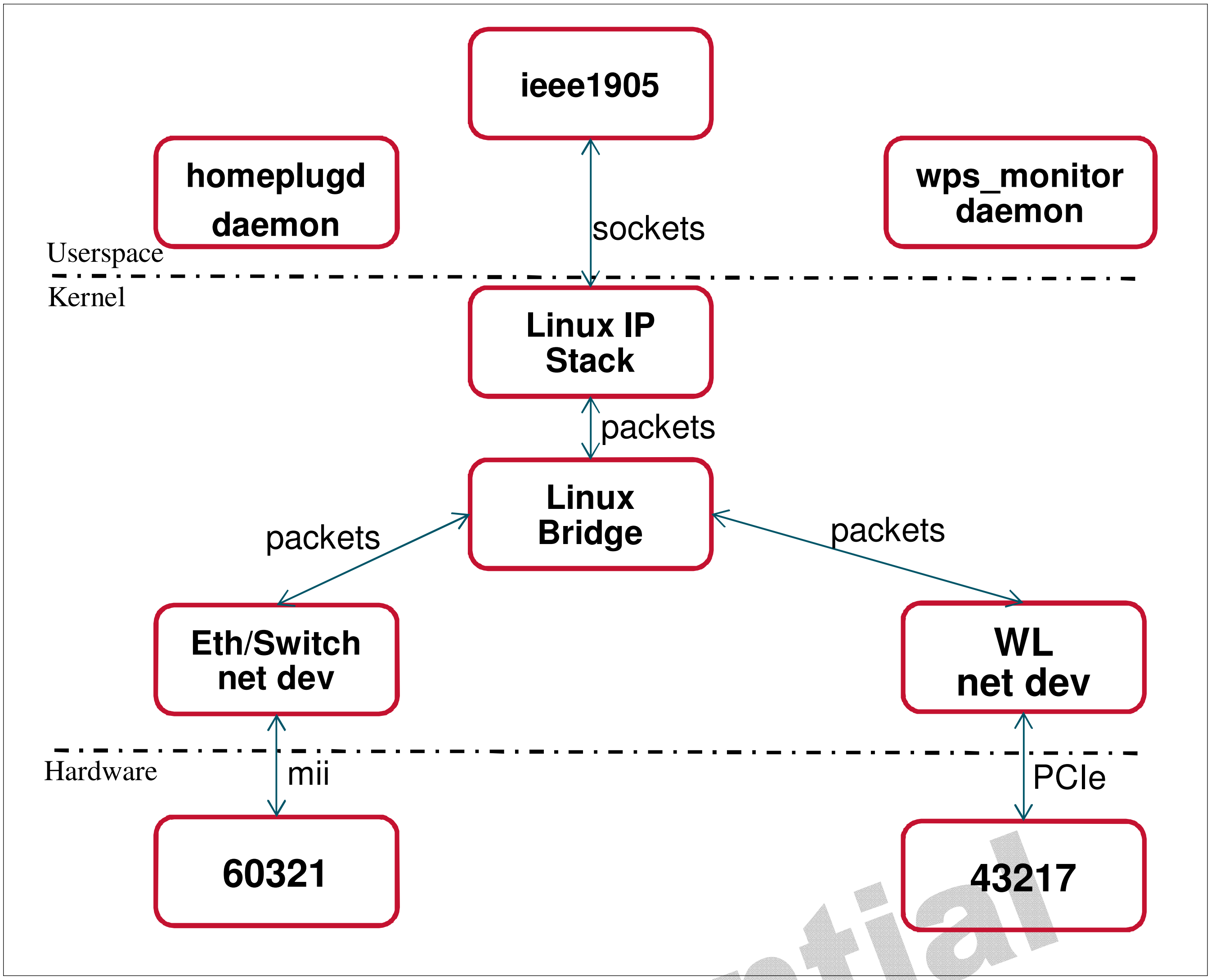


Figure 3. IEEE 1905 Protocol Data Path Block Diagram

Note that the above diagram represents the data path for IEEE 1905 protocol packets only. All user data packets are unaffected by the IEEE 1905 protocol and continue to follow the path that they have always followed in our different chips (using the Integrated Switch, FAP, Flowcache, Linux Bridge, etc. or a combination of these).

4.0 CODE RELEASES

The 1905 software is released as source code in the folder userspace/private/apps/ieee1905. It can be found in the DSL/PON "data" releases or in the separate PLC Linux release. The "data" release is used to build DSL or PON Gateway software and is delivered under the Broadcom DSL Software License Agreement (SLA). The PLC Linux release is used to build Linux software for the HPAV1 and HPAV2 extenders (Ethernet/Wi-Fi/HPAV bridge) and is delivered under the Broadcom PLC SLA since it also contains PLC Firmware.

The PLC release can be untar'ed on top of the DSL release, thus creating a single source tree capable of creating images for DSL or PON Gateways or HPAV capable products all from the same Linux source, If you want to build a DSL or PON Gateway product that included an integrated PLC interface (on the same board), you will need to perform this operation or untar'ing the PLC release on top of the DSL/PON release.

The software releases also includes the experimental Adaptive Home Networking software, which is only delivered in pre-compiled, object format. The Adaptive Home Networking software is found as flowmanager.o in the same folder as the 1905 software and as a module named flowbond.ko, under bcmdriver/broadcom/char/flowbond.

5.0 COMPILE INSTRUCTIONS

5.1 Enabling IEEE1905.1

To enable IEEE 1905 in your compile profile, run the menuconfig utility and enable, under "Other Features", the "IEEE1905 support" option. One must also ensure "Ethtool Support" is selected under "Ethernet, Switch, and VLAN Selection" for use of 1905 over the Ethernet.

5.2 Enabling Adaptive Home Networking

If you are experimenting with Adaptive Home Networking, you will also want to enable additional features using the menuconfig utility. These are "Flow Manager" and "Automatically create WDS links", found under the "Other Features" menu item and, "Flowbond", found under the "Packet Acceleration" menu item. Adaptive Home Networking is a non-standard feature that is built on top of IEEE 1905.1 and which provides features such as link aggregation, best link selection and failover. Because these are experimental, it is not currently recommended to enable them in a final product.

Once you have enabled IEEE1905, you can start compiling the software using the "make PROFILE=<profile name>" command.

For instructions on how to compile the Homeplug driver or other driver, please refer to the appropriate documentation.

5.3 Maketargets

A utility is provided to facilitate updating the profile with the proper features for IEEE1905.1 or for Adaptive Home Networking. This utility is called maketargets and uses as input a profile name along with "arch" files which are essentially patch files that can be applied against an existing profile to create a modified profile.

We have created a number of "arch" files for the purpose of enabling HPAV1, HPAV2, 1905 and AHN features. To apply an arch file, the utility is invoked as follows:

release/maketargets <profile>+<arch file1>+<arch file2>+…

The following arch files related to 1905 were created in the folder targets/arch:

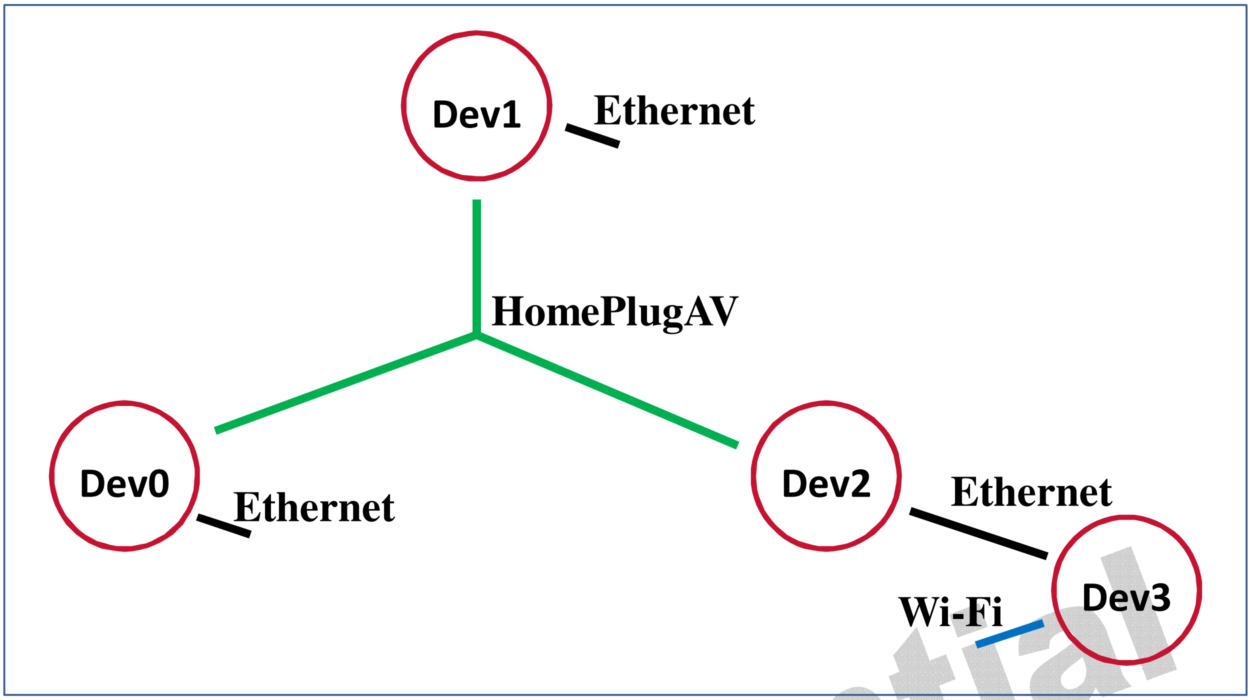
1905.arch (To enable IEEE1905.1)
FB.arch (To enable Adaptive Home Networking)
DUNA.arch (To enable the 60500/60333 support on an Gateway supporting HPAV2)
PLCSPI.arch (To enable the 60321 support on an Gateway supporting HPAV1 and SPI Flash)
PLCNAND.arch (To enable the 60321 support on an Gateway supporting HPAV1 and NAND Flash)

A number of examples are available in the file targets/definitions/profiles. Each row in this file contains a shortcut name for a profile followed by a list of arch files that can be applied using the command "release/maketargets <shortcut name>"

6.0 Data Model

6.1 Example of a 1905 Network

Consider the following network, which consists of 4 devices, three of which are directly connected to each other and one which is only connected to another device. An internal data-model to represent such network will be needed.



6.2 Internal data-model

The internal data-model of the Broadcom IEEE 1905 software is different from the standard data-model. We have chosen to store the information in a more efficient way to facilitate processing. In the Broadcom software, the following objects are defined:

Device: These represent each 1905 device found in the network. The local device, its immediate neighbors and every remote neighbor is represented by a device. Each device is uniquely identified by its assigned IEEE 1905.1 MAC address.

Interface: Each device has a number of interfaces (Ethernet, HomePlugAV, Wi-Fi). Each of these interfaces with a unique MAC address is represented by an interface object. If certain interfaces share the same MAC address, they also share the same interface object.

1905 Neighbor: This object represents how the local device is connected to its immediate neighbor devices. The neighbor 1905 object is uniquely identified by the MAC addresses of each device involved in the connection and the MAC address of each Interface involved.

Other objects are also defined to represent bridging tuple information and legacy neighbors.

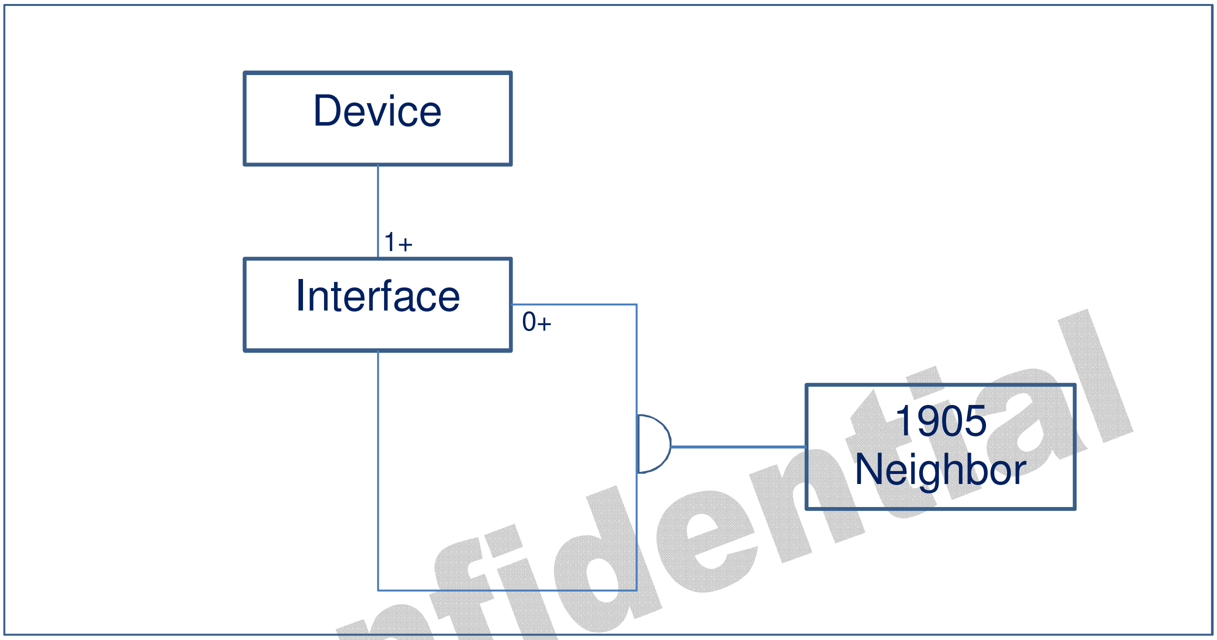
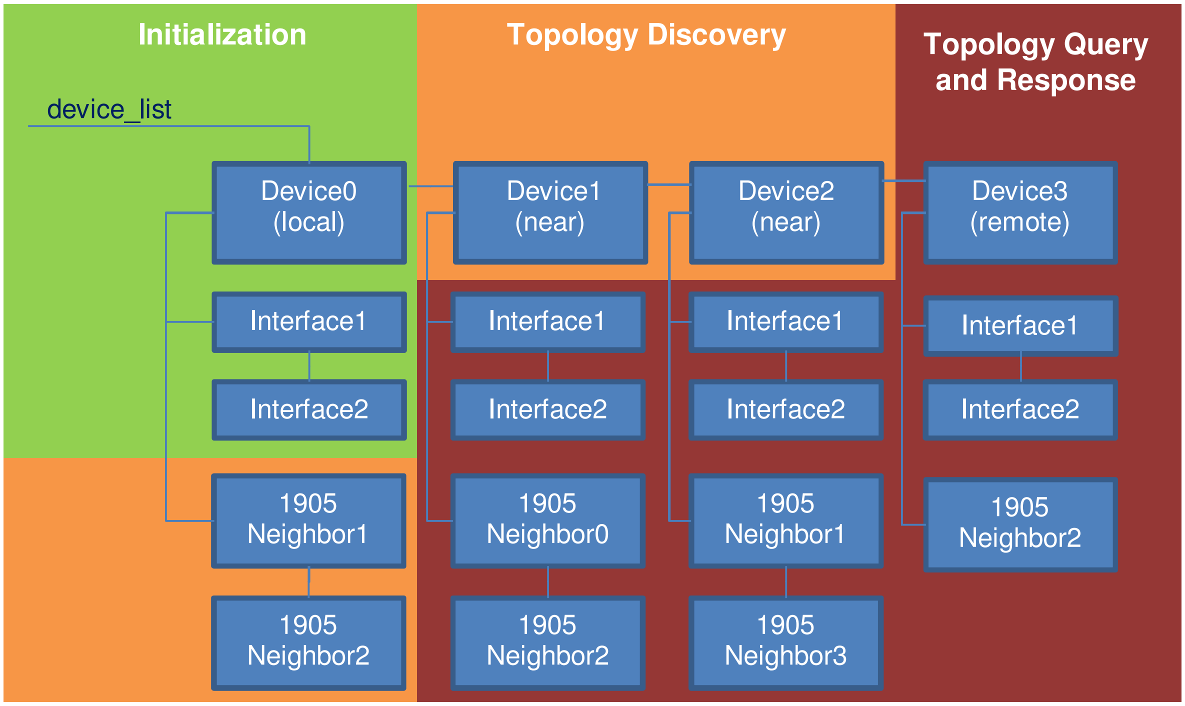


Figure 4. Internal data-model

6.3 Populating the data-model

The internal data-model is populated at different times during the Topology Discovery protocol. The first object to be created is the local device object with each of its interfaces, at initialization time. As topology discovery messages start being transmitted and received, it becomes possible to identify the presence of local (immediate, 1 hop) 1905 neighbors, which trigger the creation of neighbor objects on the local device, and the creation of new un-initialized device objects in the device list. The system then attempts to populate the newly added devices by sending topology queries to each of them. The topology response allows populating the un-initialized devices with their respective data. If a topology response indicates the presence of a remote (multi-hop) neighbor (one that was not seen with the topology discovery messages), a new un-initialized device is created for this remote neighbor and a topology query message is sent to it.



6.4 Mapping to the standard IEEE 1905.1 data-model

It is possible to map the standard IEEE 1905.1 data-model to our internal data model by applying the following translations:

Standard IEEE 1905.1 data-model object	*Broadcom Software data-model object*
Device.X_84D32A_IEEE1905.AL.	**Last device in the device list (the local device)**
Device.X_84D32A_IEEE1905.AL.Interface.{i}.	**Interface list of local device**
Device.X_84D32A_IEEE1905.AL.Interface.{i}.Link.{i}.	**1905 neighbor list of local device**
Device.X_84D32A_IEEE1905.AL.Interface.{i}.Link.{i}.Metric.	**Attribute of 1905 neighbor**
Device.X_84D32A_IEEE1905.AL.ForwardingTable.	**Attribute of the local device**
Device.X_84D32A_IEEE1905.AL.ForwardingTable.ForwardingRule.{i}.	**Currently not implemented**
Device.X_84D32A_IEEE1905.AL.NetworkTopology.	**Attribute of the local device**
Device.X_84D32A_IEEE1905.AL.NetworkTopology.ChangeLog.{i}.	**Currently not implemented**
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.	**Devices in device list, minus the last device which represents the local device**

Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.Interface.{i}.	**Interface list of each device in device list**
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.NonIEEE1905Neighbor.{i}.	**Legacy Neighbor of each device in device list, currently not implemented**
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.IEEE1905Neighbor.{i}.	**1905 neighbor list of each device in device list**
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.BridgingTuple.{i}.	**Bridging tuple list of each device in device list**
Device.X_84D32A_IEEE1905.AL.Security.	**Currently not implemented**

7.0 SHELL UTILITY

The Broadcom IEEE 1905 software includes a command line utility called i5ctl which can be used during development but is not meant to be used by a product. The command can be accessed from the Gateway shell by typing “i5ctl” or through the abbreviation “i5”, which stands for ieee1905. Help information can be obtained by typing just “i5” without any argument:

```
# i5

Usage: i5 <option>
        dm  -Display data model
        tr  -Enable function call tracing (0-Msg, 1-Tlv, 2-Dm)
        wlcfg  -Wifi Autoconfig UnitTesting
        plc -PLC actions (0 - start UKE, 1 - randomize)
        stop  -Stop ieee1905 daemon
        start  -Start ieee1905 daemon
        restart  -Stop and start ieee1905 daemon
        link  -Trigger Link Metric Queries
```

7.1 Showing the internal data model

The internal datamodel used by the ieee1905 daemon is displayed with the “i5 dm” command:

```
# i5 dm
Device: 021018746404
  Interface: 021018746402 ()
  Interface: 021018746403 ()
  Interface: 021018746401 ()
  Neighbor: LcIf 021018746401, NbAL 021018882605, NbIf 021018882601, Bridge
0, State 0, (via /0) (0/0)Mbps
  Bridging Tuple: State 0
    Fwd Interfaces:           021018746401
          021018746403
          021018746402

Device: 021018882605
  Interface: 021018882601 (eth0)
  Interface: 021018882604 (wl0)
  Neighbor: LcIf 021018882601, NbAL 021018746404, NbIf 021018746401, Bridge
0, State 0, (via eth0/6) (0/0)Mbps
```

The command displays each 1905 device found on the network and their MAC address. The last device listed is the local device. Then each interface of the device is listed. On the local device, the linux interface name is known and displayed in parenthesis. Then, the 1905 neighbors found on each interface are listed, along with the Local Interface (LcIf) on which the neighbor was seen, and the Neighbor 1905 MAC address (NbAL) and the Neighbor Interface from which this neighbor was seen (NbIF). The Brigging Tuple and Forwarding interfaces are currently under development and contain data with limited usefulness.

7.2 Enabling traces

Traces have been implemented in multiple places in the 1905 daemon software to help understand what is happening during development. The command is written to allow enabling traces per module and trace levels have been defined:

```
Disabled = 0
Errors only = 1
Normal & Errors = 2
Info, Normal & Errors = 3
```

The available modules are defined in ieee1905_trace.h and are, at this moment:

```
i5TraceMessage   = 0
i5TraceTlv       = 1
i5TraceDm        = 2
i5TraceInterface = 3
i5TraceFlow      = 4
i5TraceTimer     = 5
i5TraceSocket    = 6
i5TraceMain      = 7
i5TraceSecurity  = 8
i5TraceUdpSocket = 9
i5TracePlc       = 10
i5TraceWlcfg     = 11
i5TraceControl   = 12
i5TraceNetlink   = 13
i5TraceJson      = 14
All of the above = 255
```

```
# i5 tr 0 3
i5MessageAutoConfigSendPendingTimeout [line 728]
i5MessageApAutoconfigurationSearchSend [line 764] WSC not responding: AP
Autoconfiguration Search Retries will be delayed until later.
i5MessageCreateAutoConfigSendTimer [line 740] freqBand = 0, callCounter = 1,
timeout = 2000
```

7.3 Randomizing the HomePlugAV security key

HomePlugAV uses a security key to encrypt data on its network. Multiple HomePlugAV devices connected to each other must share the same Security key. This key is saved in NVRAM and can be randomized through the following command:

```
# i5 plc 1
```

Note that the HomePlugAV interface will restart after randomizing its key. Other devices will only be able to connect to a device with a randomized security key by performing the 1905 pairing using the security push button. Also note that the when reseting a device to default settings, the randomized key is replaced with a default key.

7.4 Triggering Link Metric Query

Link metrics need to be obtained from all devices in the 1905 network. The timing when these link metrics are needed is dependent on the application using them, so the 1905 daemon cannot decide when exactly the metrics should be obtained. The “link” command line allows triggering the gathering of link metrics on the network. The command line utility is written to wait for a response and display them in a similar manner as the data model gets displayed. To trigger this function:

```
# i5 link
```

7.5 Stop, Start and Restart commands

These commands can be used to stop, start or restart the ieee1905 daemon application. The data model information collected by the 1905 daemon is lost over a restart.

7.6 Other commands

Other debug/test level commands are also available through the “i5ctl” application but these only served specific development activities and should not be used.

8.0 API INTERFACE

The 1905 daemon API is implemented through a unix socket which is used by the i5ctl command. Thus an application developer can look at the i5ctl.c software and reuse it as a reference. The API allows extracting the data model and triggering a link metric query on the network. While the i5ctl process will block and wait for responses on the socket, an application wanting to run these commands does not have to be blocking and can keep the socket always opened to interface with it in an asynchronous way.

9.0 REFERENCES

IEEE Std 1905-2013, IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications.