Master of Science Thesis

Performance Analysis and Improvement of Topology Discovery Protocols in Home Networks

13 July, 2011

Erik German Diaz Castellanos
(4053265)

Committee Members:

Supervisors:	Prof. Dr. Ir. Sonia Heemstra de Groot
	Dr. Ir. Frank den Hartog
Other Members:	Prof. Dr. Ir. Robert Kooij

Wireless and Mobile Communication (WMC) Group
Department of Telecommunications
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Connectivity Expertise Group
Technical Sciences Expertise Center
TNO

Abstract

The growing popularity of the Internet and the increasing demand of services based on IP have influenced the evolution of home networks. From a simple network constituted by just a PC and a modem, the home network has become a complex environment providing connectivity to several devices with different capabilities. Although service providers possess tools to manage their own core and access networks, they lack the tools to get information related to home network characteristics. Due to the impact of home network configurations on delivered services, operators are interested in diagnostic tools able to gather information about the topology of the home network, the link layer technologies that are used and which are the active devices requiring home network connectivity.

The goal of this thesis is to study the currently available topology discovery protocols and evaluate their suitability for home networks. Our work focuses on two protocols that are believed by the service providers' community to be appropriate for the home network scenario. These two protocols are the Link Layer Topology Discovery (LLTD) protocol and the Link Layer Discovery Protocol (LLDP), the latter also known as IEEE 802.1AB. Our study includes the definition of a set of performance indicators for topology discovery protocols and a performance analysis of LLTD and LLDP for different conditions and topologies. We designed several experiments representing the most common home network configurations, and carried out measurements that provide the data needed for our analysis. Based on the obtained results, we then proposed a novel topology discovery architecture, called Home Network Topology Discovery (HNTD) that fulfills most of operators' requirements, in contrast to LLTD and LLDP.

Key Words: HTIP, LLTD, LLDP, IEEE 802.1AB, TR-069

Acknowledgements

This work is presented in the context of a Master thesis project in the Delft University of Technology (TU Delft) and the Netherlands Organization for Applied Scientific Research (TNO). It was carried out from October 2010 to June 2011 in the Wireless and Mobile Communications (WMC) group of the faculty Electrical Engineering, Mathematics and Computer Science (EEMCS) of TU Delft and Connectivity expertise group at TNO.

I would like to thank my supervisors Dr. Ir. Frank den Hartog and Prof. Dr. Ir. Sonia Heemstra de Groot for their assistance and guidance during the nine months of the master thesis project. I am thankful to my friends for their help during difficult times. And finally, I would like to thank my family for their continued support and love.

Contents

List of Figures

List of Tables

List of Abbreviations

AFT	Address Forwarding Table
ATM	Asynchronous Transfer Mode
AP	Access Point
CAPEX	Capital Expenditure
CCo	Central Coordinator
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
C-VLAN	Customer VLAN
Eth	Ethernet
EXT	Extension
FCS	Frame Check Sequence
FN	False Negative
FP	False Positive
FPR	False Positive Rate
HG	Home Gateway
HGI	Home Gateway Initiative
HNID	Home Network Infrastructure Device
HNTD	Home Network Topology Discovery
HNTD(C)	Home Network Topology Discovery Client
HNTD(S)	Home Network Topology Discovery Server
HopCnt	Hop Count
HP	HomePlug
HTIP	Home-network Topology Identifying Protocol
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
ITU	International Telecommunication Union
LAN	Local Area Network
LLC	Logical Link Control
LLDP	Link Layer Discovery Protocol
LLDPDU	Link Layer Discovery Protocol Unit
LLM	Light LLTD Mapper
LLTD	Link Layer Topology Discovery
LSAP	LLC Service Access Point
MAC	Medium Access Control
MIB	Management Information Base
MSAP	MAC Service Access Point
MSDU	MAC Service Data Unit

N	Negative
NF	Neighbor Field
NMS	Network Management System
ODA	Original Destination Address
OPEX	Operational Expenditure
OSA	Original Source Address
OSI	Open System Interconnection
P	Positive
PAN	Personal Area Network
PLC	Power Line Communication
QD	Quick Discovery
QoS	Quality of Service
QoSDNT	Quality of Service Diagnostics Network Tests
QoSDCTA	Quality of Service Diagnostics Cross Traffic Analysis
ROC	Receiver Operating Characteristics
SFD	Start of Frame Delimiter
SIP	Session Initiation Protocol
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
STA	Station
STP	Spanning Tree Protocol
S-VLAN	Service VLAN
TCID	Traffic Class Identifier
TDMA	Time Division Multiple Access
TDT	Topology Discovery Tests
TLV	Time, Length, Value
TN	True Negative
TP	True Positive
TPMR	Two Port MAC Relay
TPR	True Positive Rate
TTC	Telecommunication Technology Committee
VC	Virtual Circuit
VLAN	Virtual Local Area Network
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

1

Introduction

The last decade has been an exciting time for the telecommunication sector. Progress in connectivity technologies and the growth of the Internet have provided to people new means to interact between each other and the digital world. As a result of this evolution, new services have been introduced into our daily life. Of course, the adoption of these new services and tools by society has brought new challenges to service providers that need to upgrade and improve their networks to cope with the increasing amount of data exchanged every day. Also, new devices requiring Internet connection and the boom of mobility have made networks more and more complex, especially within homes.

Data networks, including the internet, are designed following the connectionless oriented approach. Data is sent using a stream of packets transmitted through different paths. In this scenario a best effort to deliver the message is made. New users demand services, such as video streaming or voice, that require a minimum of Quality of Service (QoS) to offer a suitable experience. QoS depends on the capacity of links and also on the configuration and structure of the network. In order to guarantee a suitable user experience, service providers are looking for tools to perform diagnostics of home networks to detect potential weak points that can affect the performance of offered services. To perform a good diagnostic, the tool must include mechanisms to gather information about the topology of the home network and networking technologies implemented to provide connectivity to different devices.

1.1 Motivation and Challenge

During the last two decades, Home Networks have evolved from a simple personal computer using a 56kbit/s connection for data transfer service to a group of devices using one or more broadband connections to get multiple services. This transformation has resulted in more complex networks due to an increase of the number of devices requiring interconnection and Internet connection and the coexistence of different kind of technologies in the same environment. Also, the types of devices that need to be connected vary more and more.

More complexity in home networks has brought:

- Difficulty in configuring and troubleshooting home networks
- Bottlenecks in home networks, due to unsuitable topologies than can affect QoS.
- Need of technical support for end users
- Increase of capital expenditure (CAPEX) for service providers

Instead of solving failures, service providers are interested in anticipating potential problems in home networks through the use of early diagnostic tools which could lead to a decrease of calls for home networking support and, as a result, decrease of operational expenditure (OPEX). Because of these requirements, the Home Gateway Initiative (HGI) and other bodies have been working on new diagnostics tools, demanding input from TNO and other companies and institutes.

According to HGI's diagnostics philosophy, topology information is required to help solving problems in the network. Topology information includes [8, 29]:

- A list of active devices and their capabilities
- A map of connections between devices
- Technologies used in the home network on a per-connection basis.

Although research related to topology discovery has been developed, its focus has been limited to medium and large size Ethernet networks. For topology discovery of heterogeneous home networks, much less literature is available. To test the available topology discovery protocols in home networks and to design new protocols, we first need a more deep study of available protocols.

The challenges of the present master thesis project are:

- Design of a testbed that includes a Home Gateway (HG) supporting the topology discovery protocols of interest and the common home network topologies.
- Definition of suitable performance parameters.

1.2 Thesis Goal and Structure

This thesis extensively studies two topology discovery protocols of interest for service providers that are expected to be appropriate for home networks. These protocols are Link Layer Topology Discovery (LLTD) protocol and Link Layer Discovery Protocol

(LLDP) or IEEE 802.1AB. Since a suitable performance comparison of these protocols within home networks is required, the goals of this thesis work are determined to be:

- Identify strengths and weaknesses of each protocol when used in home networks.
- Propose improvements of these protocols based on a performance comparison study.

The rest of the thesis is organized as follows:

In Chapter 2, we describe the main characteristics of home networks and the dominant home networking technologies.

In Chapter 3, an introduction about topology discovery protocols is given. First, a brief summary of related work is presented. Then, the characteristics of LLTD and LLDP are studied.

In Chapter 4, we describe the experiments done to test the operation of both LLTD and LLDP and the parameters employed to study the performance of these protocols.

In Chapter 5, we show the results of measurements and analyze the operation of these protocols based on their performance parameters.

In Chapter 6, based on the analysis of LLTD and LLDP, we propose a new architecture to perform topology discovery.

Finally, Chapter 7 concludes this report by providing a description of contributions achieved throughout this thesis project and recommendations for related future work.

2

Home Network Characteristics

Before studying the available topology discovery techniques, we need to understand which elements constitute the entity we call home network, which technologies are implemented and how they are used. First we describe the evolution of home networks and the models used to represent it. Then, we describe the main technologies used to provide connectivity to different network devices.

2.1 Evolution of Home Networks

Service providers and standardization bodies have proposed models to characterize the architecture of home networks. Throughout the years, these models have been modified to include new technologies and services.

The 1990's is the decade of massive adoption of digital personal computers and 56kbps dial- up internet access by home users. The first network architecture is characterized by its simplicity with only one personal computer connected to the Internet using the telephone network mainly for data transfer. Other devices requiring external information use their own analogue network. An initial attempt to model the home network is made in 1999 by the International Telecommunication Union (ITU) with the recommendation ITU-T G 995.1. Within this document, four types of entities are introduced to describe the architecture at that time. These entities are [1] (see figure 1):

- NT1, which is the entity that terminates the digital section of the broadband connection
- NT2, which is the entity that terminates the transport protocol for user traffic
- Terminal Adapter (TA), which is the entity that adapts the transport protocol according to the user terminal's requirements.
- User Terminal (UT), which is the entity that provides an interface to the end user.

Also, the document defines four types of interfaces (R,S,T,U) to interconnect the entities mentioned before . This architecture represents a rather operator-minded view on the home network, namely being an extension of the access network.

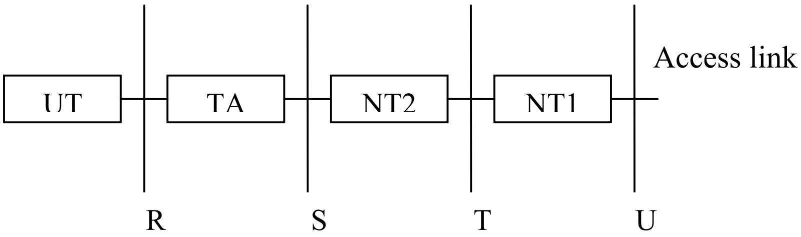


Figure 1. Home Network Model ITU-T G 995.1 [1]

The improvement of access network technologies providing greater bandwidth changes the configuration of home networks. In the mid 2000s, multiple devices exist in the home environment and are connected to the Internet using the telephone or cable network. This new scenario allows the offer of multiple services using a broadband connection. In 2004, the DSL Forum updates the home network model and defines the following entities [1] (see figure 2):

- B-NT or Broadband network termination
- The Routing Gateway
- The premises Distribution or end-user's infrastructure
- The Functional Processing Device (FPD)
- The End User Terminal (EUT)

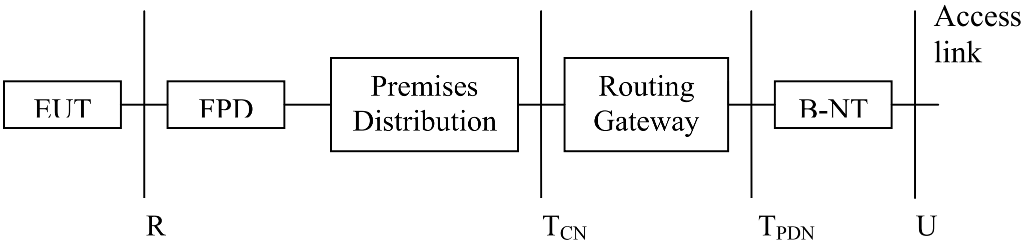


Figure 2. Home Network Model DSL Forum TR-094 [1]

As shown in figure 2, these entities are connected using interfaces known as R, T_{CN}, T_{PDN} and U. The introduction of a *Premises Distribution* entity recognizes the existence of multiple layer-2 devices providing connectivity to different devices in the home network and acknowledges the increase of complexity within home networks. The end-user's infrastructure has evolved to include new types of link technology.

The ability of delivering different services using a single broadband connection opens new business opportunities to service providers to increase their income while using the same infrastructure. The business model known as Triple Play combines two bandwidth-intensive services (Internet Access and Television) and one less bandwidth demanding service (Telephone). This business model has been adopted by most of service providers in the world and has evolved to incorporate mobile services (Quadruple play). Due to the success of Triple Play, in 2004 service providers adopted a new home network model (see figure 3) of which the key element is a residential gateway with dedicated ports or technologies to specific services [1]. As shown in figure 3, the residential gateway is modeled as a hybrid between a bridged and a routed model. Services are provided through different Virtual Circuits (VC) that constitute an Asynchronous Transfer Mode (ATM) connection. It is expected that such residential gateway evolves into a full routed model where a service is not dependent on a specific physical port or technology [28].

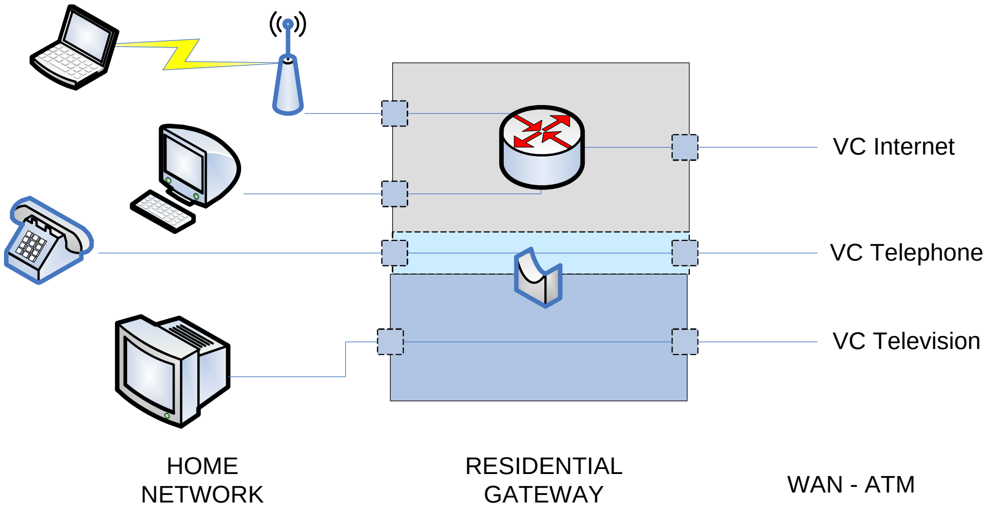


Figure 3. Triple Play Model [1]

Further research has been developed to improve the residential gateway's functionalities in order to guarantee a suitable user experience. It has been acknowledged that characteristics of the end-user's infrastructure, such as technology and topology, can also affect QoS. As a result, efforts have been carried out to improve home networking technologies and topologies. Instrumental for stimulating this research is the work done by the Home Gateway Initiative (HGI). The HGI is an organization founded in 2004 by major service providers and later joined by leading vendors of home networking products. Its goal is to gather requirements from all parties involved and standardize devices and protocols belonging to the home network environment. Although the evolution of the future residential gateway has an important role in its activities, HGI has also studied the evolution of other characteristics of infrastructure in order to guarantee QoS and provide diagnostic tools to service providers. HGI's approach describes the home network as a combination of the following entities [2]:

- Home Gateway (HG): This is the device possessing the router functionality and providing connectivity from the service provider. It is synonymous with residential gateway.
- Home Network Infrastructure Devices (HNID): These are layer 2 devices that bridge two or more segments and connect the stations with the home gateway
- Stations (STA): End-devices or interfaces between the network and the end-user.

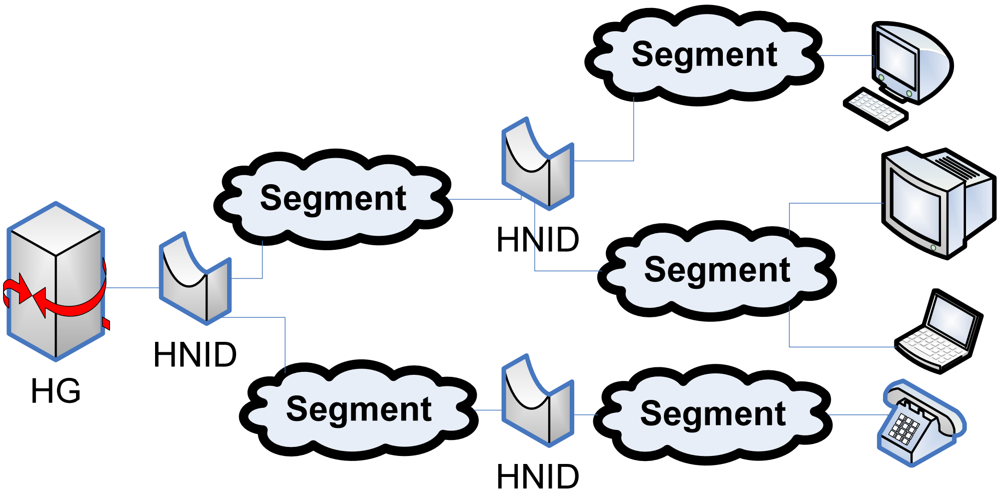


Figure 4. HGI Home Network [2]

In the model shown in figure 4, HNIDs are manageable devices able to support different kind of networking technologies for Local Area Networks (LAN). Although they are layer-2 devices, they often support an IP stack for the sake of remote management. Their fundamental interoperation is standardized by the Institute of Electrical and Electronics Engineers (IEEE) as IEEE 802.1D [3]. The different networking technologies that characterize different types of HNID are described in the following section.

2.2 Home Network technologies

The Open Systems Interconnection (OSI) model represents a communication system through a layered model as shown in figure 5.

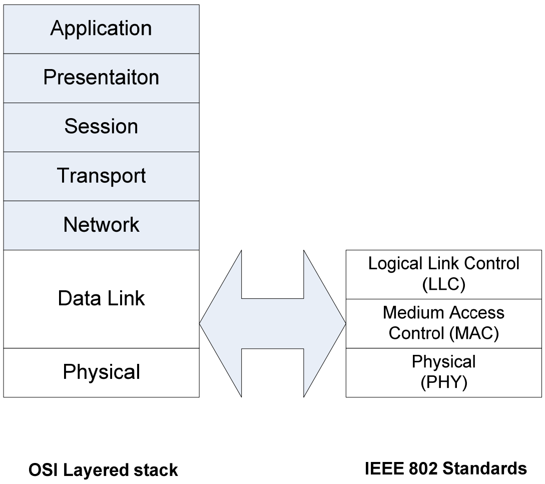


Figure 5. OSI model and its correspondence with IEEE 802 standards [4, 7, 21]

The operation of technologies described below lies on the physical layer and data link layer. Networking technologies differ by the type of physical medium and access control method. Nowadays, there are three types of technologies that have been widely deployed within home networks:

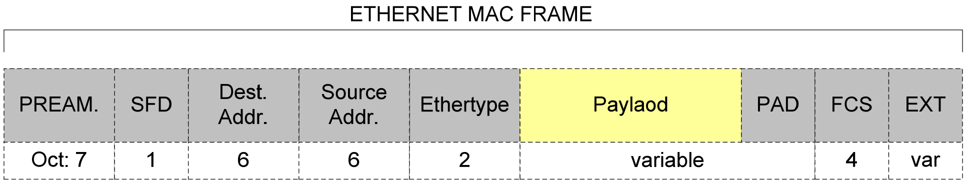
- Ethernet (Eth)
- Wireless Local Area Network (WLAN or WL)
- Power Line Communication (PLC)

Due to trends in the home automation (narrow-band in-home control services such as lighting control and automatic meter reading), a fourth type of technology known as Zigbee is being introduced in home networks.

2.2.1 Ethernet

Ethernet is one of the most widely used wired technologies within home networks. Initially developed by Xerox during the 1970's, this technology is standardized as IEEE 802.3 [4]. It relies on twisted pair copper cables as the physical medium for transmission and uses Carrier Sense Multiple Access/Collision Detection (CSMA/CD) as the medium access control method. Within home networks the most common link layer data rate is 100 Mbit/s, but Gigabit Ethernet is slowly becoming popular.

Data is sent through the medium as a sequence of packets with a specific format. IEEE 802.3 standard defines a Medium Access Control (MAC) frame format as shown in figure 6.



PREAM.	SFD	Dest. Addr.	Source Addr.	Ethertype	Paylaod	PAD	FCS	EXT
Oct: 7	1	6	6	2	variable		4	var

Figure 6. IEEE 802.3 MAC frame format

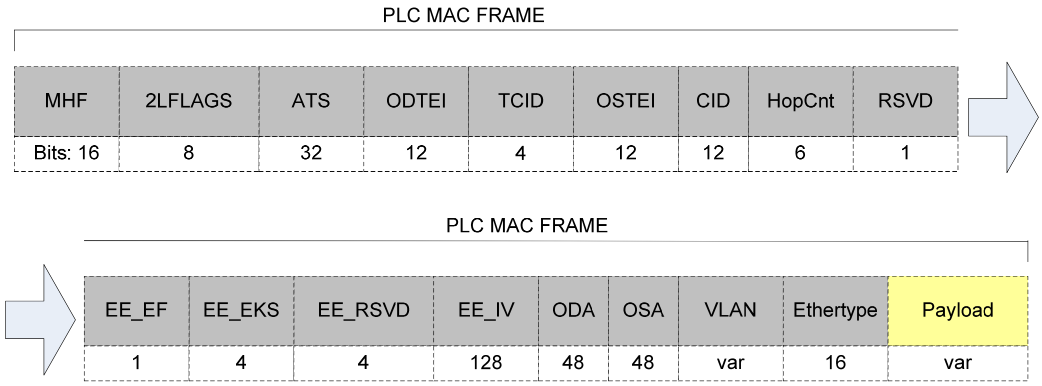
Each device in the network is identified by a unique 6 bytes long physical address, or MAC address. This physical network identification is included within the Ethernet header as destination address and source address, which provide the origin and destination of the transmitted packet. The Ethernet frame contains in its payload information from higher layers of the OSI model. The type of information or protocol that characterizes the payload is given by the Ethertype field. Other fields in the Ethernet MAC frame are the Start of Frame Delimiter (SFD), the Pad which is sequence of bytes added when the size of the payload is less than the minimum size, the Frame Check Sequence (FCS) which allows the detection of corrupted data in the frame and Extension (EXT) which is a sequence of bits transmitted before the next frame.

An Ethernet network can be constituted by several Ethernet segments or collision domains linked by one or more HNIDs known as Ethernet switches. Packets are received and transmitted by a switch according to the IEEE 802.3 standard.

2.2.2 Power Line Communication

Although Ethernet is a widely known technology, its deployment within home networks requires the upgrade of the end-user’s infrastructure by adding new wires. In a market where minimization of costs for end-users is essential, PLC has become a suitable solution for home networking. This technology uses existing low voltage electric wiring for data transmission, which means that no new wires are required. PLC is not yet well standardized. The most commonly used PLC implementation is known as HomePlug. New standards under development such as IEEE 1901 [6] and ITU-T G.hn [31,32] adopt the main features of HomePlug.

HomePlug defines the existence of a Central Coordinator (CCo) device that controls the activities of the PLC network [5]. In contrast to Ethernet, PLC combines Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) and Time Division Multiple Access (TDMA) as methods for access control to provide a physical layer transmission rate of 200 Mbit/s. Data is transmitted using the MAC frame format shown in figure 7.



PLC MAC FRAME

MHF	2LFLAGS	ATS	ODTEI	TCID	OSTEI	CID	HopCnt	RSVD
Bits: 16	8	32	12	4	12	12	6	1

PLC MAC FRAME

EE_EF	EE_EKS	EE_RSVD	EE_IV	ODA	OSA	VLAN	Ethertype	Payload
1	4	4	128	48	48	var	16	var

Figure 7. PLC frame format [6]

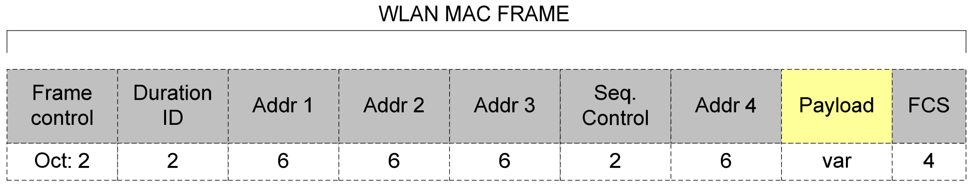
Although the PLC MAC frame format defines more fields compared to the Ethernet MAC frame format, most of these fields are used for access networks. As an example of fields used only for access networks we have the Hop Count field (HopCnt) and Traffic Class Identifier (TCID). The relevant fields for our study are the Original Source Address (OSA) which is the sender's MAC address, the Original Destination Address (ODA) which is the receiver's MAC address and the Ethertype field.

PLC technology relies on the existence of a single collision domain provided by the low voltage home wiring system that connects all devices to the HG. The typical HNID used in a PLC network links an Ethernet segment with the PLC segment.

2.2.3 Wireless Local Area Network

Unlike wired technologies, this technology provides more flexibility to the end-user thanks to the use of wireless radio spectrum to transmit data. Although WLAN has less transmission capacity compared to its wired counterparts, it has been successful due to the benefit of mobility. IEEE has standardized this technology as IEEE 802.11 [7].

An HNID known as Access Point (AP) bridges the Ethernet wired network with the wireless medium. The coverage of an AP is defined by the characteristics of the environment and obstacles. The access method is CSMA/CA and the MAC frame format is shown in figure 8. Unlike Ethernet or PLC MAC frames, WLAN MAC frame format can include more that two address fields. Besides the destination address and source address, other possible types of address fields are the Basic Service Set Identifier, transmitting station address and receiving station address. The Ethertype field, which is present in the PLC and Ethernet MAC frames format, is not included in the WLAN MAC frame format.

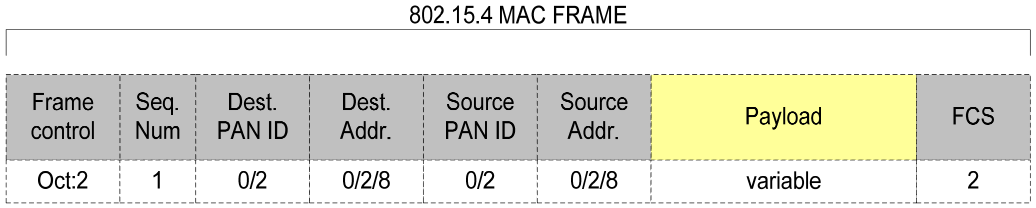


Frame control	Duration ID	Addr 1	Addr 2	Addr 3	Seq. Control	Addr 4	Payload	FCS
Oct: 2	2	6	6	6	2	6	var	4

Figure 8. IEEE 802.11 MAC frame format

2.2.4 Sensor network within home networks

The scope of home networks has been extended beyond network devices such as computers, network attached storage and video games consoles. Because of the adoption of home automation, future home networks will also include sensors and controllers that allow end-users to interact with home systems controlling environment (temperature), energy, appliances and security. The upcoming technology used to communicate between such devices is called Zigbee and its lower layers have been standardized by IEEE as IEEE 802.15.4 [21]. This technology is a wireless technology that uses CSMA/CA as access method and it is characterized by its low power consumption and low data rate (up to 250 kbit/s). Another interesting feature is its ability to form ad-hoc networks, which are networks without defined topology and without central controller. Also WLAN can operate in ad-hoc mode (thus without AP), but unlike Zigbee, it is hardly ever used as such. Figure 9 shows the frame format defined by the IEEE 802.15.4 standard. The physical addresses of devices supporting IEEE 802.15.4 can be 2 or 8 bytes long. IEEE 802.15.4 includes a second field in the MAC header to relate a device to a specific Personal Area Network (PAN). This field is called Source/Destination PAN ID. Also, the Ethertype field is not included in the frame format.



Frame control	Seq. Num	Dest. PAN ID	Dest. Addr.	Source PAN ID	Source Addr.	Payload	FCS
Oct:2	1	0/2	0/2/8	0/2	0/2/8	variable	2

Figure 9. IEEE 802.15.4 MAC frame

3

Related work

Topology discovery is a topic that has been studied for large-sized networks where administrators need detailed information about the active network devices and connections between them. Automatic processes have been developed to provide the most accurate topology information that is needed to manage the system. This chapter first explores the different techniques created to perform topology discovery based on available information from Address Forwarding Tables (AFT) and Spanning Tree Protocol (STP). Then, we describe the main characteristics of LLDP and LLTD. Finally, we describe a modified version of LLDP as currently under standardization by ITU.

3.1 Definitions

In order to explore and understand the different discovery topology mechanisms, we provide the following definitions:

- B_y^x represents a set of ports *x* in bridges *y*.
- The forwarding set for port *x* is the list of MAC addresses that can be reached through this port. F_y^x is the forwarding set for port *x* in bridge *y*. We assume that *N* devices can be reached through port *x* of bridge *y*. F_y^x is complete if there are entries for all existing *N* devices.
- The through set for port *x* is the list of MAC addresses that can be reached through other ports in the same bridge. T_y^x is the through set for port *x* in bridge *y*.
- *y(a)* is the port of bridge *y* whose forwarding set includes MAC address *a*.
- Two nodes are referred as ***directly connected*** if there are no other nodes between them

- Two switches are ***simply connected*** by ports x_1 and x_2 if they find each other within their forwarding sets, but there may be other nodes in between.
- A ***segment*** is a shared media where a device is able to receive other devices' transmissions. A segment with at least one STA is known as ***shallow segment***. On the other hand, a segment with no STAs is known as ***deep segment***. A segment is called ***intermediate segment*** if it connects at least two bridges.
- An ***island*** contains at least one shallow segments of the network, forming a maximal connected subgraph with no deep segments.
- A ***gap*** is a portion of a network that contains only deep segments. It connects several islands together.
- S_B is the set of bridges connected to a segment. S_E is the group of end-devices attached to a segment. S includes all devices (bridges and end-devices) connected to a segment and it is represented by the following expression:

$$S = S_B \cup S_E \tag{1}$$

3.2 Topology Discovery using AFT and STP

Extensive research has been carried out to discover the topology of large Ethernet networks where most of the bridges can be managed by a central entity or Network Management System (NMS) through the use of protocols such as Simple Network Management Protocol (SNMP). The goal of these techniques is to discover the switches and the list of devices identified by their MAC addresses. Little relevance is given to information related to characteristics of end-devices that can not be provided by AFTs or SPT. Due to the fact that AFTs and STP protocol are supported by all manufacturers, the following algorithms are able to operate within a multivendor environment.

3.2.1 Topology Discovery based on AFT

In [9], authors describe an algorithm to discover the layer two topology of a large Ethernet Network using the information stored within AFTs. These tables list the devices, which are identified by their MAC addresses, that can be reached through a specific port x belonging to a switch y. The forwarding set includes the neighbors and also other devices that are not connected directly to the switch.

Authors in [9] discover the topology of the Ethernet network by finding the direct connections and shared segments among switches. The following theorems define the algorithm:

- *Direct Connection Theorem*: Assume that F_i^x and F_j^z are complete. Two bridges *i* and *j* are directly connected via the link connected to port *x* on *i* and port *z* on *j*, if and only if:

$$F_i^x \cap F_j^z = 0 \tag{2}$$

$$\text{and } F_i^x \cup F_j^z = M \tag{3}$$

  where *M* is the set of all nodes in the network.

- *Shared Segment Theorem:* *S* consists of a shared segment between the bridges in S_B , if and only if for all bridges a in S_B and all MAC addresses b,c of devices in S:

$$a(b) = a(c) \tag{4}$$

  All forwarding databases are required to be complete.

The completeness requirement can be a limitation when the algorithm is implemented, especially in the case of large networks where it is not possible to guarantee complete forwarding sets. Authors in [9] also propose an algorithm to find topology based on information provided by incomplete forwarding sets. This alternative approach finds simple connections between nodes according to the following theorem:

- *Simple Connection Theorem*: The ports *x* from switch *i* and *z* from switch *y* are connected if:

$$T_i^x \cap T_y^z = 0 \tag{5}$$

Although the simple connection theorem does not require complete forwarding databases, algorithms based on it are less accurate.

In [10] and [11] the core of proposed topology discovery processes rely on theorems described before, but upgrades have been included in order to cope with redundant links and multi-subnet networks. The topology discovery process is performed by the NMS that accesses every switch through SNMP to retrieve the stored AFTs.

3.2.2 Topology Discovery based on STP

STP is used to avoid loops in the topology of an Ethernet network by blocking redundant connections. In this protocol each bridge has a unique identifier (ID) and every Ethernet link has a cost that is defined by IEEE 802.1D standard according to its link capacity. The cost of a path connecting two switches is the sum of costs of all links that belong to that path. The general operation of the protocol can be summarized in four steps:

- **Step 1**: The switch with the smallest ID value is selected as root switch.

- **Step 2**: Each switch determines the cost of each possible path from itself to the root switch and then chooses the path with the smallest cost. The port leading to the path with minimum cost becomes the root port of the switch.
- **Step 3**: Switches connected to a segment determine which one has the minimum cost path from the segment to the root switch. The selected switch becomes the designated switch and the port belonging to this switch connected to the segment becomes the designated port.
- **Step 4**: Any port that is not a designated port or a root port becomes a blocked port.

In [12], authors identify the connections between different switches using STP information. The proposed topology discovery strategy is based on the following deductions:

- **Inference 1**: If the state of the port B_y^x is blocking and the designated port B_k^l corresponding to it does not belong to bridge y, then port B_y^x and port B_k^l are connected and the connection is redundant.
- **Inference 2**: If the bridge y acts only as the designated bridge of itself, it is designated as the leaf node of the Spanning Tree.
- **Inference 3**: The root port and the designated port of the leaf node connect each other directly.
- **Inference 4**: If two ports that connect directly to each other appear more than once in connection sets, there must be one or more hubs or bridges that do not support STP.

3.3 Link Layer Discovery Protocol

Topology discovery strategies described in section 3.2 provide information about connections between network devices and their scope of implementation is limited to Ethernet networks. Information about characteristics of bridges and capabilities of end-devices is neglected. Also, topology information is not immediately available because it must be inferred from AFTs or STP databases. IEEE proposes a discovery topology protocol able to operate within networks supporting different technologies and able to provide complete topology information avoiding major data processing. In 2005, IEEE publishes a first version of the IEEE 802.1AB standard, also known as the Link Layer Discovery Protocol (LLDP), which is developed in close cooperation with CISCO systems. LLDP allows all network devices attached to a LAN to advertise their presence and major capabilities by multicasting frames that carry the desired data. The information is stored in a data base called Management Information Base (MIB), supported only by

HNIDs, which can be accessed using management protocols such as SNMP (which requires HNIDs to be IP addressable). Each HNID stores information from its neighbors, which means that every HNID has information about the local topology. The protocol runs over the data link layer and works for all IEEE 802 access protocols. The remaining of this section describes the main characteristics of the protocol and its operation according to the IEEE 802.1AB standard as published in 2009 [13].

3.3.1 LLDP entities

The elements described in this section are essential components during the operation of the protocol, which includes:

- Transmission and reception of relevant data between networked devices
- Storage of received information

3.3.1.1 LLDP Data Unit

LLDP Data Unit (LLDPDU) carries data as a sequence of variable length information elements (see figure 10). Each one of these information elements includes type, length and value fields known as TLV.

- The type field identifies the kind of information that is being sent.
- The length field indicates the length of the information string in octets.
- The value field carries the actual information.

A LLDPDU contains the following TLVs:

- Chasis ID TLV (Mandatory)
- Port ID TLV (Mandatory)
- Time to live TLV (Mandatory)
- Zero or more optional TLVs
- An end of LLDPDU TLV (Mandatory)

Both the Chasis ID TLV and Port ID TLV are used as a logical MAC service access point (MSAP) identifier. MSAP is the access point for the MAC sublayer services to the LLC sublayer. Thanks to the MSAP identifier, the recipient can recognize the sending LLDP agent/port. The Time to live TLV gives the time of validity of the information carried by the LLDPDU.

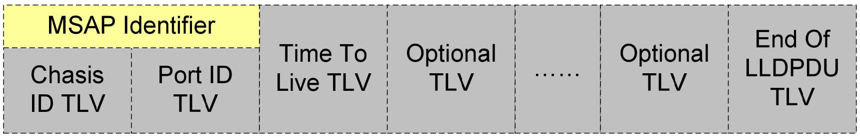


Figure 10. LLDPDU format

Optional TLVs are grouped into basic management TLV set and organizationally specific TLV set. The basic management TLV set is required in all LLDP implementations and includes the following TLVs:

- Port Description TLV
- System name TLV
- System description TLV
- System capabilities TLV
- Management address TLV

Organizationally specific extension sets are defined by standardization groups in order to improve the management capability of a network that is operating over a particular media or using a particular protocol. As an example we can mention the IEEE 802.1 organizationally specific TLV set and the IEEE 802.3 organizationally specific TLV set.

3.3.1.2 Management Information Base

The Management Information Base (MIB) is a database used to manage the entities belonging to a network. In the context of LLDP, the information regarding capabilities and characteristics of networked devices connected to a LAN are stored into two types of MIB modules known as the mandatory basic MIB and optional organizationally specific MIB extensions (see figure 11). Each type of module is divided into two sections:

- Local system MIB/Extension: This section stores the information corresponding to the local device that is going to be advertised to other devices.
- Remote system MIB/Extension: This section stores the information advertised by other devices in the same LAN and received through the incoming LLDPDU.

The basic MIB contains information obtained from the basic management TLV sets and the organizationally specific MIB extension contains obtained information from the optional TLVs defined by a third party or organization.

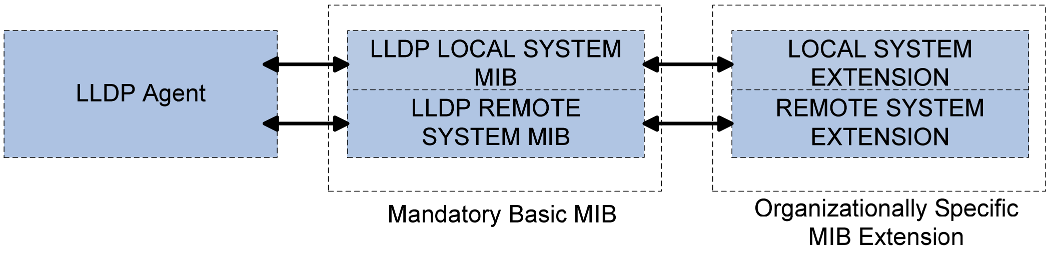


Figure 11. LLDP MIB and extensions

3.3.1.3 LLDP agent

The LLDP agent is an important entity that interacts with the Logical Link Control (LLC) sublayer and the different MIBs in a HNID. It performs the following tasks:

- It maintains current information in the LLDP local system MIB
- It extracts the information that is going to be advertised from the LLDP local system MIB and organizes the data into a proper LLDPDU format in order to be transmitted.
- It recognizes and processes received LLDP frames.
- It maintains current values in the LLDP remote system MIB.
- It notifies to MIB managers whenever a value or status change has occurred in one or more objects on the LLDP local system MIB or LLDP remote system MIB.

The LLDP agent can also be supported by end-devices that do not have management agents such as SNMP and MIBs. In this case, this agent just advertises the end-device's information obtained from a local data base.

3.3.2 Transmission and Reception principles

3.3.2.1 Addressing

Each LLDPDU is transmitted as a single MAC service request by the LLC sublayer that uses an MSAP. An LLDPDU frame is received by the LLC sublayer through the MSAP as a MAC service indication. The parameters of each service request and service indication are the destination address, the source address, the Ethertype and LLDPDU.

3.3.2.1.1 Destination and Source MAC address

The destination MAC address used by LLDP determines the scope of propagation of LLDPDU within a bridged LAN. Three group MAC address are defined by the protocol as follows:

- ***The nearest bridge group MAC address (01:80:C2:00:00:0E)****:* The propagation of LLDPDUs is constrained to a single segment.
- ***The nearest non - Two Port MAC Relay (TPMR) bridge group MAC address (01:80:C2:00:00:03)***: The propagation of LLDPDUs is constrained by all bridges other than TPMRs. This address is intended for use on bridged networks. A TPMR bridge is a two port bridge used for network maintenance purposes only (usually not present in home networks).
- ***The nearest customer bridge (01:80:C2:00:00:00):*** The propagation of LLDPDUs is constrained by customer bridges, i.e. bridges that comply with IEEE 802.1D.

The source MAC address is the individual MAC address of the sending station or port.

3.3.2.1.2 Ethertype and LLDPDU

The LLDP Ethertype value used to identify the LLDP is 0x88CC. If the MSAP used is supported by a MAC method able to support encoding of Ethertypes (example: IEEE 802.3), the LLC sublayer concatenates both LLDP Ethertype and LLDPDU to form a MAC Service Data Unit (MSDU) as shown in figure 12.

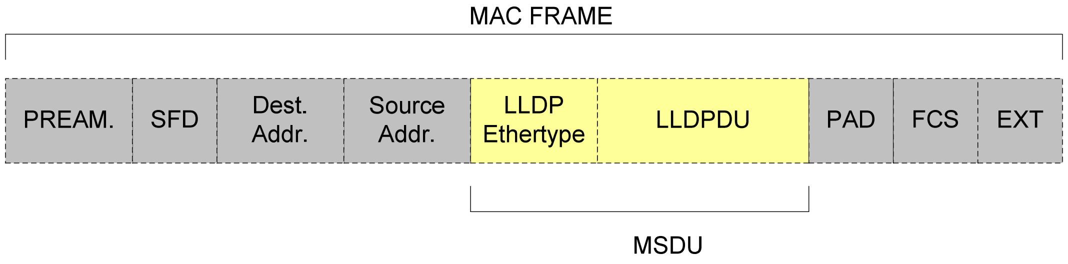


Figure 12. MAC frame and MSDU formats

When the MAC method does not directly support Ethertype encoding (example: IEEE 802.11), Ethertypes are supported only via Subnetwork Access Protocol (SNAP) encapsulation.

3.3.2.2 Transmission Principle

Transmission of LLDPDUs can me initiated either by the expiration of a transmission countdown timing counter or by a change in value of one or more of the information elements in a LLDP MIB. When the transmission cycle is initiated, the LLDP agent extracts the managed objects from the local MIB. Then, the information obtained from the local MIB is formatted into TLVs that will be inserted into an LLDPDU. Finally, the LLDPDU is passed to the LLDP transmission module. Figure 13 summarizes the steps involved during the transmission process.

1. Transmission cycle is initiated
2. The managed objects are extracted from the local MIB
3. The information from the local MIB is formatted into TLVs
4. New TLVs are inserted into an LLDPDU
5. The LLDPDU is passed to the LLDP transmission module

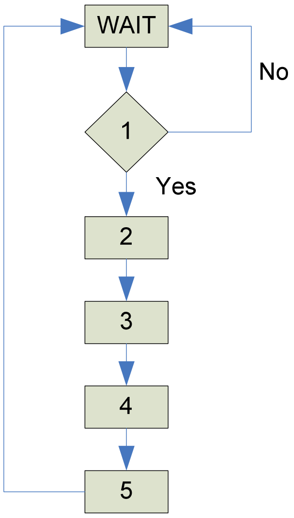


Figure 13. Transmission process

3.3.2.3 Reception Principle

The reception of a LLDP frame is carried out in three phases: frame recognition, frame validation and remote system MIB updating.

- ***Frame recognition*:** Before passing a frame to the LLDP agent, the LLC sublayer and the corresponding LLC Service Access Point (LSAP) must determine whether the received frame is an LLDP frame. This is done by checking the destination address and the Ethertype included in the header.

- ***Frame validation:*** After recognizing the incoming frame as a LLDP frame, the LLDP agent validates the format of the carried data. The LLDP frames must be properly constructed and contain the correct set of mandatory TLVs.

- ***Remote system MIB updating:*** The validated LLDP frames are used to update the information of entries in the remote systems MIB corresponding to the originating LSAP identifier and source MAC address. If an entry for the originating LSAP identifier and source MAC address does not exist, a new entry is created in the remote systems MIB.

Figure 14 summarizes the steps involved during the reception process.

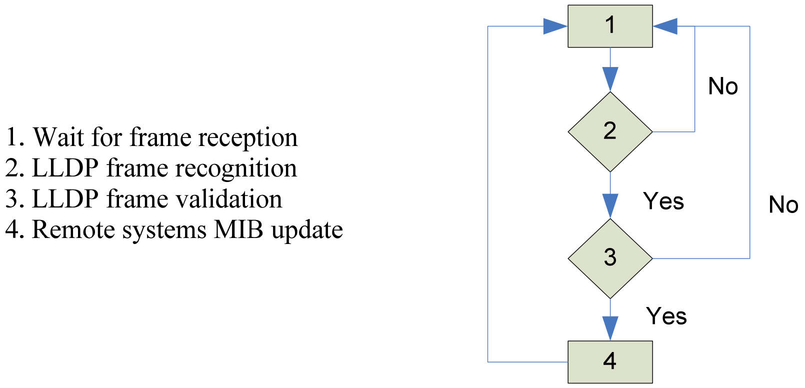

1. Wait for frame reception
2. LLDP frame recognition
3. LLDP frame validation
4. Remote systems MIB update
1
No
2
Yes
No
3
Yes
4

Figure 14. Reception process

3.3.3 Inference of network topology

Mechanisms used by LLDP allow each HNID to gather topology information about its neighbors, about the local topology. In order to obtain the complete information about the topology of the network, a NMS is required. Using SNMP, the NMS will retrieve the information about the local topology seen by each HNID. Thanks to the neighbor lists, the NMS can easily build the final topology map.

3.4 Link Layer Topology Discovery

LLDP relies on every networked device and segment to support the topology discovery process and the topology information is distributed among all HNIDs. Link Layer Topology Discovery (LLTD) protocol, which is a protocol developed by Microsoft as part of the Windows Rally set of technologies, operates based on tests performed on the network by a central entity known as mapper. The mapper tries to identify the different network devices according to their behavior or forwarding mechanism. The tests are supported by entities called responders which operate based on instructions sent by the mapper. The mapper and responder entities are included in last versions of Microsoft's operating system (Windows XP, Windows Vista, Windows 7).

LLTD works over wired as well as wireless media. The use of Microsoft's protocol is restricted by patents and licenses. However, technical specifications of the responder and frame formats are freely available to be used by manufacturers who want to implement this tool within their products. For the remaining of this section, we describe the main characteristics of the protocol and mechanism used according to LLTD's technical specifications [14, 15].

3.4.1 LLTD services

LLTD supports several services, not only for topology discovery but also for QoS diagnostics. During the operation of different services, end-devices supporting the protocol can take different types of roles. An end-device can be an enumerator, a mapper, a responder, a controller, a sink or a cross-traffic analysis initiator. Four types of services are established by the standard: Quick Discovery (QD), Topology Discovery Tests (TDT), QoS Diagnostics Network Tests (QoSDNT) and QoS Diagnostics Cross Traffic Analysis (QoSDCTA).

QD and TDT services are the ones important to carry out the topology discovery process relying on the exchange of different types of frames. For topology discovery, also the role of enumerator is relevant, besides mappers and responders. It is defined in the following section.

3.4.1.1 Quick Discovery

The information advertised by the responders includes the type of link technology used by the device to interact with the network. An end-device that takes the role of enumerator enumerates other capable-LLTD end-devices responders in the network. The enumerator discovers the responders by initiating the QD service.

3.4.1.2 Topology Discovery Tests

This service is an extension of the QD service and, as a result, it can be performed only after QD. The enumerator that initiates QD becomes the mapper and tries to associate itself with the active responders by performing TDT. There must be only one mapper associated to all responders in a broadcast domain. After the association process, the responders accept and respond to the commands sent by the mapper.

3.4.2 Transport and format of LLTD frames

The protocol relies on IEEE 802 technologies to transport the messages exchanged between the different stations. The Ethertype field within the Ethernet header is set to 0x88D9. The LLC sublayer concatenates the LLTD Ethertype and the LLTD frame to form an MSDU. The LLTD frame is formed by an LLTD Demultiplex Header and an LLTD Base and Upper Layer Header (see figure 15).

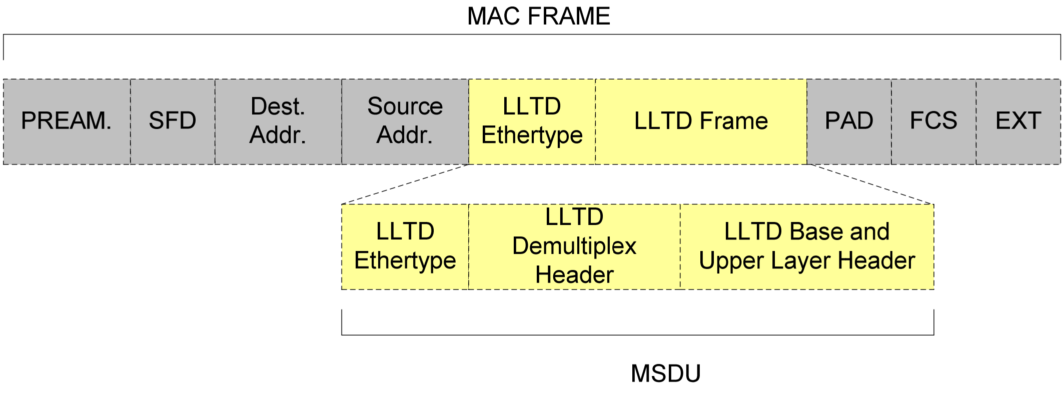


Figure 15. MAC frame and MSDU formats

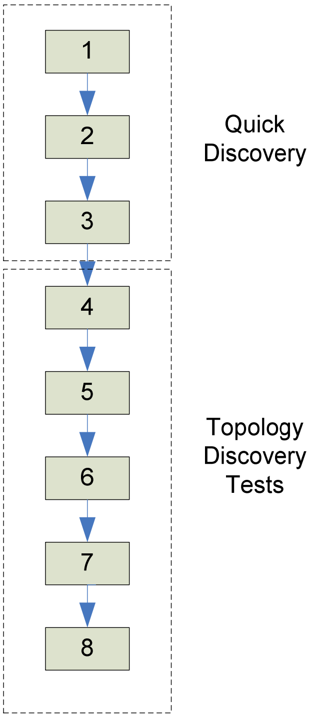
The demultiplex header carries the information about the type of service and message that is being used. Four fields within the header are defined: version, type of service, reserved and function. The version and reserved fields must be set to 0x01 and 0x00 respectively. The type of service field is assigned with a value according to the service that is being used. The value can be 0x00 for TDT, 0x01 for QD and 0x02 for QoSDNT/QoSDCTA. Services tasks include the exchange of different messages. The type of message is indicated by the value of the function field as shown in table 1 for QD and TDT. The base and upper layer header carries the messages used by LLTD services to perform their tasks. In table 1, only messages used by QD and TDT services are described.

Function field		Service	Sent by..		Purpose
Value	Meaning		M	R	
0x00	Discover	QD, TDT	X		Discovery of active responders
0x01	Hello	QD, TDT		X	Send attributes from responders
0x02	Emit	TDT	X		To request transmission of train or probe frames with specific source and destination addresses
0x03	Train	TDT		X	Allow bridges to learn the origin of a MAC address
0x04	Probe	TDT		X	Must be recorded by other responder
0x05	Ack	TDT		X	To acknowledge the reception of Emit frames
0x06	Query	TDT	X		To ask for information about probe frames received by a responder
0x07	QueryResp	TDT		X	To send a list of recordable events available since previous Query frame
0x08	Reset	QD, TDT	X		To abort mapping process
0x09	Charge	TDT	X		To prevent bandwidth amplification attacks
0x0A	Flat	TDT		X	To report charge frames count or to ask mapper to retry the Emit frame request
0x0B	QueryLargeTLV	TDT	X		To query a responder for data that is too large to be included into a Hello frame
0x0C	QueryLargeTLVResp	TDT		X	To respond to a QueryLargeTLV frame

Table 1. Messages for TDT and QD, M=Mapper, R=Responder

3.4.3 Topology Discovery Process

In previous sections we explained the different services offered by LLTD protocol to discover the network topology at the link layer level. Also, we described the transport and format of LLTD messages that are exchanged between LLTD-capable devices during the operation of the protocol. We aim at studying the characteristics and performance of the topology discovery functionality of LLTD. The topology discovery process covers only two of the services offered by the protocol: QD and TDT services. Figure 16 shows a block diagram describing the topology discovery process. The figure is self-explanatory.



1. Enumerator broadcast discovery frames
2. Responders respond reply with Hello frame
3. Enumerator find RESPONDERS in a LAN
4. Enumerator becomes MAPPER
5. Association between MAPPER and RESPONDERS
6. MAPPER sends zero or more Emit requests
7. RESPONDERS send Probe or Train frames
8. MAPPER infers topology

Figure 16. Topology Discovery process

3.4.4 Inference of Network Topology

The LLTD protocol relies on the exchange of different types of frames and operations on responders found during QD to assess the topology of a network. The technical specification published by Microsoft ([14,15]) gives a general description of responder functionalities and frame formats that allow end-devices to perform their tasks according to the assigned role. But, it does not explain the mechanisms or algorithms within the mapper that provide the intelligence required to infer the topology of a network at the link layer level.

In [16], which is a paper published before the introduction of LLTD in the market, Mircrosoft's researchers explain a topology discovery technique that does not require the assistance of HNIDs belonging to the network under test. It is based on the injection of probe packets and observation of their final destination. This topology discovery algorithm is based on two properties:

- End-devices on the same segment can be detected. Each one of them is able to see the traffic from others.

- Packets with a particular MAC source address entering a bridge on one port will prevent the switch from sending packets with that destination address to any other port.

Also, the following assumptions are made:

- There is a central controlling entity for the algorithm. This entity is similar to LLTD mapper.
- Most end-devices in the network run a daemon that injects probe packets and records received packets. This daemon is similar to LLTD responder.
- A preliminary protocol lets us discover all end-devices running daemons and establish a connection with them. The purpose of this preliminary protocol is the same as LLTD QD service.
- The algorithm uses training packets that cause a bridge to learn a particular source address.
- The algorithm uses probe packets to test whether a bridge has learnt a trained address.

Because Microsoft's algorithm is not available for studying, we assume that the method described in [16] is the source of topology discovery services included in LLTD and we use it as reference to understand LLTD mapper's operation.

According to [16], after detecting all active responders in the network, the mapping process is carried out in four phases. Figure 17, figure 18 and figure 19 illustrate the different phases.

- **Phase 1 – Segment Detection:** The sees set is the list of source addresses of received probe packets by an end-device plus the end-device's own MAC address. During segment detection phase, after exchange of probe packets, the sees sets of different end-devices are compared in order to identify a segment. Two end-devices belong to the same shallow segment if they have the same sees set. In figure 17, A, B, C, D, E, F and G are the segments found after phase 1. Segment M includes the central controlling entity that is equivalent to the LLTD mapper.

- **Phase 2 – Discovery of switches:** The algorithm is able to detect any bridge shared between multiple shallow segments by observing whether a bridge trained by one segment changes its behavior when observed by another segment. After identifying the segments, a segment tree is constructed using information about discovered switches. Figure 17 shows a possible outcome of phase 2, where the known segments are linked to the discovered switches.

- **Phase 3 – Island Discovery:** After identifying the segments and some switches, we obtain groups of islands. The main goal of this phase is to identify the switches at the edge of these islands. This phase requires information from phase 1 and phase 2. No further injected traffic is needed. Figure 18 shows the discovered islands after phase 3.

- **Phase 4 – Discovering Gaps:** The central controlling entity (equivalent to the mapper) performs a series of path crossing tests designed to determine how different paths, each one connecting two end-devices supporting the daemon (equivalent to the responder), intersect. Thanks to these tests the topology discovery algorithm described in [16] is able to identify the connections between the switches at the edge of islands found during phase 3. Figure 19 shows a possible result of phase 4.

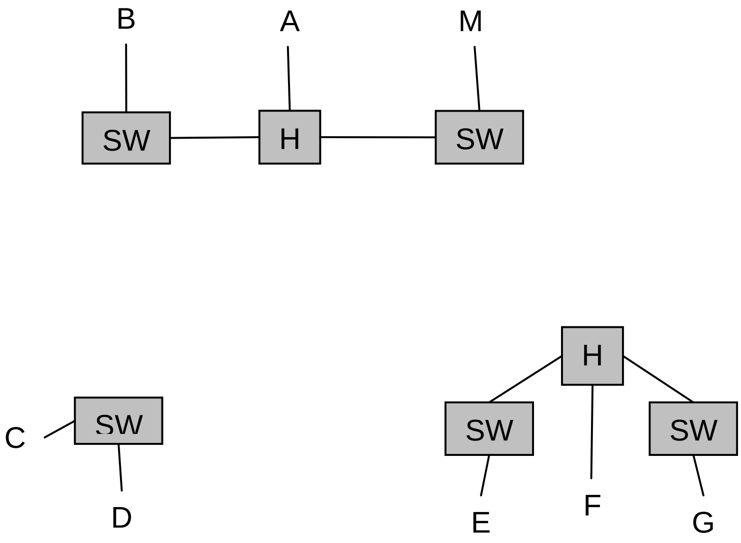


Figure 17. Phases 1 and 2, SW = Switch, H = Hub

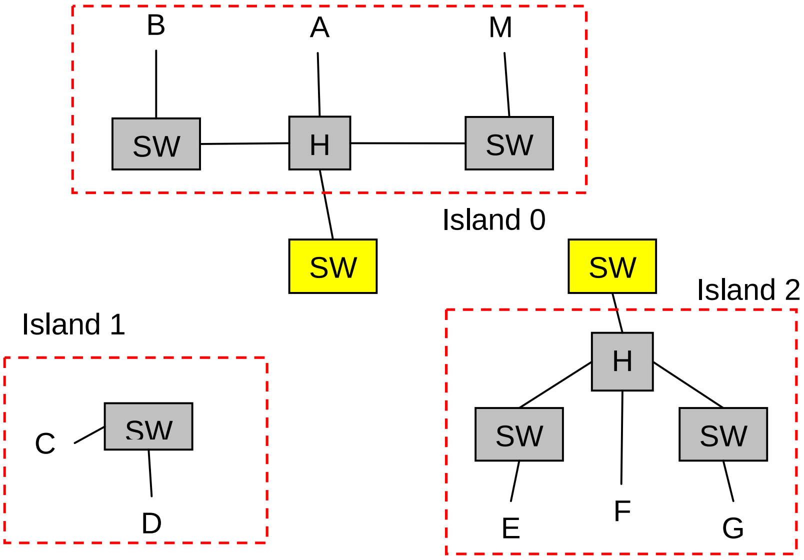


Figure 18. Phase 3, SW = Switch, H = Hub

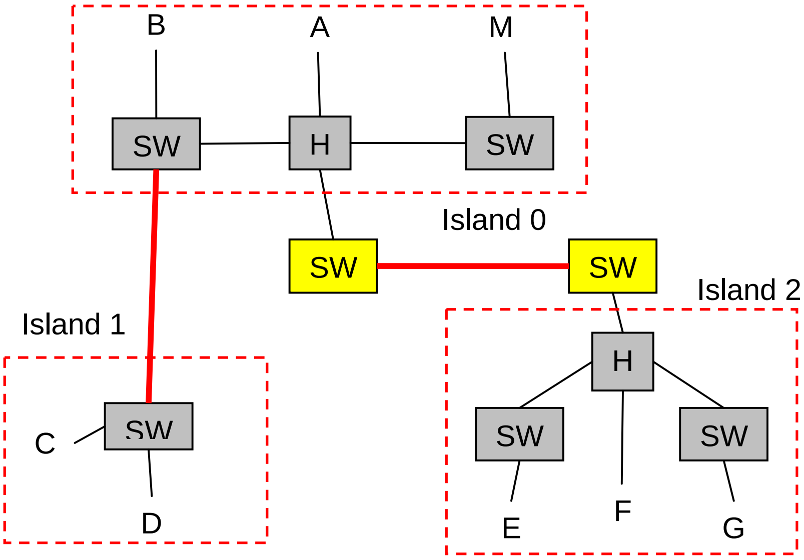


Figure 19. Phase 4, SW = Switch, H = Hub

3.5 Home-network Topology Identifying Protocol

Devices within home networks tend to be simple in functionality. Often bridges do not support the IP stack and are not manageable. Addition of these characteristics could increase the cost of a home network device for the end-user. This is a severe limitation of the use of LLDP. The Japanese Telecommunication Technology Committee (TTC) and the International Telecommunication Union (ITU) have been working on a discovery protocol derived from LLDP to reduce functionalities within HNIDs. The first draft of this modified version of LLDP, known as Home-network Topology Identifying Protocol (HTIP), is published by TTC in August 2010 [17] and is now studied by ITU.

HTIP establishes that topology information can also be retrieved from AFTs within HNIDs, instead of LLDP MIB storing neighbors' information. Based on this assumption, an NMS is able to create a topology map after receiving the AFTs from active HNIDs. Instead of using SNMP to retrieve the required information, the list of MAC addresses belonging to an AFT is sent as a sequence of TLV information elements carried by broadcasted LLDP frames. End-devices are discovered not by LLDP, but by UPnP [17]. This transmission mechanism avoids the use of a management agent, such as SNMP, or IP stack within HNIDs. Figure 20 shows the HTIP's operation.

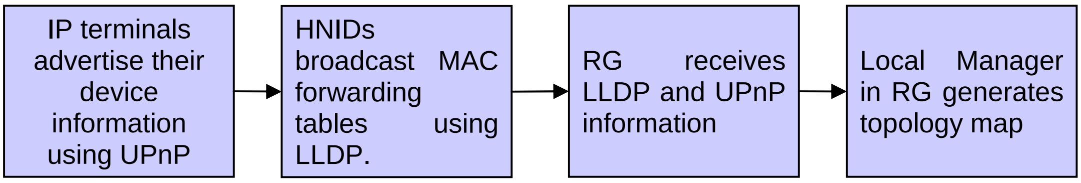

IP terminals advertise their device information using UPnP
HNIDs broadcast MAC forwarding tables using LLDP.
RG receives LLDP and UPnP information
Local Manager in RG generates topology map

Figure 20. HTIP operation

HTIP uses the vendor-specific extension field as described by IEEE 802.1AB standard with TLV type equal to 127. Figure 21 shows the format of the HTIP TLV information element.

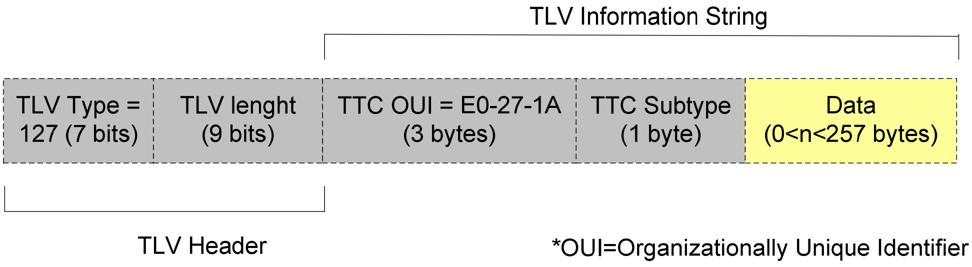


Figure 21. HTIP TLV format

The field called TTC Subtype identifies the type of data that is transmitted. Relevant values of this field are shown in table 2. In order to send AFTs (TTC Subtype = 3), the format of the data field is as shown in figure 22.

TTC Subtype	Data
1	Device Information
2	Link Information
3	MAC address list

Table 2. TTC Subtype values

As we can see, the TLV format includes the forwarding set for each available port in an HNID and also describes the type of port (link layer technology) according to IANAifType definitions [18]. An example of interface type values is given in table 3.

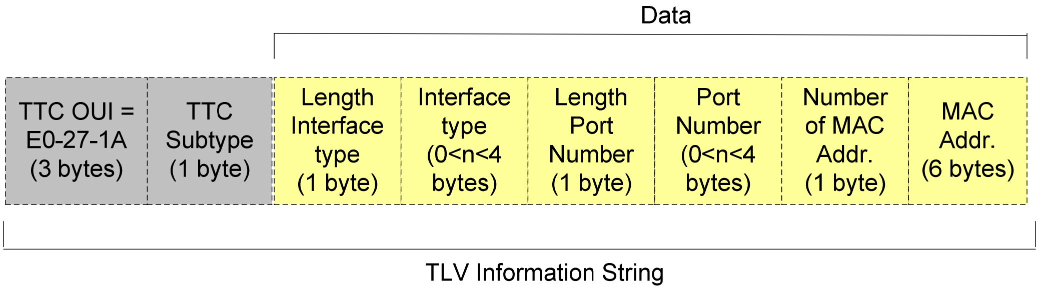

Data
TTC OUI = E0-27-1A (3 bytes)
TTC Subtype (1 byte)
Length Interface type (1 byte)
Interface type (0<n<4 bytes)
Length Port Number (1 byte)
Port Number (0<n<4 bytes)
Number of MAC Addr. (1 byte)
MAC Addr. (6 bytes)
TLV Information String

Figure 22. Address Forwarding Table TLV

Interface Type	Data
6	Wired lined, Ethernet-like interfaces
71	IEEE 802.11
174	Power Line Communications

Table 3. Interface Type Values

Although HTIP provides the mechanism to transmit information from HNIDs to the HG, the first draft of this protocol does not give a guideline to process the obtained data to infer the network topology.

3.6 Preliminary comparison

Throughout this chapter we presented different discovery topology techniques that have been developed to operate within different kind of networks. After analyzing the technical specifications, we identify three main mechanisms to carry out topology discovery:

- **Mechanism 1**: This mechanism infers topology using information from existent protocols or data bases, such as STP, AFT or UPnP
- **Mechanism 2**: This mechanism relies on advertisement mechanisms to discover devices and connections. This is the basis of LLDP and HTIP
- **Mechanism 3**: This mechanism employs tests to find network devices based on their behavior when probes are injected into the network under study. This is the basis of LLTD.

Although all described techniques are able to identify HNIDs and STAs, those which are based on AFT and/or STP's information only can not identify the type of home networking technology (Tech). Table 4 shows the main characteristics of the described topology discovery techniques. The technical specifications allow us to deduce strengths and weaknesses of different topology discovery techniques as shown in table 5.

	Mechanism			**Identifies**			**Relies on…**		**Needs NMS**	**IPR restriction**
	1	**2**	**3**	**HNID**	**STA**	**Tech**	**HNID**	**STA**		
AFT/STP	X			X	X		X		Yes	No
LLTD			X	X	X	X		X	No	Yes
LLDP		X		X	X	X	X	X	Yes	No
HTIP	X	X		X	X	X	X		No	No

Table 4. Characteristics of Topology Discovery Techniques

	LLTD	LLDP	AFT/SPT	HTIP
Strengths	Relies only on end-devices to support the protocol Does not need an NMS	LLDP is an open standard Automatic advertisements cause topology to be updated instantaneously.	Use of available data from other protocols.	Use of available data from other protocols. Does not need management agents in HNIDs
Weaknesses	IPR restricted Topology only generated after test is manually initiated	HNIDs should support management agents. (e.g. MIBs, SNMP)	HNIDs should support management agents. (e.g. MIBs, SNMP) Conceived for Ethernet networks.	An algorithm to infer network topology is not provided yet.

Table 5. Strengths and Weaknesses of Topology Discovery Techniques

From table 5 we conclude that from a qualitative point of view, none of the protocols is clearly superior over others. They all have their strengths and weaknesses. Further analysis needs to be qualitative, and is presented in the following chapters.

4

Description of Experiments

This chapter aims at describing the different experiments carried out and the parameters used to analyze the operation of protocols under study. First, we explain the implementation process and characteristics of different components included in the testbed. Then, we describe the different configurations representing the common topologies within home networks. Finally, we establish the performance parameters to be used to analyze the operation of LLTD and LLDP.

4.1 Testbed implementation

To test LLTD and LLDP, we implement a testbed that includes devices and required agents or daemons supporting the protocols under study. Following the HGI model, a home network is constituted by a Home Gateway and an end-user's infrastructure that contains several HNIDs.

4.1.1 Home Gateway

The Home Gateway is the key device for performing our analysis, because it must include the protocols of interest. Home network products manufacturers do currently not offer an appropriate home gateway supporting LLTD and LLDP. However, different network devices and software packages available in the market support at least one of the desired functionalities:

- The LLTD mapper is included in any personal computer with Microsoft operating system's Windows Vista and Windows 7.
- The LLTD responder is also included in Windows Vista and Windows 7 and can be implemented in Windows XP.

- The LLDP agent is included by CISCO's or Hewlett Packard's managed Ethernet switches for medium-sized or large-sized networks.
- An NMS is required to gather LLDP topology information. We found two NMSs able to support LLDP: NETDISCO [33] and Solarwinds Engineer's toolset [34].

The following functionalities must be included in the home gateway to be used in our study: DHCP sever, LLTD MAPPER, LLDP Agent and NMS.

Using the following devices we constructed a suitable home gateway as shown in figure 23:

1. Linksys WRT54GL (Router and DHCP Server)

2. Dell Netbook Latitude 2100 with Microsoft OS Windows Vista (LLTD MAPPER) and NMS Solarwinds Engineer's toolset (trial version).

3. CISCO SF-300-08 Ethernet Switch for small business with LLDP and SNMP agents.

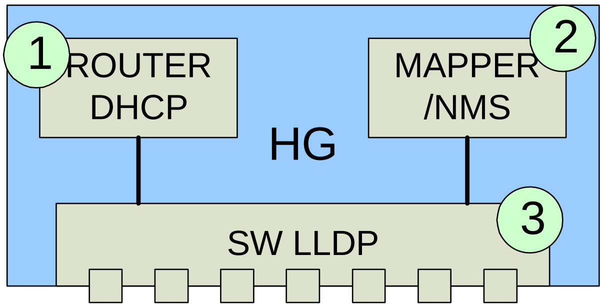


Figure 23. Home Gateway structure

The interaction between Mapper/NMS and networked devices is monitored using Wireshark installed on the Dell Netbook. Thanks to this network protocol analyzer, we are able to capture all packets sent and received by the Mapper or NMS.

4.1.2 End-user's infrastructure and end-devices

The end-user's infrastructure consists of HNIDs and physical transmission mediums deployed by the end-user to provide connectivity to end-devices. Home network product manufacturers offer three dominant types of technologies: Ethernet, WLAN and PLC. These technologies are characterized by the following HNIDs: Ethernet Switch, Wireless Access Point and Homeplugs. LLTD does not impose any requirements to HNIDs. However, LLDP needs all networked devices to support the LLDP agent. The following products are found supporting an LLDP agent:

- CISCO SF-300-08 Ethernet Switch for small business with LLDP and SNMP agents
- Hewlett Packard HP V-M200 802.11n Wireless Access Point with LLDP and SNMP agents.

Homeplugs available in the market do not support the LLDP agent or the SNMP agent. We chose homeplug Sitecom LN – 513 to include PLC technology into our testbed.

Both LLTD and LLDP require that end-devices support the LLTD responder and the LLDP agent respectively. The chosen equipment that represents a home network end-device is the Acer Netbook Aspire ONE with Windows XP which includes the LLTD responder. Several LLDP agents are available for implementation. Table 6 shows a comparison of daemons supporting LLDP functionalities.

Name	Operating System			LLDP agent mode		File Size (KB)
	Linux	Debian Package	Windows	Tx	Rx	
Lldpd	X	X		X	X	591
Openlldp	X			X	X	612
Ladvd	X			X		440
Cdpd	X		X	X		68.3
haneWIN			X	X	X	470

Table 6. LLDP agents comparison ([19],[20])

HaneWIN (trial version) and lldpd (packaged on Debian) support transmission and reception mode, and both LLDP agents are available in packages for Windows and Linux respectively. Due to these characteristics, haneWIN (trial version) and lldpd are chosen for implementation on the end-devices used for the testbed. Our netbooks run also Linux (Ubuntu) besides Windows, enabling us to install lldpd.

Table 7 summarizes the characteristics of network devices used to implement the testbed.

Device	Type	LLDP agent		LLTD	
		Tx mode	Rx mode	Responder	Mapper
HG	HG	No	yes	no	yes
Station	End-device	yes	no	yes	no
Access Point	HNID	Yes	no	no	no
Switch	HNID	yes	yes	no	no
Home Plug	HNID	no	no	no	no

Table 7. LLDP and LLTD modes

4.2 Testbed configurations

4.2.1 Basic configurations

In [27], a study of home network topologies and technologies is carried out in order to model the network dynamics in a typical Dutch household. Figure 24 presents the most common configurations.

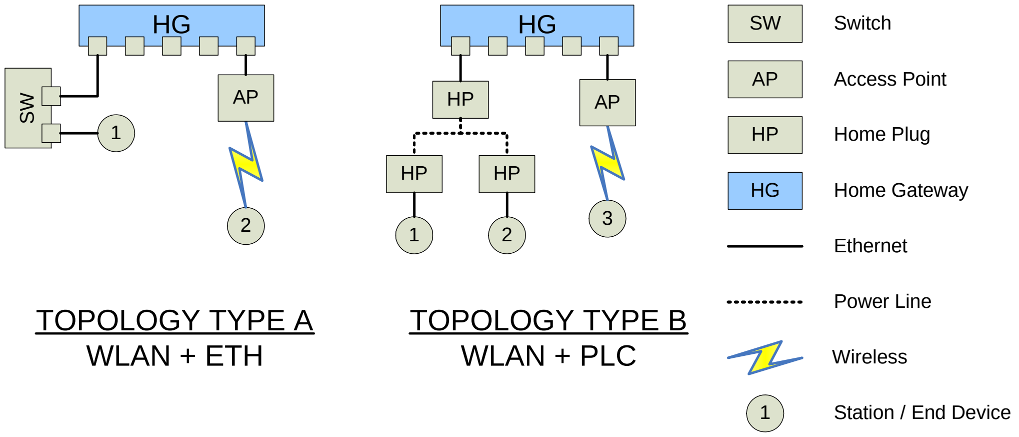


Figure 24. Common Home Network Topologies

The topology type A combines Ethernet and WLAN. In order to extend the coverage of the wired LAN, switches are incorporated into the home network (typically one or two). In the topology type B, Ethernet is replaced with PLC and the WLAN is maintained. In both situations, one AP provides the desired coverage. Based on these common home network topologies, we propose four types of testbed configurations, each one using a specific type of link layer technology. The minimum and maximum numbers of end-devices in a given configuration are one and three respectively.

4.2.1.1 Configuration Ethernet (Config Eth)

The most common configuration within home networks is a HG that provides connectivity to several devices using Ethernet as link layer technology (see figure 25). Although this technology is being challenged by others offering more flexibility in terms of mobility or installation cost, it remains popular among end-users due to its link layer capacity (100 Mbit/s) and reliability of the connectivity it offers.

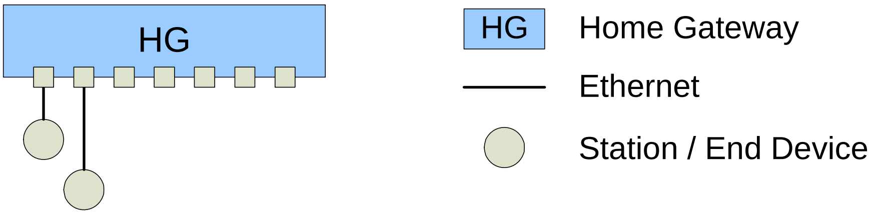


Figure 25. Testbed-Configuration Eth

4.2.1.2 Configuration Ethernet Switch (Config SW)

Ethernet is a technology where several segments can be linked through an HNID known as Ethernet switch (see figure 26). The number of ports at the HG is limited. Thanks to switches, several devices can share a single HG port. This property of Ethernet is important in future home networks where the number of end-devices is much larger than today.

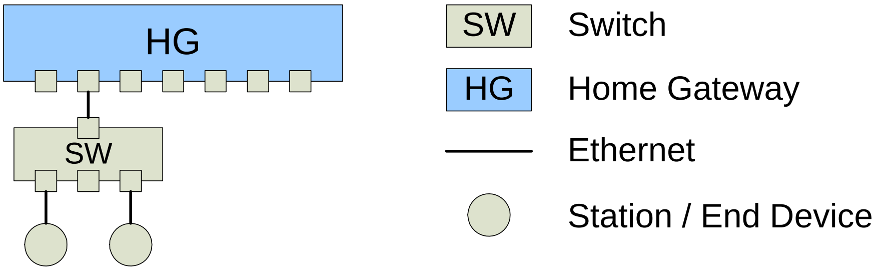


Figure 26. Testbed-Configuration SW

4.2.1.3 Configuration Power Line Communication (Config PLC)

One of the link layer technologies that challenge the dominance of Ethernet is PLC. It provides connectivity to network devices through the low-voltage electric wire system. A typical PLC network is shown in figure 27, where the HG and end-devices are connected to the low-voltage electric wire system through homeplugs.

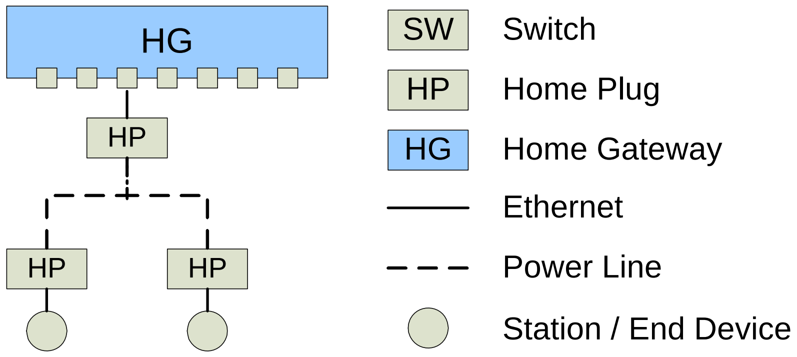


Figure 27. Testbed-Configuration PLC

4.2.1.4 Configuration Wireless LAN (Config WL)

Simplicity and flexibility of WLAN are the main reasons of the wide acceptance among end users. Nowadays, every home gateway has Ethernet ports and a WLAN port as a standard configuration. WLAN networks are often simple star configurations as shown in figure 28. Wireless devices are connected to one AP which forwards packets to or from the HG, or directly to other stations on the WLAN.

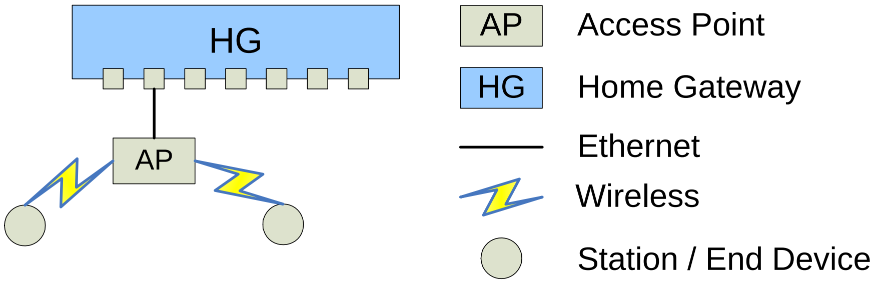


Figure 28. Testbed-Configuration WL

4.2.2 Zigbee and LLTD/LLDP compatibility issues

LLDP and LLTD are designed to work for Ethernet-like technologies. Analysis of IEEE 802.3 and IEEE 802.15.4 standards has pointed out the following sources of incompatibility of LLDP and LLTD with Zigbee:

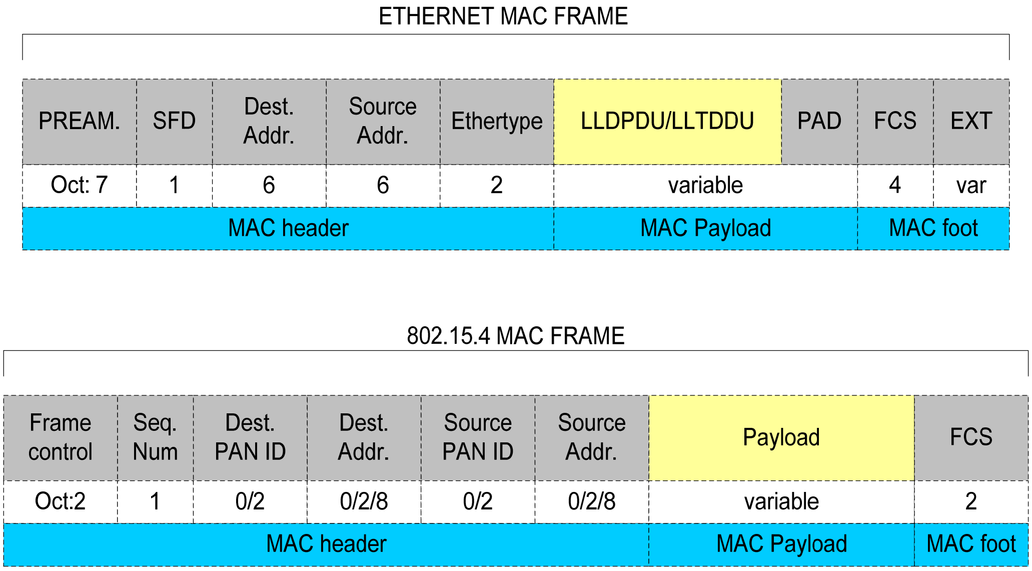
- First, the length of the MAC address in IEEE 802.15.4 is 16 or 64 bits long. IEEE 802.3 only handles 48 bits long MAC addresses.
- Second, IEEE 802.15.4 includes a second field in the MAC header to relate a device to a specific PAN. This field is called Source/Destination PAN ID. IEEE

802.3 just uses the MAC address for addressing purposes at layer 2 (see figure 29).

- Third, IEEE 802.3 uses a field called Ethertype which gives information regarding the protocol used in the payload. A similar field exists in IEEE 802.15.4 (Control Frame), but it only gives information about the MAC command frame used (see table 8).

Command frame identifier	Command name
0x01	Association request
0x02	Association response
0x03	Disassociation notification
0x04	Data request
0x05	PAN ID conflict notification
0x06	Orphan notification
0x07	Beacon request
0x08	Coordinator realignment
0x09	GTS request
0x0A-0xFF	Reserved

Table 8. Zigbee MAC command frames and [21]



ETHERNET MAC FRAME

PREAM.	SFD	Dest. Addr.	Source Addr.	Ethertype	LLDPDU/LLTDDU	PAD	FCS	EXT
Oct: 7	1	6	6	2	variable		4	var
MAC header					MAC Payload		MAC foot	

802.15.4 MAC FRAME

Frame control	Seq. Num	Dest. PAN ID	Dest. Addr.	Source PAN ID	Source Addr.	Payload	FCS
Oct:2	1	0/2	0/2/8	0/2	0/2/8	variable	2
MAC header						MAC Payload	MAC foot

Figure 29. IEEE 802.3 and IEEE 802.15.4 MAC frames

The topology discovery protocols under study are based on the Ethernet frame structure and only handle 48 bits-long MAC addresses. As a result, LLDP and LLTD cannot be used on IEEE 802.15.4 networks.

A bridge or HNID is a device that interconnects two or more segments within a LAN. According to IEEE 802.1D, which is the standard that describes the operation of these devices, the principal elements of a bridge's operation are:

- Relay and filtering of frames
- Maintenance of the information required to make frame filtering and relaying decisions
- Management of the above

Also, the standard requires that all MAC entities communicating across a bridge use 48-bit addresses.

The addition of IEEE 802.15.4 into the testbed requires a mechanism to translate MAC addresses from 16/64 bits to 48 bits (and vice versa). This functionality can not be implemented in a bridge that is IEEE 802.1D compliant. Devices exist that fulfill the role of interface between LAN and Wireless Personal Area Networks (WPAN). However, these devices are not bridges but gateways that allow internetworking between ZigBee/802.15.4 and IPv6/802.3 ([22]). This topic is out of the scope of the present master thesis project and it is not treated.

4.2.3 Congested MAC Layer

Home networks support different kind of services, some of them requiring a large part of the available bandwidth. The protocols under study are tested in non-utilized networks as well as congested networks as shown in figure 30.

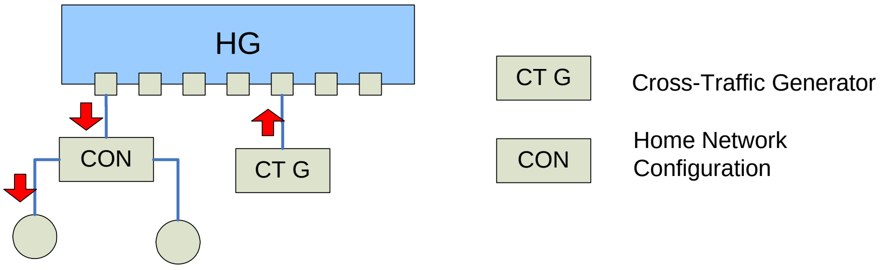


Figure 30. Testbed Congested MAC Layer

The Cross-Traffic Generator consists of a laptop running D-ITG ([23]) which is a traffic generator that sends frames to one station. The values of cross traffic are 25%, 50%, 75%

and 100% of the maximum bit rate of each link layer technology being the bottleneck in a given configuration.

4.3 Experiments and Performance Parameters

For each protocol, we perform five consecutive topology discovery processes at a rate of one process per 60 seconds. This step is repeated for every configuration for one up to three connected stations. During the operation of either protocol, Wireshark captures all packets exchanged between the HG and other devices. Based on information included in the captured packets and their timestamps, we develop our performance study.

In order to carry out a proper analysis of protocols' operation and further comparison, we have defined a set of performance indicators. They are explained in the following sections. Nobody has compared topology discovery protocols qualitatively before, so to our knowledge, we are the first to propose a set of performance indicators for this.

4.3.1 Average rate of injected traffic

The protocols under study inject traffic to the home network in order to perform the topology discovery process. This injected traffic could be probing packets or advertisement packets. By measuring this overhead traffic rate averaged over a relatively long period of time, we intend to analyze how intrusive the topology discovery technology is as a function of the number of active devices in the home network.

Wireshark can provide the evolution of injected traffic over the duration of the measurement. To calculate the average rate of injected traffic, we use:

$$\bar{f} = \frac{1}{b-a}\int_a^b f(t)dt \tag{6}$$

where $f(t)$ represents the observed traffic rate per unit of time, and $\bar{f}$ is the average value within the period *b-a* expressed in bit/s

The limits of integration depend on the time required by the protocols to perform the discovery process. In order to compare the results from different configurations, we establish a unique integration interval of 60 seconds. This value corresponds to the time between two consecutives discovery processes. We estimate the average traffic for each single topology discovery process and then calculate the mean value over the consecutive processes.

4.3.2 Discovery Time

An important performance indicator is the time required by each protocol to perform the complete topology discovery process. This discovery time can be estimated by reading the timestamp of frames captured by Wireshark.

The LLTD technical specification defines several types of frames. One of them, the Reset frame, is broadcasted when the LLTD discovery process is initiated and also when the LLTD discovery process ends. To estimate the LLTD discovery time, we read the timestamps from the first broadcasted Reset frame and from the last broadcasted Reset frame. The difference between these timestamps gives us the desired value for LLTD.

The NMS for LLDP uses the SNMP protocol to access LLDP MIBs within all HNIDs. It will send SNMP queries to all active network devices. To estimate the LLDP discovery time, we read the timestamps from the first SNMP query and from the last SNMP query sent by the NMS during the topology discovery process. The difference between these timestamps gives us the desired value for LLDP.

We measure the discovery time for each discovery process and then calculate the mean value over the consecutive processes.

4.3.3 Accuracy

The topology discovery problem can be divided into two problems:

- Discovery and classification of HNIDs within a home network according to their behavior and type of supported link layer technology.
- Creation of a proper graph representing the topology of the home network.

Our approach to estimate the accuracy of a discovery process is by analyzing these problems independently.

4.3.3.1 Classification Accuracy

A network is a collection of interconnected devices. Within a home network we can find HNIDs and Stations or End-devices.

HNIDs are layer 2 devices connecting different segments or collision domains. Within a home network a HNID can be:

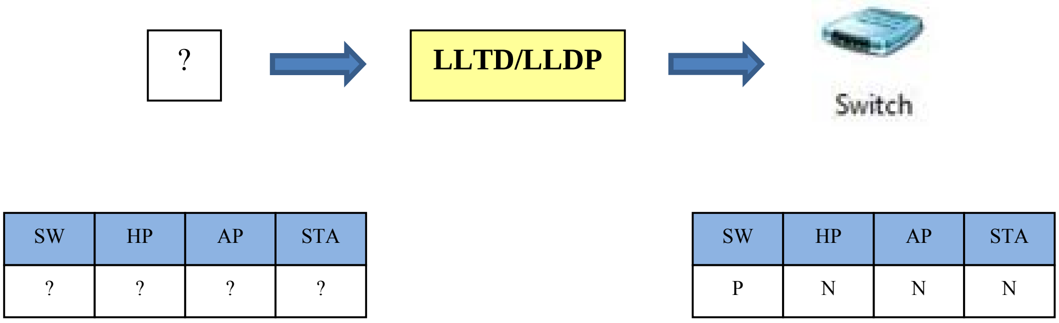
- Switch (Eth-Eth bridge)
- Access Point (Eth-WLAN bridge)
- Home Plug (Eth-PLC bridge)

In total, there are four types of devices within a home network:

- Switch (SW)
- Access Point (AP)
- Home Plug (HP)
- Station (STA)

A method used to analyze classification systems is called the Receiver Operating Characteristics (ROC) graph. This technique provides a way to visualize, organize and select classifiers based on their performance [24].

Our classifiers try to relate an unknown device to one of the four possible types according to its behavior or advertised information. The result of the match could be positive (P) or negative (N). An example is illustrated in figure 31.



SW	HP	AP	STA
?	?	?	?

SW	HP	AP	STA
P	N	N	N

Figure 31. Example of matching process

After comparing the actual type of the device and the identified type of device, we have the following possible outcomes:

- True Positive (TP): A positive match between device and type of device is correct
- False Positive (FP): A positive match between device and type of device is incorrect
- True Negative (TN): A negative match between device and type of device is correct
- False Negative (FN): A negative match between device and type of device is incorrect

Based on the number of P, N, TP and TN, an accuracy (ACC) formula is provided to evaluate different classifiers:

$$Acc_{class} = \frac{\#TP + \#TN}{\#P + \#N} \tag{7}$$

A ROC graph (see figure 32) is a two dimensional graph that represents relative trade-offs between true positives rates (TPR) and false positive rates (FPR). To calculate TPR and FPR, we use the following equations:

$$TPR = \frac{\#TP}{\#P} \tag{8}$$

and $FPR = \frac{\#FP}{\#N}$ (9)

TPR is plotted on the Y axis and FPR is plotted on the X axis. A perfect classification is represented with coordinates (0,1). Analysis from the ROC graphs establishes that a classifier A is equally good or better than a classifier B if A is to the northwest of B. This means that the following condition must be fulfilled:

$$TPR_A \geq TPR_B \wedge FPR_A \leq FPR_B \tag{10}$$

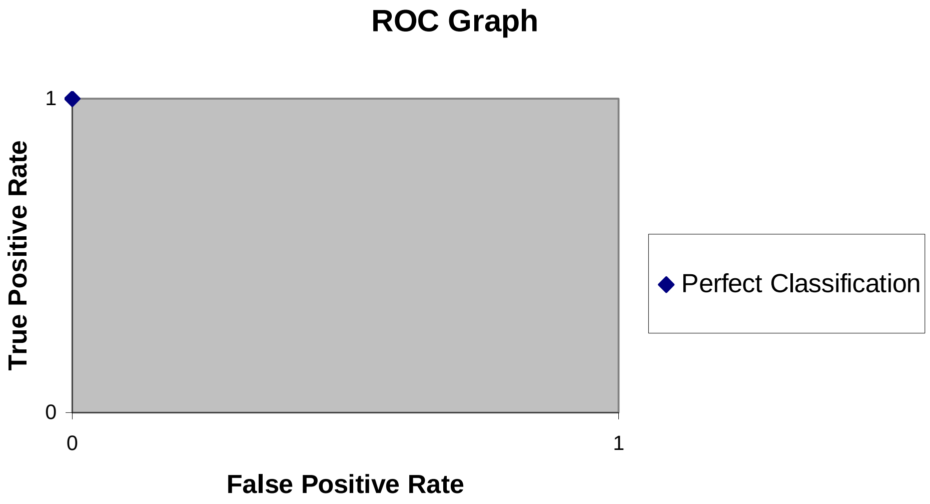


Figure 32. ROC Graph

4.3.3.2 Graph Accuracy

The classification part of both protocols identifies the type of existing network devices. Another aspect of these protocols is the identification of the network topology.

A network can be modeled by using undirected graphs [30]. Networked devices and connections among them are represented by nodes and links. A graph has a mathematical representation known as adjacency matrix. The adjacency matrix is an M by M matrix (M is the number of nodes) and its elements represent the links that exist between nodes (see figure 33). The nodes are HNIDs and stations that exist within the home network.

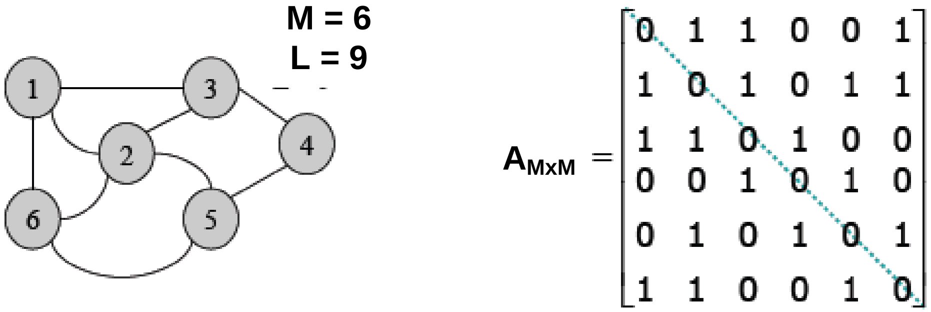


Figure 33. Example of undirected graph and adjacency matrix [30]

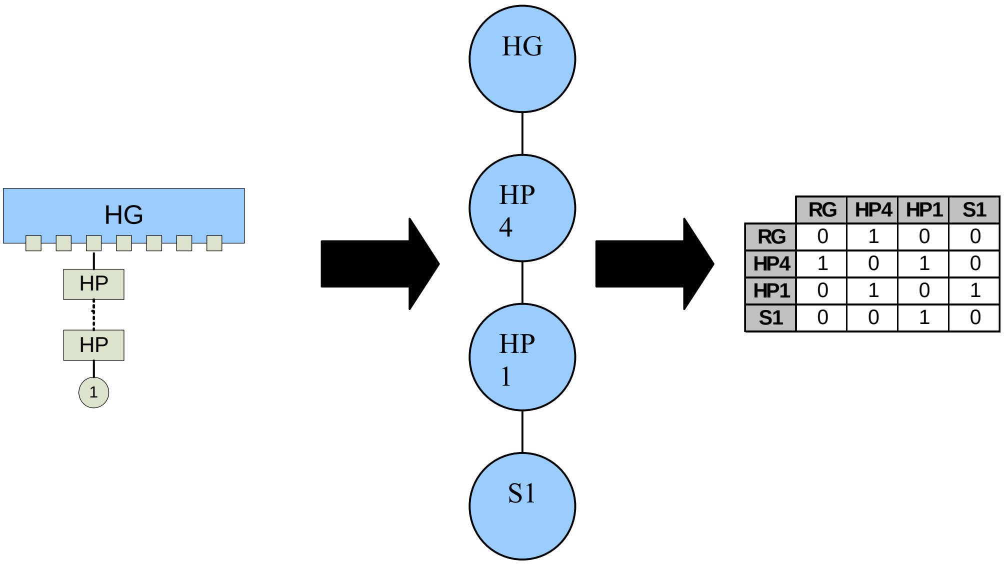
We can compare the map and the topology of the testbed by comparing the adjacency matrices of graphs representing the generated map and the real implementation. We compare both adjacency matrices and find the number of positions where values in both matrices are the same. Accuracy is calculated as:

$$Acc_{graph} = \frac{\#TP \ \ + \#TN}{M^2} \tag{11}$$

Where TP and TN are defined for links analogue to their definition for devices in the previous section

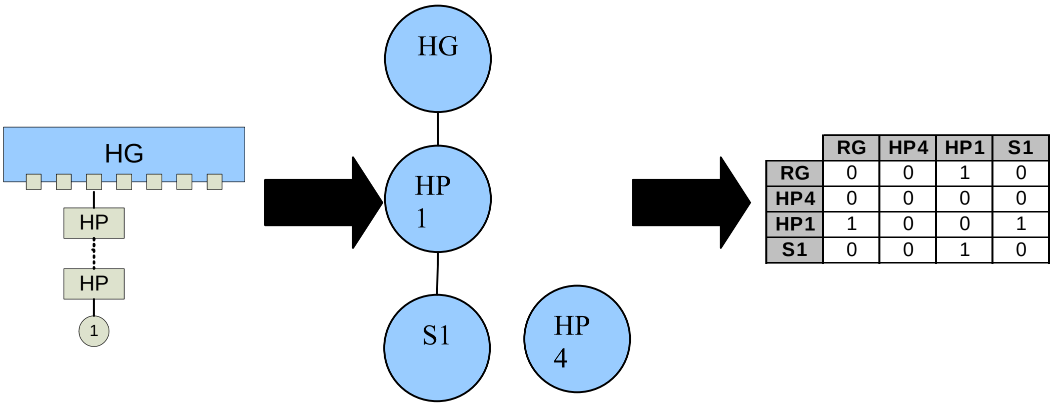
The following example shows the steps followed to calculate the topology accuracy. We use a simple configuration with one station connected to an HG using two homeplugs. The configuration, the graph representation and adjacency matrix are shown in figure 34.

Let us assume that the final map generated by a given protocol and its graph representation are as shown in figure 35. The node that we call HP4 is not included as an element of the network and it is represented as a node not linked to nodes HG, HP1 and S1. The total number of elements in one adjacency matrix is 16 and the number of positions where elements of adjacency matrices are the same is 10. According to equation (11), the topology accuracy is 62.5%.



	RG	HP4	HP1	S1
RG	0	1	0	0
HP4	1	0	1	0
HP1	0	1	0	1
S1	0	0	1	0

Figure 34. Tesbed configuration, graph and adjacency matrix



	RG	HP4	HP1	S1
RG	0	0	1	0
HP4	0	0	0	0
HP1	1	0	0	1
S1	0	0	1	0

Figure 35. Final map, graph and adjacency matrix

4.3.4 Memory Requirement

The goal of HGI is to include a topology discovery protocol in the future set of home network diagnostic tools. This tool should be supported by different home network devices such as HGs and HNIDs which usually possess limited capabilities and resources. One of the most important resources within an HG or HNID is memory. Memory must be shared among different applications and databases running within the device. To estimate the memory required by each protocol we analyze the memory space needed by daemons, the memory space required to store topology data and the memory space required to perform the tests needed to finish the task.

Windows Task Manager, which is an application included in every Microsoft's operating system, allows users to monitor the resources of their computers. This tool gives a mean to visualize the use of memory by different processes or applications (figure 36).

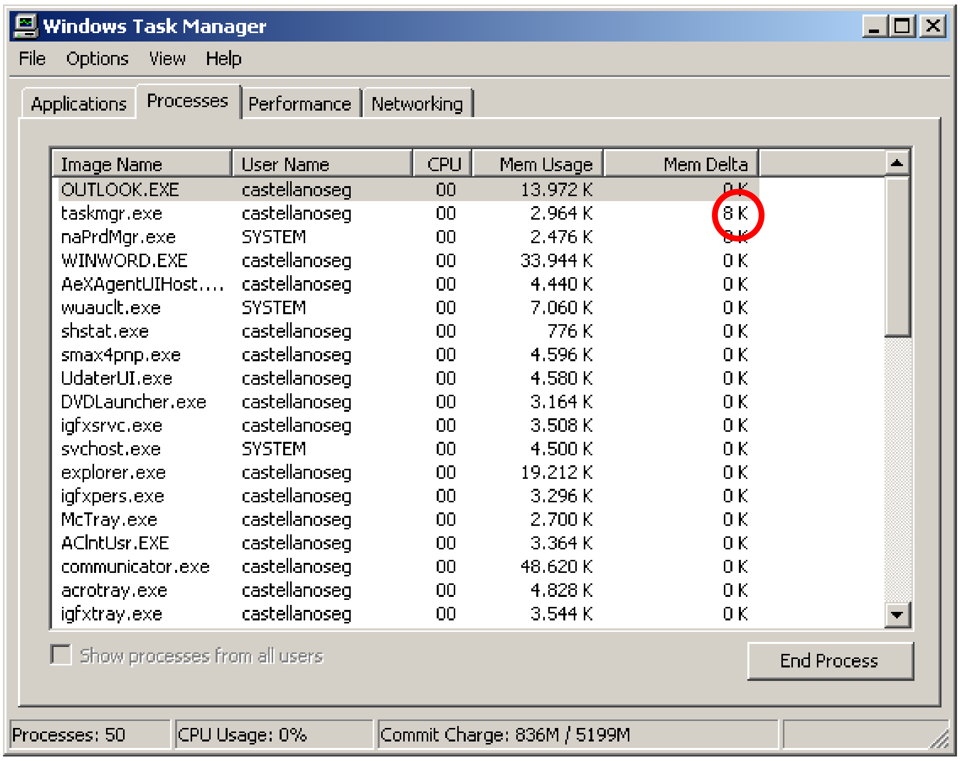


Image Name	User Name	CPU	Mem Usage	Mem Delta
OUTLOOK.EXE	castellanoseg	00	13.972 K	0 K
taskmgr.exe	castellanoseg	00	2.964 K	8 K
naPrdMgr.exe	SYSTEM	00	2.476 K	0 K
WINWORD.EXE	castellanoseg	00	33.944 K	0 K
AeXAgentUIHost....	castellanoseg	00	4.440 K	0 K
wuauclt.exe	SYSTEM	00	7.060 K	0 K
shstat.exe	castellanoseg	00	776 K	0 K
smax4pnp.exe	castellanoseg	00	4.596 K	0 K
UdaterUI.exe	castellanoseg	00	4.580 K	0 K
DVDLauncher.exe	castellanoseg	00	3.164 K	0 K
igfxsrvc.exe	castellanoseg	00	3.508 K	0 K
svchost.exe	SYSTEM	00	4.500 K	0 K
explorer.exe	castellanoseg	00	19.212 K	0 K
igfxpers.exe	castellanoseg	00	3.296 K	0 K
McTray.exe	castellanoseg	00	2.700 K	0 K
AClntUsr.EXE	castellanoseg	00	3.364 K	0 K
communicator.exe	castellanoseg	00	48.620 K	0 K
acrotray.exe	castellanoseg	00	4.828 K	0 K
igfxtray.exe	castellanoseg	00	3.544 K	0 K

Processes: 50 | CPU Usage: 0% | Commit Charge: 836M / 5199M

Figure 36. Screen dump of typical output of the Windows Task Manager

One of the options to monitor memory space is called *Mem Delta* which indicates the change of the use of memory by a specific process in KBytes. This value represents the memory space required to perform the test. *Mem usage* represents the memory space needed by daemons. The memory space required to store topology data is given by the characteristics of each protocol.

5

Results and Analysis

As mentioned in the previous chapter, we analyze the performance of LLTD and LLDP based on Discovery Time, Average rate Injected Traffic, Accuracy (classification and topology) and Memory Requirement. This chapter presents the results of experiments performed using the basic configurations. We compare the results and draw initial conclusions about the performance of LLDP and LLTD within a typical home network.

5.1 Generated Topology Maps

Each protocol has the mechanisms to visualize the topology information after the discovery process. While LLTD mapper directly generates the final map, LLDP needs an NMS to gather topology information and then generate the final map. In appendix A, we show the resulting maps of different discovery processes performed during the experiments. In the case of LLTD, the HG is represented as a combination of a mapper and a switch. In the case of LLDP, only the switch with MAC address 18:EF:63:81:A8:8E or IP address 192.168.1.221 represents the HG. It can be concluded that both protocols always generate a map and that the map is well represented and easy to read.

5.2 Discovery Time

Figure 37 shows the results of the topology discovery time measurements using the four basic configurations and up to three stations. The Y-axis represents the time required by each protocol to complete the topology discovery process. The variation of the results between measurements is less than 5%.

The results for LLTD are different for the different configurations, but always in the range of 4 - 10 s. For configuration Eth, configuration SW and configuration WL topology discovery time increases linearly, but very little (by 1% to 5%) when a new station is added. For PLC, the increment of topology discovery time is more pronounced compared to other configurations (10% to 30%).

For LLDP, results from configuration Eth, configuration SW and configuration PLC are virtually the same (all curves virtually overlap). The discovery times are also much slower than for LLTD, namely in the range of 16 – 55 s. According to results from configuration WL, topology discovery time remains the same and it is independent from the number of active stations.

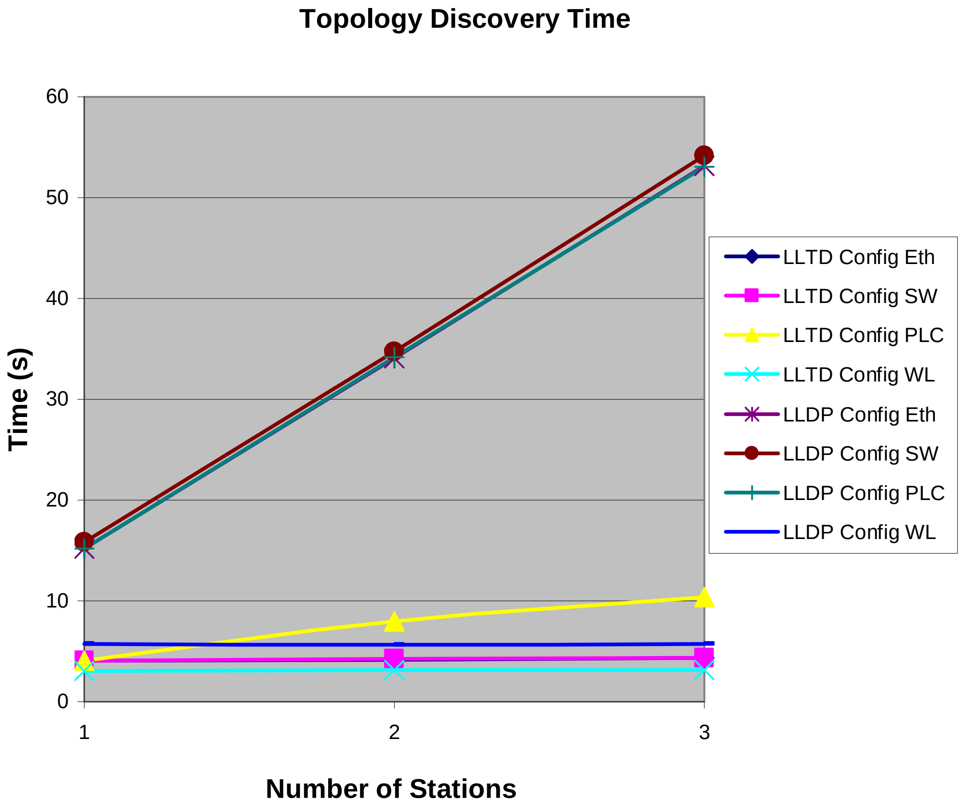


Figure 37. Topology Discovery Time LLTD/LLDP

5.3 Injected Traffic

The LLTD mapper injects probes into the network in order to trigger a behaviour of the devices in the network and from this observed behaviour it deduces the topology. This entity will create the topology map. On the other hand, LLDP topology information is distributed and stored among HNIDs in the home network. The NMS does not perform any test; it just accesses each HNID and retrieves the information about the local topology. The traffic resulting from the interaction between network devices and the mapper or the NMS can be measured. The average injected traffic for each configuration having up to three stations is shown in figure 38 where the Y-axis represents the bit rate of traffic generated by the topology discovery process. For every measurement, the standard deviation obtained from 5 observed injected traffic rate values is less than 5%.

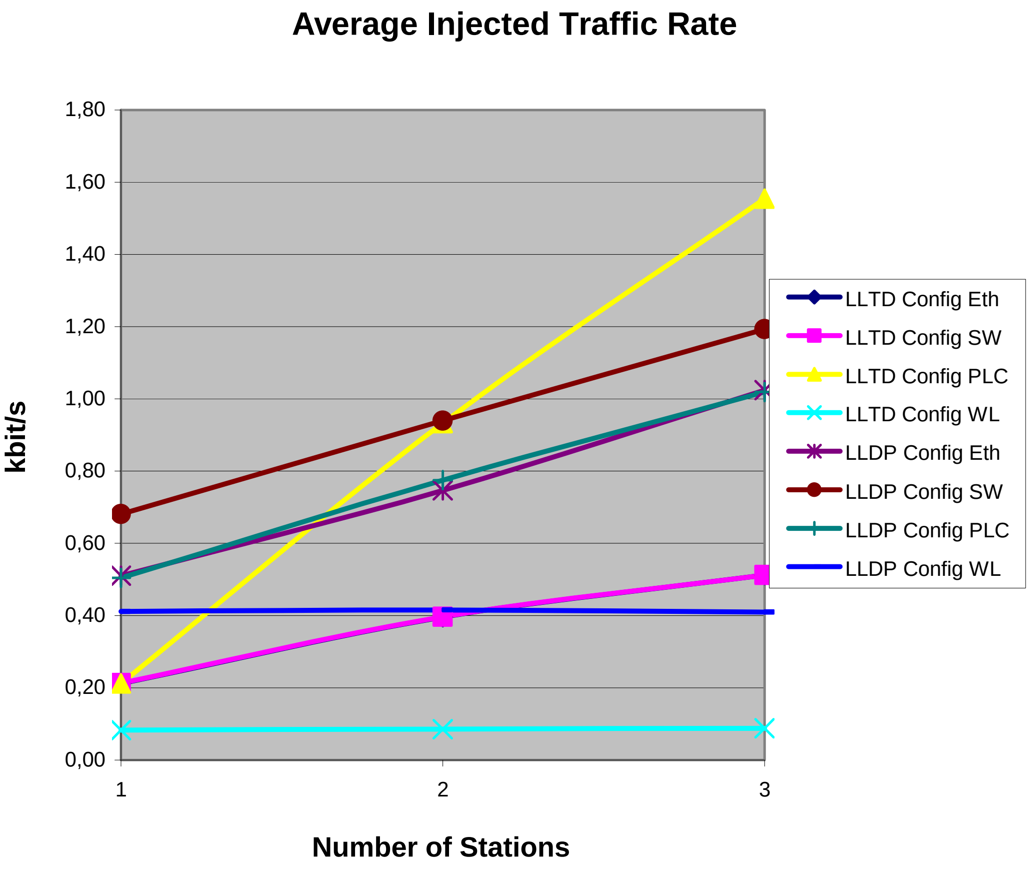


Figure 38. Average Injected Traffic Rate LLTD/LLDP

It can be observed that the average injected traffic rate for a discovery process is typically less than 2 kbits/s, which is very low. It can also be seen that the overhead traffic generated by LLTD is generally less than for LLDP with the exception for PLC in the case of plural stations.

LLTD requires only very little traffic to discover WLAN (configuration WL) and it does not depend on the number of active responders using the same segment or collision domain. For the configuration Eth somewhat more traffic is needed, but also there is no dependence on the number of active responders.

For the configuration SW, the injected traffic rate increases by adding new stations that do not share segments. In this scenario, the addition of a new station results in the addition of a new segment connecting one responder to one HNID. LLTD results from configuration Eth, configuration SW and configuration WL show that the increase of injected traffic depends on the number of segments and not on the number of active responders in the network. Configuration PLC shows an important increase of traffic with the addition of a new station using PLC as means of connectivity.

The NMS accesses each HNID with the SNMP protocol, which uses layer 3 to send and receive messages. The amount of injected traffic depends on the number of HNIDs supporting an SNMP agent and the configuration parameters of the NMS such as the number of SNMP queries and timeout. The injected IP traffic can be changed by modifying the mentioned parameters. However, it can be expected that the SNMP traffic rate is somewhat larger than the LLTD traffic most of the time because of the richer message control and the size of SNMP packets compared to the layer-2 LLTD frames. Although the NMS generates the final topology map of the network, it does not play any role when LLDP MIBs are created. In LLDP, each network device participates in the topology process by multicasting periodically LLDP frames which are layer-2 frames. One LLDP frame transmitted from every active port from network devices advertising their presence is enough to complete a discovery process. Using Wireshark, we captured LLDP frames advertised by a station and by an HNID, both supporting LLDP: the size of an LLDP frame transmitted by stations is 185 bytes and the size of a LLDP frame transmitted by HNIDs is 118 bytes.

In order to compare LLTD traffic and LLDP traffic (not SNMP traffic), we calculate the injected traffic rate per networked device and per segment. In the case of LLTD, we divide the traffic shown in figure 38 by the number of networked devices and by the number of segments that exist in each configuration used. The number of networked devices and the number of segments in each configuration are summarized in table 9.

Configuration	Case	# Network Devices (ND)	# Segments (NS)
Config Eth	1 Station	1	1
	2 Stations	2	2
	3 Stations	3	3
Config SW	1 Station	2	2
	2 Stations	3	3
	3 Stations	4	4
Config PLC	1 Station	3	3
	2 Stations	5	4
	3 Stations	7	5
Config WL	1 Station	2	2
	2 Stations	3	2
	3 Stations	4	2

Table 9. Number of Devices and Segment for every configuration

Let's call ND the number of devices and NS the number of segments. The following formula is used to calculate the traffic per device and segment for each configuration and case:

$$f_{SegDev} = \frac{\bar{f}}{NS \cdot ND} \qquad (12)$$

where f_{SegDev} represents the injected traffic per device per segment and $\bar{f}$ is the average traffic injected by the mapper.

Using equation (6) we calculate the average injected traffic rate for LLDP over 60 seconds when a LLDP agent sends one frame with size 185 bytes through one port or segment. The result is 0.0247 kbps/device/segment. Figure 39 shows the results for all configurations and for each protocol.

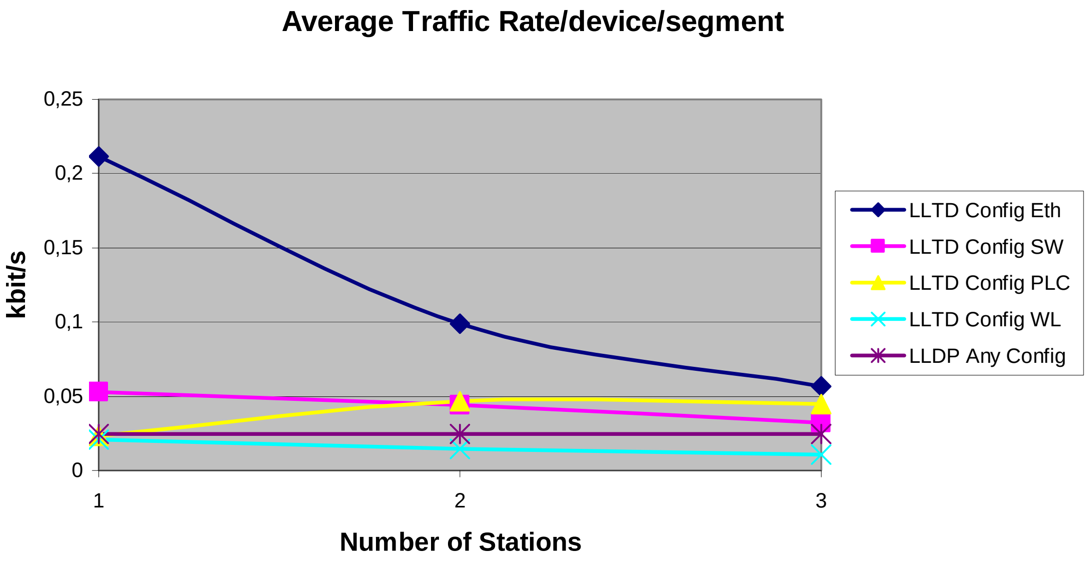


Figure 39. Average Injected Traffic Rate/device/segment

We can see that, with the exception of wireless LAN, LLDP injects on average less traffic than LLTD to perform a topology discovery process.

5.4 Accuracy

5.4.1 Classification Accuracy

Table 10 and table 11 summarize the results of LLTD and LLDP matching processes as explained in chapter 4.

Device		LLTD Matching Process			
Type	Characteristic	HP	SW	AP	STA
HP	At least one LLTD responder connected to it	N	P	N	N
	No LLTD responder connected to it	N	N	N	N
SW	At least one LLTD responder connected to it	N	P	N	N
	No LLTD responder connected to it	N	N	N	N
AP	At least one LLTD responder connected to it	N	N	P	N
	No LLTD responder connected to it	N	N	N	N
STA	Supports LLTD responder	N	N	N	P

Table 10. LLTD matching process

Device			LLDP Matching process			
Type	Characteristics		HP	SW	AP	STA
HP	No LLDP	Not compliant with 802.1D	N	N	N	N
SW	LLDP	Compliant with 802.1D	N	P	N	N
AP	LLDP (Tx only)	Compliant with 802.1D	N	N	P	N
STA	LLDP	HNID with LLDP	N	N	N	P
		HNID with LLDP (Tx only)	N	N	N	N

Table 11. LLDP matching process

After comparing the result of the matching process with the actual type of device under test, we obtain the following outcomes:

Device		LLTD Outcomes			
Type	Characteristic	HP	SW	AP	STA
HP	At least one LLTD responder connected to it	FN	FP	TN	TN
	No LLTD responder connected to it	FN	TN	TN	TN
SW	At least one LLTD responder connected to it	TN	TP	TN	TN
	No LLTD responder connected to it	TN	FN	TN	TN
AP	At least one LLTD responder connected to it	TN	TN	TP	TN
	No LLTD responder connected to it	TN	TN	FN	TN
STA	Supports LLTD responder	TN	TN	TN	TP

Table 12. LLTD Outcomes

Device			LLDP Outcomes			
Type	Characteristics		HP	SW	AP	STA
HP	No LLDP	Not compliant with 802.1D	FN	TN	TN	TN
SW	LLDP	Compliant with 802.1D	TN	TP	TN	TN
AP	LLDP (Tx only)	Compliant with 802.1D	TN	TN	TP	TN
STA	LLDP	HNID with LLDP	TN	TN	TN	TP
		HNID with LLDP (Tx only)	TN	TN	TN	FN

Table 13. LLDP Outcomes

False negatives outcomes are obtained when the unknown device is not connected to a LLTD responder device or when a HNID does not support a full transmission/reception enabled LLDP agent. This result is consequence of mechanisms and philosophies applied for each protocol. Only LLTD provides a false positive outcome when the protocol tries to classify PLC HNID. According to the map generated by the LLTD mapper, PLC technology is modeled as combination of a hub representing the PLC collision domain and a switch representing the Homeplug (see figure 40). The current version of LLTD was not designed to identify PLC as a home networking technology.

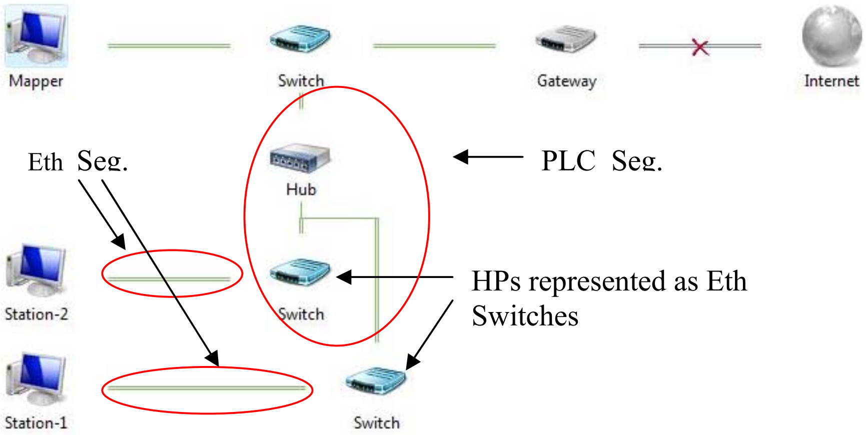


Figure 40. LLTD MAP – PLC representation

Using information from table 12, table 13, equation (7), equation (8) and equation (9), we calculate the classification accuracy, the TPR and FPR values for each protocol as shown in table 14 and figure 41. From both the ROC graph and the accuracy values, it can be established that LLDP shows better classification accuracy performance compared to LLTD.

	LLTD	LLDP
Number of TP	3	3
Number of TN	20	15
Number of FP	1	0
Number of FN	4	2
Number of P	4	3
Number of N	24	17
Accuracy	82%	90%

Table 14. Classification Accuracy Values

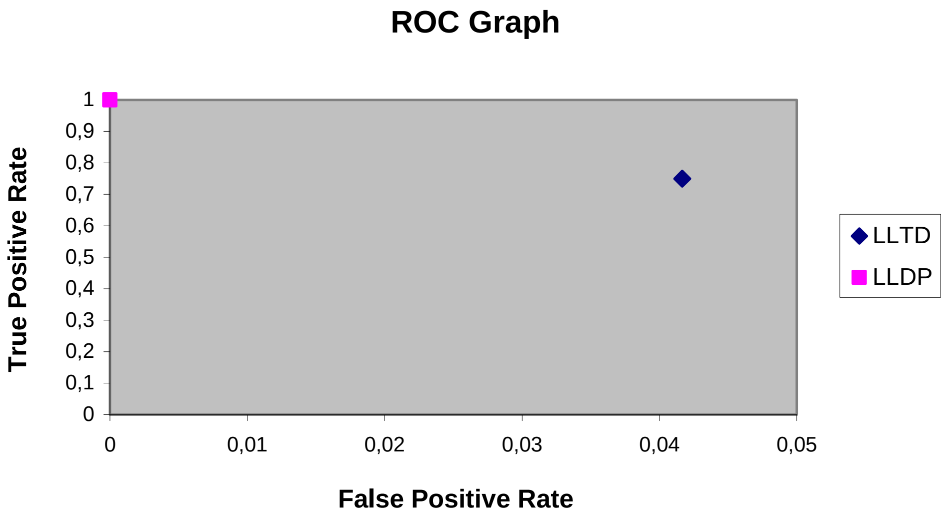


Figure 41. ROC Graph Results

5.4.2 Topology Accuracy

Topology accuracy is obtained by comparing the adjacency matrices from the original configuration and the map generated by each protocol. Appendix A includes adjacency matrices of four basic configurations, adjacency matrices from maps generated by LLTD and adjacency matrices from maps generated by LLDP. Using equation (11) we calculate the values of topology accuracy for different configurations. As shown in table 15, LLTD has an equal or better topology accuracy compared to LLDP for every configuration.

Accuracy Topology			
Configuration	**Case**	**LLTD**	**LLDP**
Config Eth	1 Station	100%	100%
	2 Stations	100%	100%
	3 Stations	100%	100%
Config SW	1 Station	100%	100%
	2 Stations	100%	100%
	3 Stations	100%	100%
Config PLC	1 Station	63%	50%
	2 Stations	72%	56%
	3 Stations	83%	59%
Config WL	1 Station	100%	78%
	2 Stations	100%	75%
	3 Stations	100%	76%

Table 15. Accuracy Topology Values

5.5 Memory requirement

We use Windows Task Manager to monitor the memory usage required by LLTD when the topology discovery process is activated. Table 16 shows the values of *Mem Delta* provided by the monitoring tool for five consecutive topology discovery processes per measurement. We observed that the LLTD mapper requires a minimum of 44 Kbytes and a maximum of 120 Kbytes to operate. LLTD service is supported by a dll extension file included in the operating system. This file is called *lltdsvc.dll* and its size is 188 Kbytes. In total, Microsoft's topology discovery tool thus requires 232 Kbytes to 308 Kbytes of memory.

Devices supporting LLDP multicast frames carrying topology data as a sequence of TLVs. Information included in most of the TLVs is retrieved from a LLDP MIB stored within HNIDs. The maximum data size of each relevant TLV is defined by the IEEE 802.1AB standard (see table 17).

If we use all mandatory and optional TLV types, and assuming a maximum data size for each field, for each neighbor device an LLDP HNID will require 1446 Bytes to store its information. Also, the LLDP HNID must support one of the LLDP daemons mentioned in table 4. In the testbed the size of the LLDP daemon used is approximately 500 Kbytes. In the worst scenario (3 stations), an LLDP HNID thus needs approximately 505 Kbytes of total memory.

Configuration	Case	Delta (KByte)					AV ± SD
		Disc. 1	Disc. 2	Disc. 3	Disc. 4	Disc. 5	
Config Eth	1 Station	56	56	56	120	56	68.8 ± 42%
	2 Stations	56	44	56	56	120	66.4 ± 46%
	3 Stations	104	56	32	52	56	60 ± 44%
Config SW	1 Station	60	56	56	56	56	56.8 ± 3%
	2 Stations	56	116	56	56	36	64 ± 47%
	3 Stations	56	56	120	56	68	71.2 ± 39%
Config PLC	1 Station	56	80	56	56	56	60.8 ± 18%
	2 Stations	56	56	56	56	56	56 ± 0%
	3 Stations	40	52	56	60	56	52.8 ± 15%
Config WL	1 Station	56	56	60	60	48	56 ± 9%
	2 Stations	56	64	56	56	56	57.6 ± 6%
	3 Stations	116	56	92	76	84	84.8 ± 26%

Table 16. LLTD memory usage

TLV name	Usage	Data Size (Bytes)
Chassis ID	Mandatory	1 to 255
Port ID	Mandatory	1 to 255
Port Description	Optional	1 to 255
System Name	Optional	1 to 255
System Description	Optional	1 to 255
System Capabilities	Optional	4
Management Address	Optional	9 to 167

Table 17. LLDP TLVs Data Size

5.6 Results obtained with a congested MAC layer

Figure 42, figure 43 and figure 44 show the results of our performance indicators when measurements are done with a congested MAC layer. Only configurations with three stations are considered. Figure 42 shows the topology discovery time (Y-axis) for different values of cross-traffic. Figure 43 shows the average injected traffic (Y-axis) for different values of cross-traffic. Figure 44 shows the topology accuracy (Y-axis) for different values of cross-traffic.

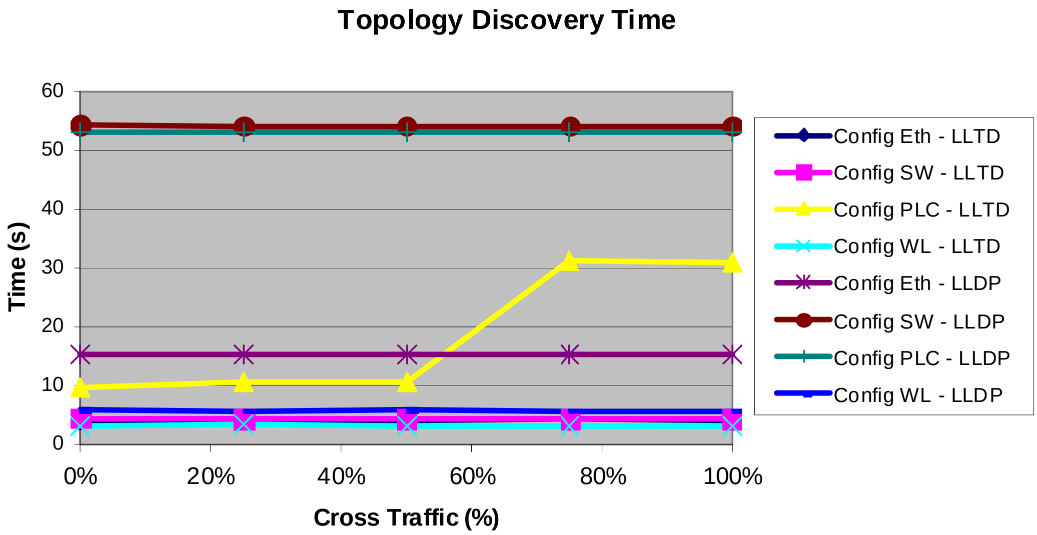


Figure 42. Topology Discovery Time LLTD/LLDP – Congested MAC layer

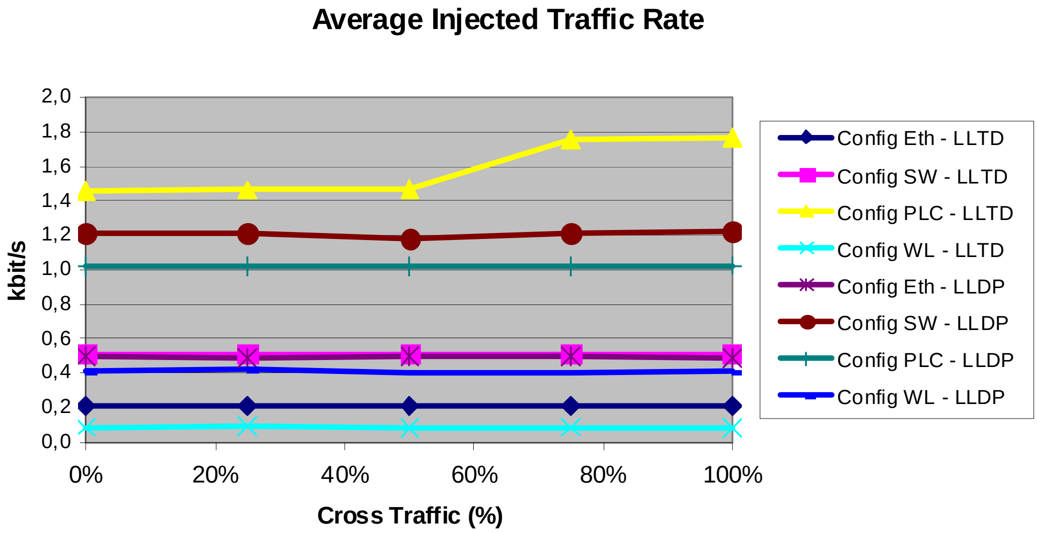


Figure 43. Average Injected Traffic Rate LLTD/LLDP – Congested MAC layer

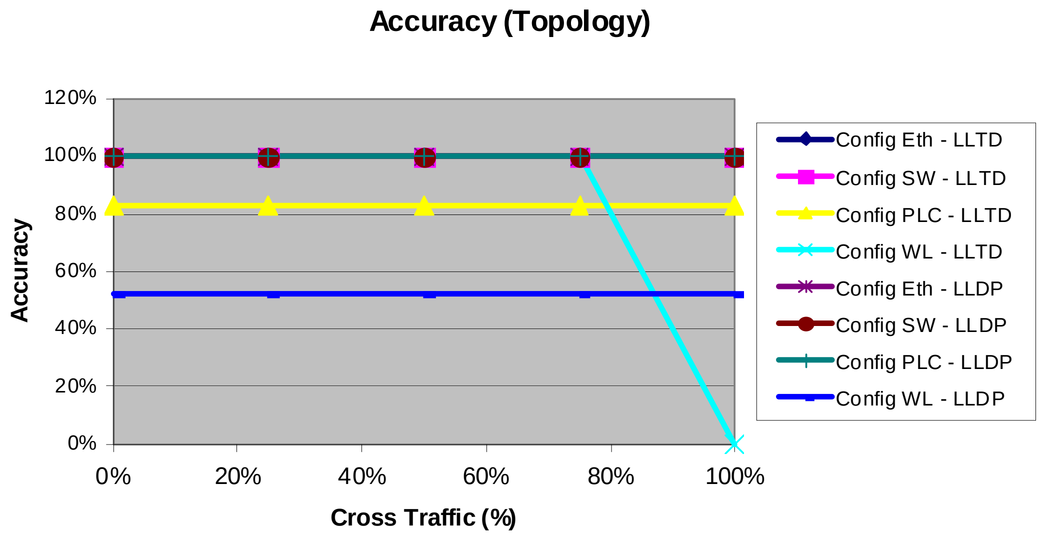


Figure 44. Topology Accuracy – Congested MAC layer

The results show that, while LLDP remains unaffected by the congested MAC layer, the values of the LLTD performance indicators can vary with the amount of cross-traffic. The topology accuracy for configuration WL, the topology discovery time for configuration PLC and the average injected traffic rate for configuration PLC show a decrease of LLTD’s performance at cross traffic rates higher than 60%.

5.7 Analysis

Throughout several meetings, HGI´s members have discussed the need of an appropriate topology discovery protocol for home networks. Observations from service providers can be summarized into the following basic requirements:

- **Requirement 1**: The topology discovery time must be less than 2 seconds.
- **Requirement 2**: The accuracy must be very close to 100% (both for classification and topology).
- **Requirement 3**: The injected traffic should not affect the operation of other services.
- **Requirement 4**: The architecture should not depend on proprietary and IPR restricted standards or protocols, unless it is within the span of control of HGI.

In this section, we analyse the obtained results from our experiments and verify the fulfilment of these service providers' requirements.

5.7.1 Basic Configurations

LLDP topology discovery times appear to be the same for configuration Eth, configuration SW and configuration PLC. This parameter does not depend on the type of the link layer technology to be identified. In the case of configuration WL, the topology discovery time is also independent of the number of stations. The final map generated by the NMS (appendix A) only shows the HNID (access point), but it does not show the stations supporting LLDP agents. According to initial conditions (table 7), the LLDP agent supported by the AP has only the transmission mode enabled. As a result, the access point's MIB does not store information from wireless stations sending LLDP frames.

For configuration Eth and configuration PLC, the LLDP average injected traffic rate is the same. Although these configurations are characterized by different technologies, they have the same number of devices supporting the IP stack: 1 to 3 stations and one HG. Configuration SW includes an IP-enabled SNMP supported Ethernet switch that thus increases the amount of traffic injected by the NMS. While configuration WL has the same number of devices supporting the IP stack as configuration Eth, the injected traffic is less compared to other scenarios. From these results we conclude that the strategy followed by the NMS is to try to access all devices included in the list of neighbors of HNIDs.

For all scenarios the average injected traffic rate by NMS or LLDP autonomous traffic represents less than 1% of any home networking technology capacity and it therefore hardly affects other services (Req. 3). On the other hand, the topology discovery time is generally larger than the time required by service providers (Req. 1).

The LLTD discovery process requires QD and TDT services to be activated. The TDT service possesses four phases: segment detection, switches detection, island edges detection and gaps detection. Thanks to inspection of captured packets by Wireshark, we are able to find out which phases are carried out during the discovery of different configurations. The most representative frames injected by the mapper are the Discover frames, Emit frames and Reset frames. Table 18 shows the statistics regarding the number of packets injected during one discovery process for every configuration.

	Config Eth			Config SW			Config PLC			Config WL		
Number of STA	1	2	3	1	2	3	1	2	3	1	2	3
Discover	9	9	9	9	9	9	9	9	9	9	9	9
Emit	9	18	27	9	18	27	9	34	79	0	0	0
Query	4	22	28	4	22	28	4	46	88	1	1	1
Reset	6	6	6	6	6	6	6	6	6	6	6	6

Table 18. Number of LLTD frames by type

For every type of configuration the mapper injects 9 Discover frames into the network which are used to find the available LLTD responders. As we can see in configuration WL, the mapper does not inject Emit frames. The discovery of WLAN and wireless stations only requires a QD service.

Configuration Eth and Configuration SW are full Ethernet networks where several shallow segments are linked using Ethernet switches. In configuration SW there is only one deep segment and no gap. According to table 18, the number of emit frames are the same for both configurations and increases with the number of shallow segments (not with the number of switches). For these configurations only QD, the TDT segment detection phase and the TDT switches detection phase are required. There are no gaps that could trigger the TDT gaps detection phase. On the other hand, the results for configuration PLC show an important increase of injected Emit frames when more than one homeplug is used in the testbed. As we can see in figure 40, the PLC segment is identified as an Ethernet Hub that links two or more deep segments. PLC is modeled as a gap linking several HNIDs. LLTD will trigger the TDT gaps detection phase to complete the discovery process. The presence of a gap within a home network will thus require more processing from the LLTD mapper. The performance of LLTD is apparently affected when the protocol is faced with topologies relying on technologies using one collision domain to connect several HNIDs. The result is an important increase of discovery time and injected traffic rate. As mentioned for LLDP, the LLTD injected traffic does not affect other services’ performance though (Req. 3). Although the LLTD topology discovery time is low compared to LLDP, in general its value is still above the service providers’ requirement (Req. 1).

The advertisement mechanism is the major strength of LLDP from the classification accuracy point of view. While LLDP devices reveal their nature to the HG, the LLTD mapper makes the best guess based on information provided by responders. This explains the better classification accuracy of LLDP. Unfortunately, the characteristics of homeplug devices and the access point weaken LLDP’s topology accuracy performance. These devices do not support a full operational LLDP agent. This situation explains the better topology accuracy performance from LLTD. Both protocols failed to reach 100%

of topology accuracy and classification accuracy as demanded by service providers (Req. 2).

The LLTD tool requires memory to support the topology discovery algorithm and to display the final map. It employs between 232 Kbytes and 308 Kbytes of memory. As conceived by Microsoft, this tool does not store the topology information or update it when there is a change in the network's configuration. LLDP bases its operation on the storage of topology information and a continuous update process which explains a larger amount of needed memory (505 Kbytes) compare to LLTD. Although not explicitly demanded by service providers, both numbers are well in line with the available memory in today's home gateways

Table 19 summarizes which requirements are fully fulfilled (+), which requirements are partially fulfilled (0) and which requirements are not fulfilled (–) by each protocol.

	Req 1	**Req 2**	**Req 3**	**Req 4**
LLTD	–	0	+	–
LLDP	–	0	+	+

Table 19. Fulfillment service providers' requirements

Thus, none of the protocols fulfills all service providers' requirements. This result has been adopted recently by HGI [35].

5.7.2 Congested MAC Layer

The LLDP topology information is stored within HNIDs and continuously updated by the LLDP agents. This mechanism provides robustness to this protocol when the MAC layer is congested, as is validated in figure 42, figure 43 and figure 44. On the other hand, LLTD must trigger the complete discovery process when a map is required. This means that the mapper must inject probes into the network and wait for replies from the responders. The tests mechanism is therefore vulnerable to delays and packet loss. Our results indicate that LLTD can show a weak performance when WLAN and PLC are used in the home network. The generated cross traffic during the experiments is apparently not enough to cause delay or packet loss in the congested Ethernet medium (configuration Eth and configuration SW).

In table 20, we analyze the flow of injected LLTD packets by the mapper for configurations using WLAN and PLC. The generated cross traffic is not enough to cause a massive loss of packets in the congested PLC medium, but it is enough to cause delays in the transmission. After analyzing the data from the measurements, we can see that under a congested PLC medium the mapper sends duplicates of Emit frames. Although ACK frames are sent, the transmission is delayed due to the congestion. As a result, the mapper does not receive the ACK frame on time and sends duplicates of Emit frames. This causes an increase of injected traffic and also an increase of discovery time.

The mapper does not receive Hello frames from stations connected to the WLAN when cross-traffic is 100% of the maximum layer-2 bit rate. Discover frames sent by the mapper or Hello frames sent by stations are lost in a congested wireless medium. Due to the dependence of LLTD on if being supported by stations, the mapper is not able to perform the topology discovery process in the situation described above. Other consequences are the decrease of the injected traffic and discovery time.

Type Packet	CT - % of max L2 bit rate									
	PLC					WLAN				
	0%	25%	50%	75%	100%	0%	25%	50%	75%	100%
Discover	9	9	9	10	10	9	9	9	9	8
Hello	4	4	4	4	4	4	4	4	5	1
Emit	61	63	62	86	87	0	0	0	0	0
Train	13	13	13	13	13	0	0	0	0	0
Probe	38	38	38	38	38	2	2	2	2	1
Ack	59	59	60	82	79	0	0	0	0	0
Query	69	71	71	87	85	1	1	1	1	0
QueryResp	69	69	71	87	85	1	1	1	1	0
Reset	6	6	6	6	6	6	6	6	6	6
Charge	120	120	121	124	126	0	0	0	0	0
Flat	0	0	0	0	2	0	0	0	0	0
QueryLargeTlv	0	0	0	0	0	0	0	0	0	0
QueryLargeTlvResp	0	0	0	0	0	0	0	0	0	0

Table 20. Injected LLTD packets – Congested PLC and WLAN medium

5.7.3 Strengths and Weaknesses

Table 21 summarizes the observed strengths and weaknesses of LLTD and LLDP.

	LLTD	LLDP
Strengths	• Relies only on end-devices to operate. • Does not need an NMS • Requires less time to generate a final map compared to LLDP/NMS	• Open standard • Automatic advertisements cause topology to be updated instantaneously. • Advertisement mechanism provides the best classification accuracy. • Performance is independent of the type of technology. • Less injected layer 2 traffic compared to LLTD. • Robust when MAC layer is congested.
Weaknesses	• IPR restricted • Topology only generated after test is manually initiated. • Performance is affected by the presence of gaps or by a congested MAC layer. • All end-devices must support the responder functionality	• HNIDs should support managements agents (e.g. MIBs, SNMP). • All networked devices should support an LLDP agent. • The use of SNMP to access all HNIDs delays the discovery process.

Table 21. Strengths and weaknesses

6

Improvement

The protocols under study differ in their operation and behavior under different conditions. A common weakness is their dependence on specific daemons to discover the topology of a network. As they are conceived, these protocols do not use other protocols to improve the final result. As mentioned in [8], an HGI recommended diagnostic tool could be able to use different sources of information in order to achieve the desired result. Following the HGI's guidelines, in this chapter we propose a topology discovery architecture aiming at providing a suitable solution for all common home network topologies.

6.1 Requirements for improved architecture

To the four basic requirements mentioned in section 5.7, we add 5 new requirements based on our analysis of strengths and weaknesses of LLDT and LLDP. The requirements for the improved architecture are:

- **Requirement 1**: The topology discovery time must be less than 2 seconds.
- **Requirement 2**: The accuracy should be very close to 100% (both for classification and topology).
- **Requirement 3**: The Injected traffic should not affect the operation of other services.
- **Requirement 4**: The architecture should not depend on proprietary and IPR restricted standards or protocols, unless they are within the span of control of HGI.
- **Requirement 5**: The architecture should be able to use different sources of information to complete the task.

- **Requirement 6**: The architecture should be able to handle end-devices not supporting an LLTD responder or LLDP agent.
- **Requirement 7**: The architecture should avoid the need for management agents, such as SNMP agent or TR-069 agent, within HNIDs. Having an IP stack within HNIDs should not be a requirement for topology discovery operation.
- **Requirement 8**: The topology information should be centralized and stored in the Home Gateway. It should be accessible through TR-069.
- **Requirement 9**: The topology information should be automatically updated when there is a change in the network's configuration.

6.2 Proposal of architecture

As stated in [9], much topology information can be inferred from AFTs stored within HNIDs. Thanks to layer 2 forwarding mechanisms, we can discover the active networked devices by their MAC addresses. The challenge of using AFTs to infer the topology of the network is to find the connections between HNIDs based on Direct Connection and Shared Connection theorems or the Simple Connection theorem. A disadvantage of the method described in [9] is the inability of identifying the type of link technology because this information is not included in AFTs. LLDP ([13]) and its modified version HTIP ([17]) provide a mean to identify the direct and shared connections that can be deduced from the neighbor list and also the type of technologies used within the network. We propose therefore to combine the AFTs information, the LLDP neighbor list and the HTIP frame format into a protocol which we call Home Network Topology Discovery (HNTD).

The architecture shown in figure 45 defines the various HNTD's entities that will cooperate to centralize AFTs and neighbor lists information from all HNIDs into the home gateway as required by HNTD protocol.

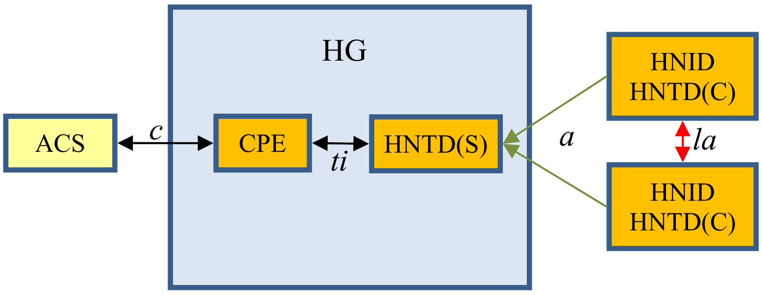


Figure 45. Our architecture for topology discovery with the HNTD protocol

The entities of the proposed architecture are:

- A TR-069 compliant Auto Configuration Server (ACS)
- Customer Premises Equipment (CPE)
- HNTD Client (HNTD(C))
- HNTD Server (HNTD(S))

The interfaces are:

- *a* advertisement: Interface between HNIDs and the HG
- *la* (local advertisement): Interface between two or more neighbor HNIDs
- *ti* topology information: Interface between HNTD(S) and CPE
- *c* CWMP: Interface between CPE and ACS, as standardized by TR-069.

6.2.1 Definition of the new entities

6.2.1.1 Home Network Topology Discovery Client

HNTD(C) is an LLDP agent modified to send an HNID's information to other HNIDs and to the HG within the home network under test.

Two main processes are defined:

- HNTD(C) advertises the presence of an HNID and also receives advertised information from neighboring HNIDs. This interaction is represented by interface *la*.
- HNTD(C) sends the list of neighbors and the AFT from an HNID to the HG. This interaction is represented by interface *a*.

The frames used are LLDP_Hello and LLDP_Info (see table 22).

6.2.1.2 Home Network Topology Discovery Server

HNTD(S) is an LLDP agent modified to include some of LLTD functionalities to interact with other networked devices using layer-2. Also HNTD(S) receives a neighbor list and the AFT from HNIDs (interface *a*). The frames used are LLDP_Hello, LLDP_Ack and LLDP_Query (see table 22). HNTD(S) shares topology information with the CPE (interface *ti*) in order to make it available to service provider's ACS. The CPE may be the HG or the HG may include the CPE functionality (as defined in TR-069) as well as the HNTD(S). In case of the latter, interface *ti* is internal.

Type	Description	DST MAC Address	Propagation
LLDP_Hello	Frame used to advertise HNID presence	01:80:C2:00:00:0e	Multicast within segment
		FF:FF:FF:FF:FF:FF	Broadcast
LLDP_Ack	Frame used to acknowledge the reception of a LLDP_Info frame.	HNID of interest	Unicast
LLDP_Query	Frame used to ask for transmission of forwarding tables from HNIDs	FF:FF:FF:FF:FF:FF	Broadcast
LLDP_Info	Frame used to send neighbor list and MAC forwarding/association table.	HG	Unicast

Table 22. Frames New Architecure

6.2.2 Operation

HNTD protocol can be described as a five phases process. Figure 46 shows the operation of HNTD(C) and HNTD(S).

Phase 1 – To advertise HNID presence

HNIDs advertise their presence by sending LLDP_Hello frames with destination MAC address 01:80:C2:00:00:0e (interface *la*). HNIDs generate a list of neighbors. The transmission of LLDP_Hello frames to advertise HNID presence is periodic (autonomous traffic).

Phase 2 – To gather HNIDs information

HNIDs will broadcast their AFTs periodically, using an LLDP_Info frame, until receiving an LLDP_ACK frame (interface *a*). We use the same format as HTIP with a slight change in the data format to send AFT information. We intend to include a neighbor field (NF), as shown in figure 47, for every MAC address indicating if the device is a neighbor (0xFF) or not (0x00).

After receiving an LLDP_ACK frame, HNIDs will broadcast an LLDP_Hello frame periodically. A new LLDP_info frame will be generated only when there is a change in the AFT. The HG can initiate the transmission of AFTs from HNIDs by sending an LLDP_Query frame. This action takes place when one of the bridges does not advertise its presence.

Phase 3 – New Neighbor List for each HNID

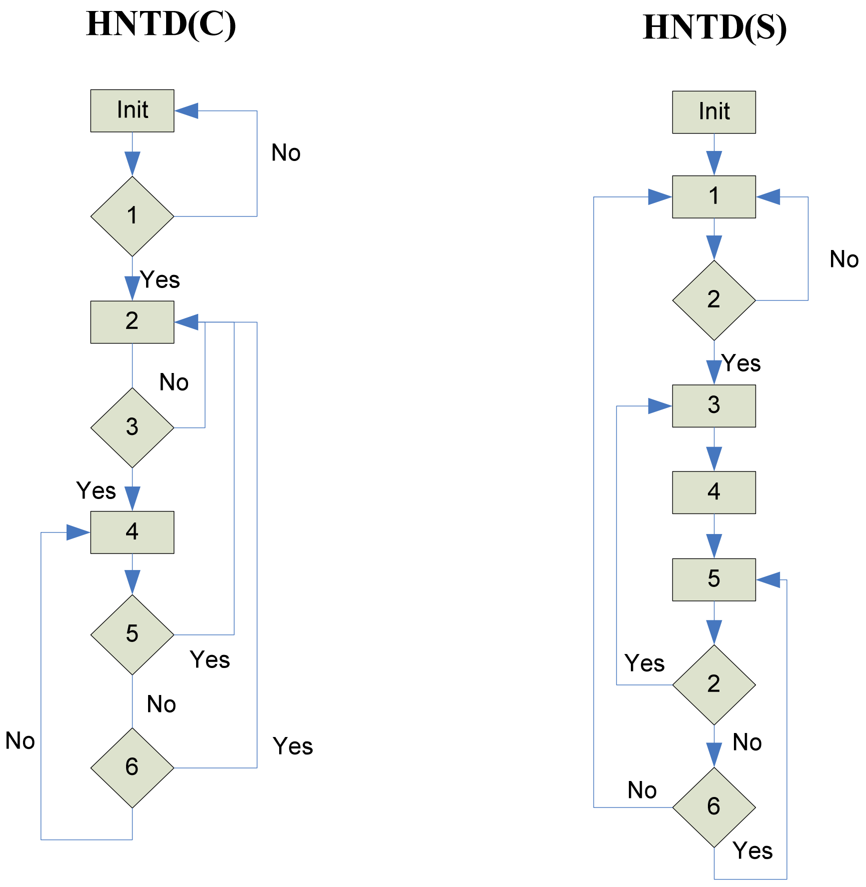
After receiving AFTs and neighbor lists, the HG performs a topology discovery algorithm that creates a new neighbor list for each HNID that includes neighbor HNIDs as well as neighbor stations. The algorithm is explained in the following sections.

Phase 4 – To store New Neighbor List using TR-069 Data Model

Neighbor lists for all HNIDs are stored in the TR-069 Data model of the CPE module.

Phase 5 – To send information ACS

Topology information stored in the HG is sent to the ACS when requested. Based on the received information, the ACS generates a layer 2 topology map.



1. Reception LLDP_Query HG
2. Transmission of LLDP_Info
3. Reception of LLDP_Ack
4. Transmission LLDP_Hello
5. Reception LLDP_Query
6. Change of ATF

1. Transmission LLDP_Query
2. Reception LLDP_Info
3. Transmission LLDP_Ack
4. Topology Algorithm
5. Store Data
6. Reception LLDP hello

Figure 46. HNTD state machines

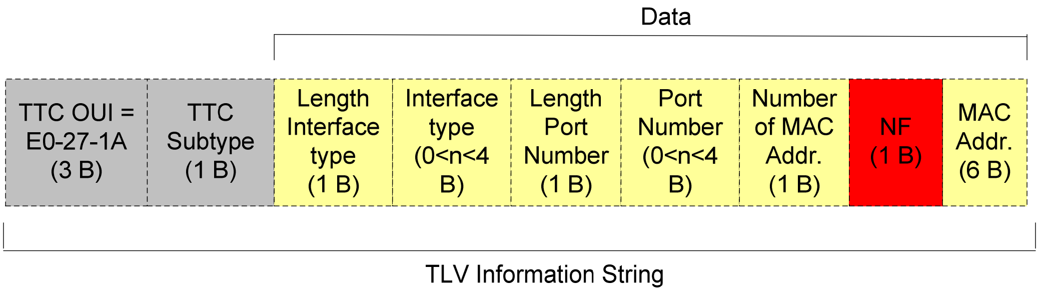


Figure 47. TLV format to send AFT and neighbor list

6.2.3 Topology Discovery Process

The architecture described above is a proposal to establish mechanisms to gather the required information (AFTs and neighbors list), to perform topology discovery. After receiving the information, the HG needs to process the data in order to store the topology information in a proper format. In the following paragraphs, we describe an algorithm able to infer network topology using the neighbor list and AFTs.

Assumption:

- All HNIDs support the HNTD(C) agent.

As an example, we have three bridges (B1, B2, B3), one HG and 6 stations. The AFTs are shown in figure 48.

AFT											
		Nodes									
Bridge	Port	HG	B1	B2	B3	A	B	C	D	E	F
B1	1	X		X	X		X	X	X	X	X
	2					X					
B2	1	X	X		X	X			X	X	X
	2						X				
	3							X			
B3	1	X	X	X		X	X	X	X		X
	2									X	
HG	1		X		X	X				X	X
	2			X			X	X			
	3								X		

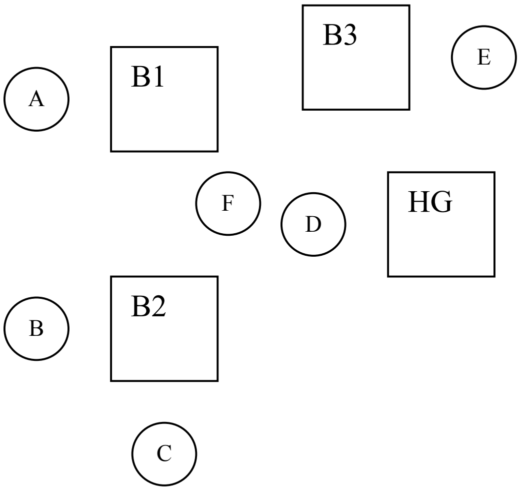


Figure 48. Example Address Forwarding Tables

Where X means the node can be reached through the port that is being analyzed.

Step 1 – Connections between bridges

The bridges advertise their presence to their neighbors. This means that a bridge is able to detect other bridges with shared segment or direct connection. As shown in figure 49, we start building our neighbor list based on the received AFTs. N represents a direct connection or shared segments between bridges. The numbers on the link represent the relevant port numbers.

NEIGHBOR LIST											
		Nodes									
Bridge	Port	HG	B1	B2	B3	A	B	C	D	E	F
B1	1	N		X	N		X	X	X	X	X
	2					X					
B2	1	N	X		X	X			X	X	X
	2						X				
	3							X			
B3	1	N	N	X		X	X	X	X		X
	2									X	
HG	1		N		N	X				X	X
	2			N			X	X			
	3								X		

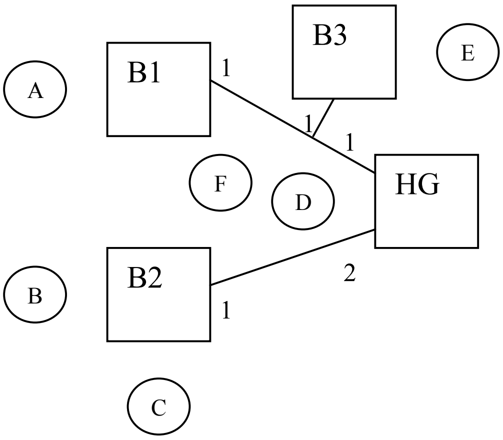


Figure 49. Example Neighbor List – Step 1

Step 2 – Connections between bridges and stations

For every bridge i, each port j that is not connected to another bridge has a group of stations in its forwarding set. These stations are neighbors of bridge i through port j. As shown in figure 50, the neighbor list is updated with the stations that are neighbors to every bridge. N also represents a direct connection or shared segments between stations and bridges.

NEIGHBOR LIST											
		Nodes									
Bridge	Port	HG	B1	B2	B3	A	B	C	D	E	F
B1	1	N		X	N		X	X	X	X	X
	2					N					
B2	1	N	X		X	X			X	X	X
	2						N				
	3							N			
B3	1	N	N	X		X	X	X	X		X
	2									N	
HG	1		N		N	X				X	X
	2			N			X	X			
	3								N		

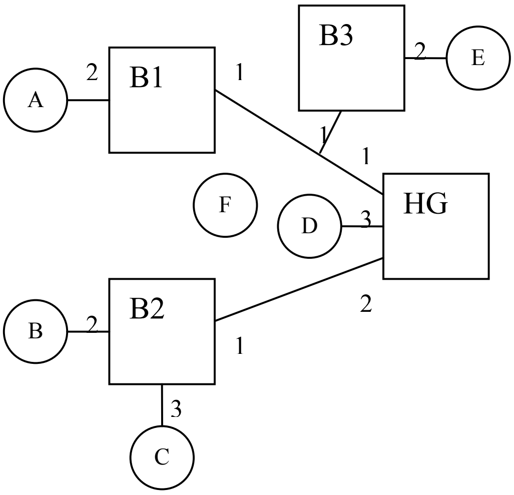


Figure 50. Example Neighbor List - Step 2

Step 3 – Shared segments between bridges and stations

In some scenarios, it is possible to find stations sharing a segment with bridges. In this step we find these stations to include them in the final neighbor list.

We first find the ports from bridges sharing a segment. Then we find the intersection between forwarding sets of these ports. In the example, we have two segments shared by bridges:

- The segment 1 connecting port 1 of B1, port 1 of HG and port 1 of B3
- The segment 2 connecting port 1 of B2 and port 2 of HG.

We find the intersection of forwarding sets for each situation:

- The segment connecting port 1 of B1, port 1 of HG and port 1 of B3

$$F_{B1}^{1} \cap F_{HG}^{1} \cap F_{B3}^{1} = F \qquad (12)$$

where F_{B1}^{1} ={HG,B2,B3,B,C,D,E,F} F_{HG}^{1}={B1,B3,A,E,F} F_{B3}^{1}={HG,B1,B2,A,B,C,D,F}

- The segment connection port 1 of B2 and port 2 of HG.

$$F_{B2}^{1} \cap F_{HG}^{2} = \{HG,B1,B3,A,D,E,F\} \cap \{B2,B,C\} = \Phi \qquad (13)$$

While segment 2 has no stations, in segment 1 we find station F connected to B1, HG and B3. We complete the table with this new information. Any entry in the neighbor list that is not marked as a neighbor is removed. The final neighbor list is shown in figure 51.

NEIGHBOR LIST											
		Nodes									
Bridge	Port	HG	B1	B2	B3	A	B	C	D	E	F
B1	1	N			N						N
	2					N					
B2	1	N									
	2						N				
	3							N			
B3	1	N	N								N
	2									N	
HG	1		N		N						N
	2			N							
	3								N		

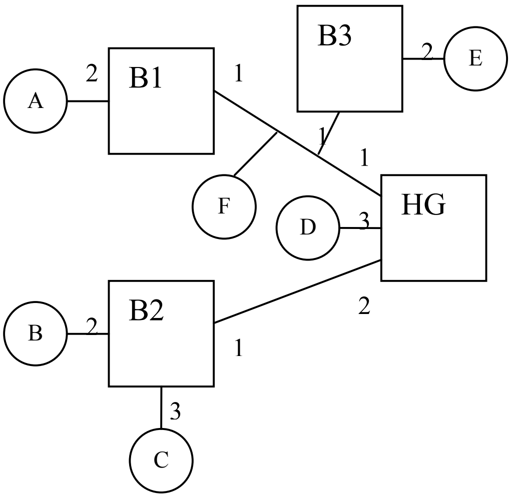


Figure 51. Example Neighbor List – Step 3

6.2.4 End-device information

The architecture shown in figure 46 and the topology discovery process described before give us an idea about the active bridges in the home network and how they are connected. Any device that does not provide an AFT is assumed to be an end-device identified by its MAC address. If further information about end-devices' characteristics is required, new entities must be included in the proposed architecture.

In [26], authors propose an architecture to carry out discovery of end-devices by retrieving information from different protocols and agents such as ARP, DHCP, UPnP and Session Initiation Protocol (SIP). The different devices are classified according to the following definitions:

- Type D which is a device managed by DHCP only.
- Type U which supports UPnP.
- Type CD is a typical device remotely managed by CWMP including a DHCP client stack
- Type CU is a UPnP device that can be remotely managed using CWMP
- Type C is a device remotely managed by an ACS, but without DHCP client stack
- Type S is a Session Initiation Protocol (SIP) device which User Agent (UA) can be uniquely discovered.

The ACS can access end-devices data thanks to a Managed Devices Data Base (MDDB) entity in the TR-069 data model of the CPE which centralizes all relevant information. This architecture can be a good extension of HNTD to include end-device information other than MAC address.

The dominance of Microsoft over the personal computer market has guaranteed the presence of its operating system within most home networks in the world. This means that Microsoft's tools, like LLTD, can be an important source of information about devices in the home network devices. In order to achieve an interaction between the HG and available LLTD responders in the home network, a new entity called Light LLTD Mapper (LLM) is included in the architecture described in [26]. LLM performs QD service to gather information from stations supporting a Microsoft LLTD responder. It sends Discovery frames and receives Hello frames from responders. End-device information is shared by the MMDB entity. Figure 52 shows our final architecture to perform end-device discovery together with topology discovery.

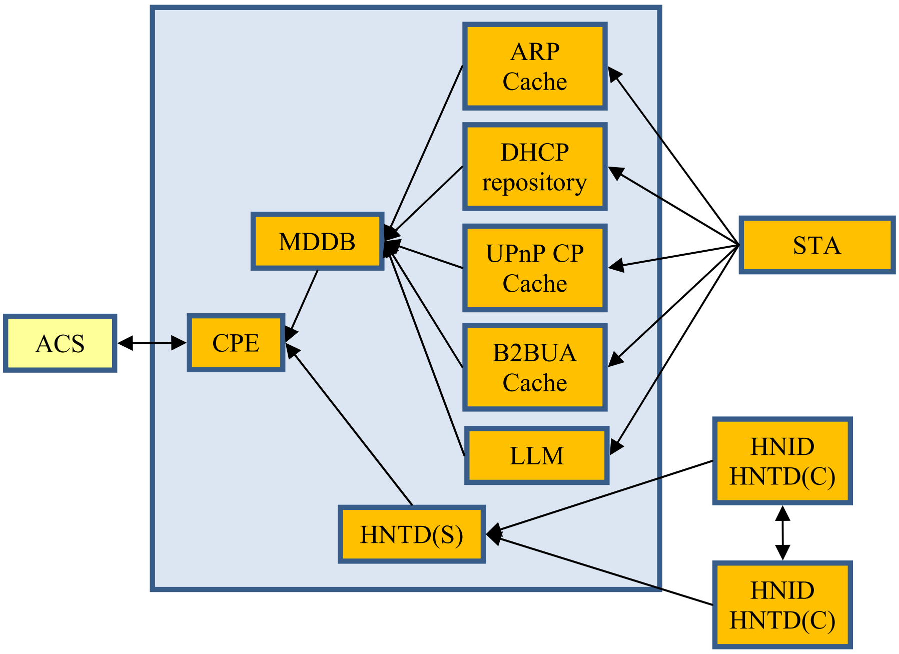


Figure 52. End-device and topology discovery architecture

6.3 Performance Analysis

Based on the HNTD architecture's characteristics, we are able to analyze discovery time, injected traffic and accuracy performance parameters. The evaluation of memory requirement needs the implementation of the proposed architecture, which is future work.

6.3.1 Discovery Time

The proposed HNTD architecture has two types of operations:

- A normal operation (NO), when there are no topology changes and the HNIDs just transmit LLDP_Hello frames.
- An update operation (UO), when there are topology changes and the HNIDs transmit LLDP_Info frames.

During the normal operation phase, data processing is not required and topology information is always available. This makes the discovery time virtually zero.

Conditions during the update operation phase are quite different. Three main steps define the topology discovery process of the HNTD architecture. First, when necessary, AFTs are updated by HNIDs when devices are joining or leaving the network. Second, The

AFTs must be sent to the HG using the frame format described in 6.2.2. Finally, the AFTs' data is processed by the HG using the proposed algorithm in section 6.2.3 to infer the link layer network topology.

Topology information is automatically updated when a change of AFTs is detected. HNIDs will immediately update AFTs when a new device joins the network. The problem lies when a device leaves the network. Update mechanisms define an aging time of AFT's entries that can be set by the end-user. Usually the value of aging time is greater than 2 seconds. As an example, the minimum aging time in CISCO switch SF-300-08 is 10 seconds.

Each HNID sends an LLDP_Info frame that does not exceed 1500 bytes. The time required to transmit this amount of data through any modern home networking technology is negligible and it does not affect the calculation of topology discovery time.

The HNTD algorithm's processing time depends on hardware characteristics of the HG. During the update operation phase, the topology discovery time can thus exceed 2 seconds. More study on this is needed.

6.3.2 Injected Traffic

The size of an LLDP_Hello frame will be the same as a regular LLDP frame. For our analysis we assume that the size of an LLDP_Hello frame is 185 bytes as measured during the experiments. During the normal operation phase, the injected traffic would be the same as LLDP. LLDP_Info frames carry the same information as LLDP_Hello frames plus the AFTs and the list of neighbors included as a sequence of TLVs. For each type of configuration the length of the LLDP_Info frame depends on the number of MAC addresses listed in the AFT. Table 23 gives the LLDP_Info frame's length and the average traffic calculated using equation (6) for every configuration.

During the normal operation (NO) phase, the injected traffic would be the same as for LLDP and less than for LLTD (with the exception of WLAN). Figure 53 shows the results for LLTD, LLDP and the HNTD architecture (for update as well as normal operation). In general, the proposed architecture does not exceed the traffic generated by LLTD and fulfils the service providers' requirement (Req. 3) easily. Especially for configuration Eth, HNTD performs much better than LLTD.

Configuration	Case	HNID info (bytes)	Extra TLV (bytes)	Frame Size (bytes)	Av Traffic (kbit/s)
Config Eth	1 Station	0	0	0	0
	2 Stations	0	0	0	0
	3 Stations	0	0	0	0
Config SW	1 Station	185	44	229	0.0305
	2 Stations	185	66	251	0.0335
	3 Stations	185	88	273	0.0364
Config PLC	1 Station	185	44	229	0.0305
	2 Stations	185	51	236	0.0315
	3 Stations	185	58	243	0.0324
Config WL	1 Station	185	44	229	0.0305
	2 Stations	185	51	236	0.0315
	3 Stations	185	58	243	0.0324

Table 23. LLDP_Info frame's length and corresponding average traffic

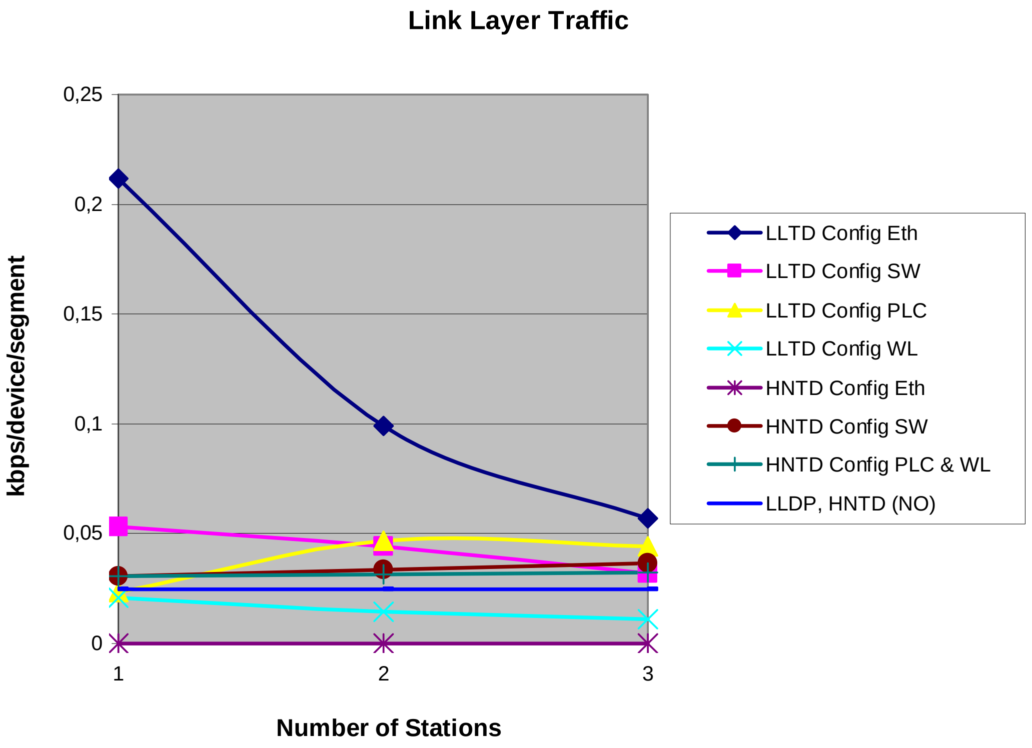


Figure 53. Average Injected Traffic

6.3.3 Accuracy

6.3.3.1 Classification Accuracy

In the case of HNTD, each HNID is capable of advertising its presence and capabilities to the HG. Every MAC address listed in the AFTs that does not belong to a HNID is automatically classified as an end-device. Advertisement mechanism adopted by HNID and AFTs eliminate the possibility of false positive or false negative outcomes. Table 24 summarizes the matching process and the outcomes for HNTD.

Device Type	HNTD matching process				HNTD outcomes			
	HP	SW	AP	STA	HP	SW	AP	STA
HP	P	N	N	N	TP	TN	TN	TN
SW	N	P	N	N	TN	TP	TN	TN
AP	N	N	P	N	TN	TN	TP	TN
STA	N	N	N	P	TN	TN	TN	TP

Table 24. HNTD matching process and outcomes

Based on the information provided by table 24, we calculate the classification accuracy value of HNTD and its coordinates in the ROC graph. We compare these results with LLDP's and LLTD's results (see table 25 and figure 54). HNTD exceeds LLTD and LLDP's accuracy performance and matches the performance of LLDP in the ROC graph.

In order to be correctly classified by LLTD or LLDP, end-devices must support a LLDP or LLTD daemon. HNTD does not have any strict condition over stations and it only needs information from AFTs to find them. In this sense, the classification capability of HNTD is greatly improved compared to other protocols and matches Req. 2.

	LLTD	LLDP	HNTD
Number of TP	3	3	4
Number of TN	20	15	12
Number of FP	1	0	0
Number of FN	4	2	0
Number of P	4	3	4
Number of N	24	17	12
Accuracy	82%	90%	100%

Table 25. Accuracy Values LLTD, LLDP, HNTD

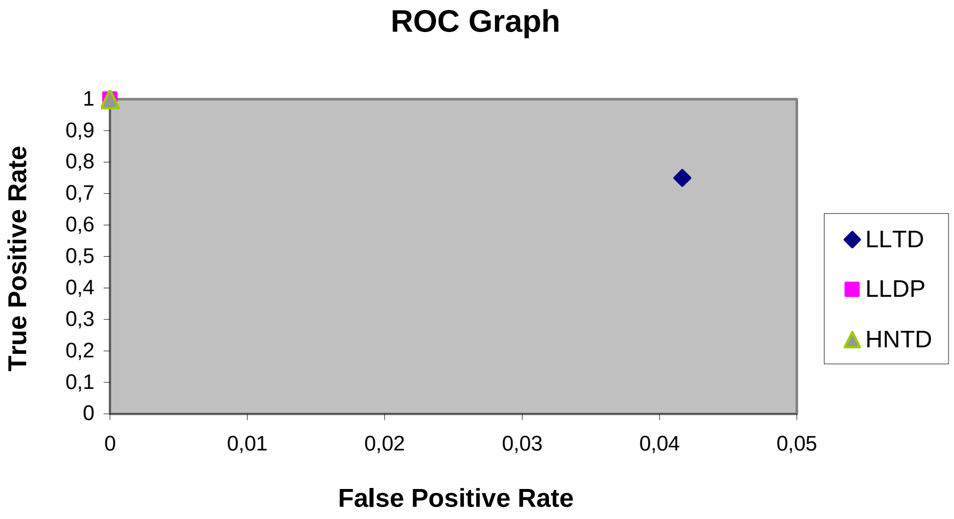


Figure 54. ROC Graph LLTD, LLDP, HNTD

6.3.3.2 Topology Accuracy

We calculate topology accuracy values of HNTD using AFTs of different HNIDs during the experiments. Appendix B includes the matlab code used to simulate the topology discovery algorithm, the AFTs and topology results for all configurations. Figure 55 compares topology accuracy results for all protocols. HNTD reaches 100% accuracy for all configurations.

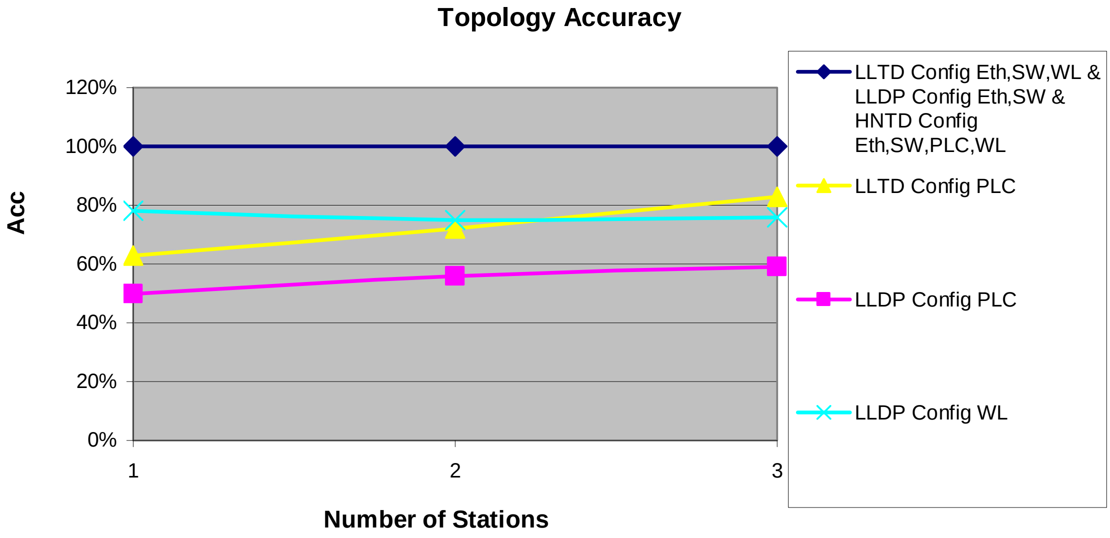


Figure 55. Topology Accuracy HNTD, LLDP, LLTD

6.3.4 Analysis of results

Table 26 shows which requirements are fully fulfilled (+), which requirements are partially fulfilled (0) and which requirements are not fulfilled (–) by LLTD protocol, LLDP and HNTD protocol.

	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9
LLTD	–	0	+	–	–	–	+	0	–
LLDP	–	0	+	+	–	–	–	–	+
HNTD	0	+	+	+	+	+	+	+	+

Table 26. Fulfillment service providers' extended requirements

7

Conclusions and future work

Throughout this master thesis report we explored different aspects of topology discovery within home networks, we analyzed the performance of LLDP and LLTD, and we proposed the HNTD topology discovery architecture for home networks. This chapter presents the main conclusions and recommendations for future work.

7.1 Conclusions of the thesis

In the past, topology discovery mechanisms have been studied and developed for medium-sized and large-sized Ethernet networks, typically using information available from address forwarding tables and the spanning tree protocol. Little attention was paid to topology discovery tools designed for home networks containing multiple home networking technologies. Microsoft addressed this issue and introduced LLTD to the market as part of the Windows operating system. In the meantime, IEEE standardized a topology discovery protocol known as LLDP which was specifically designed for local area networks. Due to the potential of LLTD and LLDP, service providers have been demanding further analysis of these protocols and comparison of their operation within a home network. In order to carry out this study, we defined four performance indicators: discovery time, average injected traffic rate, accuracy and memory requirement. The accuracy of the topology discovery protocols is analyzed from the classification point of view (which is the ability to identify the device type correctly) and from the graph identification point of view (which is the ability to identify links between the different networked devices correctly). For the accuracy analysis, we used graph theory and receiver-operating-characteristics techniques. As far as we know, this is the first attempt to establish a comparison framework to evaluate the operation and performance of topology discovery protocols for heterogeneous consumer networks. The assessment of LLTD and LLDP, based on the performance indicators we designed, drew the following conclusion regarding their operation:

- The advertisement mechanism employed by LLDP provides a more accurate classification result compared to test mechanism employed by LLTD.
- LLTD does not perform well when the network under test contains collision domains that link more than two bridges or HNIDs.
- LLDP relies on the presence of manageable HNIDs which are not common within home networks. Experiments have shown that this characteristic makes LLDP less suitable for use by service providers compared to LLTD, in terms of topology discovery time and memory requirement.
- Although LLTD performs slightly better than LLDP, mainly in terms of discovery time and accuracy, none of the protocols fulfills the service providers' requirements satisfactorily.
- Robustness was not one of the service providers' requirements, but we tested the protocols on this anyway, by flooding the home network and monitoring the performance indicators. Above 60% cross traffic, LLDP performance significantly better than LLTD.

Based on our experimental results and further analysis, we proposed a new architecture that combines different elements of the studied topology discovery techniques and largely fulfils the requirements for use by service providers. We dubbed the resulting protocol Home Network Topology Discovery (HNTD).

7.2 Future work recommendations

- The HNTD protocol must be implemented and tested under real conditions. The development of a daemon can be done based on the description given in this master thesis. Although not mentioned explicitly in the text, we designed the architecture of HNTD such that it can be easily constructed from the current open source LLDP implementations. That would then also allow further evaluation of the memory requirements of the HNTD architecture. The HNTD daemons' size and the memory that needs to be allocated in HG to store topology information can then be assessed.
- A mechanism to meet the topology discovery time requirement when a device is removed from the home network must be invented and included in the HNTD protocol.
- Solutions to include non-IP network segments such as Zigbee into the topology discovery process must be designed and evaluated. The gateway linking the LAN and WPAN could transmit the list of non-IP devices according to a frame format to be specified.

- At the moment, the first draft of HTIP's technical specifications only describes the mechanisms used to exchange information between networked devices. The algorithm to process the data and infer network topology is not available yet. When completed, HTIP must be implemented and tested following the comparison framework presented in this document.
- This thesis addresses the topology discovery problem which is one of components of HGI's recommended diagnostic tool set. Although topology information is important to understand the characteristics of a network, it does not provide by itself solutions to solve all service providers' problems. Methods or systems must be developed to intelligently exploit this information according to the service providers' needs. An expert system could be designed to identify topologies that could affect QoS. Also, a logging system could store topology information periodically in order to evaluate the changes of a home network's topology over time.
- The success of HNTD is depending on its adoption in HNIDs. To accelerate the adoption, HNTD must be standardized and the software and source code must be made freely available under an attractive licensing scheme. Future protocol architectures may be developed to lessen the requirement of HNID support.
- Although the set of configuration we used for assessing the protocols' performance is based on realistic home network topologies and is fairly complete, it should be analyzed how much the result differ for less common configurations.

Bibliography

[1] M. Castrucci et al., *Deliverable 6.1 Omega Architecture Model*, Seventh Framework Programme, December 2008.

[2] F.T.H. den Hartog, F. Fang, *QoS and unicast/muticast requirements on Home Network Infrastructure Devices*, HGI Draft Document HGI01628R01, February 2011.

[3] *IEEE 802.1D: Media Access Control (MAC) Bridges*, June 2004.

[4] *IEEE 802.3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, December 2008.

[5] *HomePlug AV White Paper*, HomePlug Power Line Alliance, 2005.

[6] *IEEE std 1901: IEEE Standard for Broadband over Power Line Networks – Medium Access Control and Physical Layer Specifications*, December 2010.

[7] *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 2007.

[8] B. Bissell et al., *Use cases and business requirements on diagnostics in the home network*, HGI Draft Document HGI01150, 2008.

[9] B. Lowekamp et al., *Topology Discovery for Large Ethernet Networks*, SIGCOMM'01, August 2001.

[10] T.Q. Yan, X. Lu, *Physical topology discovery in networks with redundant links*, International Conference on Apperceiving Computing and Intelligence Analysis, 2008.

[11] U. Uzair et al., *An efficient Algorithm for Ethernet Topology Discovery in Large Multi-subnet Networks*, International Conference on System of Systems Engineering, 2007.

[12] H. Peng et al., *Physical Topology Discovery Based on Spanning Tree Protocol*, International Conference on Computer Application and System Modeling, 2010.

[13] *IEEE 802.1AB: Station and Media Access Control Connectivity Discovery*, September 2009.

[14] *Link Layer Topology Discovery (LLTD) Protocol Specification*, Microsoft, August 2010.

[15] *LLTD: Link Layer Topology Discovery Protocol*, Microsoft, September 2006

[16] R. Black, A. Donnelly, C. Fournet, *Ethernet Topology Discovery without Network Assistance*, 12th IEEE International Conference on Network Protocols, 2004.

[17] *JJ-300.00: Home-network Topology Identifying Protocol (HTIP)*, Telecommunication Technology Committee Standard, August 2010.

[18] http://www.iana.org/assignments/ianaiftype-mib, May 2011.

[19] *Comparison of LLDP daemons*, http://www.kempgen.net/voip/lldp-agents.html, May 2011.

[20] haneWIN LLDP agent, http://www.hanewin.net/lldp-e.htm, May 2011

[21] *IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, September 2006.

[22] R.C. Wang, R.S. Chang, H.C. Chao, *Internetworking Between Zigbee/802.15.4 and IPv6/802.3*, IPv6'07, August 2007.

[23] A. Botta, A. Dainotti, A.Pescape, *Multiprotocol and multiplatform traffic generation and measurement*, INFOCOMM 2007 DEMO Session, May 2007.

[24] T. Fawcett, *ROC Graphs: Notes and Practical Considerations for Researchers*, HP Laboratories, March 2004.

[25] A. Barthel, *Analysis, Implementation and Enhancement of Vendor dependent and independent Layer 2 Network Topology Discovery*, Master Thesis, Chemnitz University of Technology, April 2005. Improve NMS

[26] FTH. den Hartog et al., *Remote discovery and management of end-user devices in heterogeneous private networks*, CCNC, 2009.

[27] E. Diaz Castellanos, *Characterizing Dutch Home Network Dynamics*, TNO Internship Report, September 2010.

[28] FTH. den Hartog et al, *Convergence of Residential Gateway Technology*, IEEE Communications Magazine, May 2004.

[29] D. Thorne and H.W. Bitzer, *HGI DRAFT 0.4 – Home Gateway and Home Network Diagnostics Requirements*, December 2010.

[30] P. Van Mieghem, *Lecture Notes Complex Networks*, TU Delft, 2011

[31] *Recommendation ITU-T G.9960: Unified high-speed wire-line based home networking transceivers – System architecture and physical layer specification*, June 2010.

[32] *Recommendation ITU-T G.9961: Data link layer (DLL) for unified high-speed wire-line based home networking transceivers*, June 2010.

[33] http://www.netdisco.org/, May 2011.

[34] http://www.solarwinds.com/products/toolsets/engineer.aspx, July 2011.

[35] E. Diaz Castellanos, FTH. den Hartog, *Performance comparison of LLTD and IEEE 802.1AB*, HGI Document HGI01678R01, February 2011.

Appendix A

A.1 Adjacency matrices basic configurations

Configuration Eth

	RG	S1
RG	0	1
S1	1	0

	RG	S1	S2
RG	0	1	1
S1	1	0	0
S2	1	0	0

	RG	S1	S2	S3
RG	0	1	1	1
S1	1	0	0	0
S2	1	0	0	0
S3	1	0	0	0

Configuration SW

	RG	SW	S1
RG	0	1	0
SW	1	0	1
S1	0	1	0

	RG	SW	S1	S2
RG	0	1	0	0
SW	1	0	1	1
S1	0	1	0	0
S2	0	1	0	0

	RG	SW	S1	S2	S3
RG	0	1	0	0	0
SW	1	0	1	1	1
S1	0	1	0	0	0
S2	0	1	0	0	0
S3	0	1	0	0	0

Configuration WL

	RG	AP	S1
RG	0	1	0
AP	1	0	1
S1	0	1	0

	RG	AP	S1	S2
RG	0	1	0	0
AP	1	0	1	1
S1	0	1	0	0
S2	0	1	0	0

	RG	AP	S1	S2	S3
RG	0	1	0	0	0
AP	1	0	1	1	1
S1	0	1	0	0	0
S2	0	1	0	0	0
S3	0	1	0	0	0

Configuration PLC

	RG	HP4	HP1	S1
RG	0	1	0	0
HP4	1	0	1	0
HP1	0	1	0	1
S1	0	0	1	0

	RG	HP4	HP1	HP2	S1	S2
RG	0	1	0	0	0	0
HP4	1	0	1	1	0	0
HP1	0	1	0	1	1	0
HP2	0	1	1	0	0	1
S1	0	0	1	0	0	0
S2	0	0	0	1	0	0

	RG	HP4	HP1	HP2	HP3	S1	S2	S3
RG	0	1	0	0	0	0	0	0
HP4	1	0	1	1	1	0	0	0
HP1	0	1	0	1	1	1	0	0
HP2	0	1	1	0	1	0	1	0
HP3	0	1	1	1	0	0	0	1
S1	0	0	1	0	0	0	0	0
S2	0	0	0	1	0	0	0	0
S3	0	0	0	0	1	0	0	0

A.2 LLTD Topology Maps

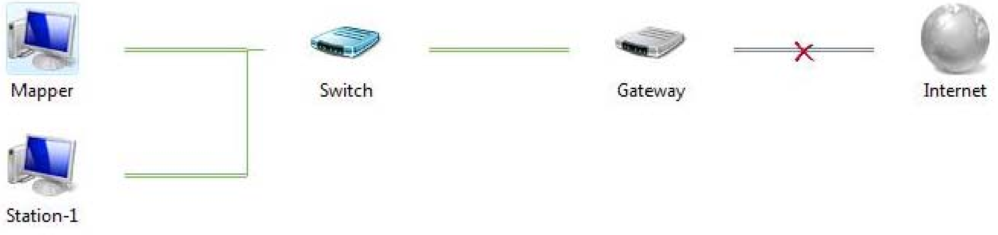


Figure 56. LLTD map, Config Eth 1 – One Station

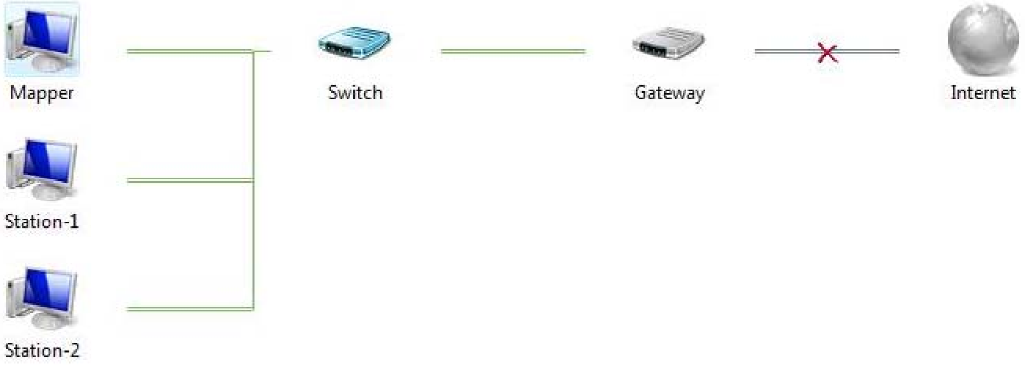


Figure 57. LLTD map, Config Eth – Two Stations

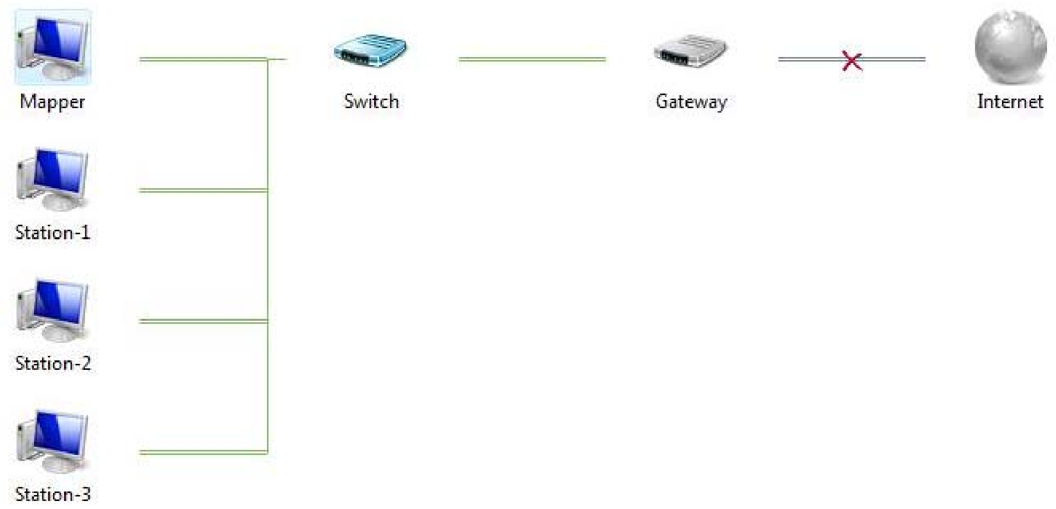


Figure 58. LLTD map, Config Eth – Three Stations

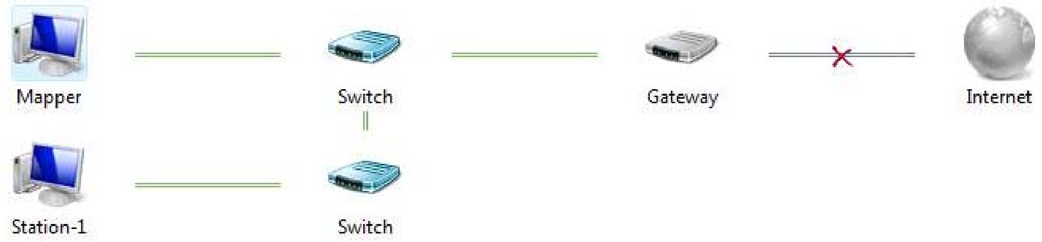


Figure 59. LLTD map, Config SW – One Station

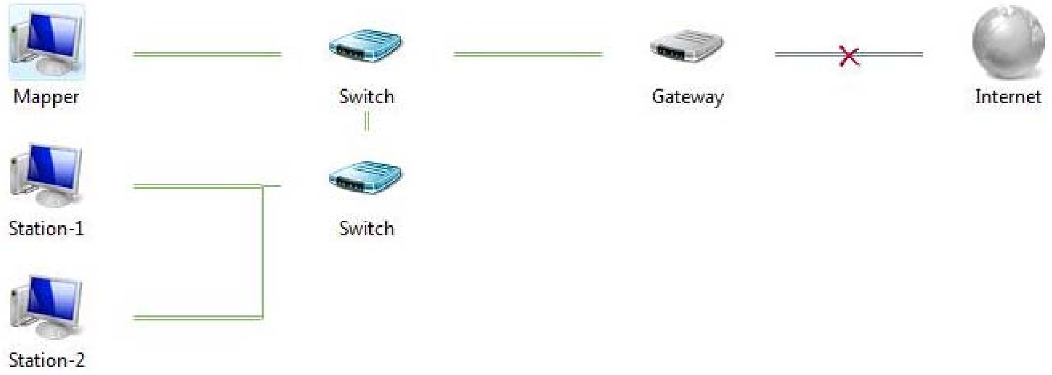


Figure 60. LLTD map, Config SW – Two Stations

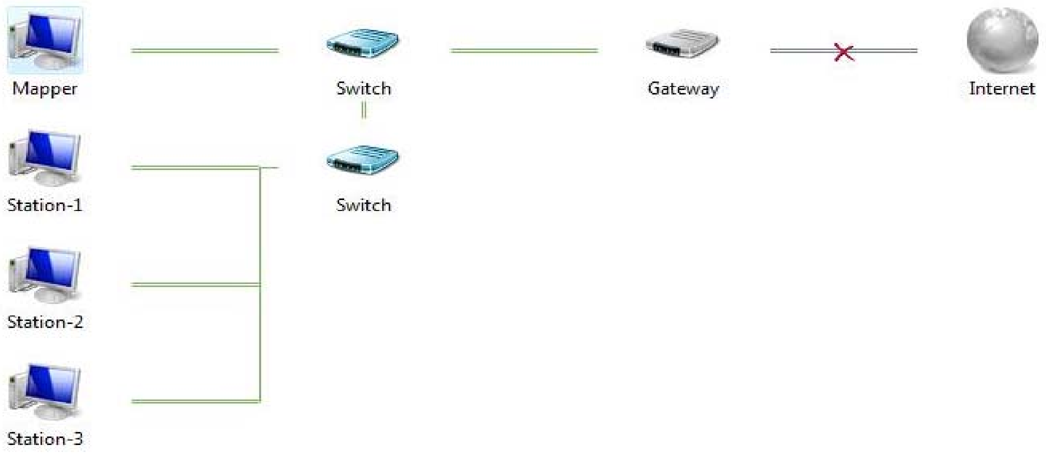


Figure 61. LLTD map, Config SW – Three Stations

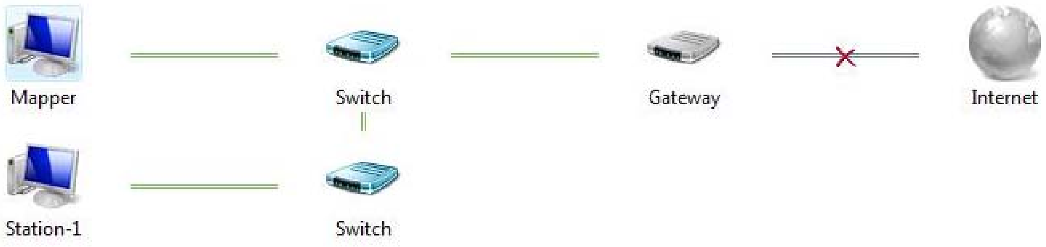


Figure 62. LLTD map, Config PLC – One Station

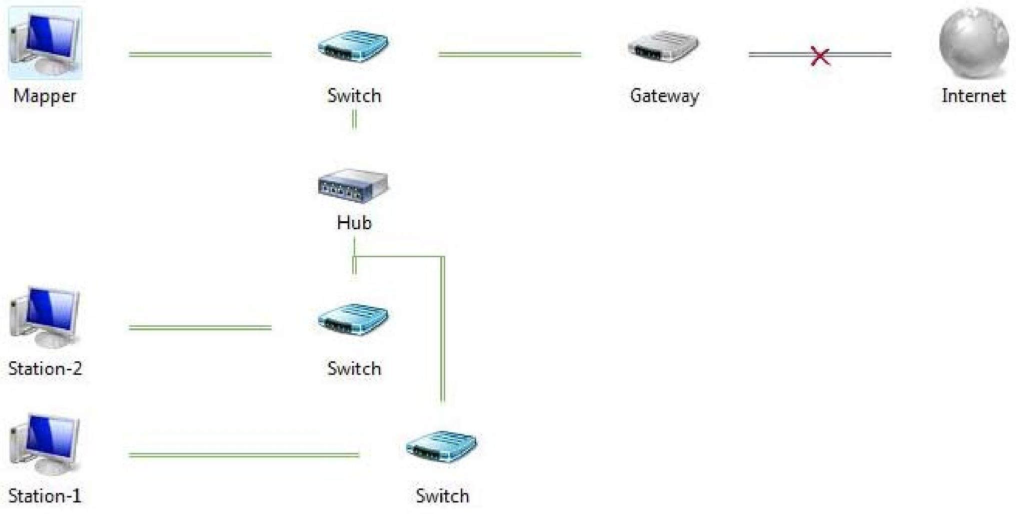


Figure 63. LLTD map, Config PLC – Two Stations

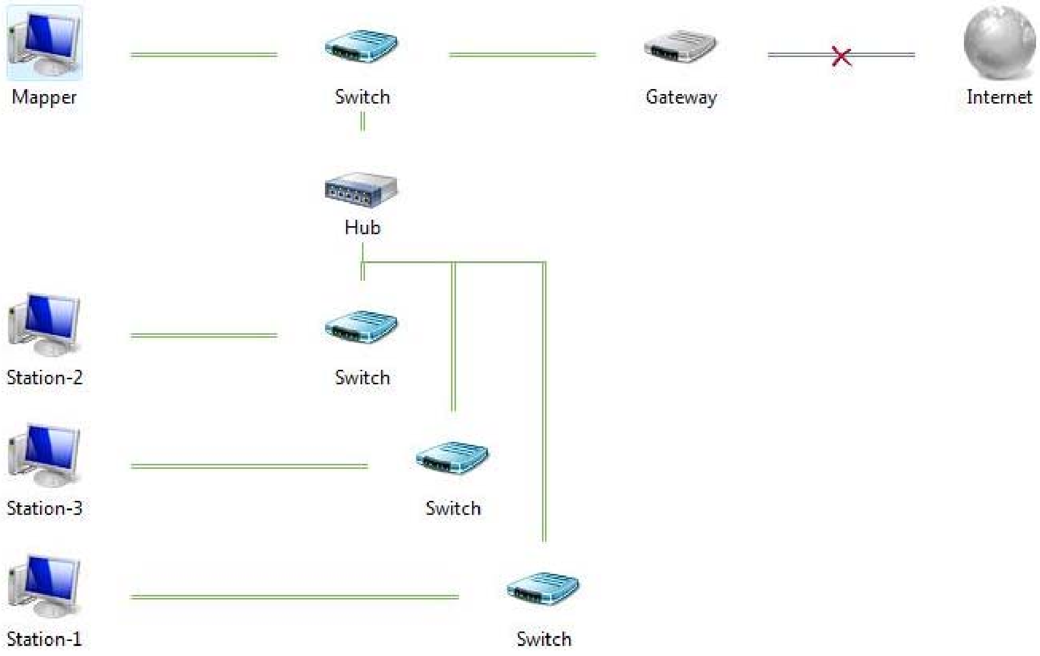


Figure 64. LLTD map, Config PLC – Three Stations

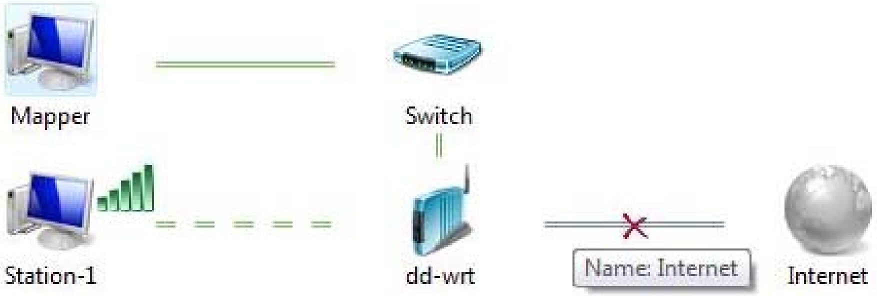


Figure 65. LLTD map, Config WL – One Station

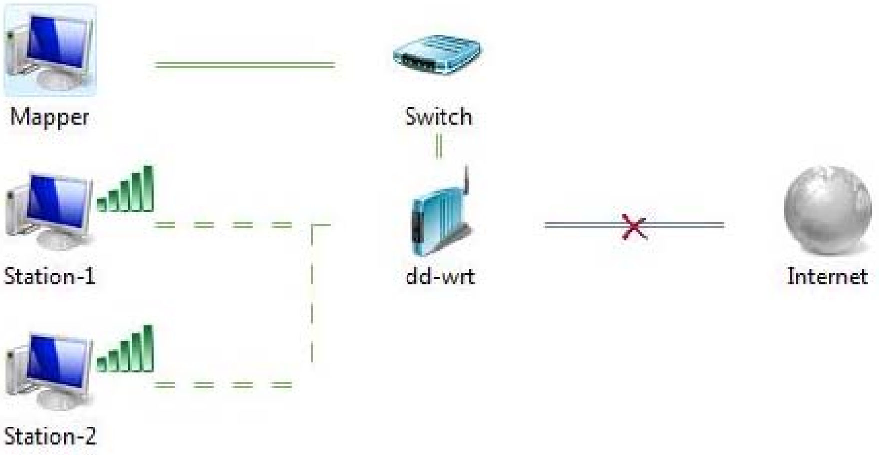


Figure 66. LLTD map, Config WL – Two Stations

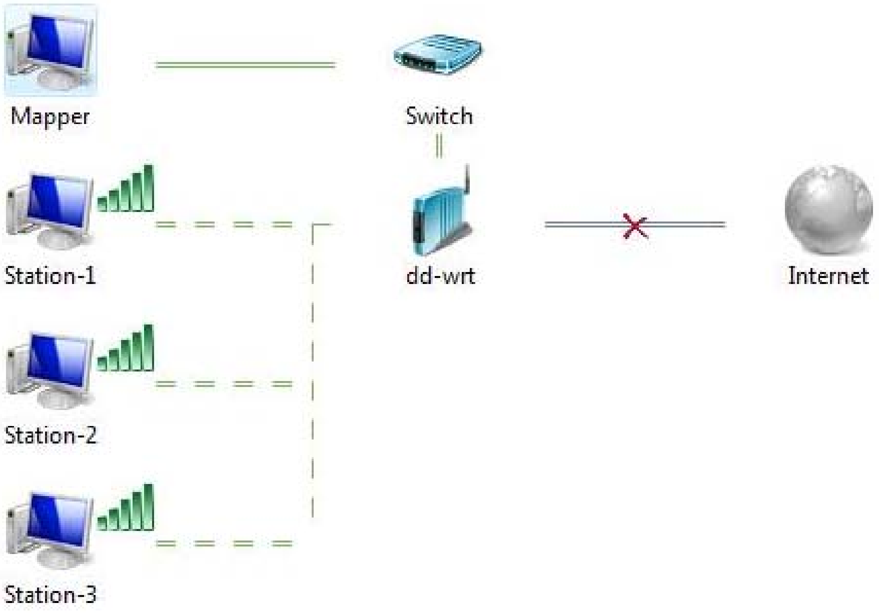


Figure 67. LLTD map, Config WL – Three Stations

A.3 Adjacency matrices from LLTD

Configuration Eth

	RG	S1
RG	0	1
S1	1	0

	RG	S1	S2
RG	0	1	1
S1	1	0	0
S2	1	0	0

	RG	S1	S2	S3
RG	0	1	1	1
S1	1	0	0	0
S2	1	0	0	0
S3	1	0	0	0

Configuration SW

	RG	SW	S1
RG	0	1	0
SW	1	0	1
S1	0	1	0

	RG	SW	S1	S2
RG	0	1	0	0
SW	1	0	1	1
S1	0	1	0	0
S2	0	1	0	0

	RG	SW	S1	S2	S3
RG	0	1	0	0	0
SW	1	0	1	1	1
S1	0	1	0	0	0
S2	0	1	0	0	0
S3	0	1	0	0	0

Configuration WL

	RG	AP	S1
RG	0	1	0
AP	1	0	1
S1	0	1	0

	RG	AP	S1	S2
RG	0	1	0	0
AP	1	0	1	1
S1	0	1	0	0
S2	0	1	0	0

	RG	AP	S1	S2	S3
RG	0	1	0	0	0
AP	1	0	1	1	1
S1	0	1	0	0	0
S2	0	1	0	0	0
S3	0	1	0	0	0

Configuration PLC

	RG	HP4	HP1	S1
RG	0	0	1	0
HP4	0	0	0	0
HP1	1	0	0	1
S1	0	0	1	0

	RG	HP4	HP1	HP2	S1	S2
RG	0	0	1	1	0	0
HP4	0	0	0	0	0	0
HP1	1	0	0	1	1	0
HP2	1	0	1	0	0	1
S1	0	0	1	0	0	0
S2	0	0	0	1	0	0

	RG	HP4	HP1	HP2	HP3	S1	S2	S3
RG	0	0	1	1	1	0	0	0
HP4	0	0	0	0	0	0	0	0
HP1	0	0	0	1	1	1	0	0
HP2	0	0	1	0	1	0	1	0
HP3	0	0	1	1	0	0	0	1
S1	0	0	1	0	0	0	0	0
S2	0	0	0	1	0	0	0	0
S3	0	0	0	0	1	0	0	0

A.3 LLDP Topology Maps

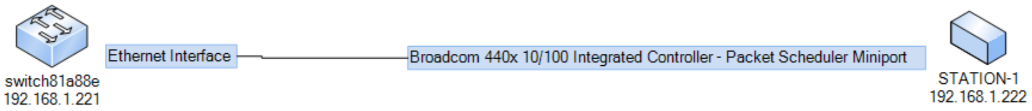

switch81a88e
192.168.1.221
Ethernet Interface
Broadcom 440x 10/100 Integrated Controller - Packet Scheduler Miniport
STATION-1
192.168.1.222

Figure 68. LLDP map, Config Eth, PLC – One Station

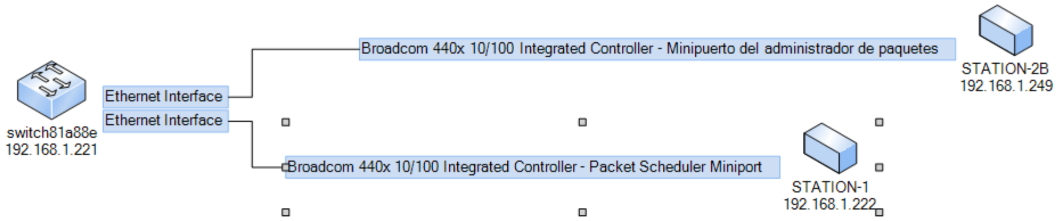

switch81a88e
192.168.1.221
Ethernet Interface
Ethernet Interface
Broadcom 440x 10/100 Integrated Controller - Minipuerto del administrador de paquetes
STATION-2B
192.168.1.249
Broadcom 440x 10/100 Integrated Controller - Packet Scheduler Miniport
STATION-1
192.168.1.222

Figure 69. LLDP map, Config Eth, PLC – Two Stations

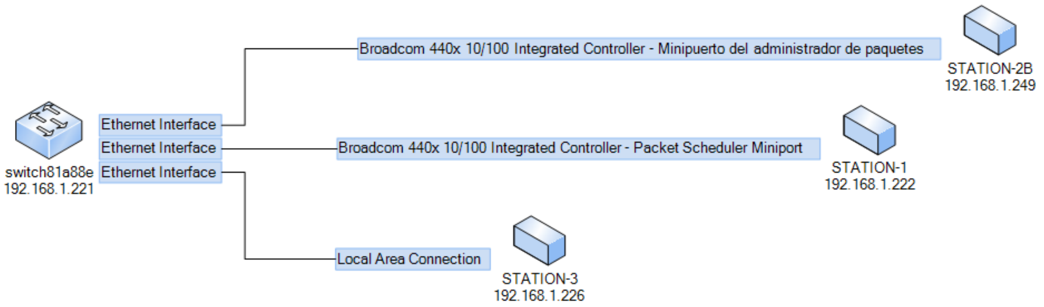


Figure 70. LLDP map, Configurations Eth, PLC – Three Stations

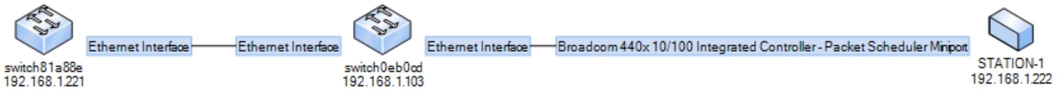


Figure 71. LLDP map, Config SW – One Station

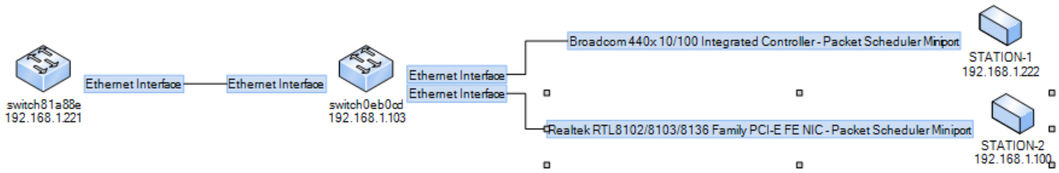


Figure 72. LLDP map, Config SW – Two Stations

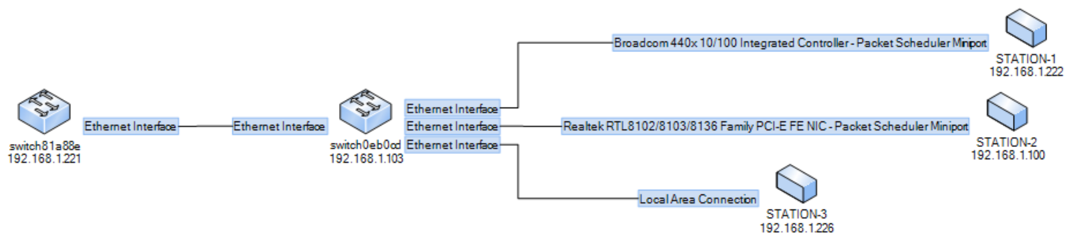


Figure 73. LLDP map, Config SW – Three Stations

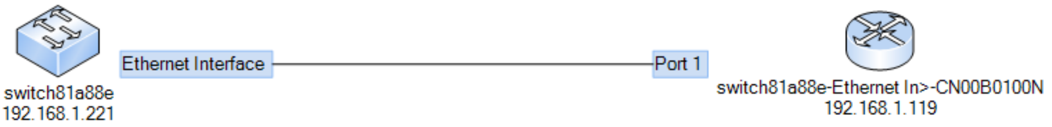


Figure 74. LLDP map, Configu WL – One, Two and Three Stations

A.4 Adjacency matrices from LLDP

Configuration Eth

	HG	S1
HG	0	1
S1	1	0

	HG	S1	S2
HG	0	1	1
S1	1	0	0
S2	1	0	0

	HG	S1	S2	S3
HG	0	1	1	1
S1	1	0	0	0
S2	1	0	0	0
S3	1	0	0	0

Configuration SW

	HG	SW	S1
HG	0	1	0
SW	1	0	1
S1	0	1	0

	HG	SW	S1	S2
HG	0	1	0	0
SW	1	0	1	1
S1	0	1	0	0
S2	0	1	0	0

	HG	SW	S1	S2	S3
HG	0	1	0	0	0
SW	1	0	1	1	1
S1	0	1	0	0	0
S2	0	1	0	0	0
S3	0	1	0	0	0

Configuration WL

	HG	AP	S1
HG	0	1	0
AP	1	0	0
S1	0	0	0

	HG	AP	S1	S2
HG	0	1	0	0
AP	1	0	0	0
S1	0	0	0	0
S2	0	0	0	0

	HG	AP	S1	S2	S3
HG	0	1	0	0	0
AP	1	0	0	0	0
S1	0	0	0	0	0
S2	0	0	0	0	0
S3	0	0	0	0	0

Configuration PLC

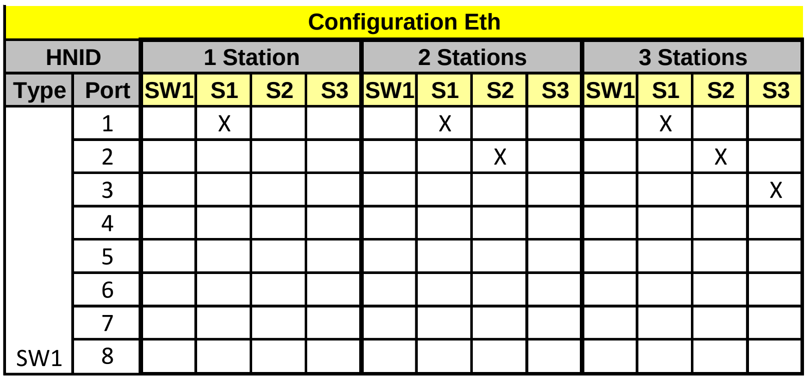
	HG	HP4	HP1	S1
HG	0	0	0	1
HP4	0	0	0	0
HP1	0	0	0	0
S1	1	0	0	0

	HG	HP4	HP1	HP2	S1	S2
HG	0	0	0	0	1	1
HP4	0	0	0	0	0	0
HP1	0	0	0	0	0	0
HP2	0	0	0	0	0	0
S1	1	0	0	0	0	0
S2	1	0	0	0	0	0

	HG	HP4	HP1	HP2	HP3	S1	S2	S3
HG	0	0	0	0	0	1	1	1
HP4	0	0	0	0	0	0	0	0
HP1	0	0	0	0	0	0	0	0
HP2	0	0	0	0	0	0	0	0
HP3	0	0	0	0	0	0	0	0
S1	1	0	0	0	0	0	0	0
S2	1	0	0	0	0	0	0	0
S3	1	0	0	0	0	0	0	0

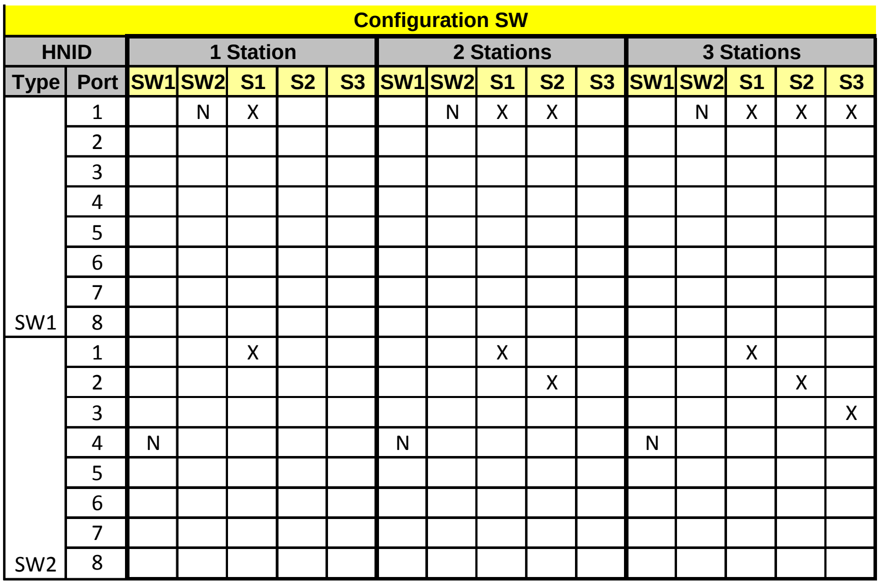
Appendix B

B.1 HNIDs’ AFTs for basic configurations



Configuration Eth													
HNID		1 Station				2 Stations				3 Stations			
Type	Port	SW1	S1	S2	S3	SW1	S1	S2	S3	SW1	S1	S2	S3
SW1	1		X				X				X		
	2							X				X	
	3												X
	4												
	5												
	6												
	7												
	8												

Figure 75. HNIDs’ AFTs Config Eth



Configuration SW																	
HNID		1 Station					2 Stations					3 Stations					
Type	Port	SW1	SW2	S1	S2	S3	SW1	SW2	S1	S2	S3	SW1	SW2	S1	S2	S3
SW1	1		N	X				N	X	X			N	X	X	X
	2															
	3															
	4															
	5															
	6															
	7															
	8															
SW2	1			X					X					X		
	2									X					X	
	3															X
	4	N					N					N				
	5															
	6															
	7															
	8															

Figure 76. HNIDs’ AFTs Config SW

Configuration PLC																									
HNID		1 Station								2 Stations								3 Stations							
Type	Port	SW1	HP1	HP2	HP3	HP4	S1	S2	S3	SW1	HP1	HP2	HP3	HP4	S1	S2	S3	SW1	HP1	HP2	HP3	HP4	S1	S2	S3
SW1	1					N	X							N	X	X						N	X	X	X
	2																								
	3																								
	4																								
	5																								
	6																								
	7																								
	8																								
HP1	1						X								X								X		
	2					N						N		N						N	N	N			
HP2	1															X								X	
	2										N			N					N		N	N			
HP3	1																								X
	2																		N	N		N			
HP4	1	N								N								N							
	2		N								N	N							N	N	N				

Figure 77. HNIDs' AFTs Config PLC

Configuration WL																
		1 Station					2 Stations					3 Stations				
HNID	Port	SW1	AP	S1	S2	S3	SW1	AP	S1	S2	S3	SW1	AP	S1	S2	S3
SW1	1		N	X				N	X	X			N	X	X	X
	2															
	3															
	4															
	5															
	6															
	7															
	8															
AP	1	N					N					N				
	2			X					X	X				X	X	X

Figure 78. HNIDs' AFTs Config WL

B.2 Matlab Code for Topology Discovery Algorithm

```
% The first part of the code provides AFTs with neighbor list to the
topology discovery algorithm

%Forwarding Set (FS) is represented as a cell array. Each element of
the
%cell array is the address of a network device preceded by two bytes
indicating if it is a neighbor or not.

%For every bridge, FS from all ports are concatanated to form one
AFT that
%is going to be send to the HG

%AFT from all bridges are included in one matrix (AFTs)
```

```
NetworkDevices = [{'SW1'} {'SW2'} {'0AP'} {'HP4'} {'HP1'} {'HP2'}
{'HP3'} {'0S1'} {'0S2'} {'0S3'}];

AFTsMatrix = zeros(10,10);
AdjacencyMatrix = zeros(10,10);
Configuration = 'ConfigWL3STA';

switch Configuration
   case 'ConfigEth1STA'
       %Config Eth - Case 1 Station
       %AFT SW1
       FSSW1P1 = [{'000S1'}]
       PortsSW1 =[{FSSW1P1}]
       AFTSW1 = [{'SW1'} {PortsSW1}]
       AFTs = AFTSW1;

   case 'ConfigEth2STA'
       %Config Eth - Case 2 Stations
       %AFT SW1
       FSSW1P1 = [{'000S1'}]
       FSSW1P2 = [{'000S2'}]
       PortsSW1 =[{FSSW1P1} {FSSW1P2}]
       AFTSW1 = [{'SW1'} {PortsSW1}]
       AFTs = AFTSW1;

   case 'ConfigEth3STA'
       %Config Eth - Case 3 Stations
       %AFT SW1
       FSSW1P1 = [{'000S1'}]
       FSSW1P2 = [{'000S2'}]
       FSSW1P3 = [{'000S3'}]
       PortsSW1 =[{FSSW1P1} {FSSW1P2} {FSSW1P3}]
       AFTSW1 = [{'SW1'} {PortsSW1}]
       AFTs = AFTSW1;

   case 'ConfigSW1STA'
       %Config SW - Case 1 Station
       %AFT SW1
       FSSW1P1 = [{'FFSW2'} {'000S1'}];
       PortsSW1 =[{FSSW1P1}];
       AFTSW1 = [{'SW1'} {PortsSW1}];
       %AFT SW2
       FSSW2P1 = [{'000S1'}];
       FSSW2P4 = [{'FFSW1'}];
       PortsSW2 =[{FSSW2P1} {FSSW2P4}];
       AFTSW2 = [{'SW2'} {PortsSW2}];
       AFTs = [AFTSW1 ; AFTSW2]
   case 'ConfigSW2STA'
       %Config SW - Case 2 Stations
       %AFT SW1
       FSSW1P1 = [{'FFSW2'} {'000S1'} {'000S2'}];
       PortsSW1 =[{FSSW1P1}];
       AFTSW1 = [{'SW1'} {PortsSW1}];
       %AFT SW2
       FSSW2P1 = [{'000S1'}];
```

```
        FSSW2P2 = [{'000S2'}];
        FSSW2P4 = [{'FFSW1'}];
        PortsSW2 =[{FSSW2P1} {FSSW2P2} {FSSW2P4}];
        AFTSW2 = [{'SW2'} {PortsSW2}];
        AFTs = [AFTSW1 ; AFTSW2]

    case 'ConfigSW3STA'
        %Config SW - Case 3 Stations
        %AFT SW1
        FSSW1P1 = [{'FFSW2'} {'000S1'} {'000S2'} {'000S3'}];
        PortsSW1 =[{FSSW1P1}];
        AFTSW1 = [{'SW1'} {PortsSW1}];
        %AFT SW2
        FSSW2P1 = [{'000S1'}];
        FSSW2P2 = [{'000S2'}];
        FSSW2P3 = [{'000S3'}];
        FSSW2P4 = [{'FFSW1'}];
        PortsSW2 =[{FSSW2P1} {FSSW2P2} {FSSW2P3} {FSSW2P4}];
        AFTSW2 = [{'SW2'} {PortsSW2}];
        AFTs = [AFTSW1 ; AFTSW2]

    case 'ConfigPLC1STA'
        %Config PLC - Case 1 Station
        %AFT SW1
        FSSW1P1 = [{'FFHP4'} {'000S1'}];
        PortsSW1 =[{FSSW1P1}];
        AFTSW1 = [{'SW1'} {PortsSW1}];
        %AFT HP1
        FSHP1P1 = [{'000S1'}];
        FSHP1P2 = [{'FFHP4'}];
        PortsHP1 =[{FSHP1P1} {FSHP1P2}];
        AFTHP1 = [{'HP1'} {PortsHP1}];

        %AFT HP4
        FSHP4P1 = [{'FFSW1'}];
        FSHP4P2 = [{'FFHP1'}];
        PortsHP4 =[{FSHP4P1} {FSHP4P2}];
        AFTHP4 = [{'HP4'} {PortsHP4}];
        AFTs = [AFTSW1 ; AFTHP1 ; AFTHP4];

    case 'ConfigPLC2STA'
        %Config PLC - Case 2 Stations
        %AFT SW1
        FSSW1P1 = [{'FFHP4'} {'000S1'} {'000S2'}];
        PortsSW1 =[{FSSW1P1}];
        AFTSW1 = [{'SW1'} {PortsSW1}];
        %AFT HP1
        FSHP1P1 = [{'000S1'}];
        FSHP1P2 = [{'FFHP2'} {'FFHP4'}];
        PortsHP1 =[{FSHP1P1} {FSHP1P2}];
        AFTHP1 = [{'HP1'} {PortsHP1}];
        %AFT HP2
        FSHP2P1 = [{'000S2'}];
        FSHP2P2 = [{'FFHP1'} {'FFHP4'}];
        PortsHP2 =[{FSHP2P1} {FSHP2P2}];
        AFTHP2 = [{'HP2'} {PortsHP2}];
        %AFT HP4
        FSHP4P1 = [{'FFSW1'}];
        FSHP4P2 = [{'FFHP1'} {'FFHP2'}];
        PortsHP4 =[{FSHP4P1} {FSHP4P2}];
        AFTHP4 = [{'HP4'} {PortsHP4}];
        AFTs = [AFTSW1 ; AFTHP1 ; AFTHP2 ; AFTHP4];

    case 'ConfigPLC3STA'
        %Config PLC - Case 3 Stations
        %AFT SW1
        FSSW1P1 = [{'FFHP4'} {'000S1'} {'000S2'} {'000S3'}];
        PortsSW1 =[{FSSW1P1}];
        AFTSW1 = [{'SW1'} {PortsSW1}];
        %AFT HP1
        FSHP1P1 = [{'000S1'}];
        FSHP1P2 = [{'FFHP2'} {'FFHP3'} {'FFHP4'}];
        PortsHP1 =[{FSHP1P1} {FSHP1P2}];
        AFTHP1 = [{'HP1'} {PortsHP1}];
        %AFT HP2
        FSHP2P1 = [{'000S2'}];
        FSHP2P2 = [{'FFHP1'} {'FFHP3'} {'FFHP4'}];
        PortsHP2 =[{FSHP2P1} {FSHP2P2}];
        AFTHP2 = [{'HP2'} {PortsHP2}];
        %AFT HP3
        FSHP3P1 = [{'000S3'}];
        FSHP3P2 = [{'FFHP1'} {'FFHP2'} {'FFHP4'}];
        PortsHP3 =[{FSHP3P1} {FSHP3P2}];
        AFTHP3 = [{'HP3'} {PortsHP3}];
        %AFT HP4
        FSHP4P1 = [{'FFSW1'}];
        FSHP4P2 = [{'FFHP1'} {'FFHP2'} {'FFHP3'}];
        PortsHP4 =[{FSHP4P1} {FSHP4P2}];
        AFTHP4 = [{'HP4'} {PortsHP4}];
        AFTs = [AFTSW1 ; AFTHP1 ; AFTHP2 ; AFTHP3 ; AFTHP4];

    case 'ConfigWL1STA'
        %Config WL - Case 1 Station
        %AFT SW1
        FSSW1P1 = [{'FF0AP'} {'000S1'}];
        PortsSW1 =[{FSSW1P1}];
        AFTSW1 = [{'SW1'} {PortsSW1}];
        %AFT AP
        FS0APP1 = [{'FFSW1'}];
        FS0APP2 = [{'000S1'}];
        Ports0AP =[{FS0APP1} {FS0APP2}];
        AFT0AP = [{'0AP'} {Ports0AP}];
        AFTs = [AFTSW1 ; AFT0AP];

    case 'ConfigWL2STA'
        %Config WL - Case 2 Stations
        %AFT SW1
        FSSW1P1 = [{'FF0AP'} {'000S1'} {'000S2'}];
        PortsSW1 =[{FSSW1P1}];
        AFTSW1 = [{'SW1'} {PortsSW1}];
        %AFT AP
        FS0APP1 = [{'FFSW1'}];
```

```
       FS0APP2 = [{'000S1'} {'000S2'}];
       Ports0AP =[{FS0APP1} {FS0APP2}];
       AFT0AP = [{'0AP'} {Ports0AP}];
       AFTs = [AFTSW1 ; AFT0AP];

   case 'ConfigWL3STA'
       %Config WL - Case 3 Stations
       %AFT SW1
       FSSW1P1 = [{'FF0AP'} {'000S1'} {'000S2'} {'000S3'}];
       PortsSW1 =[{FSSW1P1}];
       AFTSW1 = [{'SW1'} {PortsSW1}];
       %AFT AP
       FS0APP1 = [{'FFSW1'}];
       FS0APP2 = [{'000S1'} {'000S2'} {'000S3'}];
       Ports0AP =[{FS0APP1} {FS0APP2}];
       AFT0AP = [{'0AP'} {Ports0AP}];
       AFTs = [AFTSW1 ; AFT0AP];

end


%The second part of the code corresponds to the topology discovery
algorithm

%L(1) gives the number of HNIDs
L = size(AFTs);
flag = 0;

for i = 1:L(1);
    %M(2) gives the number of ports for every HNID
    M = size(AFTs{i,2});
    I = FindIndex(AFTs{i,1});
    for j = 1:M(2);
        %N(2) gives the number of elements in the port's forwarding
set
        N = size(AFTs{i,2}{j});
        for k = 1:N(2);
            J = FindIndex(AFTs{i,2}{j}{k}(3:5));
            AFTsMatrix(I,J) = 1;
            if strcmp(AFTs{i,2}{j}{k}(1:2),'FF')== 1;
                AdjacencyMatrix(I,J) = 1;
                flag = 1;
            end
        end
        if flag == 0
            for k = 1:N(2);
                J = FindIndex(AFTs{i,2}{j}{k}(3:5));
                AdjacencyMatrix(I,J) = 1;
            end
        else
            flag = 0;
        end
    end
end
```

```
% The third part of the code shows the graph of the final neighbor
list
gObj = biograph(AdjacencyMatrix,NetworkDevices);
gObj = view(gObj);
```

B.3 Results Topology Discovery Algorithm for HNTD

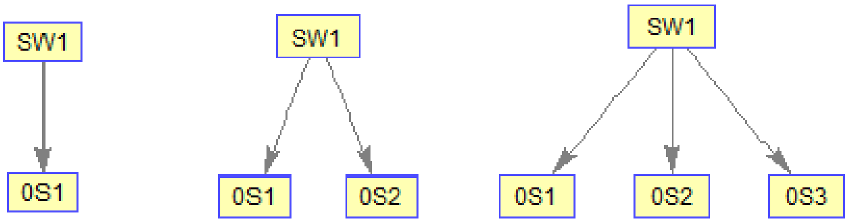


Figure 79. Graphs for Config Eth

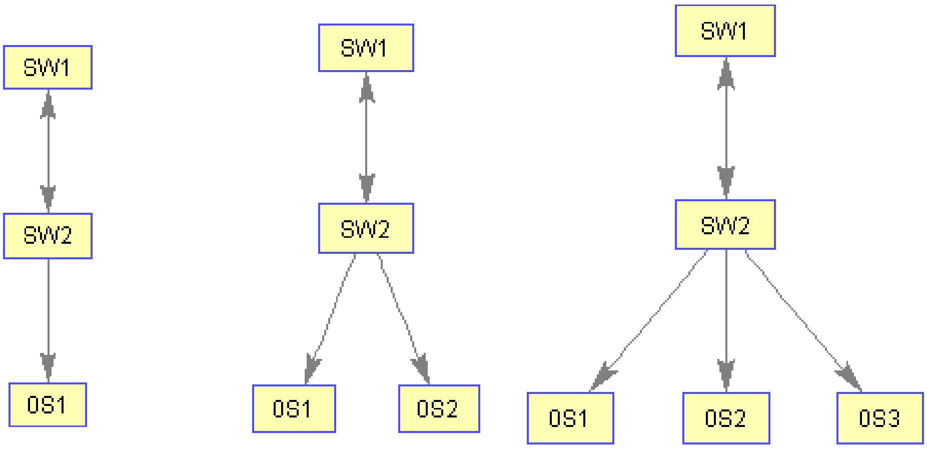


Figure 80. Graphs for Config SW

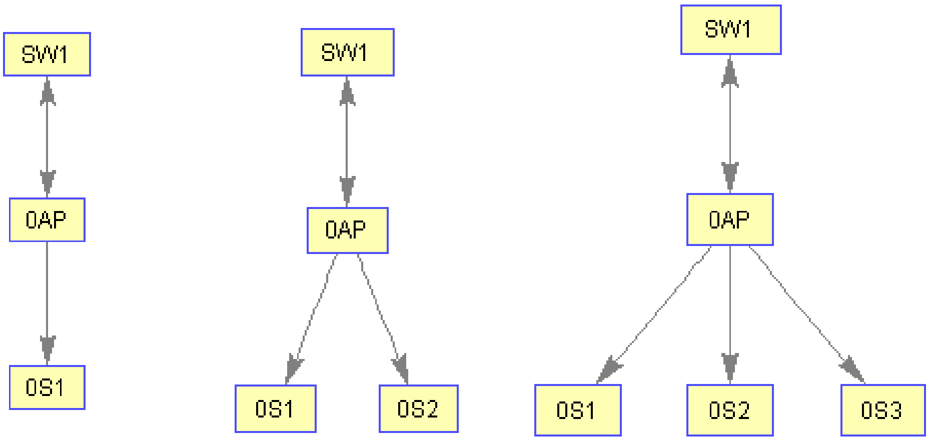


Figure 81. Graphs for Config WL

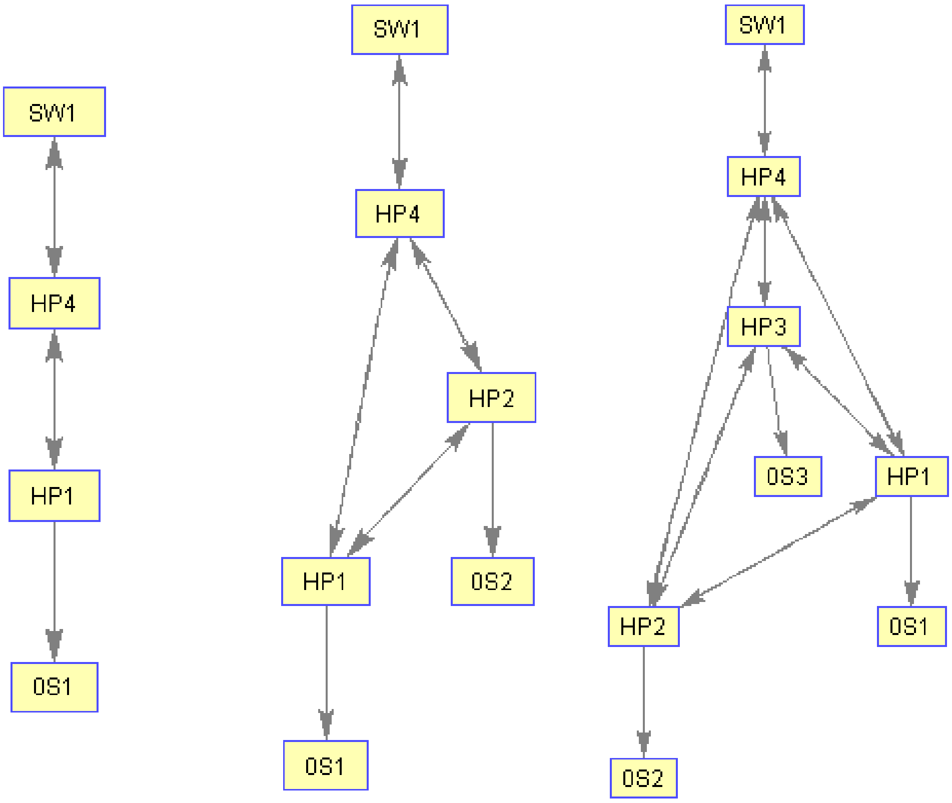


Figure 82. Graphs for Config PLC

Configuration Eth

	RG	S1
RG	0	1
S1	1	0

	RG	S1	S2
RG	0	1	1
S1	1	0	0
S2	1	0	0

	RG	S1	S2	S3
RG	0	1	1	1
S1	1	0	0	0
S2	1	0	0	0
S3	1	0	0	0

Configuration SW

	RG	SW	S1
RG	0	1	0
SW	1	0	1
S1	0	1	0

	RG	SW	S1	S2
RG	0	1	0	0
SW	1	0	1	1
S1	0	1	0	0
S2	0	1	0	0

	RG	SW	S1	S2	S3
RG	0	1	0	0	0
SW	1	0	1	1	1
S1	0	1	0	0	0
S2	0	1	0	0	0
S3	0	1	0	0	0

Configuration WL

	RG	AP	S1
RG	0	1	0
AP	1	0	1
S1	0	1	0

	RG	AP	S1	S2
RG	0	1	0	0
AP	1	0	1	1
S1	0	1	0	0
S2	0	1	0	0

	RG	AP	S1	S2	S3
RG	0	1	0	0	0
AP	1	0	1	1	1
S1	0	1	0	0	0
S2	0	1	0	0	0
S3	0	1	0	0	0

Configuration PLC

	RG	HP4	HP1	S1
RG	0	1	0	0
HP4	1	0	1	0
HP1	0	1	0	1
S1	0	0	1	0

	RG	HP4	HP1	HP2	S1	S2
RG	0	1	0	0	0	0
HP4	1	0	1	1	0	0
HP1	0	1	0	1	1	0
HP2	0	1	1	0	0	1
S1	0	0	1	0	0	0
S2	0	0	0	1	0	0

	RG	HP4	HP1	HP2	HP3	S1	S2	S3
RG	0	1	0	0	0	0	0	0
HP4	1	0	1	1	1	0	0	0
HP1	0	1	0	1	1	1	0	0
HP2	0	1	1	0	1	0	1	0
HP3	0	1	1	1	0	0	0	1
S1	0	0	1	0	0	0	0	0
S2	0	0	0	1	0	0	0	0
S3	0	0	0	0	1	0	0	0

Figure 83. Adjacency Matrices