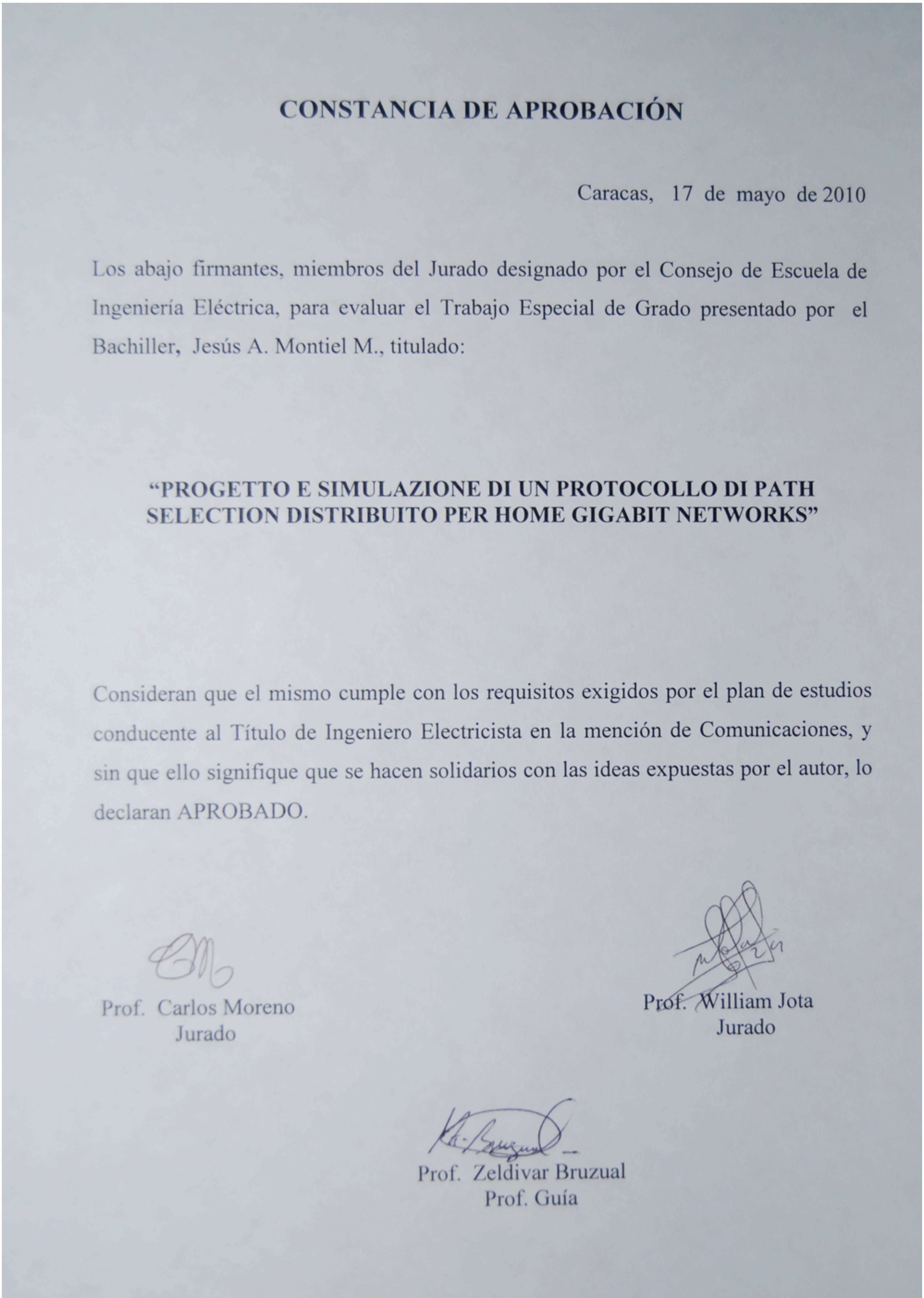
TRABAJO ESPECIAL DE GRADO

DISEÑO Y SIMULACION DE UN PROTOCOLO DE PATH SELECTION DISTRIBUIDO PARA REDES DOMESTICAS CON CAPACIDAD GIGABIT

Prof. Guia: Ing. Zeldivar Bruzual
Tutor: Prof. Francesco Delli Priscoli

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Montiel M., Jesús A.
para optar al Título de
Ingeniero Electricista

Caracas, 2010



CONSTANCIA DE APROBACIÓN

Caracas, 17 de mayo de 2010

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller, Jesús A. Montiel M., titulado:

“PROGETTO E SIMULAZIONE DI UN PROTOCOLLO DI PATH SELECTION DISTRIBUITO PER HOME GIGABIT NETWORKS”

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Comunicaciones, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.

Prof. Carlos Moreno
Jurado

Prof. William Jota
Jurado

Prof. Zeldivar Bruzual
Prof. Guía

Montiel M., Jesús A.

DISEÑO Y SIMULACION DE UN PROTOCOLO DE PATH SELECTION DISTRIBUIDO PARA REDES DOMESTICAS CON CAPACIDAD GIGABIT

Tutor Académico: Prof. Francesco Delli Priscoli. Co-tutor Académico: Dr. Ing. Vincenzo Suraci. Tesis. Roma. Università di Roma "La Sapienza". Facoltà di Ingegneria. UCV. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Mención: Comunicaciones. 2010. 98h + Anexos.

Palabras claves: Path Selection, Distribuido, HWMP, Home Gigabit, QoS, Redes Heterogéneas a malla

Resumen. El presente trabajo plantea estudiar y adaptar un protocolo de *Path Selection* Distribuido, para la comunicación dentro de redes de tipo malla con tecnologías heterogéneas. Para esto se propuso utilizar y adaptar el protocolo *Hybrid Wireless Mesh Protocol* (HWMP) el cual es a su vez el protocolo propuesto para el estándar IEEE 802.11s. Se codificó el protocolo utilizando el programa OPNET Modeler, implementando el *Path Selection Engine* y al mismo tiempo, trabajando con parámetros de los otros *Engines* (*Monitoring, Forwarding, QoS*) de la arquitectura de nivel I-MAC. Se definió la métrica a utilizar (banda, retardo de propagación y *hop count*), la política para la selección del *Path* y un método de cooperación eficiente entre la parte proactiva y reactiva. Además de esto, se estudiaron parámetros que pudiesen ser optimizados para mejorar las prestaciones del protocolo cuando este es utilizado en redes domésticas. Para garantizar la *QoS* se añadió un nuevo campo en los paquetes de control que llevan la métrica requerida para la aplicación a transmitir. En las simulaciones se estudiaron las prestaciones generales del protocolo cuando éste es implementado en una red heterogénea a malla en ambiente doméstico. Estas prestaciones incluían: *overhead* del protocolo, tiempo de convergencia y establecimiento del *Path*, tiempo de respuesta a fallas y por último, tiempo de recuperación de un *Path* debido a cambios de posición.

Montiel M., Jesús A.

DESIGN AND SIMULATION OF A DISTRIBUTED PATH SELECTION PROTOCOL FOR HOME GIGABIT NETWORKS

Tutor: Prof. Francesco Delli Priscoli. Co-tutor Académico: Dr. Ing. Vincenzo Suraci. Thesis. Roma. Università di Roma "La Sapienza". Facoltà di Ingegneria. UCV. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Mención: Comunicaciones. 2010. 98h + Anexos.

Keywords: Path Selection, Distributed, HWMP, Home Gigabit, QoS, Mesh Networks

Abstract. This thesis work proposes to study and adapt a Distributed Path Selection protocol for communication within mesh networks with heterogeneous technologies. For this end, it was proposed to use and adapt the Hybrid Wireless Mesh Protocol (HWMP) which is in turn the proposed protocol for the IEEE 802.11s standard. The protocol was programmed using the OPNET Modeler, implementing the Path Selection Engine while working with the other Engines parameters (Monitoring, Forwarding, QoS) of the architecture of the I-MAC level. Was defined the metric to be used (band, propagation delay and hop count), the policy for selecting the Path and an efficient cooperation method between the proactive and reactive components of HWMP. In addition, the parameters that could be optimized to improve the performance of the protocol when it is used in home networks were studied. To ensure the QoS, a new field was added in packages bearing the target metric required for the application to be transmitted. In the simulations the overall performance of the protocol when it is implemented in a heterogeneous mesh network in the home environment was studied. These performance parameters included: overhead of the protocol, convergence time and establishment of the Path, response time to failures and finally, recovery time of a path due to changes in position.

Università Degli Studi Di Roma "La Sapienza"
Facoltà di Ingegneria

Corso Di Laurea Specialistica In Ingegneria Delle Telecomunicazioni

Tesi di Laurea:

Progetto e Simulazione di un Protocollo di Path Selection Distribuito per Home Gigabit Networks

Relatore:
Prof. Francesco Delli Priscoli

Correlatore:
Dr. Ing. Vincenzo Suraci

Laureando:
Jesús Alberto Montiel Maillo

ANNO ACCADEMICO 2009 - 2010

A mis padres y hermanos

"Non ho mai incontrato un uomo così ignorante
dal quale non abbia potuto imparare qualcosa"
Galileo Galilei

"Innovation distinguishes between
a leader and a follower"
Steve Jobs

"Importa
si no te rindes"
Stephen Hawkings

Index

List Of Figures

List Of Acronyms

OMEGA: hOME Gigabit Access

MBSS: Mesh Basic Service Set

STA: Station

AODV: Ad-Hoc On Demand Distance Vector

OLSR: Optimised Link State Routing

HWMP: Hybrid Wireless Mesh Protocol

AP: Access Point

LOS: Line Of Sight

VLC: Visible Light Communication

UWB: Ultra Wide Band

FTTH: Fiber To The Home

ARP: Address Resolution Protocol

UMTS: Universal Mobile Telecommunication System

MAC: Medium Access Control

WMN: Wireless Mesh Network

LED: Light Emitting Diode

PLC: Power Line Communication

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

IP: Internet Protocol

HAN: Home Area Network

HWO: Hybrid Wireless Optics

DSL: Digital Subscriber Line

QoS: Quality of Service

BSS: Basic Service Set

LTE: Long Term Evolution

Introduction

The way people communicate has evolved at an impressive speed, especially in the last two decades. Back in the early 1970's people used the public telephone along with letters and telegraphs as the main forms of communication. The arrival of the fax machine in the late 70's gave business and home users a new way of quickly transmitting information. It was not until the early 90's that the internet, the network that would eventually revolutionise the way people communicate, started its exponential growth to become what it represents today. In its early days, the internet offered little usability. In fact, it was limited to pages created mainly by commercial and government entities. Later on, the internet slowly became a tool for the end user to communicate directly to people all over the world by means of email, instant messages, forums and live audio/video chats. As the users multimedia requirements grew, so did the required bandwidth necessary to fulfil them. For sometime people were limited to dial-up connections ranging from 28.8 kbps to 56 kbps, but years later broadband connections made it possible for the end user to use new services such as high definition video streaming, live video chat and video conferencing at higher resolutions, remote data storage, high definition online gaming, just to name a few. The internet today has become one of the most important expanding markets in the world. In fact, every day new services are offered, further raising the bar for the connection speed requirements.

Years ago, our only access to the internet or any other network was a desktop computer, but as of today, the way we access a network has expanded to many other devices. Today, the number of network capable devices ranges from the typical desktop computer, to printers, music players, laptops, gaming consoles, digital cameras, HDTV sets and even refrigerators. Furthermore, nowadays services such as the home lighting, thermostat control or the security system can now be controlled remotely, whether the user is inside or outside the house. It is clear that every day more and more network capable devices are brought into the market, therefore increasing their number on a given network. Since today's home users tend to have at least one network capable device per person, considering a home with 4 family members and considering also other secondary devices (e.g. refrigerator, network controlled system) that also accesses the network, we end up having a considerable amount of network capable devices. In reality, as time passes by, a standard home will have so many network capable devices together that a new proposal for Home Networking had to be taken into consideration.

Home networks as of today suffer from two main problems. First, the initial infrastructure and networking experience required for building and setting up a network that covers the entire home is considerable. This in turn requires the placement of at least one wireless access point and the introduction of cabling throughout the house in order to obtain complete coverage. Second, even a completely covered network-wise home is limited in terms of the bandwidth available by the single technology implemented and this is not sufficient to satisfy the increasing requirements for the numerous network devices that may eventually share the network. The OMEGA network proposal intends to revolutionise the way people communicate, either when communicating inside the Home Network or to external networks such as the Internet. Moreover, OMEGA will create a Home Network environment that uses the existing home infrastructure without the need to install additional cabling. In addition, since in the future home networking will become a service as common as electricity and gas, OMEGA intends to develop an easy to use and set-up home network that will eventually become a normal home service such as those mentioned. The key aspects that the OMEGA network offers are:

- *Technology Independent:* Allows devices to connect to the OMEGA home network independently of the technology.
- *MAC Layer Convergence:* The idea of making the OMEGA network converge at a MAC level is to make it transparent to the end user so that no additional software is needed, facilitating the installation of the network.
- *Hybrid Architecture:* A central node will have the responsibility of handling the network but the it will also be distributed in order to increase its mobility and capacity.

MAC Layer Convergence was already obtained through the use of the Inter-MAC layer, which is inserted between the IP layer and the MAC layer, allowing the use of different technologies in a simple and efficient way. The issue that still needs a solution concerns finding the most efficient way for devices inside OMEGA to find paths and communicate. Having a hybrid network such as OMEGA allows devices to communicate with each other and share information without the need for the OMEGA Gateway to intervene. Therefore, an OMEGA network with network capable devices such as laptops, smartphones, music players and even common household products such as microwaves and refrigerators that communicate with each other independently using different technologies make OMEGA a heterogeneous home meshed network. Moreover, considering that in the near future mobile devices will dominate the majority of network capable devices, with more and more wireless technology standards coming into perspective, ranging from the already in use WiFi standards 802.11b/g/n, WiMax 802.16 and UMTS (3G mobile telecommunication technology), to still under research UWB 802.15.3 and 802.15.4, LTE (4G mobile telecommunication technology), VLC (Visible Light Communication) and HWO (Hybrid Wireless Optics), it is highly probable that many of the devices in OMEGA will be mobile devices, consequently making of OMEGA a wireless/wired heterogeneous home meshed network.

Therefore, OMEGA has to offer connectivity to fixed and mobile devices with an efficient Path Selection algorithm that provides low latency and low overhead while maintaining QoS requirements inside an heterogeneous home meshed network with Gbps Capacity. To this end, a distributed Path Selection algorithm had to be found in order to efficiently meet all these requirements. Over the last few years, many distributed algorithms have been proposed for this type of situation where devices communicate with each other independently. Some of the most notable are AODV, which is a distributed reactive path selection protocol, and OLSR, which is a distributed proactive path selection protocol. AODV as all reactive protocols, do not discover a route until a flow is initiated, creating an initial delay to start communication that can be as high as the network is large. On the contrary, OLSR as all a proactive protocols suffer no such delay since paths are established proactively. However, on the downside these paths result to be sub-optimal and might not meet the QoS requirements. In order to satisfy a low initial delay while creating an optimum path to the destination that sustains the QoS requirements a different approach had to be considered about the Path Selection inside OMEGA.

In this context, this thesis proposes an adaptation of the *Hybrid Wireless Mesh Protocol* (HWMP) as the Path Selection protocol for the OMEGA network. HWMP is the default path selection protocol for IEEE 802.11s, and being a hybrid protocol, it shares both reactive and proactive modes and benefits to find paths to a destination. The work carried out in this thesis is comprised of the study, project and implementation in OPNET modeler of HWMP inside an OMEGA network with the final objective of studying its performance and the possible parameters that could be optimised in order to fully adapt it to the network involved. To arrive at this objective, a new Path Selection Engine inside OPNET was created to accurately represent both reactive and proactive components of HWMP. This engine interacts with previously created engines such as the Forwarding engine, QoS engine and Monitoring engine. Further, everything involving the path metrics, metrics propagation and path decision criteria based on the metrics was defined. Also, a cooperation method between the proactive and reactive components of HWMP was implemented in order to take full advantage of the Hybrid capability of this protocol.

This thesis is divided into five chapters.

The *First Chapter* presents the OMEGA project, introducing the context in which this thesis has evolved, indicating as well a technical description of the proposed network. Further, it presents the consortium involved in this project with a reference to the work in which the University of Rome is in charge.

In the *Second Chapter* the problem of network convergence is presented along with the Inter-MAC solution proposed by the University of Rome. Thereafter, presenting the Inter-MAC requirements as well as the proposed architecture for this convergence layer.

The *Third Chapter* starts by giving an overview of mesh networks and the different path selection approaches. Thereon, giving an overview of the most common reactive and proactive path selection protocols, AODV and OLSR respectively. Subsequently, a detailed explanation of HWMP is presented.

The *Fourth Chapter* begins by introducing OPNET modeler as a simulation tool. Thereafter, it continues by giving an overview of the Path Selection engine (HWMP) was implemented inside OPNET and how it interacts with the other engines.

And finally the *Fifth Chapter* presents the results obtained after various simulations were made on several different scenarios. First the scenarios considered are presented, giving information on the technologies that each network device presents and the reason why each scenario was ultimately chosen. Finally, the results involving overhead, convergence and path set-up times, link failure response times and others are depicted and analysed.

Chapter One:The OMEGA Project

1.1. Project Goal

The OMEGA project will set a global standard for ultra broadband home area networks. The new standard will enable transmission speeds of one gigabit per second (1 Gbps) via heterogeneous communication technologies, including power line communications and wireless connections. Thus, OMEGA aims to make home area networks as easy to use as electricity from the socket, putting an end to the coverage limitations as well as the wiring clutter in the home. With OMEGA's gigabit home network, users will get easy access to high- bandwidth information and communication services such as telepresence, 3D gaming, enhanced interactivity, virtual reality, high definition video as well as e-health applications and services for the exchange of user-generated business or multimedia content.

1.2. Background

Home networks at gigabit speed are a pivotal technology for realising the EU's vision of the future Internet. The demand for gigabit home networks is driven by the emerging future Internet services running over new high-speed optical access networks and the rapidly growing number of communicating devices in the home. Current home networks suffer from the fact that many devices are limited to gross transmission rates of 54 megabit per second in case of wireless links, or require troublesome wiring to achieve higher rates. Thus, current home networks are at risk of becoming a bottleneck, when fed by high-speed optical access networks, which offer 100 megabit per second or more, both down and upstream. The future Internet will offer extremely high bandwidth in core and access networks. Home area networks play a key role in realising the benefits of this high bandwidth and making it tangible for the users by providing critical access to this infrastructure for end devices within the home. Extending access into the home and to individual devices is the only way to ensure the success of the future Internet. Future home area networks must enrich the lives of users, for example by allowing visual communications with their friends or relatives and by enabling interactive experiences through entertainment. Furthermore, home networks should also support citizens in maintaining their independence as they age, for example by offering remote healthcare and by allowing them to communicate with their family to reduce any sense of isolation they may have. In short, users must have the ability to control their virtual as well as their physical environment via home networks. Users will require such networks to be simple to install, without any new wiring, and easy enough to use so that information services running on the home area network will be just another utility, like electricity, water, and gas. The OMEGA project is centred on the needs of the user: gigabit radio frequency and optical links, combined with more robust local-area radio frequency and visible-light communications will provide wireless connectivity within the home and its surroundings. Combined with power line communications this provides a communications backbone in the home without new wires. A technology-independent MAC layer will control this network and provide services as well as connectivity to any number of devices the user wishes to connect in the home network in any room of a house or apartment. Furthermore, this MAC layer will allow the service to follow the user from device to device. In order to make this vision come true, substantial progress is required in the fields of power line, optical-wireless and radio frequency physical layers, in protocol design, and in system architecture.

1.3. Technical Approach

The OMEGA home network will deliver Gbit/s capacity and low latency within the home and to the access network, with either wireless transmission or transmission using existing wired home infrastructure, thus enabling access to and the development of new and innovative services.

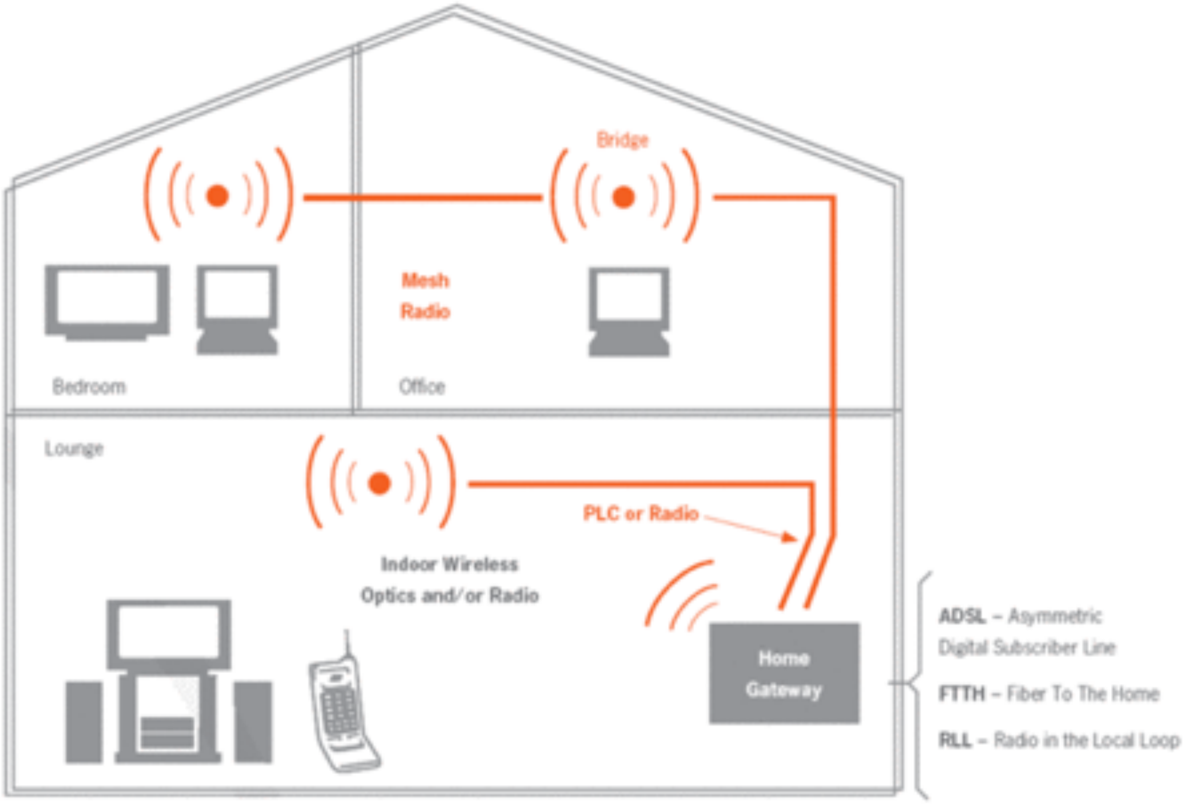


Figure 1 - Ultra-Broadband Home Area Network - OMEGA.

Figure 1 illustrates the network concept. Data enters the home and is routed by the home gateway. The gateway in turn is connected to OMEGA hardware, which can deliver Gbit/s data transmission by use of either line-of-sight (LOS) 60-GHz RF or LOS optical-wireless links. Room-area communications is provided through ultra wide band (UWB) and broadcasting by use of visible-light communications (VLC). To extend UBB penetration, the gateway can also use low frequency RF to connect to terminals, or use power-line communications (PLC) beyond state of the art 100 Mbps net to connect to OMEGA bridges within the house. For the RF channels 60GHz systems can provide Gbit/s data capacity, but require 'almost' LOS between terminals. For fixed devices such as TVs this is sufficient, but under conditions of blocking a wide-area lower-frequency (3-10 GHz) UWB channel can provide both coverage and capacity. In addition, compatible legacy 802.11(g) and (n) WiFi systems will be used to achieve this. LOS optical wireless can provide Gbit/s communications. Devices can have a low complexity compared with their RF counterparts, and their complexity does not increase with increasing data rate. This provides a potentially attractive alternative to LOS RF, especially as data rates climb beyond Gbit/s to 10 Gbit/s.

The adoption of LED lighting makes VLC and attractive option for data broadcasting. With little additional complexity, LED sources can be modulated to provide both illumination and data transmission, and a proof-of-concept link will be integrated within OMEGA. This technique is rather LOS, so diffuse "fallback" schemes will be investigated, creating an hybrid wireless optics (HWO) subsystem, aiming to provide some robustness as well as UBB transmission capability. In addition, HWO will be combined with RF systems to provide wide-area coverage.

PLC will be used not only to link the devices to the gateway but also to connect the bridges within the OMEGA HAN. This will allow room-to-room connectivity for the channels constrained to a single 'space', e.g. UWB and HWO, and it will increase the available data rate to 'wired' nodes of the network.

The aforementioned access technologies represent a rather large and heterogeneous set of channels to be addressed by a single project, but this rather broad scope is necessary to provide Gbit/s capability over a wide range of channel conditions that intrinsically cannot be predicted by the provider of the HAN, since each consumer's home is different, and so is the communication technology available. A very small portion of new homes in Europe will be designed for UBB HANs in the next decade, so OMEGA must easily fit into existing housing over the wide range of 'traditional' designs in the EU. No home will be the same, with different interference conditions and impairments, and also as a range of different devices will be independently chosen by the consumer. The broad range of transmission standards envisaged for OMEGA will ensure UBB performance in a wide range of conditions and circumstances, with any number of devices and terminals.

These heterogeneous channels provide a robust communications capability, based on the complementary nature of the channels and the diversity of the available links. However, to realise the benefits of the channels convergence has to be at the "heart" of OMEGA.

Convergence will be implemented at several levels:

- The RF and Optical channels will have a MAC layer that uses these resources optimally in order to provide the best QoS to the user. This will be done taking into account interference and other factors impairments. This approach allows specific interdependencies, often complex, of the RF and optical channels to separately be taken into account.
- A specific inter-MAC that will communicate with the above technology-dependent MACs, using a 'technology-neutral' syntax that will accommodate new standards and devices as they become available. This layer will manage all aspects of the OMEGA UBB HAN network, including QoS, load balancing, intra and inter-technology handover, and it will ensure that terminals will be "always best connected". The syntax developed will also be "future-proof" by allowing newly introduced standards that conform to the syntax, such as plastic or single mode optical fibre transmission, to attach to it.

The Gbit/s capability of OMEGA, combined with the focus on future-proof convergence, will seamlessly deliver services such as HD video to the user throughout the home, act as an enabler for FTTH and other second-generation roll-outs by allowing service providers direct access to terminals, with no 'bottleneck', thus providing new business opportunities. Furthermore, OMEGA will enrich the lives of the consumer by providing simple access to the information and services of the future Internet.

1.4 OMEGA Terminology

In this paragraph will be shown the terminology used in OMEGA project, divided into their membership group.

1.4.1 Network infrastructure

OMEGA Network: it is an Ultra Broadband home network, or Gigabit home network. It integrates different heterogeneous telecommunication technologies (e.g. PLC, Wi-Fi, UWB, HWO) to provide connectivity between home network elements and devices. The integration of different telecommunication technologies is realised by mean of the Inter-MAC layer.

Home Network Segment: it is a network inside the home realised using only one specific technology. For example, it can be a Wi-Fi network, or a PLC network. In the OMEGA project, different Home Network Segments are melt in a unique network to realise the OMEGA Network.

1.4.2 Network Elements

OMEGA Device (OD): it is a network element that implements the Inter-MAC layer and the protocols defined within the OMEGA framework. It is able to connect to all the others OMEGA devices belonging to the same OMEGA network. Examples of OMEGA devices are: OMEGA Home gateway, OMEGA Extender, OMEGA enabled end-devices.

OMEGA Home Gateway (OHG): it is the boundary element of the OMEGA network. It interconnects the OMEGA network with the access network. All the information flows that enter or exit the OMEGA network pass through the OHG. It implements the Inter- MAC layer and can support all the PHY and MAC technologies considered within the OMEGA project: PLC, Radio and HWO. It also implements the protocols defined within the OMEGA framework.

OMEGA Extender (OE): it is a network entity that is used to extend the OMEGA network coverage, interconnecting different heterogeneous Home Network segments. It does not implement all the protocol stack, but it has only PHY, MAC and Inter-MAC layers. It can support any combination of the PHY and MAC technologies considered within the OMEGA project: PLC, Radio and HWO.

Home Manager (HM): it is a logical entity that implements all the functionalities for the management and control of the OMEGA network. It can be located in a unique network element, in case of centralised approach for network control and management, or can be distributed in several network elements, in case of a distributed approach.

Legacy Device (LD): it is a common device, realised without taking into consideration the OMEGA framework. It does not implement the Inter-MAC layer. Legacy devices must be taken into consideration in the OMEGA project to offer backward compatibility to users. Thus, OMEGA framework must consider the possible presence of legacy devices inside the OMEGA network.

End-Device (ED): it can be an OMEGA device or a legacy device, depending on the set of functionalities that it implements. It offers users the capability to connect to the OMEGA network and to benefit from all its services.

1.5 OMEGA Functionality

1.5.1 Generic Home Networks Functionalities

LAN connectivity: the home network must support connectivity between OMEGA devices. The LAN connectivity can be established over heterogeneous technologies. The OMEGA project will provide support for Ethernet, UWB, Wi-Fi, PLC, 60 GHz, and Free Space Optics.

WAN Connectivity: allows the devices to be connected to the WAN. Usually, only one element in the OMEGA network will implement this functionality, i.e. the OMEGA Gateway. WAN connectivity can be achieved using at least one technology: xDSL, FTTH, WiMAX, HSDPA, etc.

Data transfer: includes all the functionalities needed to support and realise the reception and the transmission of data.

Network attachment: is the functionality used to manage the provisioning of addresses at IP layer inside the home network and, in case, providing IP masquerading towards the external network. It can be realised using a server/client model.

Local Device management: is the functionality that allows the home network administrator to manage the devices of the network. In this way, the home network administrator is able to insert some policies in order to manage the network (i.e., parental control, energy consumption profiles, etc.). It can be realised using a server/client model.

Remote Device management: is the functionality that allows a remote (from the WAN) management of the devices belonging to the home network. In this way the network operator is able to insert some policies in order to manage the network (i.e., QoS parameters setting).

Operation and Maintenance: supports processes for periodic diagnosis of Generic OMEGA functional units. It performs the routine actions including measurements and tests for performance control, and adjustments and part reconfiguration, in order to prevent faults from occurring and to reconfigure worn parts before they cause OMEGA system failure.

I.5.2 OMEGA Network Functionalities

PHY connectivity: The ability of two adjacent OMEGA devices to communicate with each other over a physical media. The OMEGA project will provide support for UWB, WiFi, PLT, 60 GHz, and Free Space Optics by the OMEGA network.

Path selection: The functionality to compute one or multiple best paths between OMEGA source device(s) and OMEGA destination device(s) based on a certain optimality criterion. Dedicated path selection algorithms are used.

Inter-MAC Forwarding: assures layer 2 frame transfer. An OMEGA device will transfer incoming layer 2 frames to a neighbouring OMEGA device based on the destination and other information contained in the layer 2 frames and the forwarding information contained in the forwarding table. Some OMEGA devices might be configured not to forward traffic of other devices.

Load-Balancing: defines how to distribute traffic, going from a source node to a destination node, among two or more existing paths between the two nodes.

Connection Admission Control: decides whether or not to accept a new connection in the OMEGA network, in order to support QoS to accepted flows.

QoS mapping: decision to which QoS service class the session flow belongs to on the basis of the TSpec and RSpec related to the particular session or on the basis of DSCP bits of the IP header or the TID bits of the Wi-Fi MAC header, etc.

QoS marking and handling: is in charge of marking data packets (e.g. setting the OMEGA class of service bits) in order to differentiate each packet on the basis of the QoS class of service it belongs to. In addition, it is in charge of handling such data packets and managing the forwarding with the aim to fulfil QoS requirements (including QoS scheduling and shaping).

Monitoring: to monitor network links and devices availability.

Security: it includes all the functionalities implemented in order to provide a secure access to the network resources and to provide a secure transmission of data across the network.

Signalling agent capability: additional functionality is performed at session level to allow the establishment of a connection among two different OMEGA devices that want to communicate. The signalling agent functionality is then in charge of informing the application about the possibility to start or not to start the transmission of application data.

Device Discovery functionality: used to discover the presence of other devices and identify their functions and associated capabilities (e.g. available PHY interfaces, energy mode). Once a new device is discovered, information about the new device is available for the Inter-MAC functional block that can use such information for its functionalities.

Energy Saving functionality: used to manage the energy profiles defined by the manufacturer of the OMEGA device and set through the local device management client functional block. Energy Saving policies employed by an OMEGA device are sent to the operation and maintenance module in order to differentiate the operational ways of that particular OMEGA device. In fact, different levels of energy profiles can be brought to different ways of operation by Inter-MAC control functionalities in the Inter-MAC functional block.

User input support: this functionality is referred to the possibility to run user's application and to provide an interface with the user of the device.

Serve legacy devices: includes the functionalities used to allow legacy devices to use the services offered by the OMEGA network. The goal is to provide connectivity to legacy devices to the OMEGA network and to preserve the QoS experienced by the legacy device for a specific flow.

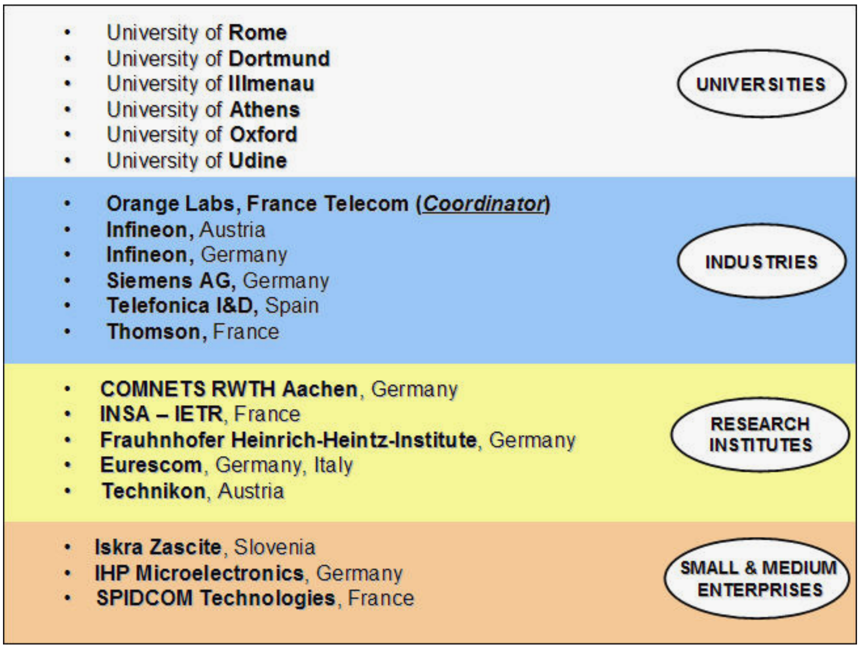
Handover: this functionality is related to the possibility to setup and tear down links, even dynamically.

Intra-home mobility management: this functionality is referred to the possibility to support session continuity while users are switching between different physical access technologies.

1.6 Consortium

OMEGA is an Integrated Project in the ICT area funded by the European Commission under the Seventh Research Framework Programme (FP7). The project is running for three years from January 2008 to December 2010. OMEGA will develop a user-friendly home area network capable of delivering high-bandwidth services and content at a transmission speed of one gigabit per second. The interdisciplinary project consortium consists of 20 European partners from industry and academia, as shown in Figure 2.

The OMEGA consortium comprises 5 manufacturers (Thomson, Infineon Germany and Austria, Siemens AG, Spidcom), 2 operators (France Telecom and Telefónica), 8 universities (Aachen, Dortmund, Rennes, Udine, Athens, Rome, Ilmenau, Oxford), 2 SMEs (Thyia, Technikon), 2 R&D centres (IHP and Heinrich-Hertz Institute), and 1 R&D programme management company (Eurescom). All partners are from EU Member States and thereof one from a new member state (Slovenia).



- University of **Rome**
- University of **Dortmund**
- University of **Illmenau**
- University of **Athens**
- University of **Oxford**
- University of **Udine**

UNIVERSITIES

- **Orange Labs, France Telecom (*Coordinator*)**
- **Infineon**, Austria
- **Infineon**, Germany
- **Siemens AG**, Germany
- **Telefonica I&D**, Spain
- **Thomson**, France

INDUSTRIES

- **COMNETS RWTH Aachen**, Germany
- **INSA – IETR**, France
- **Frauhnhofer Heinrich-Heintz-Institute**, Germany
- **Eurescom**, Germany, Italy
- **Technikon**, Austria

RESEARCH INSTITUTES

- **Iskra Zascite**, Slovenia
- **IHP Microelectronics**, Germany
- **SPIDCOM Technologies**, France

SMALL & MEDIUM ENTERPRISES

Figure 2 - Project Consortium.

The University of Rome through the "Dipartimento di Informatica e Sistemistica" (Department of Computer and System Sciences - UoR-CRAT) of the Faculty of Engineering has the following responsibilities in the OMEGA project broken down by work packages (WPs):

WP5 – Inter-MAC, Convergence layer

UoR-CRAT will lead the design of the inter-MAC layer, identifying its architecture, its functionalities, the structure of its components, and their exposed interfaces. UoR-CRAT will also investigate the definition of a semantic language to be used within the signalling and management plane, and as a common ontology shared by all WPs. UoR-CRAT will contribute to the design of the Inter-MAC protocol dealing with QoS, intelligent access control, scheduling, path selection, and security by use of an innovative approach derived from advanced control theories like robust control, game theory, and enforcement learning. The developed procedures and algorithms will be assessed by means of dedicated simulation tests in terms of reliability, efficiency, complexity, performances, and energy saving.

WP6 – Architecture, security and continuity of service

UoR-CRAT will lead the design of the Gigabit Home Network architecture, identifying the network components, their functionalities and interfaces for each architectural plane, management, signalling and transport. The overall architecture will be improved recursively with a feedback approach taking into account incremental design improvements of the technology-independent Inter-MAC layer and the technology-dependent MAC layers.

University of Rome is working in cooperation with companies such as France Telecom, Siemens, Infineon for control and management of OMEGA technology. University of Rome plans to validate the research that will be made in QoS and resource management areas using them to achieve convergence purposes (Inter-MAC convergence layer). In addition, the University of Rome intends to exploit the results of this project for didactic and teaching purposes. In particular, many master degree theses are expected to profit from the documentation and the background coming from the project in question. Moreover, project results will be exploited to upgrade and update the programs of several courses and to hold thematic seminars on these matters. Participation on this project will allow new generation engineers to acquire know-how on telecommunications and informatics and more specifically on Inter-MAC and on Next Generation Networks.

Chapter Two: Inter-MAC, a Convergence Layer

II.1 Inter-MAC as a Convergence Layer in Heterogeneous Networks

There has been vast amounts of investigation put on converging different telecommunications technologies onto a single network, as an attempt to create a more robust and usable network. The majority of the work has been done assuming IPv6 as a common base for next generation networks. Many open and lightweight middleware architectures have been designed to handle QoS, resource management and security aspects in heterogeneous networks, again assuming heterogeneous IPv6 networks. OMEGA is considered to be a next generation network, requiring to handle large amounts of data for services such as High Definition 3D video streaming (HDTV), online 3D gaming and HD video conferences, all through a high speed broadband internet connection and therefore creating a high demanding network in terms of resources. As a result, these services require a disruptive approach for the management of resources in a Home Gigabit Access Network where convergence should be achieved maintaining simplicity, scalability and backwards compatibility. The OMEGA project aims to solve middleware/IPv6 heterogeneous limitations enabling the cooperation of telecommunications technologies by developing an Inter-MAC convergence layer located between layer two (MAC level) and layer three (Network Protocol level). This approach is completely new because, to the best of our knowledge, there has not been any research efforts so far concerning the convergence of wired and wireless technologies in a challenging scenario of a multimedia and high demanding in terms of data rate Home Gigabit Networks (HAN).

II.2 The Inter-MAC Layer

The Inter-MAC layer is the proposed layer for convergence in the OMEGA network. The placement of this layer is below the network layer, and above the technology-dependent MAC layers. Having a technology-independent layer will provide a robust communication capability based on the diversity of available links. This Inter-MAC layer controls the different MAC layers by means of flexible and extensible technology-dependent Inter-MAC adapters and provides functionalities like path selection, guaranteed quality of service, security, energy efficiency and is backwards compatible.

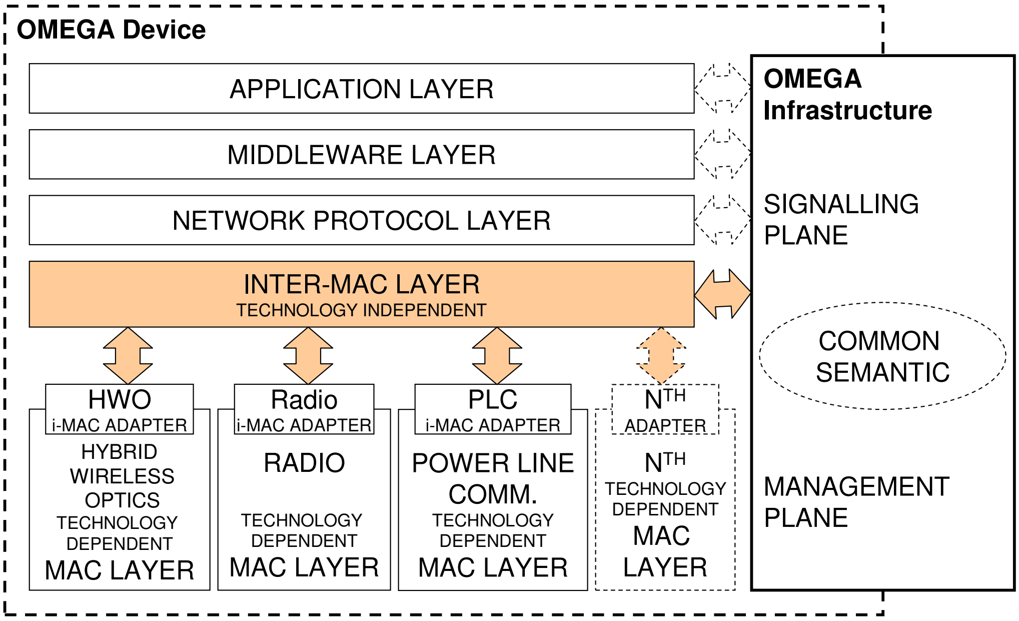


Figure 3 - Schematics of the Inter-MAC in the protocol stack.

As shown on figure 3, the Inter-MAC layer of the OMEGA architecture plays the key role for all inter-network communication aspects. It defines a set of common semantic elements to allow information to be exchanged between each technology dependent MAC layer and the inter-MAC. Furthermore, it provides a set of management plane elements that can be accessed by each other layer to optimise its function. It also accesses management plane elements of other layers to optimise its own function. Moreover, signalling plane elements are used to establish and to release connections, execute path selection as well as to set the QoS and security parameters. Both, the management plane and signalling elements can be used under a common management policy that allows it to control the overall OMEGA network. The Inter-MAC integrates arbitrary heterogeneous communication technologies in a single (home) network by providing a providing path selection and QoS support other these technologies. This will decrease the administration effort for the user. The Inter-MAC is also in charge of responding to network changes, like link failures or decrease of QoS, guaranteeing session continuity and transparency. A monitoring system collects and sorts the information gathered from the technology dependent MAC layers and the Inter-MAC peers.

The Inter-MAC provides path selection functionalities in order to automatically develop a hybrid mesh network supporting multi-hop connectivity over dissimilar technologies like radio, PLC and HWO. QoS Control is provided by the Inter-MAC in order to cope with the different QoS requirements between the higher network layers and the MAC protocols (Net-to-MAC adaptation) and between different MAC protocols (MAC-to-MAC adaptation). The QoS control is also in charge of managing the network resources under the premise to guarantee the agreed levels of multi-hop Quality of Service.

The fact that the Inter-MAC layer is technology independent will make the OMEGA network future proof and compatible with new technologies. The Gbps capability of OMEGA, combined with the Inter-MAC convergence features, will:

- Seamlessly deliver services as HD video to the user throughout the home.
- Act as enabler for second generation roll-outs (FTTH) by allowing service providers direct access to terminals, with no bottleneck.
- Improve the quality of life of the consumer by providing simple access to the information and services of the future Internet.

II.3 Inter-MAC Requirements

The Inter-MAC architecture provides a feasible solution to opposing requirements. On one hand it must guarantee the QoS required by innovative, gigabit services such as HDTV, 3DTV, on-line gaming, file sharing and home services such as VoIP, video conferencing and web navigation. On the other hand it must setup a hybrid wireless/wired meshed multi-hop network guaranteeing the interoperability of heterogeneous home technologies such as VLC, IRC, PLC, 60 GHz, 802.11n, UWB, legacy gigabit Ethernet and other future technologies. To yield a high acceptance of the Inter-MAC concept it must provide QoS on a flow-by-flow basis independent of the physical networking technologies involved, and independent of the specific operating system, assuming to interface uniquely with the L3 network protocol. Legacy devices should retain their original qualities and performance after the integration with Inter-MAC enabled devices.

On one hand the Inter-MAC architecture must be capable and efficient enough to be realised either as a hardware device, a software application (that can be run even on an embedded processor) or a custom hardware-software system. On the other hand it should be scalable to address different home scenarios with up to 50 network nodes with mobility and changing link qualities. Most scenarios, however, will not contain more than 20 network nodes. Further, the Inter-MAC has to generate low operating costs in terms of energy costs, system setup/configuration effort (e.g. authentication, accounting) and system maintenance effort.

II.4 Overview of the Inter-MAC Layer

The Inter-MAC architecture is divided into data plane, control plane, and management plane.

The *data plane* is responsible for transferring the data packets. It manages the packets arriving at a device, both form the upper network layer and the lower MAC layer. The received packets are arranged into an output queue according to their priority and the filtering rules, e.g. QoS requirements. It also transports the control and management-related data of higher layers between the respective entities.

The *control plane* performs short-term actions which allow to manage the data plane behaviour. As an example the requests of higher layer entities are handled and a flow to the desired destination with appropriate QoS will be established if possible. Furthermore it maintains and releases the flow, e.g. in case of failure.

The *management plane* is concerned with long-term actions which describe the behaviour of the device itself. This is defined by means of policies (e.g. for low-energy path selection) by the network administrator. These policies are stored in the policy repository in the management plane shown in figure 5.

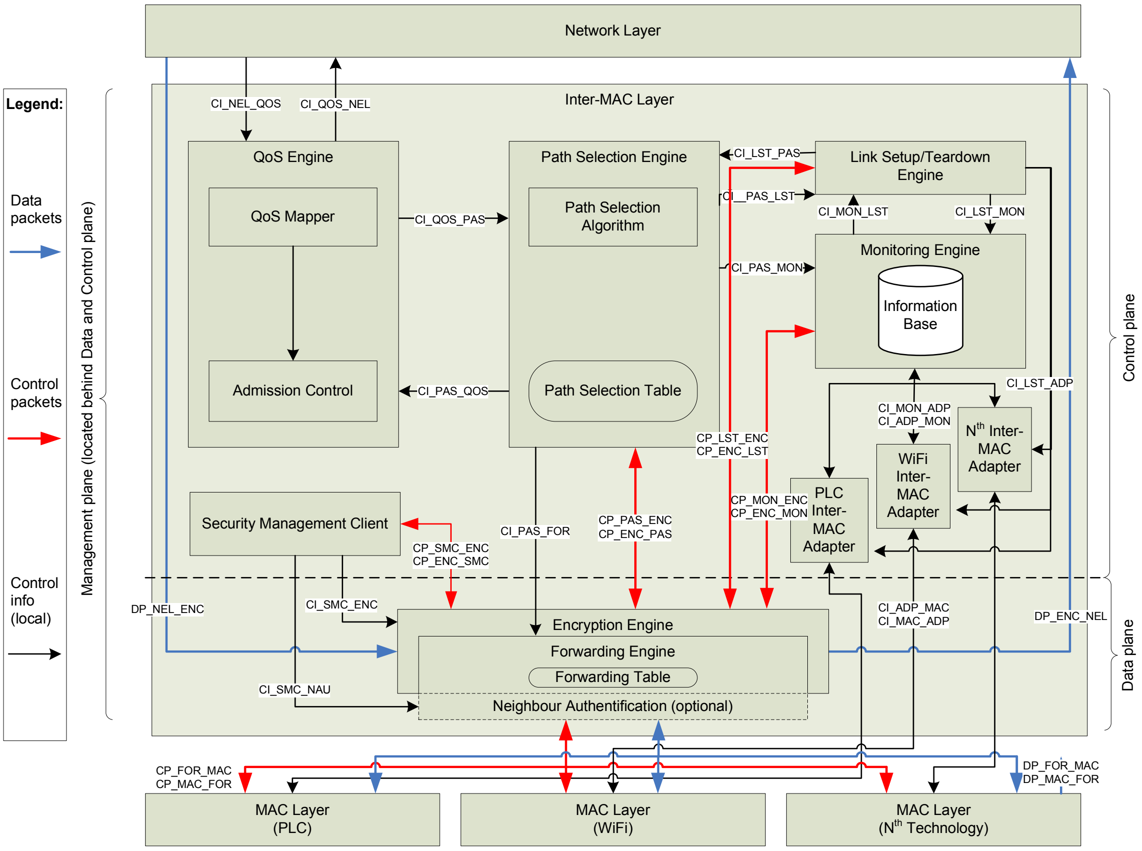


Figure 4 - Inter-MAC architecture, data and control plane.

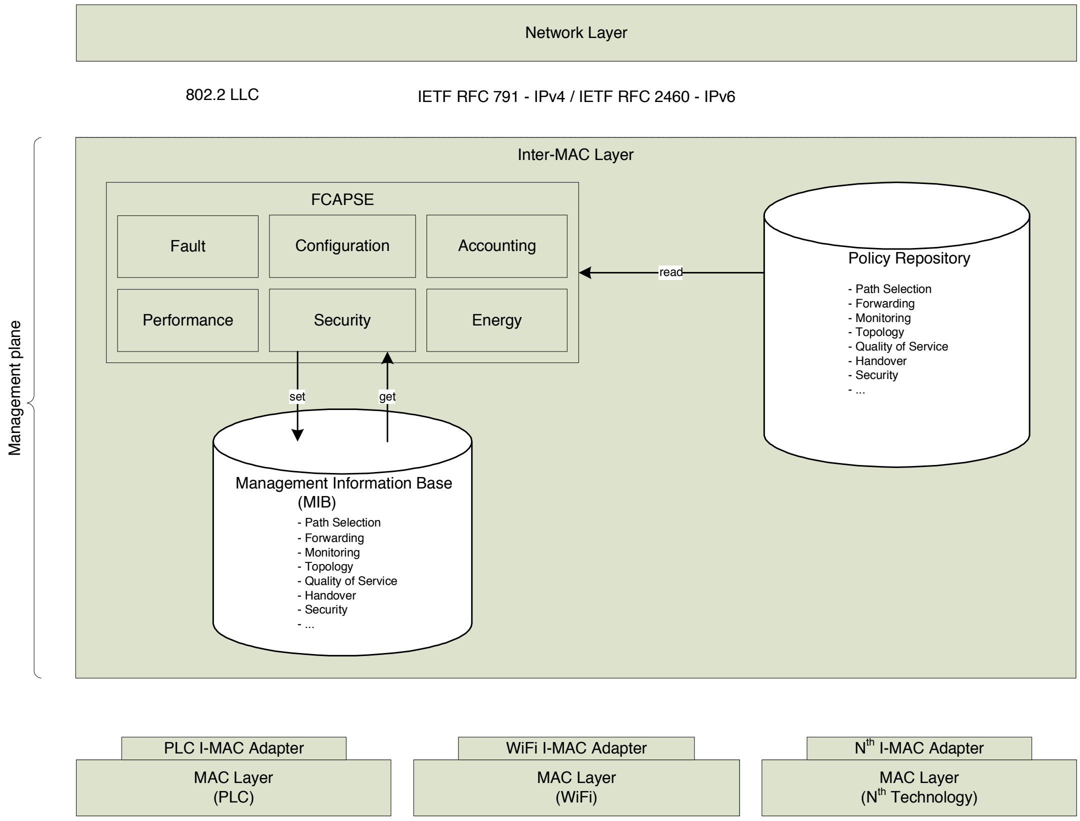


Figure 5 - Inter-MAC architecture, management plane.

II.5 The Inter-MAC Layer Architecture

II.5.1 Control Plane

The tasks of the control plane can be grouped into several engines, these are:

- *Monitoring Engine*, gathers network information to support configuration, performance and fault management of the OMEGA network.
- *Path Selection Engine*, individuate the paths towards every needed destination and memorise such paths into a table that is present in every OMEGA node.
- *QoS engine*, is responsible of the mapping between QoS classes of higher layer and the ones of Inter-MAC. It also handles Admission control inside the OMEGA network.
- *Link Setup/Teardown Engine*, facilitates vertical as well as horizontal handover.
- *Inter-MAC adapter*, enables the collaboration between MAC and Inter-MAC layers.

Each engine is detailed in the next paragraphs with at least a general description, functional description and interactions with other engines wherever considered appropriate.

II.5.1.1 Monitoring Engine:

The Inter-MAC monitoring engine is responsible for gathering network information which is mainly supposed to support configuration, performance and fault management of the OMEGA network. The monitoring engine is part of every OMEGA device (as an entity of the Inter-MAC) and directly exchanges control information with the Inter-MAC adapters of the OMEGA technologies that are associated with that device.

In case of a highly evolved technology, e.g. IEEE 802.11, a lot of information obtained on MAC and PHY level can be reused on Inter-MAC level. The Inter-MAC adapters are used to extract the relevant information from the MAC and PHY MIBs. Or the monitoring engine can trigger measurement routines directly. The measurements can be processed on-demand, regularly, or subscription based. The set of parameters and link metrics which can be gathered via the Inter-MAC adapters depends on the corresponding technology. It is obvious that an emerging technology like HWO which is still topic of fundamental research is not able to provide such a variety of parameters compared to WLAN. In this case the Inter-MAC adapters should either support a common basic subset of measurement parameters or the Inter-MAC can initiate the transmission of measurement packets itself.

The monitoring engine directly relates to the path selection as well as to the link setup/ teardown engine and the encryption engine. Figure 6 gives an overview of the monitoring engine within the Inter-MAC.

The monitoring engine does not explicitly implement a service that is offered to a higher layer, but it provides the Inter-MAC entities with valuable information which support the algorithms on Inter-MAC level. The path selection engine requires knowledge about the neighbourhood of the local station. Therefore the Inter- MAC MIB contains a list of all adjacent stations seen by the local station. Furthermore, it lists the capabilities for each of these stations regarding the available technologies and related QoS metrics. In order to obtain such knowledge about the neighbourhood, an OMEGA neighbour discovery protocol is necessary.

The link setup/teardown engine requires the indication of parameter or metric changes. For example, the monitoring engine will be able to generate events, which indicate the change of a QoS parameter for a link crossing a certain threshold. For the link setup/teardown engine this could mean to prepare a switch-over of the corresponding link to another technology. This approach is similar to IEEE 802.21 which defines events like Link_Detected, Link_Down, Link_Parameters_Report or Link_Going_Down. Such events are also useful for the path selection engine to indicate the necessity of a possible path re-calculation.

Management Information Base (MIB): The Inter-MAC MIB mainly contains two branches. One of them is the information with regard to the local station itself and the other one is the information which can be obtained from the neighbourhood. Attributes of the Inter-MAC such as the capabilities of the local OMEGA device, adjacent devices or active links at the local device can be seen as managed objects. Furthermore, the Inter-MAC MIB monitors the parameters of active links at the local station. This information should as possible be monitored and stored within the MIB for adjacent stations as well.

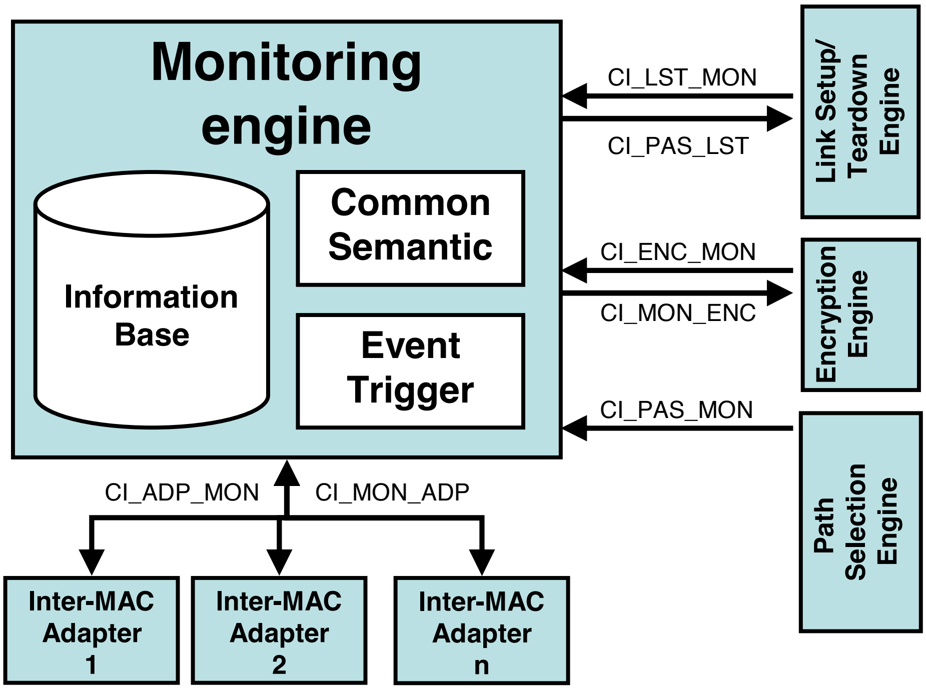


Figure 6 - Inter-MAC monitoring engine.

II.5.1.2 Path Selection Engine

II.5.1.2.1 Description

The path selection engine finds paths to every needed destination and builds up a detailed path selection table in each OMEGA node. It stores the available paths and its cost between any source destination pair taking into account QoS requirements from higher layers. This can be tight constrains like delay, throughput, jitter and leads to a variant of multi-constrained routing. The path selection entity is also responsible for continuous path maintenance. This includes the recovery from link breaks or recalculation of paths in case the path metric crosses a certain quality threshold.

II.5.1.2.2 Functionalities

In general, path selection and routing are synonymous. Path selection finds the best end-to-end path from a source to a destination. Path selection often refers to routing on layer 2 with MAC addresses. It works with nodes and links between these nodes. Cost metrics are assigned to links. In

OMEGA, path selection is technology independent and also works with links of heterogeneous technologies.

The key functions of path selection are:

- ➡ To find the best paths for flows between a source and a destination.
- ➡ To maintain the paths of the network devices.
- ➡ To assist other path selection engines of other devices in setting up paths.
- ➡ To inform other devices of changes in the network topology.
- ➡ To determine where to forward received packets (e.g. appropriate interface, next hop).

The non-functional requirements of path selection are described in the following overall design goals:

- Optimisation – Select the best path depending on the flow requirements and based on the link metrics.
- Scalability – Efficient operation for required number of nodes.
- Low overhead – Only few resources are taken for signalling.
- Low complexity – Easy implementation and integration.
- Robustness and stability – Correctness even when confronted by unusual or unforeseen circumstances (e.g. hardware failures, high load conditions).
- Flexibility – Quick adaptation to a variety of network changes (e.g. changes in node or link availability, bandwidth, delay).
- Rapid convergence – Convergence is the process of agreement by all nodes on available routes. Routing algorithms that converge slowly can cause data to be undeliverable.

The work carried out in this thesis will regard the specific topic of Path Selection within the OMEGA network, focusing on the distributed approach of path selection, studying several proposed protocols and thereafter concentrating on Hybrid Wireless Mesh Protocol (HWMP), a protocol that offers a Hybrid Proactive and Reactive approach.

II.5.1.3 QoS Engine

II.5.1.3.1 Description

The QoS engine manages the QoS mapping between the QoS of the Inter-MAC and the higher layers, and manages the Admission control of new flows inside the OMEGA network. The QoS engine hosts two sub- engines: the QoS mapper and the Admission control. The QoS mapper translates application layer QoS parameters to one of the six Inter-MAC service classes (Emergency, Conferencing, Streaming, Less Sensitive Traffic, Best Effort, Other). The Admission control is in charge to accept, reject and eventually drop the flows running in the OMEGA network.

II.5.1.3.2 Functionalities

The QoS mapper collects the QoS requirements coming from a QoS middleware, a session layer or an application through the CI_NEL_QOS interface and calculates which Inter-MAC class of service best matches with the provided QoS requirements. The Inter-MAC classes of service are then mapped in the lookup table of the Forwarding Engine to the QoS classes of the respective PHY-MAC class. The QoS mapper shall be capable of supporting QoS signalling generated at session/application layers. Application/session layer QoS frameworks are not standardised, therefore each middleware solution and each session protocol requires an OMEGA (software) driver to be developed. Hereafter, the cases of Session Initiation Protocol (SIP) and Universal Plug and Play (UPnP) QoS v3 will be used to exemplify the QoS mapper actions.

The signalling messages generated at session/application layer to initialise/negotiate/terminate a flow are collected by the QoS mapper module through the CI_NEL_QOS interface (see Figure 4 for a reference on the interfaces). The QoS mapper must be aware of the semantic of these messages, i.e., it must be capable of interpreting them. For this reason, appropriate software drivers are required to guarantee the compatibility with present and future upper layer QoS frameworks. Based on the traffic specifications defined in the upper layer signalling messages, the QoS mapper decides the most suitable class among the OMEGA QoS classes of service.

For instance the SIP protocol is widely used to support VoIP calls. The INVITE SIP message is received as a QoS Flow Request by the QoS mapper module, which, according-to its mapping rules, maps the VoIP call onto the Conferencing OMEGA class of service. The initialisation request is then forwarded to the Admission control module, which decides upon the admission of the new flow. In case of positive answer, the VoIP call packets are routed over a path in the OMEGA network which guarantees the requested QoS. The same procedure is triggered when the BYE SIP message (which terminates the call) is received by the QoS mapper.

Once the QoS mapper identifies the Inter-MAC class of service to be associated to the flow, the Admission control sub-engine is triggered in order to accept or reject the flow. The result of flow acceptation or rejection is sent back by the QoS engine to the Network Layer through the CI_QOS_NEL interface.

Admission control sub-engine determines based on the Path Selection response whether a certain flow can be admitted, dropped or rejected. Through the CI_QOS_PAS interface the Admission control advises the Path Selection Engine that a new flow should be served or an existing flow has to be stopped. Through the CI_PAS_QOS interface the Admission control receives the acceptation or rejection response of a new flow or receives a dropping warning for an existing flow whom QoS can be no more guaranteed. Once the Admission control asks to the path selection to provide a feasible path for this new flow, the path selection will evaluate the state of the network and discover the existence of suitable paths able to serve the QoS required by the new flow. Path selection will be performed only for a small set of high priority QoS Classes. That means that low priority (i.e. best effort) flows do not need to be admitted: they are always accepted and will rely on the best effort done by the underlying priority queues. If no network resources are available to host the new flow, it will be rejected, otherwise it will be accepted. Once a flow has been accepted, if the network status changes so that the path selection is no more able to find suitable paths for the required QoS of that flow, a dropping message will be sent to the QoS Admission control. Starting from that moment any QoS will be guaranteed to that flow.

II.5.1.4 <u>Link Setup/Teardown Engine</u>

The link setup/teardown service purpose is to improve the user experience of mobile devices by facilitating handover between physical interfaces whether or not they are of different types of technology. For the sake of simplicity hereafter the "link setup/teardown" concept will be referred as "handover".

The link setup/teardown module is responsible for making the necessary preparations in case of alteration of the path currently being used. Also, it is responsible for assisting path selection by reporting events that imply a new path calculation. It serves to path selection service using information provided by monitoring engine and the OMEGA information base.

II.5.1.5 Inter-Mac Adapter

The Inter-MAC adapter is a key element of the Inter-MAC architecture, since it enables the collaboration of different transmission technologies (PHY and MAC) and the Inter-MAC. It is understood as a software driver which can be loaded at a certain device in order to activate the Inter-MAC functionality for the corresponding technology. The driver has to be provided by the technology supplier which implements the translation of technology-dependent MAC service primitives to generic Inter-MAC primitives and vice versa. Furthermore it has to provide link metrics which can be taken into account for QoS control in order to keep track of the OMEGA network status. Furthermore the Inter-MAC may mediate between OMEGA service classes and the MAC protocol access categories. Figure 7 gives an overview of the functionality of the Inter-MAC adapter. An additional feature could support the translation of the Inter-MAC traffic specification to any technology-dependent traffic specification.

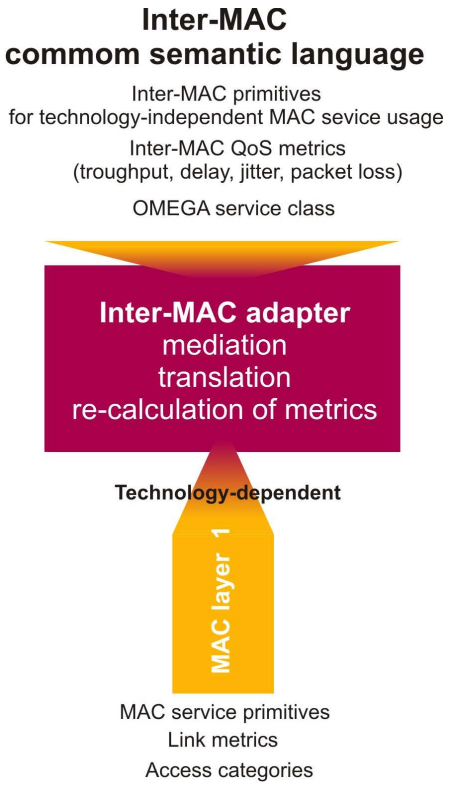


Figure 7 - Overview of the Inter-MAC adapter functionality.

In conclusion, the Inter-MAC adapter represents the possibility for new technology to participate in the OMEGA network and to gain as much OMEGA functionality as possible.

II.5.2 Data Plane

The data plane is composed by the following engines:

- Forwarding Engine, handles the incoming and outgoing forwarding of packets.
- Encryption Engine, cipher outgoing messages for data as well as for control plane activities.

Each engine is detailed in one of the next paragraphs with at least a general description, functional description is included wherever appropriate.

Refer to figure 4 for a picture of the data plane engines and how they fit into the overall Inter-MAC.

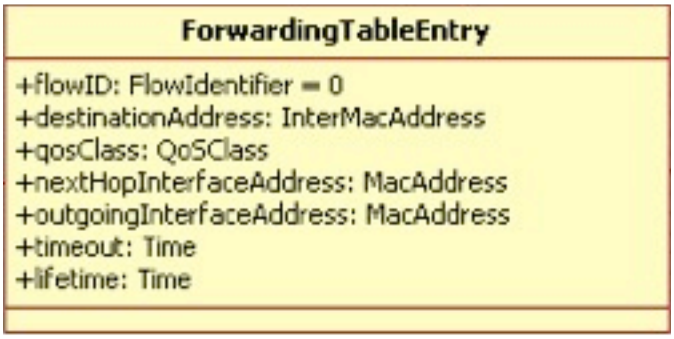
II.5.2.1 Forwarding Engine

When an OMEGA device receives a frame, the MAC layer de-encapsulate the payload to deliver it to the InterMAC layer. The call primitive used by the MAC layer is different according to the value of the Ethertype field:

Ethertype value in Mac header	Called entity
x9999	InterMAC (dataplane)
x0800 or x0806	Omega Legacy Device Adapter (dataplane)
x88E1	InterMAC adapter for PLC (controlplane)

The InterMAC dataplane analyses the InterMAC header to make the forwarding decision:

- If the InterMAC destination address is the local address or the broadcast address, then the frame is delivered to the upper layer specified in the ProtocolID field. If the frame is a control frame (control flag set to 1 and no ProtocolID field), the frame is delivered to the control plane.
- If the InterMAC destination address is not the local address, the device makes a lookup in the forwarding table to determine the next-hop. The structure of the forwarding table is shown in the next figure:



ForwardingTableEntry

+flowID: FlowIdentifier = 0
+destinationAddress: InterMacAddress
+qosClass: QoSClass
+nextHopInterfaceAddress: MacAddress
+outgoingInterfaceAddress: MacAddress
+timeout: Time
+lifetime: Time

Figure 8 - Forwarding Table Entry.

- The value "destinationAddress" contains either InterMAC address or Mac address for legacy devices.
- The value "qosClass" contains the value Best Effort, Elastic, Real Time or Low Latency.
- The value "nextHopInterfaceAddress" and "outgoingInterfaceAddress" contain the MAC addresses for the next-hop.
- The value "flowID" contains the flowID assigned to this entry.
- The value "lifetime" refers to the maximum time an entry is shall exist in the Forwarding Table.
- The value "timeout" refers to the moment when an entry is removed from the Forwarding Table. It is updated when an entry is used.

II.5.2.2 Encryption Engine

II.5.2.2.1 Description

The main functionality of the encryption engine is to cipher outgoing messages for data as well as for control plane activities. More complex means such as key exchange are handled by the security manager client. The cipher algorithms to be supported by the encryption engine are not yet defined. Promising candidates are the Advanced Encryption Standard (AES) for bulk, since it can be realised efficient in software and hardware and Gigabit throughput can be achieved. For supporting key exchange and digital signatures elliptic curve cryptography is considered since it provides higher security with less computational effort than standard RSA approaches. For supporting digital signatures a secure hash function will be provided by the encryption engine. Keys to be used, when en-/decrypting data, are retrieved from the security management client. Keys for currently active flows are stored for efficiency reasons in the encryption engine.

II.5.2.2.2 Functionalities

Bulk data en/decryption: messages originating from the device on which the encryption engine is running as well as messages destined for that device are en- and decrypted respectively. For this a symmetric cipher algorithm will be applied. For applying the correct keys the calling engine needs to identify the source or sink address.

Support for data integrity: the encryption engine implements a secure hash function, which is the prerequisite for checking data integrity. For bulk data message authentication codes are applied, i.e. after calculating the hash value of the packet to be sent the hash value is en/decrypted using the key of the respective source/sink pair. For control messages the same procedure is applied except that the private/public key pairs are used.

Authentication support: for authentication the same means as used for data integrity are provided by the encryption engine. Additional functionality such as checking black lists of certificates etc. are provided by the security management client.

II.5.3 Management Plane

The management plane provides the standard FCAPS management functions enhanced with energy management (FCAPSE) to allow for green IT, saving resources of battery powered devices.

- Fault management
- Configuration management
- Accounting management
- Performance management
- Security management
- Energy management

The management plane is composed by:

- *Policy Repository*, includes a database of policies that can influence the Inter-MAC core functionality.
- *Management Information Base*, do not provide functionalities, it is only a structured database.

It is important to remember that the retrieving and updating parameters of Inter-MAC components and associated OMEGA devices is done in an SNMP-like fashion, i.e. get and set primitives are used to access parameters of the Inter-MAC engines, which are represented in a data repository. The latter is denoted as management information base which is taken from the SNMP world but is still not meant as the only way to realise this functionality.

II.5.3.1 Policy Repository

In order to realise different strategies in the Inter-MAC core functionality, e.g. for packet forwarding, policies are used. These policies are stored in the policy repository in the management plane. The policies are used to configure the behaviour of the Inter-MAC engines. For example the Admission control engine may use different access policies e.g. flows generated by parents will be granted even though bandwidth is already fully booked by flows generated by children. Other policies might influence the path selection procedure by providing higher priority to wired connections. The format of the policies is not yet defined but XML is a good candidate since it provides easy access to both human and machine readers.

II.5.3.2 Management Information Base

The Management Information Base can be divided into two parts: i.e. the Internet MIB and the Agent MIB. The former defines a tree structure which determines which data is located in which sub tree whereas the latter defines which data is represented by a certain agent. In the Inter-MAC context such an agent might be defined per Inter-MAC engine. The management model applied is a client server architecture in which a manager acts as a client and the management agent running on the managed object acts as a server.

II.6 Inter-MAC Interfaces

Inter-MAC will communicate through the use of specific interfaces, with technology-independent adapters when communication with other nodes in the network and when communicating with the various technology-dependent MAC layers. The Inter-MAC is technology independent and controls multiple technologies in Gigabit Networks by means of proper adapters. It also provides services as well as connectivity to all the devices in the house.

II.7 Inter-MAC Frame Format

Since the OMEGA network needs to support different technologies, a new frame header is added to packets in order to be able to address them. When data arrives to the Inter-MAC module, a packet is encapsulated with the Inter-MAC header and then its sent downstream to the corresponding MAC layer. Similarly, in reception a packet is decapsulated once received, revealing the IP packet. This section will present the current Inter-MAC header and a brief explanation of its content.

- Protocol ID: for de-framing & Control Plane frame type identification.
- Sequence Number: identify the frame in case of duplicated frame (flooding, broadcast...).
- Hop Limit: limits the number of times a packet can be forwarded.
- QoS Class: DSCP for prioritisation.
- Broadcasting Flag: to mark IP broadcasts (Destination. *.*.*.255) and Control Plane broadcasts (Destination. 0xffffffffffff).
- Control Plane Flag: to mark frames that are forwarded to the Control Plane instead of the IP layer.
- Probe Flag: Probe frames are marked to enable time-stamp injection (probe frames may have any priority).
- FlowID: identify a flow. It must be unique in the network. The identifier is a hash of some field of the payload (IP, TCP/UDP header) like IP source address, IP destination address, TCP/UDP source ports.
- Inter-MAC Addresses: Addresses of OMEGA Nodes (6 bytes).
- Legacy MAC Addresses: for frames with legacy destination or legacy source.
- The flags: identify the type of the frame: control frame, data frame and the presence of optional fields (payload encryption, legacy address, timestamp).

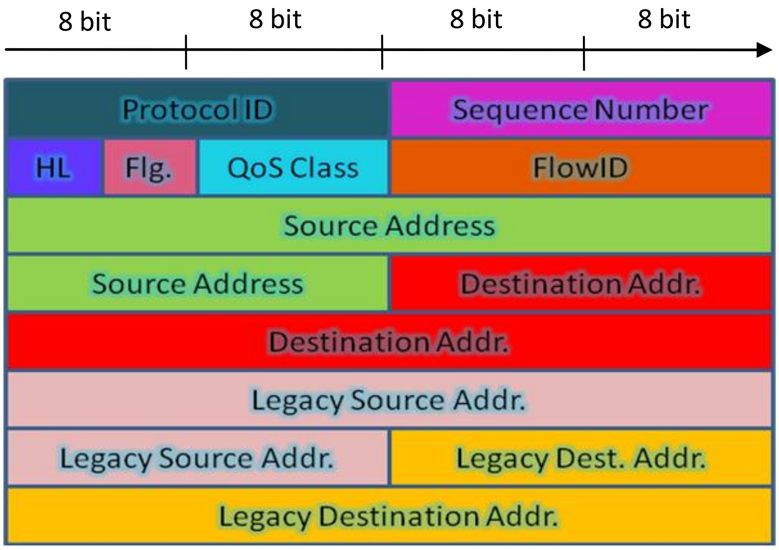
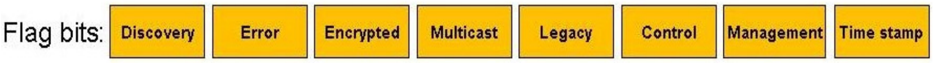


Figure 9 - Inter-MAC header.



Chapter Three: Path Selection in Meshed Networks

Path Selection in OMEGA has the important role of creating, maintaining, updating and releasing paths through the network so that the information can arrive at its destination in the most expedited possible manner while assuring a low delay, low overhead, QoS and the integrity of the data itself. Gigabit networks such as OMEGA will have demanding applications such as High Definition Video Conferencing, On-Demand High Definition Video Streaming, online Video Games and other services yet to imagine. In order to be able to support these and other services, a Path Selection protocol must be as simple and robust as possible while supporting differentiated QoS, Multicast, etc. The word Path Selection usually refers to Routing based on layer 2 addressing (MAC Addresses) so from now on the terms Routing and Path Selection are treated indifferently. This chapter begins by giving an introduction to Meshed Networks, the topology of network that represents the OMEGA network, therefore commenting on the different approaches to Path Selection, exposing some Path Selection protocols with further focusing on *Hybrid Wireless Mesh Protocol* or HWMP, the Path Selection protocol proposed for the OMEGA network.

III.1 Routing in Meshed Networks

III.1.1 Introduction

Mesh networking is a type of networking wherein each node in the network may act as an independent router, regardless of whether it is connected to another network or not. It allows for continuous connections and reconfiguration around broken or blocked paths by "hopping" from node to node until the destination is reached. A mesh network whose nodes are all connected to each other is a fully connected network. Mesh networks differ from other networks in that the component parts can all connect to each other via multiple hops, and they generally are not mobile. Mesh networks are self-healing: the network can still operate when one node breaks down or a connection goes bad. As a result, the network may typically be very reliable, as there is often more than one path between a source and a destination in the network. Although mostly used in wireless scenarios, this concept is also applicable to wired networks and software interaction.

III.1.2 Wireless Mesh Networks

Wireless Mesh Network (WMN) is a promising wireless technology for several emerging and commercially interesting applications, e.g. broadband home networking, community and neighbourhood networks, coordinated network management, intelligent transportation systems. It is gaining significant attention as a possible way for Internet service providers (ISPs) and other end-users to establish robust and reliable wireless broadband service access at a reasonable cost. WMNs consist of mesh routers and mesh clients as shown in Figure 10. In this architecture, while static mesh routers form the wireless backbone, mesh clients access the network through mesh routers as well as directly meshing with each other.

Different from traditional wireless networks, WMN is dynamically self-organised and self-configured. In other words, the nodes in the mesh network automatically establish and maintain network connectivity. This feature brings many advantages for the end-users, such as low up-front cost, easy network maintenance, robustness, and reliable service coverage. In addition, with the use of advanced radio technologies, e.g. multiple radio interfaces and smart antennas, network capacity in WMNs is increased significantly. Moreover, the gateway and bridge functionalities in mesh routers enable the integration of wireless mesh networks with various existing wireless networks, such as wireless sensor networks, wireless-fidelity (Wi-Fi), and WiMAX. Consequently, through an integrated wireless mesh network, the end-users can take the advantage of multiple wireless networks. Some of the benefits and characteristics of wireless mesh networks are highlighted as follows:

- *Increased Reliability:* In WMNs, the wireless mesh routers provide redundant paths between the sender and the receiver of the wireless connection. This eliminates single point failures and potential bottleneck links, resulting in significantly increased communications reliability. Network robustness against potential problems, e.g. node failures, and path failures due to RF interferences or obstacles, can also be ensured by the existence of multiple possible alternative routes. Therefore, by utilising WMN technology, the network can operate reliably over an extended period of time, even in the presence of a network element failure or network congestion.

- *Low Installation Costs:* Recently, the main effort to provide wireless connection to the end-users is through the deployment of 802.11 based Wi-Fi Access Points (APs). To assure almost full coverage in a metro scale area, it is required to deploy a large number of access points because of the limited transmission range of the APs. The drawback of this solution is highly expensive infrastructure costs, since an expensive cabled connection to the wired Internet backbone is necessary for each AP. On the other hand, constructing a wireless mesh network decreases the infrastructure costs, since the mesh network requires only a few points of connection to the wired network. Hence, WMNs can enable rapid implementation and possible modifications of the network at a reasonable cost, which is extremely important in today's competitive market place.

- *Large Coverage Area:* Currently, the data rates of wireless local area networks (WLANs) have been increased, e.g. 802.11n which has a real world performance of 160Mbps or faster, by utilising MIMO (multiple input/multiple output) which uses multiple antennas to send and receive digital data in multiple simultaneous radio streams, thus multiplying total performance. Although the data rates of WLANs are increasing, for a specific transmission power, the coverage and connectivity of WLANs decreases as the end-user becomes further from the access point. On the other hand, multi-hop and multi-channel communications among mesh routers and long transmission range of WiMAX towers deployed in WMNs can enable long distance communication without any significant performance degradation.

- *Automatic Network Connectivity:* Wireless mesh networks are dynamically self-organised and self-configured. In other words, the mesh clients and routers automatically establish and maintain network connectivity, which enables seam- less multi-hop interconnection service. For example, when new nodes are added into the network, these nodes utilise their meshing functionalities to automatically discover all possible routers and determine the optimal paths to the wired Internet. Furthermore, the existing mesh routers reorganise the network considering the newly available routes and hence, the network can be easily expanded.

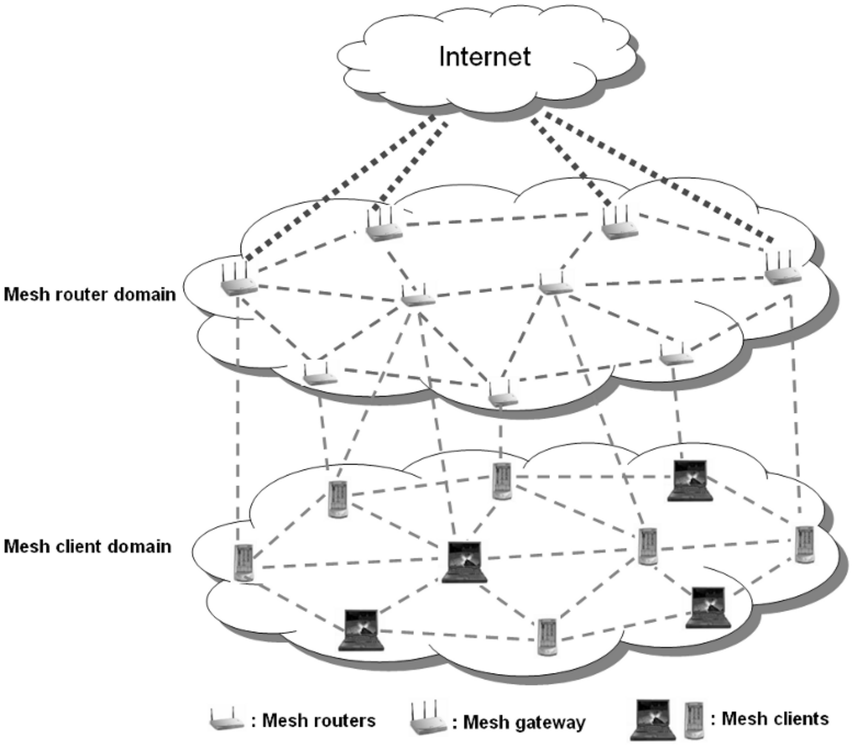


Figure 10 - Wireless Mesh Network Architecture.

III.1.3 Challenges

- *Interoperability and Integration of Heterogeneous Networks:* Existing networking technologies have limited capabilities of integrating different wireless networks. Thus, to increase the performance of WMNs and to provide the interoperability between the products from different manufacturers, the integration capabilities of multiple wireless interfaces and the corresponding gateway/bridge functions of network routers should be improved.

- *Scalability:* The deployed mesh network must be able to deal with large network topologies without increasing the number of network operations exponentially. In addition, the network performance should not degrade as the number of hops between the sender and the receiver increases. To provide the scalability in WMNs, there is a need for scalable MAC, routing and transport layer protocols with minimum overhead.

- *Dynamic Network Connectivity and Self-Configuration:* In WMNs, to eliminate the single point failures and potential bottleneck links, the wireless back-bone needs to provide redundant paths between the sender and the receiver, i.e. mesh connectivity. However, the topology and connectivity of the network can vary frequently because of the route failures and energy depletions. Therefore, to take all the advantages of autonomous mesh connectivity, efficient network self-configuration, topology control and power management algorithms are required.

- *Mobility Support:* To support mobile mesh clients in WMNs, it is necessary to design advanced physical layer and networking techniques, which adapt to the fast fading conditions commonly associated with the mobile users. In addition to these advanced techniques, low latency handover and location management algorithms are also required to improve the quality of service during mobility.

There has been a lot of research focused on finding the most efficient path selection algorithm that creates an optimal path for a packet on a given meshed network. Important aspects taken into consideration are the initial delay required to update the configuration of the network and the overall overhead generated to achieve this. Some of the best known algorithms are: Ad-Hoc On Demand Distance Vector (AODV), Optimised Link State Routing (OLSR) and Hybrid Wireless Mesh Protocol (HWMP). HWMP is the proposed Path Selection protocol for the IEEE 802.11s standard (IEEE 802.11s extends the IEEE 802.11 MAC standard by defining an architecture and protocol that support both broadcast/multicast and unicast delivery using "radio-aware metrics over self-configuring multi-hop topologies").These algorithms are discussed starting from section III.3.

III.2 Path Selection Approaches

III.2.1 Centralised Approach

In the centralised approach, each network device periodically reports its link status information to a central instance. In most cases, this is the gateway due to its special role inside the network. In the end all information about the network devices is collected in the central instance. Therefore it is aware of the whole topology of the network.

As a consequence, the central instance is easily able to calculate configurations of the network devices. For instance, it computes the path from a network device to the desired destination. Afterwards the calculated path must be propagated back to the affected devices to configure them appropriately.

For illustration: a network device recognises a link break and reports this to the central instance. The central instance reacts with a recalculation of alternative paths. Afterwards the central instance sends the calculated configurations to the relevant network devices to rearrange the affected flow over alternative paths.

Advantages:

- ➡ The central instance has a global view of the network topology. This means that the central node knows all configurations and resources in the network, so that it can make a decision on path selection and Admission control that yields a global optimum.

Disadvantages:

- A central instance is a single point of failure. In the case that this device fails, the network devices are not able to handle any changes in the network. Therefore an always-on central instance is needed.
- Continuous connectivity of all devices to the central instance is needed. If some nodes of the network get disconnected from the central instance and it occurs a failure in the network, it could happen that they cannot communicate with each other any longer.
- A continuous or event driven information flooding is needed towards the central instance. This increases the latency of path selection setup and the signalling overhead significantly.
- A further disadvantage is low scalability in terms of network dimension. This means that the network must be "reasonably" small because of the single point of failure bottleneck.

III.2.2 Distributed Approach

In the distributed approach, each network device is able to calculate paths or parts of paths according to a well-specified path selection algorithm by themselves. For instance, path selection information is calculated at each device (e.g. distance vector routing or link state routing). Depending on the path selection algorithm just a "local view" or the complete network topology is required.

Advantages:

- ➡ One of the main benefits of this approach is that no single point of failure exists. So the network is able to work even without a central instance or gateway. This makes the system very reliable and robust.
- ➡ Communication between devices does not depend on existing connectivity to a central instance. For instance, a group of nodes that is temporarily disconnected from the central instance can still communicate with each other in a multi-hop fashion.
- ➡ This approach is highly scalable in case of large and dynamic networks.

Disadvantages:

- ▸ It may happen that only a sub-optimal "global" solution is found.

The distributed approach is preferred because of its robustness. Distributed Path Selection algorithms can be subdivided into three categories: Proactive, Reactive (or On-Demand) and Hybrid.

III.2.2.1 <u>Distributed Proactive Approach</u>

The path selection information to all nodes is maintained on all nodes in the network at all times. This is achieved by regular status announcements and event-driven path selection information distribution.

Advantages:

- ➡ Lower path setup latency.
- ➡ Path is immediately available.

Disadvantages:

- ▸ Path selection overhead even if no communication is needed (periodic distribution of routing information, usually by flooding).
- ▸ Stale routing information in highly dynamic topologies.
- ▸ Scalability problems with larger networks.

An example of this type of approach is Optimised Link State Routing (OLSR) protocol. This protocol will be explained ahead in this chapter.

III.2.2.2 <u>Distributed Reactive Approach</u>

In this approach (also called On-Demand Routing), the path selection information is only maintained if needed. This means the path calculation is executed on-demand and only on the nodes which are needed. The path selection information is maintained only for the time when it is needed.

Advantages:

- ➡ No overhead for route maintenance, only if a route is actually needed.
- ➡ Faster reaction on mobility or topology changes.

Disadvantages:

- ▸ Larger path setup latency.
- ▸ Delay at the start of communication.
- ▸ Path discovery requires flooding.

An example of this type of approach is Ad-Hoc On-Demand Distance Vector (AODV) protocol. This protocol will be explained ahead in this chapter.

III.2.2.3 Distributed Hybrid Approach

The routing is proactive for some destinations (e.g. home gateway, neighbourhood) and reactive for other destinations (e.g. long distances).

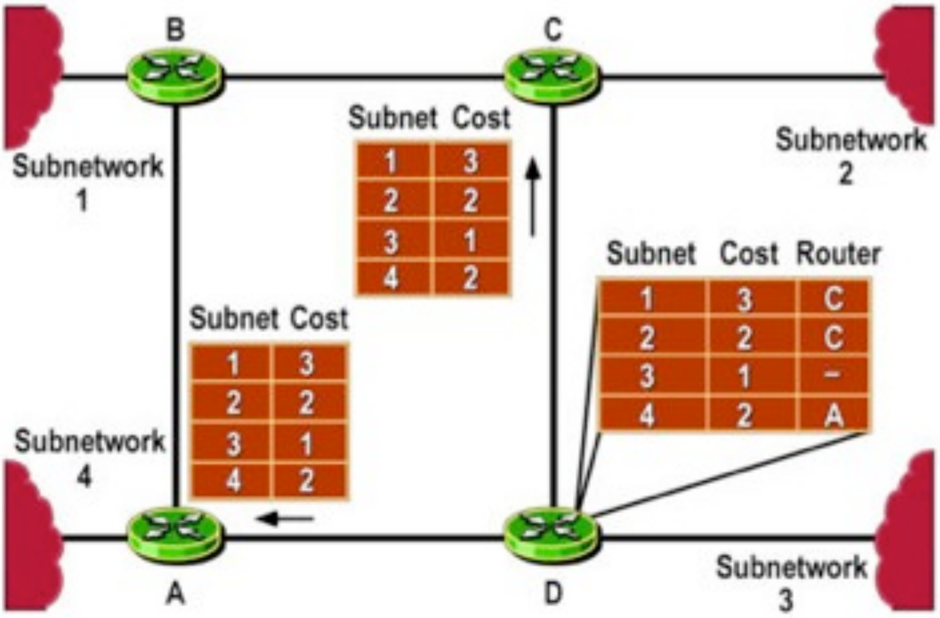
Advantages:
- Reduces impact of disadvantages of proactive and reactive routing protocols.
- No route setup latency for short distance connections or gateway connections.
- Lower routing overhead due to reactive routing for further away destinations.
- Good scalability for larger networks.

Disadvantages:
- More complex.

An example of this type of approach is Hybrid Wireless Mesh Protocol (HWMP). This protocol, proposed for the IEEE 802.11s standard, will be the main point of focus of this thesis and will be explained in detail ahead in this chapter.

III.2.3 Distance Vector Protocol

A distance vector routing protocol can best be understood by recalling the meaning of the word vector. A vector is a number with two components: magnitude and direction. In a network, we might say it has cost and direction or distance and direction.



Subnet	Cost
1	3
2	2
3	1
4	2

Subnet	Cost	Router
1	3	C
2	2	C
3	1	–
4	2	A

Subnet	Cost
1	3
2	2
3	1
4	2

Figure 11 - Distance Vector Routing Protocol

In a distance vector protocol, neighbouring routers (i.e., routers attached to the same subnetwork) exchange routing vectors. Each router pulls from its routing table a list of all known subnetworks, and some metric relating the goodness or cost of the path to that subnetwork. This information is transmitted to all neighbours. Neighbours are any systems connected to the same subnetwork.

Upon receiving an update from a neighbour, the routing protocol begins the process of updating the local routing table. For each subnetwork listed in the update, the routing protocol extracts the cost information from the update and adds to it the cost of the link from the neighbour that sent the update to the receiving router. It then examines the current routing table to determine if the subnetwork is listed and if it is, the cost to reach that network using the current route. If the subnetwork is not listed in the table, the routing protocol adds the new subnetwork including the port on which the update was received and the address of the router that sent the update. This router is the best known path to the new subnetwork.

If the subnetwork already appears in the table, the routing protocol compares the current cost to the cost it calculated via the updating router. If the cost listed in the routing table is higher than or equal to the newly calculated cost, the routing protocol does not alter the table. However, if the router that transmitted the update is reporting a lower cost route, the routing protocol updates the routing table entry for that subnetwork with the new router's address, the port on which the update was received, and the newly calculated cost. The router that transmitted the update now represents the best known route to the indicated subnetwork.

III.2.4 Link State Protocol

The link state protocol is performed by every switching node in the network (i.e. nodes that are prepared to forward packets; in the Internet, these are called routers). The basic concept of link state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical hop from it to every possible destination in the network. The collection of best next hops will then form the node's routing table. This contrasts with distance-vector routing protocols, which works by having each node share its routing table with its neighbours. In a link state protocol the only information passed between nodes is connectivity related. Link state algorithms are sometimes characterised by the phrase "Each router tells the world about its neighbours".

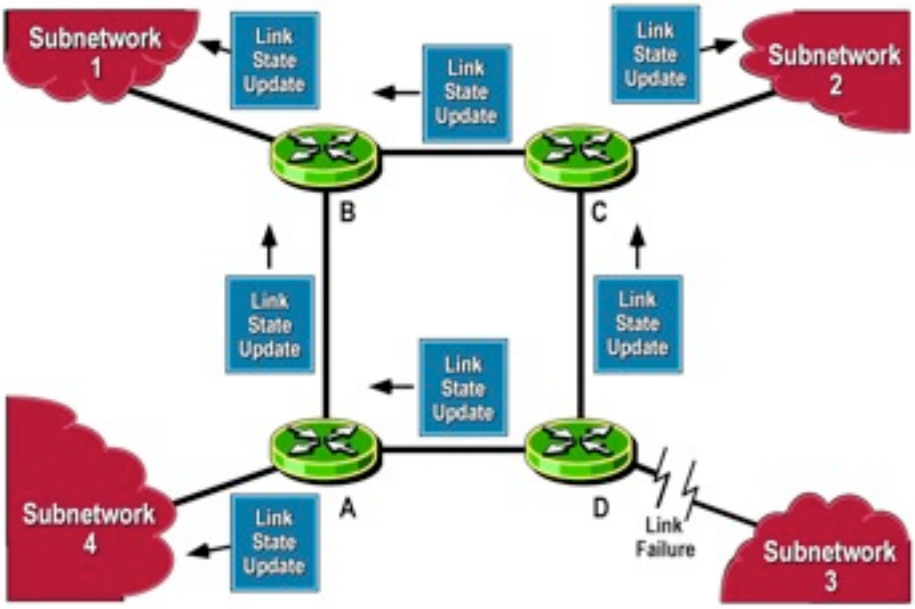


Figure 12 - Link State Routing Protocol

III.2.4.1 Distributing the maps

This description covers only the simplest configuration; i.e. one with no areas, so that all nodes do have a map of the entire network. The hierarchical case is somewhat more complex; see the various protocol specifications.

As previously mentioned, the first main stage in the link state algorithm is to give a map of the network to every node. This is done with several simple subsidiary steps.

Determining the neighbours of each node

First, each node needs to determine what other ports it is connected to, over fully-working links; it does this using a simple reachability protocol which it runs separately with each of its directly-connected neighbours.

Distributing the information for the map

Next, each node periodically and in case of connectivity changes makes up a short message, the link state advertisement, which:

- Identifies the node which is producing it.
- Identifies all the other nodes to which it is directly connected.
- Includes a sequence number, which increases every time the source node makes up a new version of the message.

This message is then flooded throughout the network. As a necessary precursor, each node in the network remembers, for every other node in the network, the sequence number of the last link state message which it received from that node. With that in hand, the method used is simple. Starting with the node which originally produced the message, it sends a copy to all of its neighbours. When a link state advertisement is received at a node, the node looks up the sequence number it has stored for the source of that link state message. If this message is newer (i.e. has a higher sequence number), it is saved, and a copy is sent in turn to each of that node's neighbours.

This procedure rapidly gets a copy of the latest version of each node's link state advertisement to every node in the network. Networks running link state algorithms can also be segmented into hierarchies which limit the scope of route changes. These features mean that link state algorithms scale better to larger networks.

Creating the map

Finally, with the complete set of link state advertisements (one from each node in the network) in hand, it is obviously easy to produce the graph for the map of the network. The algorithm simply iterates over the collection of link state advertisements; for each one, it makes links on the map of the network, from the node which sent that message, to all the nodes which that message indicates are neighbours of the sending node.

No link is considered to have been correctly reported unless the two ends agree; i.e. if one node reports that it is connected to another, but the other node does not report that it is connected to the first, there is a problem, and the link is not included on the map.

Notes about this stage

The link state message giving information about the neighbours is recomputed, and then flooded throughout the network, whenever there is a change in the connectivity between the node and its neighbours, e.g. when a link fails. Any such change will be detected by the reachability protocol which each node runs with its neighbours.

III.2.4.2 Calculation the routing table

Calculating the shortest paths

Each node independently runs an algorithm over the map to determine the shortest path from itself to every other node in the network; generally some variant of Dijkstra's algorithm is used.

Basically, a node maintains two data structures: a tree containing nodes which are "done", and a list of candidates. The algorithm starts with both structures empty; it then adds to the first one the node itself. The algorithm then repetitively:

- Adds to the second (candidate) list all nodes which are connected to the node just added to the tree (excepting of course any nodes which are already in either the tree or the candidate list).
- Of the nodes in the candidate list, moves to the tree (attaching it to the appropriate neighbour node already there) the one which is the closest to any of the nodes already in the tree.
- Repeat as long as there are any nodes left in the candidate list. (When there are none, all the nodes in the network will have been added to the tree.)

This procedure ends with the tree containing all the nodes in the network, with the node on which the algorithm is running as the root of the tree. The shortest path from that node to any other node is indicated by the list of nodes one traverses to get from the root of the tree, to the desired node in the tree.

Filling the routing table

With the shortest paths in hand, filling in the routing table is trivial. For any given destination node, the best next hop for that destination is the node which is the first step from the root node, down the branch in the shortest-path tree which leads toward the desired destination node.

To create the routing table, it is only necessary to walk the tree, remembering the identity of the node at the head of each branch, and filling in the routing table entry for each node one comes across with that identity.

Optimisations to the algorithm

The algorithm described above was made as simple as possible, to aid in ease of understanding. In practice, there are a number of optimisations which are used. Most importantly, whenever a change in the connectivity map happens, it is necessary to recompute the shortest-path tree, and then recreate the routing table.

III.2.5 Distance Vector vs. Link State Routing Protocol

There are two major differences between Distance Vector routing protocols and Link State routing protocols. Distance Vector exchanges the routing updates periodically whether the topology is change or not, this will maximise the convergence time which increases the chance of routing loops while the Link State routing protocols send triggered change based updates when there is a topology change. After initial flood, pass small event based triggered link state updates to all other routers. This will minimise the convergence time that's why there is no chance of routing loops. Secondly, the Distance Vector routing protocols rely on the information from their directly connected neighbours in order to calculate and accumulate route information. Distance Vector routing protocols require very little overhead as compared to Link State routing protocols as measured by memory and processor power while the Link State routing protocols do not rely solely on the information from the neighbours or adjacent router in order to calculate route information. Instead, Link State routing protocols have a system of databases that they use in order to calculate the best route to destinations in the network. An extra feature of Link State routing protocol is that they can detect media types along with other factors. This could increase the overhead as compare to Distance Vector routing protocols in order to measure by processor power and memory. The other differences of both types of routing protocols are as follows:

Distance Vector:

- ➡ Distance Vector routing protocols are based on Bellman-Ford algorithms.
- ➡ Distance Vector routing protocols uses hop count and composite metric.
- ➡ Distance Vector routing protocols support discontiguous subnets.

Link State:

- ‣ Link State routing protocols are based on Dijkstra algorithms.
- ‣ Link State routing protocols are very much scalable supports infinite hops.

III.3 Ad-Hoc On Demand Distance Vector (AODV)

The Ad hoc On-Demand Distance Vector (AODV) algorithm enables dynamic, self-starting, multihop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. AODV allows mobile nodes to respond to link breakages and changes in network topology in a timely manner.

The operation of AODV is loop-free, and by avoiding the Bellman-Ford “counting to infinity” problem offers quick convergence when the ad hoc network topology changes (typically, when a node moves in the network). When links break, AODV causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link.

One distinguishing feature of AODV is its use of a destination sequence number for each route entry. The destination sequence number is created by the destination to be included along with any route information it sends to requesting nodes. Using destination sequence numbers ensures loop freedom and is simple to program. Given the choice between two routes to a destination, a requesting node is required to select the one with the greatest sequence number.

III.3.1 Overview of AODV

AODV is a reactive protocol that determines routes solely on-demand. It is based on the distance vector technology. Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs) are the message types defined by AODV. These message types are received via UDP, and normal IP header processing applies. So, for instance, the requesting node is expected to use its IP address as the Originator IP address for the messages. For broadcast messages, the IP limited broadcast address (255.255.255.255) is used. This means that such messages are not blindly forwarded. However, AODV operation does require certain messages (e.g. RREQ) to be disseminated widely, perhaps throughout the ad hoc network. The range of dissemination of such RREQs is indicated by the TTL in the IP header. Fragmentation is typically not required.

As long as the endpoints of a communication connection have valid routes to each other, AODV does not play any role. When a route to a new destination is needed, the node broadcasts a RREQ to find a route to the destination. A route can be determined when the RREQ reaches either the destination itself, or an intermediate node with a 'fresh enough' route to the destination. A “fresh enough" route is a valid route entry for the destination whose associated sequence number is at least as great as that contained in the RREQ. The route is made available by sending a unicast RREP control packet back to the origination of the RREQ. Each node receiving the request caches a route back to the originator of the request, so that the RREP can be unicast from the destination along a path to that originator, or likewise from any intermediate node that is able to satisfy the request.

Nodes monitor the link status of next hops in active routes. When a link break in an active route is detected, a RERR message is used to notify other nodes that the loss of that link has occurred. The RERR message indicates those destinations (possibly subnets) which are no longer reachable by way of the broken link. In order to enable this reporting mechanism, each node keeps a "precursor list", containing the IP address for each its neighbours that are likely to use it as a next hop towards each destination. The information in the precursor lists is most easily acquired during the processing for generation of a RREP message, which by definition has to be sent to a node in a precursor list. If the RREP has a nonzero prefix length, then the originator of the RREQ which solicited the RREP information is included among the precursors for the subnet route (not specifically for the particular destination).

A RREQ may also be received for a multicast IP address. For example, the originator of such a RREQ for a multicast IP address may have to follow special rules. However, it is important to enable correct multicast operation by intermediate nodes that are not enabled as originating or destination nodes for IP multicast addresses, and likewise are not equipped for any special multicast protocol processing. For such multicast-unaware nodes, processing for a multicast IP address as a destination IP address MUST be carried out in the same way as for any other destination IP address.

AODV is a routing protocol, and it deals with route table management. Route table information must be kept even for short-lived routes, such as are created to temporarily store reverse paths towards nodes originating RREQs. AODV uses the following fields with each route table entry:

- Destination IP Address
- Destination Sequence Number
- Valid Destination Sequence Number flag
- Other state and routing flags (e.g. valid, invalid, repairable, being repaired)
- Network Interface
- Hop Count (number of hops needed to reach destination)
- Next Hop
- List of Precursors
- Lifetime (expiration or deletion time of the route)

Managing the sequence number is crucial to avoiding routing loops, even when links break and a node is no longer reachable to supply its own information about its sequence number. A destination becomes unreachable when a link breaks or is deactivated. When these conditions occur, the route is invalidated by operations involving the sequence number and marking the route table entry state as invalid.

III.3.2 Maintaining Sequence Numbers

Every route table entry at every node MUST include the latest information available about the sequence number for the IP address of the destination node for which the route table entry is maintained. This sequence number is called the "destination sequence number". It is updated whenever a node receives new (i.e., not stale) information about the sequence number from RREQ, RREP, or RERR messages that may be received related to that destination. AODV depends on each node in the network to own and maintain its destination sequence number to guarantee the loop-freedom of all routes towards that node. A destination node increments its own sequence number in two circumstances:

- ➡ Immediately before a node originates a route discovery, it must increment its own sequence number. This prevents conflicts with previously established reverse routes towards the originator of a RREQ.
- ➡ Immediately before a destination node originates a RREP in response to a RREQ, it MUST update its own sequence number to the maximum of its current sequence number and the destination sequence number in the RREQ packet.

The only other circumstance in which a node may change the destination sequence number in one of its route table entries is in response to a lost or expired link to the next hop towards that destination. The node determines which destinations use a particular next hop by consulting its routing table. In this case, for each destination that uses the next hop, the node increments the sequence number and marks the route as invalid.

Whenever any fresh enough (i.e., containing a sequence number at least equal to the recorded sequence number) routing information for an affected destination is received by a node that has marked that route table entry as invalid, the node should update its route table information according to the information contained in the update.

A node may change the sequence number in the routing table entry of a destination only if one of the following applies:

- It is itself the destination node, and offers a new route to itself.
- It receives an AODV message with new information about the sequence number for a destination node.
- The path towards the destination node expires or breaks.

III.3.3 Generating Route Request (RREQ)

A node disseminates a RREQ when it determines that it needs a route to a destination and does not have one available. This can happen if the destination is previously unknown to the node, or if a previously valid route to the destination expires or is marked as invalid. The Destination Sequence Number field in the RREQ message is the last known destination sequence number for this destination and is copied from the Destination Sequence Number field in the routing table. If no sequence number is known, the unknown sequence number flag MUST be set. The Originator Sequence Number in the RREQ message is the node's own sequence number, which is incremented prior to insertion in a RREQ. The RREQ ID field is incremented by one from the last RREQ ID used by the current node. Each node maintains only one RREQ ID. The Hop Count field is set to zero.

Before broadcasting the RREQ, the originating node buffers the RREQ ID and the Originator IP address (its own address) of the RREQ for a certain T time. In this way, when the node receives the packet again from its neighbours, it will not reprocess and re-forward the packet.

An originating node often expects to have bidirectional communications with a destination node. In such cases, it is not sufficient for the originating node to have a route to the destination node; the destination must also have a route back to the originating node. In order for this to happen as efficiently as possible, any generation of a RREP by an intermediate node for delivery to the originating node should be accompanied by some action that notifies the destination about a route back to the originating node. Figure 13 shows the broadcasting of the RREQ control packet.

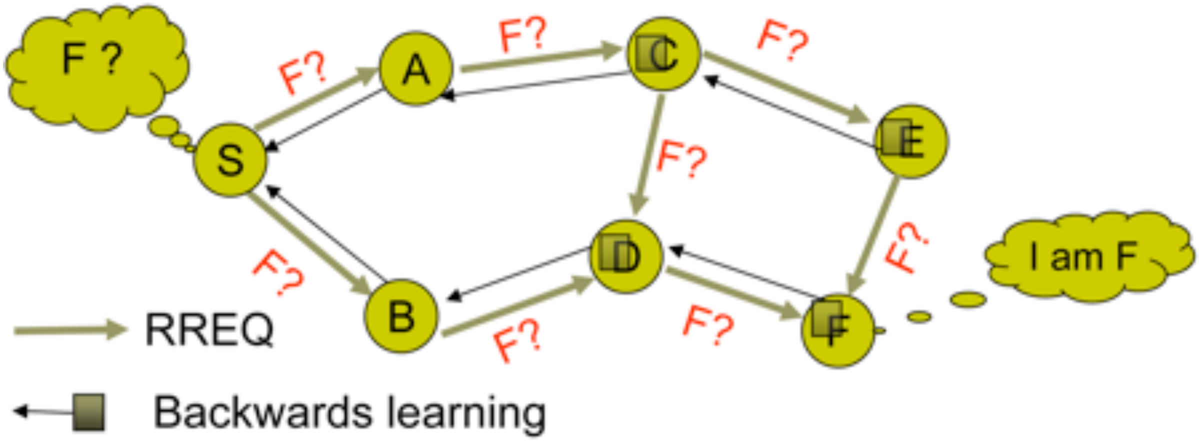


Figure 13 - Generating RREQ in AODV.

III.3.4 Generating Route Replies (RREP)

A node generates a RREP if either:

- It is itself the destination.
- It has an active route to the destination, the destination sequence number in the node's existing route table entry for the destination is valid and greater than or equal to the Destination Sequence Number of the RREQ (comparison using signed 32-bit arithmetic), and the "destination only" flag is NOT set.

When generating a RREP message, a node copies the Destination IP Address and the Originator Sequence Number from the RREQ message into the corresponding fields in the RREP message. Processing is slightly different, depending on whether the node is itself the requested destination, or instead if it is an intermediate node with an fresh enough route to the destination.

Once created, the RREP is unicast to the next hop toward the originator of the RREQ, as indicated by the route table entry for that originator. As the RREP is forwarded back towards the node which originated the RREQ message, the Hop Count field is incremented by one at each hop. Thus, when the RREP reaches the originator, the Hop Count represents the distance, in hops, of the destination from the originator. Figure 14 shows the generation of the RREP in Unicast back to the source.

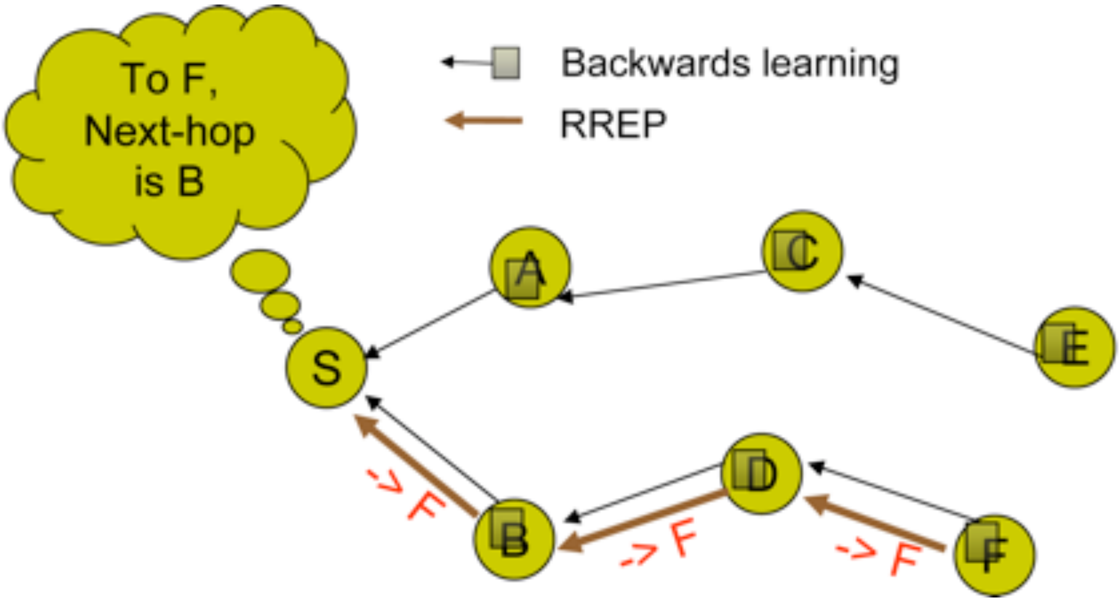


Figure 14 - Generating RREP in AODV.

III.3.5 Link Breakage

When the link breakage occurs then the host can try to locally repair the link if the destination is no further than specified amount of hops. In order to repair the link the host increase the destination sequence number and broadcasts the RREQ message to the host. The TTL for the IP header must be calculated, so that locally repair process would not spread throughout the network. The host waits for the RREP messages to its RREQ message for specified amount of time. If the RREP message is not received, then it changes the routing table status for the entry to invalid. If host receives the RREP message then the hop count metric is compared. If the hop metric from the message is greater than the previous one then the RERR with the N field set up is broadcasted. The N field in the RERR message indicates that the host has locally repaired the link and the entry in the table should not be deleted. The received RREP message is handled as original RREP message. The repairing of the link before the data is sent to unavailable host is a proactive repairing. Proactive repairing can be inefficient because the risk of repairing the routes that are not used anymore. So the proactive repairing can be used basing on the local traffic and the workload of the network.

III.3.6 Advantages and Disadvantages

The main advantage of this protocol is that routes are established on-demand and destination sequence numbers are used to find the latest route to the destination. The connection setup delay is lower. One of the disadvantages of this protocol is that intermediate nodes can lead to inconsistent routes if the source sequence number is very old and the intermediate nodes have a higher but not the latest destination sequence number, thereby having stale entries. Also multiple RouteReply packets in response to a single RouteRequest packet can lead to heavy control overhead. Another disadvantage of AODV is that the periodic beaconing leads to unnecessary bandwidth consumption.

III.4 Optimised Link State Routing (OLSR)

The Optimised Link State Routing Protocol (OLSR) is developed for mobile ad hoc networks. It operates as a table driven, proactive protocol, i.e., exchanges topology information with other nodes of the network regularly. Each node selects a set of its neighbour nodes as "multipoint relays" (MPR).

In OLSR, only nodes, selected as such MPRs, are responsible for forwarding control traffic, intended for diffusion into the entire network. MPRs provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required.

Nodes, selected as MPRs, also have a special responsibility when declaring link state information in the network. Indeed, the only requirement for OLSR to provide shortest path routes to all destinations is that MPR nodes declare link-state information for their MPR selectors. Additional available link-state information may be utilised, e.g. for redundancy.

Nodes which have been selected as multipoint relays by some neighbour node(s) announce this information periodically in their control messages. Thereby a node announces to the network, that it has reachability to the nodes which have selected it as an MPR. In route calculation, the MPRs are used to form the route from a given node to any destination in the network. Furthermore, the protocol uses the MPRs to facilitate efficient flooding of control messages in the network.

A node selects MPRs from among its one hop neighbours with "symmetric", i.e., bi-directional, linkages. Therefore, selecting the route through MPRs automatically avoids the problems associated with data packet transfer over uni-directional links (such as the problem of not getting link-layer acknowledgments for data packets at each hop, for link-layers employing this technique for unicast traffic).

OLSR is developed to work independently from other protocols. Likewise, OLSR makes no assumptions about the underlying link-layer.

III.4.1 Overview of OLSR

OLSR is a table-driven, proactive routing protocol for mobile ad hoc networks. The protocol inherits the stability of a link state algorithm and has the advantage of having routes immediately available when needed due to its proactive nature. OLSR is an optimisation over the classical link state protocol, tailored for mobile ad hoc networks.

OLSR minimises the overhead from flooding of control traffic by using only selected nodes, called MPRs, to retransmit control messages. This technique significantly reduces the number of retransmissions required to flood a message to all nodes in the network. Secondly, OLSR requires only partial link state to be flooded in order to provide shortest path routes. The minimal set of link state information required is, that all nodes, selected as MPRs, must declare the links to their MPR selectors. Additional topological information, if present, MAY be utilised e.g. for redundancy purposes.

OLSR may optimise the reactivity to topological changes by reducing the maximum time interval for periodic control message transmission. Furthermore, as OLSR continuously maintains routes to all destinations in the network, the protocol is beneficial for traffic patterns where a large subset of nodes are communicating with another large subset of nodes, and where the [source, destination] pairs are changing over time. The protocol is particularly suited for large and dense networks, as the optimisation done using MPRs works well in this context. The larger and more dense a network, the more optimisation can be achieved as compared to the classic link state algorithm.

OLSR is designed to work in a completely distributed manner and does not depend on any central entity. The protocol does not require reliable transmission of control messages: each node sends control messages periodically, and can therefore sustain a reasonable loss of some such messages. Such losses occur frequently in radio networks due to collisions or other transmission problems.

Also, OLSR does not require sequenced delivery of messages. Each control message contains a sequence number which is incremented for each message. Thus the recipient of a control message can, if required, easily identify which information is more recent - even if messages have been re-ordered while in transmission.

Furthermore, OLSR provides support for protocol extensions such as sleep mode operation, multicast routing etc. Such extensions may be introduced as additions to the protocol without breaking backwards compatibility with earlier versions. OLSR does not require any changes to the format of IP packets. Thus any existing IP stack can be used as is: the protocol only interacts with routing table management.

III.4.2 Control Messages

OLSR uses three kinds of control messages: HELLO, Topology Information (TC), and Multiple Interface Declaration (MID).

A Hello message is sent periodically to all of a node's neighbours. Hello messages contain information about a node's neighbours, the nodes it has chosen as MPRs (i.e., the MPRSelector set), and a list of list of neighbours whom bidirectional links have not yet been confirmed.

Every node periodically floods, using the multipoint relying mechanism, the network with a TC message. This message contains the node's MPRSelector set. A MID message is used for announcing that a node is running OLSR on more than one interface. The MID message is flooded throughout the network by the MPRs.

III.4.3 Multipoint Relays

The idea of multipoint relays is to minimise the overhead of flooding messages in the network by reducing redundant retransmissions in the same region. Each node in the network selects a set of nodes in its symmetric 1-hop neighbourhood which may retransmit its messages. This set of selected neighbour nodes is called the "Multipoint Relay" (MPR) set of that node. The neighbours of node N which are not in its MPR set, receive and process broadcast messages but do not retransmit broadcast messages received from node N.

Each node selects its MPR set from among its 1-hop symmetric neighbours. This set is selected such that it covers (in terms of radio range) all symmetric strict 2-hop nodes. The MPR set of N, denoted as MPR(N), is then an arbitrary subset of the symmetric 1-hop neighbourhood of N which satisfies the following condition: every node in the symmetric strict 2-hop neighbourhood of N must have a symmetric link towards MPR(N). The smaller a MPR set, the less control traffic overhead results from the routing protocol.

Each node maintains information about the set of neighbours that have selected it as MPR. This set is called the "Multipoint Relay Selector set" (MPR selector set) of a node. A node obtains this information from periodic HELLO messages received from the neighbours.

A broadcast message, intended to be diffused in the whole network, coming from any of the MPR selectors of node N is assumed to be retransmitted by node N, if N has not received it yet. This set can change over time (i.e., when a node selects another MPR-set) and is indicated by the selector nodes in their HELLO messages.

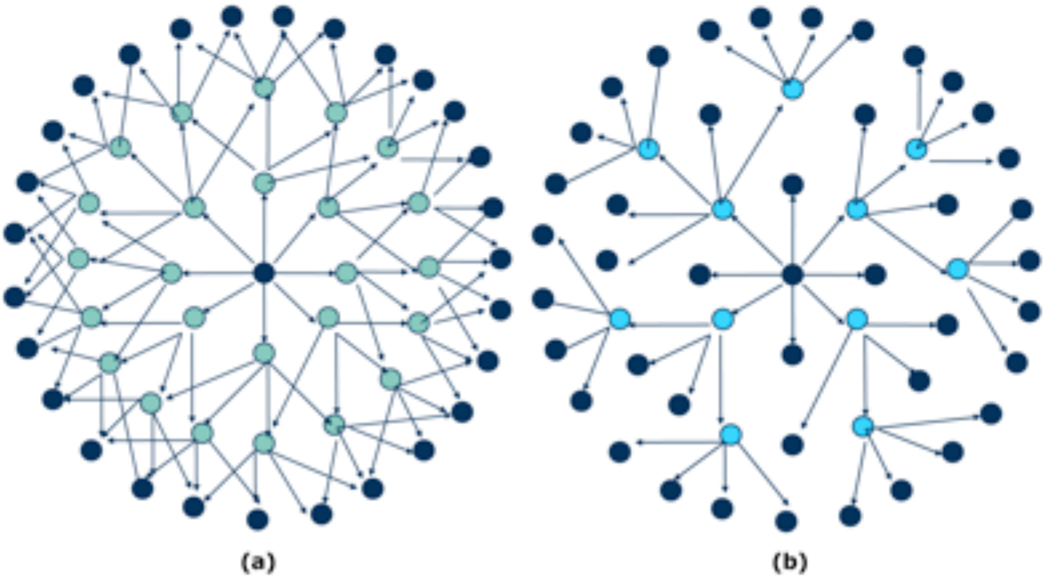


Figure 15 - Illustrates: (a) the situation where all neighbours retransmit a broadcast, (b) where only the MPRs of a node retransmit the broadcast.

III.4.4 Neighbour Discovery

As links in a ad-hoc network can be either unidirectional or bidirectional, a protocol for determining the link status is needed. In OLSR, HELLO messages serve, among others, this purpose. HELLO messages are broadcasted periodically for neighbour sensing. When a node receives a HELLO message in which its address is found, it registers the link to the source node as symmetric.

As an example of how this protocol works, consider two nodes A and B which not yet have established links with each other. First, A broadcasts an empty HELLO message. When B receives this message and does not find its own address in it, it registers in the routing table that the link to A is asymmetric. Then B broadcasts a HELLO message declaring A as an asymmetric neighbour. Upon receiving this message and finding its own address in it, A registers the link to B as symmetric. A then broadcasts a HELLO message declaring B as a symmetric neighbour, and B registers A as a symmetric neighbour upon reception of this message.

Upon receiving a HELLO message in which the node's address is not contained, the node registers in the routing table that the link to the source node is asymmetric. The node then sends a HELLO message containing the source node's address, and when the source node receives this message and find its own address in it, it registers

III.4.5 Topology Information

Information about the network topology is extracted from topology control (TC) packets. These packets contain the MPRSelector set of a node, and are broadcasted by every node in the network, both periodically and when changes in the MPRSelector set is detected. The packets are flooded in the network using the multipoint relaying mechanism. Every node in the network receives such TC packets, from which they extract information to build a topology table.

III.4.6 Route Calculation

The shortest path algorithm is used for route calculations, which is initiated when a change is detected in either of the following: the link set, the neighbour set, the two-hop neighbour set, the topology set, or the Multiple Interface Association Information Base.

To calculate the routing table, information is taken from the neighbour set and the topology set. The calculation is an iterative process, in which route entries are added starting from one-hop neighbours, increasing the hop count each time through.

III.4.7 Advantages and Disadvantages

OLSR is, like AODV, a distributed routing protocol, meaning that it does not need central administrative system to handle its routing process. The proactive characteristic of the protocol provides that the protocol has all the routing information to all participated hosts in the network. However, as a drawback OLSR protocol needs that each host periodic sends the updated topology information throughout the entire network, this increase the protocols bandwidth usage. But the flooding is minimised by the MPRs, which are only allowed to forward the topological messages. The reactiveness to the topological changes can be adjusted by changing the time interval for broadcasting the Hello messages. It increases the protocols suitability for ad-hoc network with the rapid changes of the source and destinations pairs. Further, the OLSR protocol does not require that the link is reliable for the control messages, since the messages are sent periodically and the delivery does not have to be sequential. Due to the OLSR routing protocol simplicity in using interfaces, it is easy to integrate the routing protocol in the existing operating systems, without changing the format of the header of the IP messages. The protocol only interacts with the host's Routing Table.

Another drawback of the OLSR protocol is that by being a proactive protocol, uses power and network resources in order to propagate data about possibly unused routes. While this is not a problem for wired access points, and laptops, it makes OLSR unsuitable for "green" networks which require a high grade in power-efficiency. Furthermore, being a link-state protocol, it requires a reasonably large amount of bandwidth and CPU power to compute optimal paths in the network, further influencing energy consumption.

III.5 Hybrid Wireless Mesh Protocol (HWMP)

III.5.1 Overview

The Hybrid Wireless Mesh Protocol (HWMP) is a mesh path selection protocol that combines the flexibility of reactive path selection with proactive topology tree extensions. The combination of reactive and proactive elements of HWMP enables efficient path selection in a wide variety of mesh networks (with or without access to the infrastructure).

HWMP uses a common set of protocol primitives, generation and processing rules inspired by Ad Hoc On Demand Distance Vector (AODV) protocol adapted for MAC address-based path selection and link metric awareness.

HWMP supports two modes of operation depending on the configuration. These modes provide different levels of functionality:

- *Reactive mode:* The functionality of this mode is always available. It allows mesh STAs to communicate using peer-to-peer paths. The mode is used in situations where there is no root mesh STA configured. It is also used if there is a root mesh STA configured and an reactive path can provide a better path to a given destination in the mesh.

- *Proactive tree building mode:* In this mode, additional proactive tree building functionality is added to the reactive mode. This can be performed by configuring a mesh STA as root mesh STA using either the proactive PREQ or RANN mechanism.

These modes are not exclusive: reactive and proactive modes are used concurrently, because the proactive modes are extensions of the reactive mode. One example of concurrent usage of reactive and proactive mode is for two mesh STAs that are part of the same mesh BSS to begin communicating using the proactively built tree but subsequently to perform an reactive discovery for a direct path between each other.

The use of an Hybrid protocol on the OMEGA network allows communication to begin immediately thanks to the use of the proactive component in HWMP, reducing the initial delay that other path selection protocols as AODV introduce. Moreover, since the reactive component of the protocol begins after the initial data has begun transmission, it allows discovery of an optimal path for the flow. Also, any traffic that is addressed to an external network (e.g. Internet Traffic) will always have the optimal path since the OMEGA gateway (root node) builds a proactive tree creating optimal paths from every node to itself, further reducing packet overhead since the reactive component does not have to be initiated.

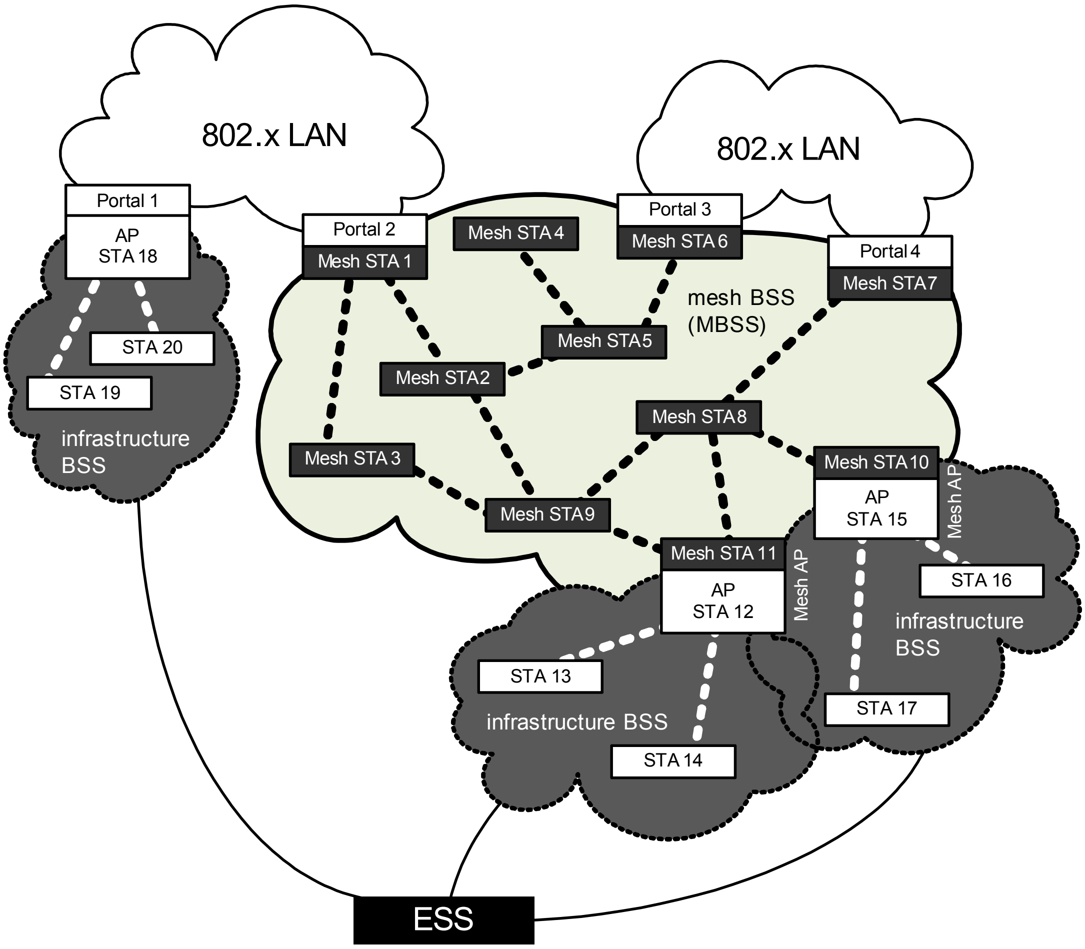


Figure 16 - Example MBSS containing mesh STAs, mesh APs and portals.

All HWMP modes of operation utilise common processing rules and primitives. HWMP information elements are the Path Request (PREQ), Path Reply (PREP), Path Error (PERR) and Root Announcement (RANN). The metric cost of the links determines which paths HWMP builds. In order to propagate the metric information between mesh STAs, a metric field is used in the PREQ, PREP and RANN elements.

Path selection in HWMP uses a sequence number mechanism to assure that mesh STAs can distinguish current path information from stale path information at all times in order to maintain loop-free connectivity. Each mesh STA maintains its own HWMP sequence number, which is propagated to other mesh STAs in HWMP information elements.

III.5.2 On Demand Path Selection Mode

If a source mesh STA needs to find a path to a destination mesh STA using the reactive path selection mode, it broadcasts a PREQ with the target mesh STA specified in the list of targets and the metric field initialised to the initial value of the active path selection metric.

When a mesh STA receives a new PREQ it creates or updates its path information to the originator mesh STA and propagates the PREQ to its neighbour peer mesh STAs if the PREQ contains a greater HWMP sequence number, or the HWMP sequence number is the same as the current path and the PREQ offers a better metric than the current path. Each mesh STA may receive multiple copies of the same PREQ that originated at the originator mesh STA, each PREQ traversing a unique path.

Whenever a mesh STA propagates a PREQ, the metric field in the PREQ is updated to reflect the cumulative metric of the path to the originator mesh STA. After creating or updating a path to the originator mesh STA, the target mesh STA sends an individually addressed PREP back to the originator mesh STA.

The PREQ provides "Target Only" (TO) and "Reply and Forward" (RF) flags that allow path selection to take advantage of existing paths to the target mesh STA by allowing an intermediate mesh STA to return a PREP to the originator mesh STA. If the TO flag is set to 1, only the target mesh STA sends a PREP. The effect of setting the TO flag to 0 is the quick establishment of a path using the PREP generated by an intermediate mesh STA, allowing the forwarding of data frames with a low path selection delay. The effect of setting the RF flag to 1 (in addition to setting the TO flag to 0) is the selection (or validation) of the best path after the path selection procedure has completed. If the RF flag is set to 1 (and the TO flag to 0), the first intermediate mesh STA that has a path to the target sends a PREP and propagates the PREQ with the TO flag set to 1 to avoid all other intermediate mesh STAs on the way to the target sending a PREP. If the RF flag is set to 0 (and the TO flag to 0), the PREQ is not propagated by the mesh STA.

Intermediate mesh STAs create a path to the target mesh STA on receiving the PREP, and also forward the PREP toward the originator. When the originator receives the PREP, it creates a path to the target mesh STA. If the target mesh STA receives further PREQs with a better metric, then the target updates its path to the originator to the new path and also sends a new PREP to the originator along the updated path. A bidirectional, best metric end-to-end path is established between the originator and target mesh STA.

III.5.3 Proactive Tree Building Mode

III.5.3.1 General

There are two mechanisms for proactively disseminating path selection information for reaching the root mesh STA. The first method uses a proactive Path Request (PREQ) element and is intended to create paths between the root mesh STA and all mesh STAs in the network proactively. The second method uses a Root Announcement (RANN) element and is intended to distribute path information for reaching the root mesh STA but the actual paths to the root mesh STA can be built on-demand.

III.5.3.2 Proactive PREQ Mechanism

The PREQ tree building process begins with a proactive Path Request element sent by the root mesh STA, with the Target Address set to all ones, the TO flag set to 1 and the RF flag set to 1. The PREQ contains the path metric (set to the initial value of the active path selection metric by the root mesh STA) and an HWMP sequence number. The proactive PREQ is sent periodically by the root mesh STA, with increasing HWMP sequence numbers.

A mesh STA receiving a proactive PREQ creates or updates its forwarding information to the root mesh STA, updates the metric and hop count of the PREQ, records the metric and hop count to the root mesh STA, and then transmits the updated PREQ. Information about the presence of and distance to available root mesh STA(s) is disseminated to all mesh STAs in the network.

Each mesh STA may receive multiple copies of a proactive PREQ, each traversing a unique path from the root mesh STA to the mesh STA. A mesh STA updates its current path to the root mesh STA if and only if the PREQ contains a greater HWMP sequence number, or the HWMP sequence number is the same as the current path and the PREQ offers a better metric than the current path to the root mesh STA. The processing of the proactive PREQ is the same as in the reactive mode previously described.

If the proactive PREQ is sent with the "Proactive PREP" bit set to 0, the recipient mesh STA may send a proactive PREP if a bidirectional path is required (for example, if the mesh STA has data to send to the root mesh STA that requires establishing a bidirectional path with the root mesh STA). During the time a bidirectional path is required, the recipient mesh STA shall send a proactive PREP even if the "Proactive PREP" bit is set to 0.

If the PREQ is sent with a "Proactive PREP" bit set to 1, the recipient mesh STA shall send a proactive PREP. The proactive PREP establishes the path from the root mesh STA to the mesh STA.

III.5.3.3 Proactive RANN Mechanism

The root mesh STA periodically propagates a RANN element into the network. The information contained in the RANN is used to disseminate path metrics to the root mesh STA. Upon reception of a RANN, each mesh STA that has to create or refresh a path to the root mesh STA sends an individually addressed PREQ to the root mesh STA via the mesh STA from which it received the RANN. The individually addressed PREQ follows the same processing rules defined in the reactive mode.

The root mesh STA sends a PREP in response to each PREQ. The individually addressed PREQ creates the reverse path from the root mesh STA to the originator mesh STA, while the PREP creates the forward path from the mesh STA to the root mesh STA.

III.5.4 HWMP Propagation

The term "propagate" is used to describe the means by which information elements are not transmitted "as is" across the network but are processed and modified along the way. Many HWMP information elements are intended to be processed and propagated across a mesh by mesh STAs. Each propagation is subject to certain rules or limitations. Certain parameters in the HWMP information elements are updated during the propagation.

The originator of an HWMP information element sets the initial value of HWMP TTL. The mesh STA that receives the HWMP information element shall propagate it if the HWMP TTL value is greater than 1. Before propagating the HWMP information element, the mesh STA decrements the HWMP TTL value. In general, the propagation of an HWMP information element is not subject to a delay.

III.5.5 HWMP Sequence Numbering

HWMP uses sequence numbers to prevent the creation of path loops and to distinguish stale and fresh path information. Each mesh STA keeps its own HWMP sequence number that it increments, uses in HWMP information elements, and processes according to the HWMP rules. HWMP sequence numbers for other mesh STAs are maintained in the forwarding information.

An HWMP sequence number is included in the PREQ, PREP, PERR, and RANN elements. The HWMP sequence number in the forwarding information is updated whenever a mesh STA receives new (i.e., not stale) information about the HWMP sequence number from a PREQ, PREP, or PERR that may be received relative to that target mesh STA or destination mesh STA. HWMP depends on each mesh STA in the network to own and maintain its HWMP sequence number to guarantee the loop-freedom of all paths towards that mesh STA. A mesh STA increments its own HWMP sequence number in two circumstances:

- Immediately before an originator mesh STA starts a path discovery, it shall increment its own HWMP sequence number. This prevents conflicts with previously established reverse paths towards the originator mesh STA. However, it might be advantageous not to increment the HWMP sequence number too frequently.

- Immediately before a target mesh STA originates a PREP in response to a PREQ, it shall update its own HWMP sequence number to maximum (current HWMP sequence number, target HWMP sequence number in the PREQ) + 1. The target HWMP sequence number is relevant when a link was broken along the path and the stored sequence number was increased at an intermediate mesh STA.

In general, when a mesh STA receives an information element with an HWMP sequence number that is less than the last received HWMP sequence number for that mesh STA, it discards the received information element. If they are the same, the outcome (information element processed or not) depends on the type of the information element and some additional conditions.

The only circumstance in which a mesh STA may change the HWMP sequence number of another mesh STA independently of the reception of an HWMP element is in response to a lost or expired link to the next hop towards that destination mesh STA. The mesh STA determines which destinations use a particular next hop by consulting its forwarding information. In this case, for each destination that uses the next hop, the mesh STA increments the HWMP sequence number in the forwarding information and marks the path as invalid. Whenever any forwarding information containing an HWMP sequence number greater or equal to the recorded HWMP sequence number for an affected destination is received by a mesh STA that has marked that forwarding information as invalid, the mesh STA shall update its forwarding information according to the information contained in the update.

III.5.6 Forwarding Information

The forwarding information to a destination consists of at least a destination mesh STA address, the HWMP sequence number (SN) of the destination, the Next Hop address, the path metric, the number of hops, the precursor list, and the lifetime of this forwarding information.

An entry in the precursor list contains the precursor mesh STA address and the lifetime of this entry. If an existing entry in a precursor list is updated, the lifetime is the maximum of the current and the updated value. If the lifetime of a precursor expires, it will be deleted from the precursor list. Precursors are used to identify legitimate transmitters of individually addressed frames and for the notification of link failures.

PREQ elements and PREP elements create or update the forwarding information of the mesh STAs that process these information elements as follows:

➡ The mesh STA may create or update its forwarding information to the transmitter of the information element if the path metric improves.

➡ The mesh STA may create or update its forwarding information to the originator mesh STA, if it received a PREQ, or to the target mesh STA, if it received a PREP, and one of the following conditions is met:
  - The Originator or Target HWMP sequence number > HWMP sequence number in the forwarding information for this Originator or Target mesh STA.
  - The Originator or Target HWMP sequence number = HWMP sequence number in the forwarding information for this target mesh STA AND the updated path metric is better than the path metric in the forwarding information.

III.5.7 Information Elements

III.5.7.1 PREQ Element

The Path Request (PREQ) element is used for discovering a path to one or more target mesh STAs, building a proactive (reverse) path selection tree to the root mesh STA, and confirming a path to a target mesh STA (optional).

Element ID	Length	Flags	Hop-count	Time to Live	PREQ ID	Originator mesh STA Address	Originator HWMP Sequence Number	Originator Proxied Address	Lifetime
Octets: 1	1	1	1	1	4	6	4	0 or 6	4

Metric	Target Count	Per Target Flags #1	Target Address #1	Target HWMP Sequence Number #1	...	Per Target Flags #N	Target Address #N	Target Sequence Number #N
4	1	1	6	4	...	1	6	4

Figure 17 - Path Request Element (PREQ).

B0	B1	B2	B3 B5	B6	B7
Portal Role	Addressing Mode	Proactive PREP	Reserved	AE	Reserved
Bits: 1	1	1	3	1	1

Figure 18 - PREQ Flags Field Format.

III.5.7.2 PREP Element

The Path Reply (PREP) element is used to establish a forward path to a target and to confirm that a target is reachable.The PREP is issued in response to a PREQ.

Element ID	Length	Flags	Hop-count	Time to Live	Target mesh STA Address	Target HWMP Sequence Number	Target Proxied Address	Lifetime
Octets: 1	1	1	1	1	6	4	0 or 6	4

Metric	Originator mesh STA Address	Originator HWMP Sequence Number
4	6	4

Figure 19 - Path Reply Element (PREP).

III.5.7.3 PERR Element

The PERR element is used for announcing a broken link to all traffic sources that have an active path over this broken link. The active forwarding information associated with the unreachable destinations should no longer be used for forwarding.

Element ID	Length	TTL	Number of Destinations N	Flags #1	Destination Address #1	HWMP Sequence Number #1	Reason Code #1	...
Octets: 1	1	1	1	1	6	4	2	

Figure 20 - Path Error Element (PERR).

III.5.8 Reactive and Proactive Modes Cooperation Method

Although HWMP provides two routing modes, the standard defines only the frame formats and operations of sending and receiving message frames, and it does not present an efficient method to cooperate two routing modes. Next is presented an efficient method of cooperation between the Proactive and Reactive modes in HWMP.

III.5.8.1 <u>Faults and Merits of the Two Routing Modes</u>

The operating method and the characteristics of the reactive routing mode of HWMP are very similar to the existing AODV, except that HWMP uses layer 2 routing. In the reactive routing mode, if a source node has no routing path to a destination node, it broadcasts a PREQ message inside the mesh network. The destination node that received the PREQ message sends a unicast PREP message back to the source node and then a bidirectional routing path between the source and the destination nodes is established. During this procedure, the PREQ ID and the destination sequence number are used to prevent sending duplicated messages and to establish loop-free routing paths.

The reactive routing mode always provides the optimum routing paths by establishing its path when data transmission is required. However, such a method results in high initial latency. Figure 21 shows the problem of increasing the initial latency when using the reactive routing mode.

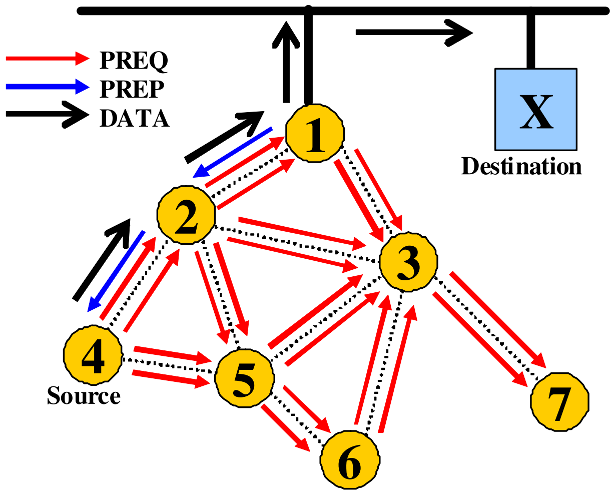


Figure 21 - The problem of initial High Latency on the Reactive Mode in HWMP.

In the tree-based proactive routing mode of HWMP, one mesh node is selected as a tree root node and a tree is created to proactively establish optimum routing paths between the root node and all nodes in the mesh network. The root node broadcasts either PREP or RANN messages periodically to create and maintain the tree. During this tree creation, only the root node can know the routing information to all nodes in the mesh network, and other nodes maintain only routing paths to the root node and their child nodes.

The proactive routing mode has merits and faults opposite to the reactive routing mode. When the proactive routing mode is used, the mesh node that intends to send data delivers it to its parent node immediately without searching for a routing path to the destination node. The node that receives the data checks whether the destination is one of its child nodes. If so, it forwards it to the destination node, otherwise it forwards the data to its parent node repetitively. When the destination node exists in the external network, data is delivered to the root node eventually and the root node sends it to the external network through the gateway node. As explained above, when the destination node is in the external network, the proactive routing mode can guarantee minimum initial latency in sending data, because it does not require the procedure of searching for an optimum routing path to the destination, unlike the reactive routing mode. Therefore, the proactive routing mode is very useful when most of the data traffic heads for the external network.

However, inversely, if the destination node exists in the same mesh network, the data transmission throughput can decrease due to data being sent through non optimum routing paths. Figure 22 shows the problem that can occur in the case of data transmission between two nodes in the mesh network. Assume that the mesh node 1 is the root node, the tree is created as shown in Figure 22, and that mesh nodes 4 and 6 are a source and a destination, respectively. First, the source node delivers data to its parent node 2. Next, node 2 forwards data to its parent node 1, again because the destination does not belong to its child nodes. Finally, the root node 1 sends data to the destination by using routing information already stored in its routing table. Although the optimum routing path with the routing metric of hop count is node 4 → 5 → 6, the routing path is established as node 4→2→1→3→ 6. That is, even if there exists a better routing path between the source and the destination, all data from the source are forwarded through a non optimum routing path. Consequently, as the efficiency of data transmission drops, data transmission throughput decreases.

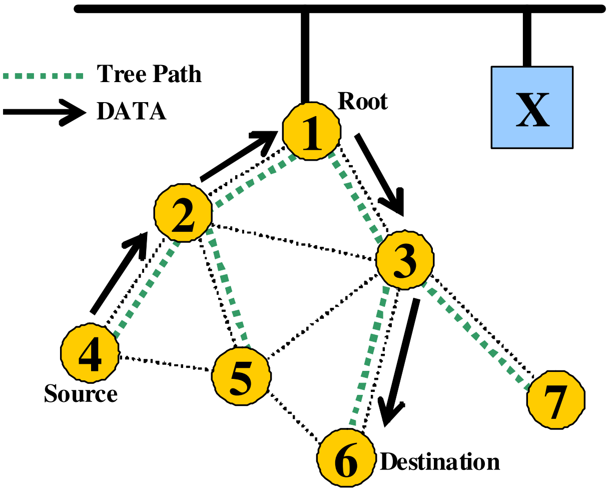


Figure 22 - The problem of non optimum paths in the proactive routing mode.

III.5.8.2 <u>The Cooperation Method</u>

The proposed cooperation method, referred to as the hybrid routing mode is designed in such a way that the proactive routing mode is the default and the reactive routing mode is used when communicating between mesh nodes in the same network.

First, communication starts normally taking the path of the proactive tree by default. Once a packet arrives to a node, it checks whether it is addressed to one of its child nodes, to itself or to a parent node.

- In case it is addressed to a child node the data packet is normally forwarded and the reactive mode does not start.
- In case it is addressed to itself the data packet is accepted and the reactive mode does not start.
- In case the data packet is addressed to the parent node one of two things can happen:
  - If the parent node is a child to the Root node, one the steps shown above are taken.
  - If the parent node IS the Root node, data is forwarded to the appropriate child node using the information from the forwarding tables. At the same time, a control packet is sent back to the source to indicate to this node that a Reactive PREQ must be started.

Note that if the destination node is the Root node, the proactive mode remains the only mode used. Figure 23 below shows an example of application of this hybrid routing mode.

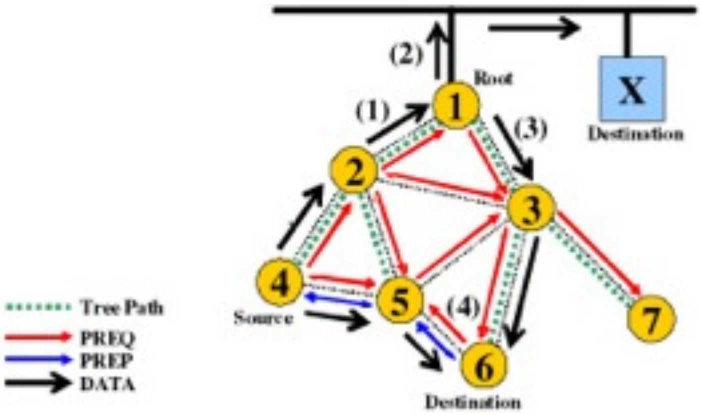


Figure 23 - Example of operation of the Hybrid Routing Mode.

Although the reactive routing mode always provides the optimum routing paths for data transmission, the initial latency can be very high when communicating with a destination node in an external network. On the other hand, the proactive routing mode shows low initial latency when communicating with a destination node in an external network, but the data transmission throughput can decrease when communicating with a destination node in the same mesh network because of the use of non optimum routing paths. By using the already set-up tree of the proactive mode for the first data frame, the initial delay caused by the path set-up time of the reactive mode is reduced while maintaining the optimum path for subsequent data frames sent through the mesh network.

Chapter Four: Implementation Of HWMP In OPNET Modeler

IV.1 About OPNET Technologies, Inc.

OPNET Technologies, Inc. is a leading provider of solutions for managing applications and networks. The company was founded in 1986 and went public in 2000. It is headquartered in Bethesda, Maryland. OPNET was Alain Cohen's (co-founder and current CTO & President) graduate project for a networking course while he was at the Massachusetts Institute Of Technology MIT. OPNET stands for Optimised Network Engineering Tools.

IV.1.1 OPNET Solutions

OPNET solutions address application performance management; network planning, engineering, and operations; and network R&D.

- ACE Live, ACE Analyst, and Panorama are OPNET's application performance management (APM) solutions. They are used for performance monitoring and troubleshooting throughout the application lifecycle to ensure compliance with application response-time SLAs.
- OPNET solutions for Network engineering, operations, and planning include the following
- IT Guru Network Planner helps enterprises plan their network for new applications or technologies.
- SP Guru Network Planner and SP Guru Transport Planner optimise service provider network designs for capacity, cost, QoS, and survivability.
- Sentinel is OPNET's network configuration auditing solution and it is used to ensure network integrity, security, and policy compliance.
- NetMapper complements OPNET's Sentinel and Guru solutions by automatically creating content-rich network diagrams in Microsoft Visio format.
- OPNET nCompass, a software solution for the NOC, enables operators to visualise real-time network topology, traffic, events, and status in a single view.
- OPNET Modeler, a network modelling and simulation software solution, is one of OPNET's flagship solutions and also its oldest product.

IV.2 General Description Of The OPNET Modeler

IV.2.1 About The OPNET Modeler

OPNET Modeler provides a comprehensive development environment supporting the modelling of communication networks and distributed systems. Both behaviour and performance of modelled systems can be analysed by performing discrete event simulations. The OPNET Modeler environment incorporates tools for all phases of a study, including model design, simulation, data collection, and data analysis.

IV.2.2 OPNET Modeler Architecture

OPNET Modeler provides a comprehensive development environment for modelling and performance evaluation of communication networks and distributed systems. The package consists of a number of tools, each one focusing on particular aspects of the modelling task. These tools fall into three major categories that correspond to the three phases of modelling and simulation projects:

- Model Specification and Modelling Communications with Packets.
- Data Collection and Simulation.
- Analysis.

These phases are necessarily performed in sequence. They generally form a cycle, with a return to Specification following Analysis. Specification is actually divided into two parts: initial specification and re-specification, with only the latter belonging to the cycle, as illustrated in figure 24.

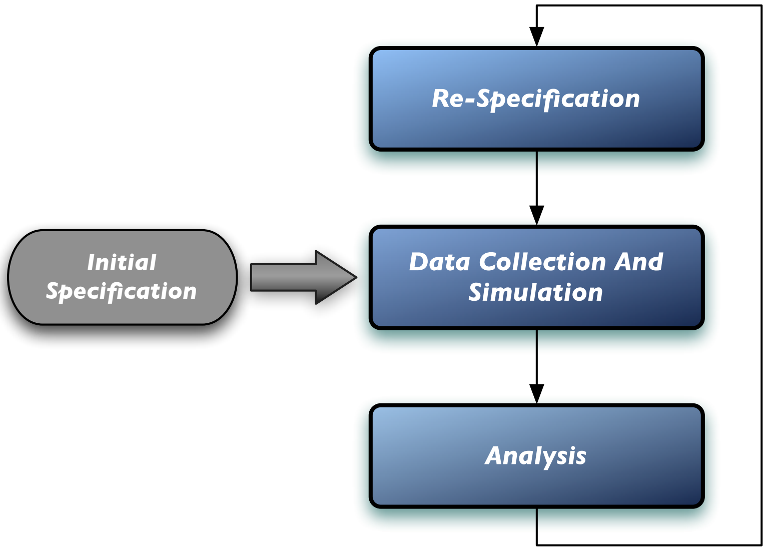


Figure 24 - Simulation Project Cycle.

IV.2.2.1 Model Specification

Model specification is the task of developing a representation of the system that is to be studied. OPNET Modeler supports the concept of model reuse so that most models are based on lower level models developed beforehand and stored in model libraries. OPNET Modeler is able to offer specific capabilities to address the diverse issues encountered in networks and distributed systems. To present the model developer with an intuitive interface, these editors handle the required modelling information in a manner that is parallel to the structure of real network systems. Therefore, the model-specification editors are organised hierarchically. Models built in the Project Editor rely on elements specified in the Node Editor; in turn, when working in the Node Editor, you use models defined in the Process Editor and External System Editor. The Process Editor expresses process models in a language called Proto-C, which is specifically designed to support development of protocols and algorithms. Proto-C is based on a combination of state transition diagrams (STDs), a library of high-level commands known as Kernel Procedures, and the general facilities of the C or C++ programming language. The remaining editors are used to define various data models, typically tables of values, that are later referenced by process or node-level models.

IV.2.2.2 Data Collection And Simulation

The objective of most modelling efforts is to obtain measures of a system's performance or to make observations concerning a system's behaviour. OPNET Modeler supports these activities by creating an executable model of the system. Provided that the model is sufficiently representative of the actual system, OPNET Modeler allows realistic estimates of performance and behaviour to be obtained by executing simulations. Several mechanisms are provided to collect the desired data from one or more simulations of a system. Discrete event simulations are capable of producing many types of output. However, in most cases, the types of data directly supported by discrete event simulations are Output Vectors and Output Scalars. In addition to numerical forms of output, discrete event simulations can provide a visual analysis of a network model's behaviour by providing an animated representation of the simulated scenario.

IV.2.2.3 Analysis

The third phase of the simulation project involves examining the results collected during simulation. Typically, much of the data collected by simulation runs is placed in output vector files. OPNET Modeler provides a graphing and numerical processing environment in the Results Browser of the Project Editor.

IV.2.3 OPNET Modeler Editors

OPNET Modeler presents its capabilities in the form of distinct environments called editors. Each editor allows you to do some set of related functions within a window that is contained in the overall graphical environment. Figure 25 shows the three basic modeller editors: Project, Node and Process Editors.

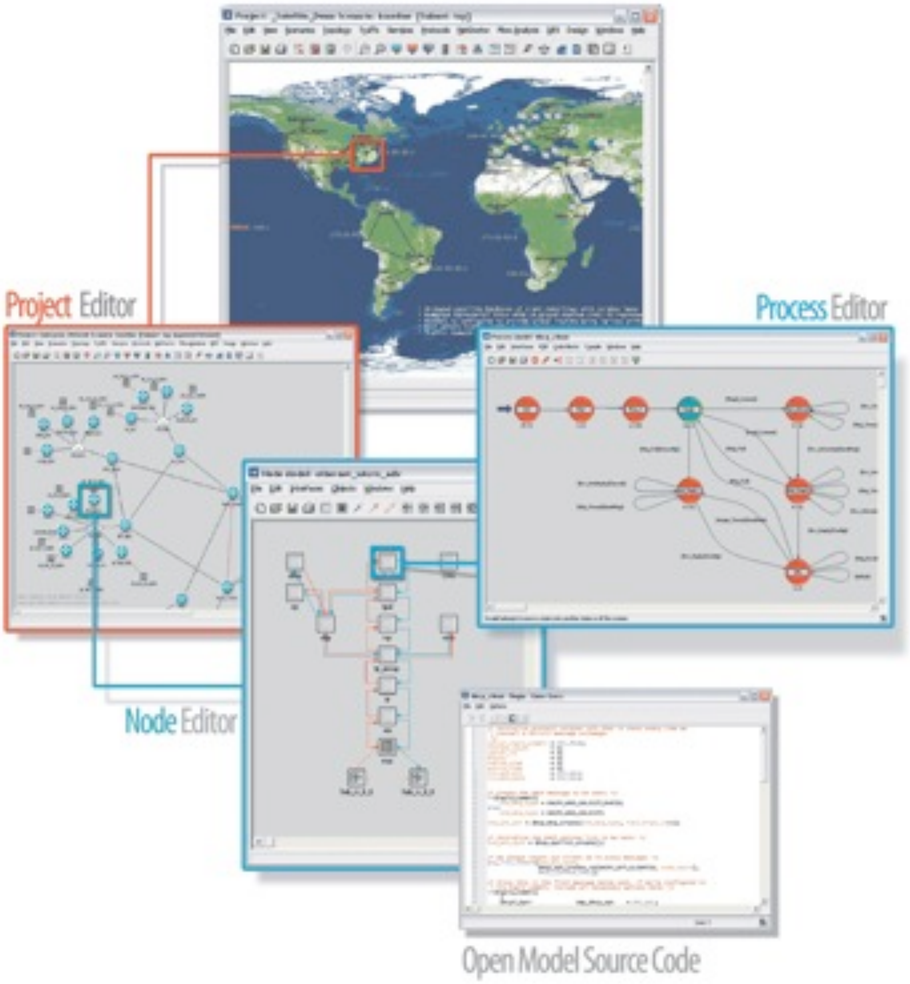


Figure 25 - OPNET Modeler Editors

IV.2.3.1 Project Editor

The Project Editor is used to construct and edit the topology of a communication network model. It also provides basic simulation and analysis capabilities. The Network Domain in which the Project Editor works is the highest modelling level, in the sense that it encompasses objects that are defined in the other modelling domains. The network model therefore specifies the entire system to be simulated. A network model contains only three fundamental types of objects: subnetworks, nodes, and links. There are several varieties of nodes and links, each offering different basic capabilities. In addition, each node or link is further specialised by its model, which determines its functions and behaviour.

IV.2.3.2 Node Editor

The Node Editor is used to specify the structure of device models. These device models can be instantiated as node objects in the Network Domain (such as computers, packet switches, and bridges). In addition to the structure, the node model developer defines the interface of a node model, which determines what aspects of the node model are visible to its user. This includes the attributes and statistics of the node model. Nodes are composed of several different types of objects called modules. At the node level, modules are black boxes with attributes that can be configured to control their behaviour. Each one represents particular functions of the node's operation and they can be active concurrently. Several types of connections support flow of data between the modules within a node.

IV.2.3.3 Process Editor

The Process Editor is used to specify the behaviour of process models. Process models are instantiated as processes in the Node Domain and exist within processor and queue modules. Processes can be independently executing threads of control that do general communications and data processing functions. They can represent functionality that would be implemented both in hardware and in software. In addition to the behaviour of a process, the process model developer defines the model's interfaces, which determines what aspects of the process model are visible to its user. This includes the attributes and statistics of the process model. Process models use a finite state machine (FSM) paradigm to express behaviour that depends on current state and new stimuli.

IV.2.3.4 Link Editor

The Link Editor provides operations for creating and editing link models.

IV.2.3.5 Packet Format Editor

The Packet Format Editor provides a way to specify the collection of fields contained by a formatted packet. Each field has attributes specifying its name, type, size, and default value. Modelling each field's size allows the overall size of the packet to be calculated.

IV.2.3.5 ICI Editor

ICI formats (Interface Control Information formats) define the internal structure of ICIs. ICIs are collections of data used to formalise interrupt-based inter-process communication.

IV.2.3.6 PDF Editor

The PDF Editor is used to create, edit, and view probability density functions (PDFs).

IV.3 Implementation Of HWMP In OPNET Modeler

IV.3.1 The OMEGA Network In OPNET Modeler

The OMEGA network in OPNET Modeler was created using the three modeller editors: Project, Node and Process editors. In the Project editor the scenario representing the actual network environment was created, inserting network capable devices in different positions in order to accurately represent an OMEGA network. The network elements used in this thesis utilise three main communication technologies: WLAN, Ethernet and PLC. At the time of implementation, the other technologies that are meant to be used within OMEGA were not available due to the fact that they were either not yet implemented in OPNET or are still in under research and development. The three available technologies used in order to demonstrate the performance of HWMP have the following characteristics: WiFi (11 Mbps DS - 801.11b to 54 Mbps Extended Range - 802.11g), Gigabit Ethernet (1000BaseX) and PLC (Bus connection at a 200 Mbps). Figure 26 shows at the bottom layer how the before mentioned technologies are implemented.

The OMEGA architecture in OPNET Modeler was already implemented by other thesis students, but since the main work of this thesis uses many of the already implemented functionalities, a brief description of all that is needed for the Path Selection engine to work will be given.

The Inter-MAC layer is represented by a node inside the Node model. The initial idea was to place this node between the IP layer and the MAC layer, so that every technology passes through the Inter-MAC defined model. The choice to insert the node between the IP and ARP process models is due to the subsequent independence of each ARP table in interfacing with its own MAC layer. In fact, one main feature of the Inter-MAC is to communicate at the same time with MAC layers that implement different technologies; it is therefore important that each MAC has its own ARP table to which communicate. The main idea of the Inter-MAC layer is to make the lower technology-dependant layers transparent to all the upper layers above the Inter-MAC. The Inter-MAC node model will therefore be inserted just below the IP Node model and above all ARP node models. In order to complete the nodes insertion into the OSI stack we must connect it using OPNET's connections. There are three types of connections in node models: packet streams, which support the flow of data packets between modules, statistic wires, which support the transmission of numerical state information between modules, and logical associations, which indicate a binding between two modules allowing certain pairs of modules to do a function together. Also, in OPNET, processes can communicate with different mechanisms:

- Packet-based Communication
- Interface Control Information (ICI)
- Event States

Packets are transferred using packets streams. After inserting the Inter-MAC node model two packet streams connect each pair of nodes, one for upstream packets and one for downstream packets. Interface Control Information structures or ICIs are data objects that resemble packets in that they consist of a list of data items. However, ICIs are simpler data structures than packets because they contain storage only for user-defined values. In addition, there is no notion of size or ownership for an ICI as there is for a packet. In fact, it is common for ICIs to be concurrently shared and modified/examined by multiple entities. So the Inter-MAC has to recognise and read all the ICIs that the IP layer has defined for the ARP, and must also recognise the ICIs that the ARP layer has defined to communicate with the IP layer. Figure 26 shows the resulting node model that implements the Inter-MAC layer.

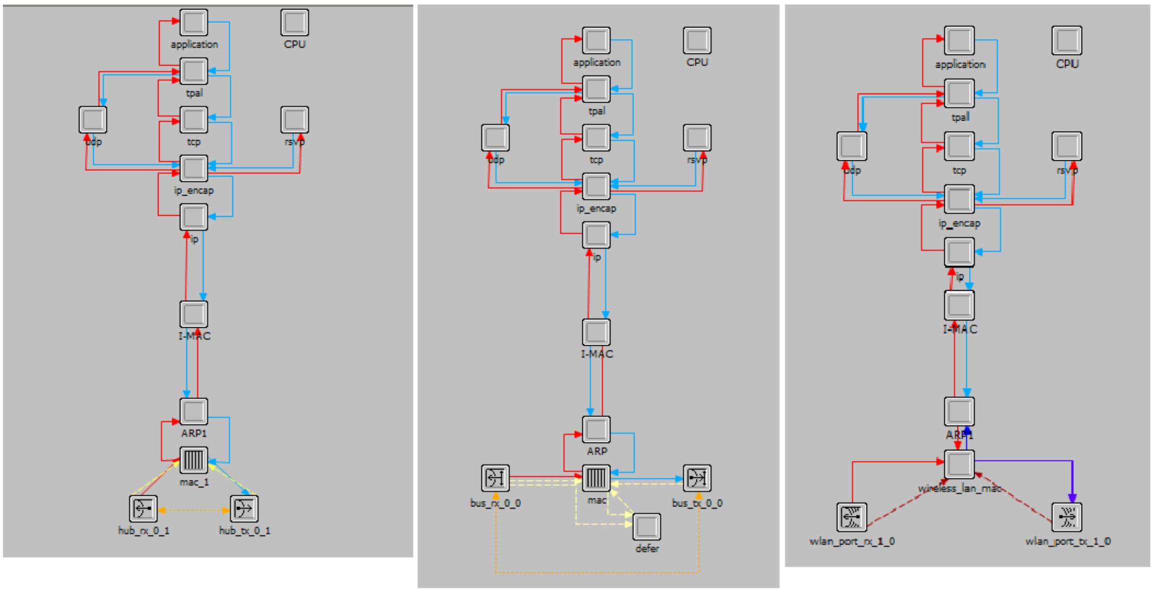


Figure 26 - Node Model of a Ethernet, PLC and WLAN workstations.

The next step is creating models that implement more than one technology. In order to do this, the lower layer for each technology must be "copied" and put into place. After this, the ARP node model must be connected by means of Packet Streams to the upper layer Inter-MAC node model. The Inter-MAC code was implemented in a modular modality, meaning that when connecting to it one of the blocks below it (ARP, MAC and transmitter processes) the network node can correctly execute its functionalities. Figure 27 shows a resulting node that implements all three available technologies: WLAN, Ethernet and PLC.

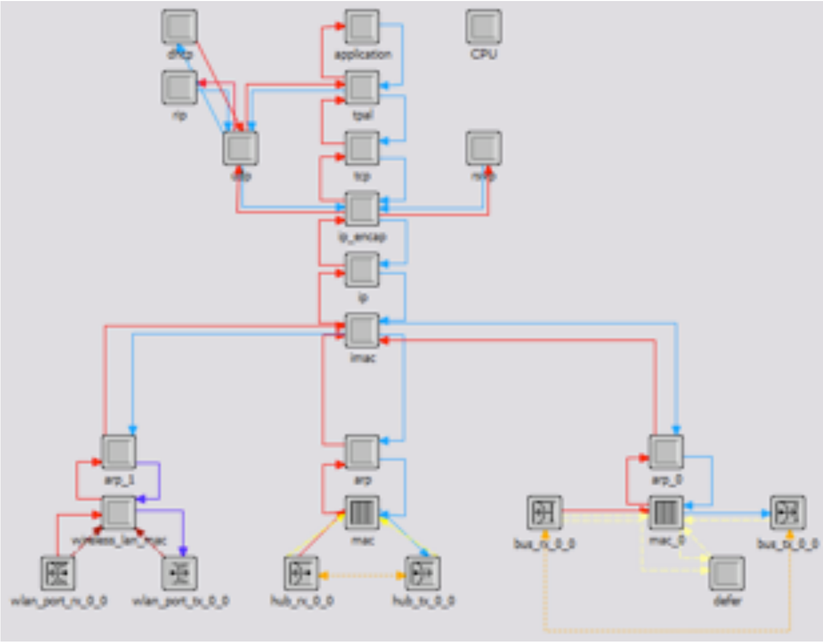


Figure 27 - Node Model implementing WLAN, Ethernet and PLC.

IV.3.2 The Inter-MAC Process Model

As shown in the general description about the OPNET Modeler Editors, the Process Model is used to represent a functionality that could be implemented both in hardware and in software. A Process Model is defined as a finite-state machine, in which events arriving to the node that implements it causes a change of state or triggers a condition. There are two kinds of states: Forced (shown in green) and Unforced (shown in red). Forced states enter and exit the state without the need of another event to trigger the exit, they execute the enter and exist executives as one. Instead, unforced states execute the enter executives and pauses, exiting only when a proper arriving event triggers it. While it's waiting, it can respond to other events that can trigger the execution of other functions. Upon the arrival of a proper event, it exist the state and executes the exit executives.

The Inter-MAC architecture was previously implemented by other thesis students using a parent process model (The Inter-MAC process model) and other child processes created by the parent process. This child processes are the Inter-MAC engines which will be discussed in the following paragraphs. Shown in figure 28, the Inter-MAC parent process has a series initial unforced states that are needed in order to give time to the IP layer to build its tables. At the idle state a node can be interrupted by a series of events which will be triggered by self-made schedules or arriving packets.

The basic function of this process model is to handle packets coming from from the upper and lower layers. Also, it creates the Inter-MAC engines that are needed in order to process the incoming packet and creates a shared memory that all the other processes can read and write. All packets that arrive from the ARP layer are handled by the forced state Dispatch which, depending on the type of packet that arrives (Control packet, Data packet, Probe packet), calls for a specific set of functions by invoking the proper Inter-MAC engine. The next paragraphs are dedicated to giving a brief overview of the functions that the Forwarding, QoS and Monitoring engines perform, in order to better understand the Path Selection engine which implements the Hybrid Wireless Mesh Protocol and is the main work of this thesis.

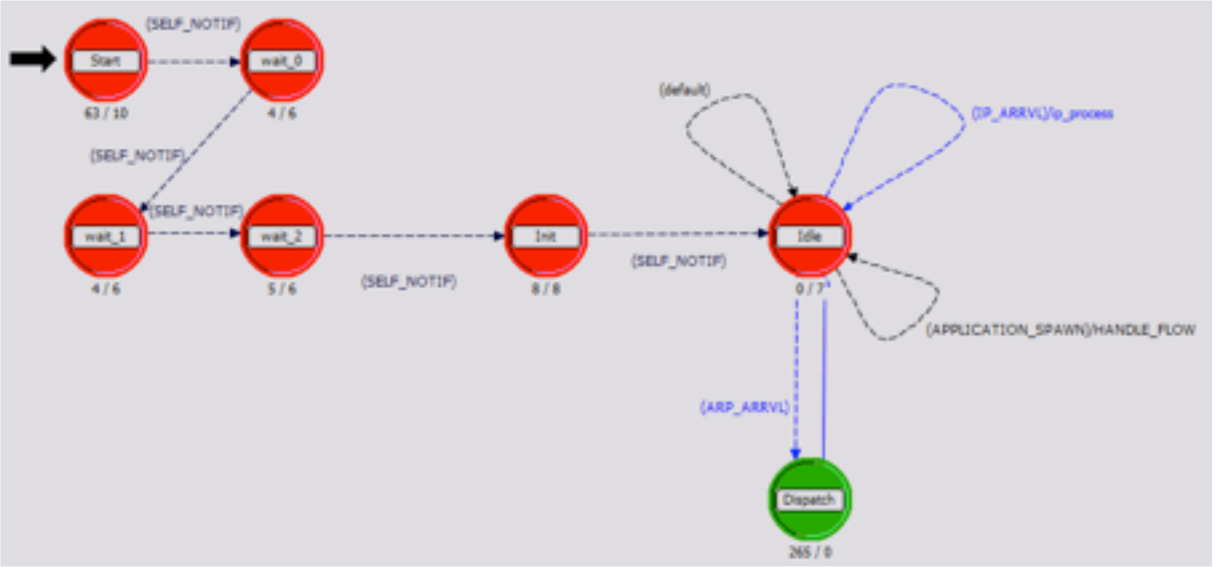


Figure 28 - Process Model of the Inter-MAC.

IV.3.3 The Inter-MAC Engines

IV.3.3.1 Monitoring Engine

As shown in chapter two, the Inter-MAC monitoring engine is responsible for gathering network information which is mainly supposed to support configuration, performance and fault management of the OMEGA network. The Monitoring engine process model consists on a Forced state that is used to initialise the engine and two other Unforced states. This process model is in charge of gathering information about the network so that the other engines who use this information can make an appropriate decision. The monitoring engine sends probes every 100ms with a one hop lifetime. The idea of these probe frames is to gather information about the band and delay involved on a certain link. Specifically, each probe gathers information about the available bandwidth (band used by a flow, nominal band and free band), the delay of a link (average, maximum and minimum) and the jitter (which is calculated after several probe frames have been sent). All the information gathered by this engine is stored as an entry on the Local Link Table, which is itself stored on the shared memory that was previously created by the parent process. Further, this engine gathers information about which flows are using a certain link at a given time so that, when a link fails, it already knows which flows are affected. Then, it proceeds to inform the Path Selection engine that a link has gone down sending to it the list of flows that have been affected. After informing the Path Selection engine that a link has gone down, this engine releases the band that was occupied by the flows that were transiting it. Figure 29 shows the process model for this engine.

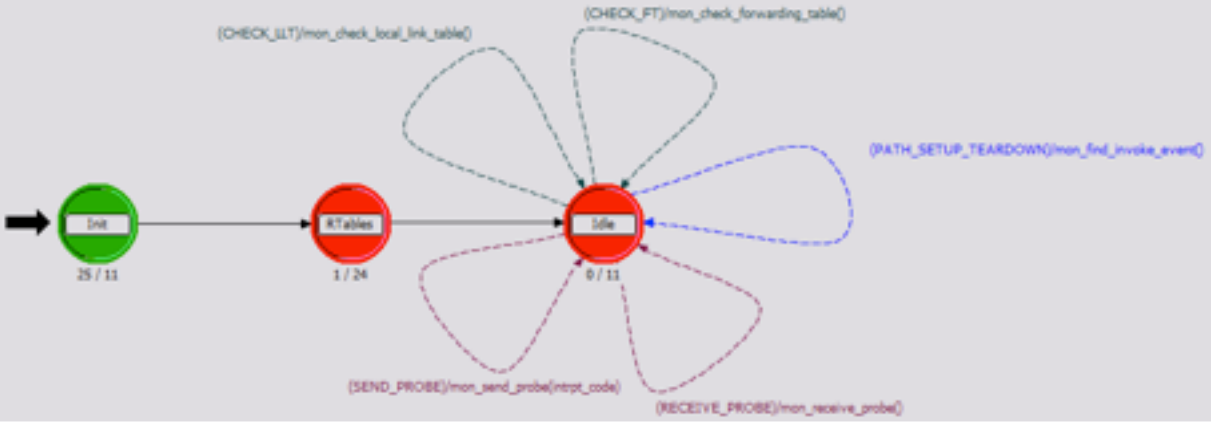

(CHECK_LLT)/mon_check_local_link_table()
(CHECK_FT)/mon_check_forwarding_table()
(PATH_SETUP_TEARDOWN)/mon_find_invoke_event()
Init
25 / 11
RTables
1 / 24
Idle
0 / 11
(SEND_PROBE)/mon_send_probe(intrpt_code)
(RECEIVE_PROBE)/mon_receive_probe()

Figure 29 - Process Model of the Monitoring Engine.

IV.3.3.2 QoS Engine

The QoS engine process model consists on a Forced state that initialises the engine and an Unforced idle state. This engine has the purpose of accepting a flow if the path for it is available. If it's not, it stores the application and then invokes the Path Selection engine with the appropriate parameters that it needs in order to find a path to the destination. Once the path has been found, an interrupt unlocks the stored application and starts the data flow. Figure 30 shows the process model for this engine.

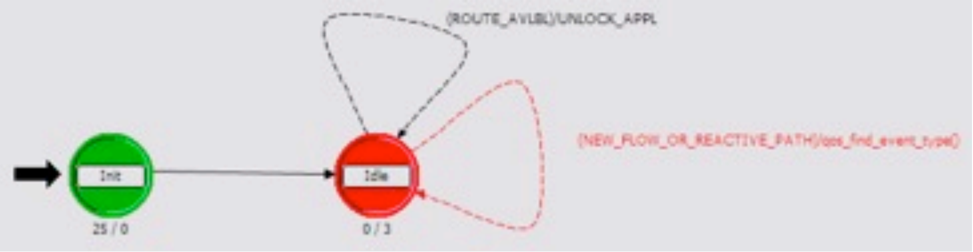

(ROUTE_AVLBL)/UNLOCK_APPL
(NEW_FLOW_OR_REACTIVE_PATH)/qos_find_event_type()
Init
25 / 0
Idle
0 / 3

Figure 30 - Process Model of the QoS Engine.

IV.3.3.3 Forwarding Engine

The Forwarding engine process model consists on a Forced state that initialises the engine and an Unforced idle state. The main purpose of this engine is to address incoming packets to the right outgoing interface. In order to do this, the Forwarding process model maintains a Forwarding Table in which is stored in each entry the information needed for a packet to be address to a destination node; these entries are differentiated by whether they were created by the Proactive or Reactive mode of HWMP. When this engine is invoked by the Path Selection engine, it searches the Forwarding Table for an entry with the input destination address and then forwards the packet to the right outgoing interface. In case the packet received is a broadcast packet, it sends it to all outgoing interfaces except the one it was received from. Figure 31 shows the process model for this engine.

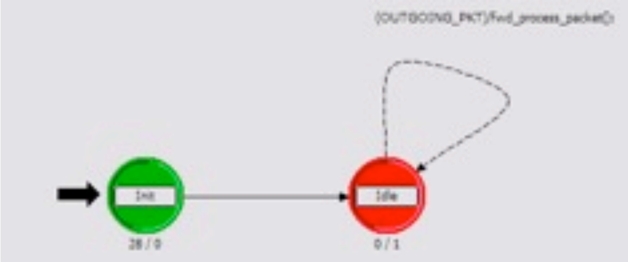


Figure 31 - Process Model of the Forwarding Engine.

IV.3.3.4 Path Selection Engine

The Path Selection Engine is the main work of this thesis, it was created as a child process of the Inter-MAC parent process. In the subsequent paragraphs, the functionality of the created process model will be explored, explaining in detail how the Path Selection protocol HWMP was implemented and the considerations that had to be taken.

IV.3.4 Path Selection Engine Implementation

IV.3.4.1 The HWMP Process Model

The Path Selection engine process model consists in two Forced states and two Unforced states. The Start state is used basically to initialise the process model; it initialises the Path Request table, the Path Discovery table and Preq_sent table. Also, it initialises the sequence_number and the preq_id variables. The subsequent state starts the Proactive mode of HWMP and then continues to the Idle state. Once on the Idle state, there are five different interrupts that can be triggered depending on the action that the Path Selection engine is required to do. Figure 32 shows the process model created for the Path Selection engine.

As said before, the Path Selection engine implements the *Hybrid Wireless Mesh Protocol*, a distributed Path Selection protocol that combines Reactive and Proactive modes to find paths for data throughout a meshed network such as OMEGA. As was detailed on the previous chapter, the Proactive mode of HWMP creates a path to every node from the OMEGA gateway from the time the OMEGA gateway is turned on. This allows to significantly reduce the initial delay for new flows that want to transit through the network. Further, the Reactive mode of HWMP allows a network device to find the optimum path to its destination once the flow has already started, allowing the best possible performance while maintaining a low delay. The subsequent paragraphs give an overview of how each of the modes were implemented, as well as highlighting the problems encountered and the solutions found.

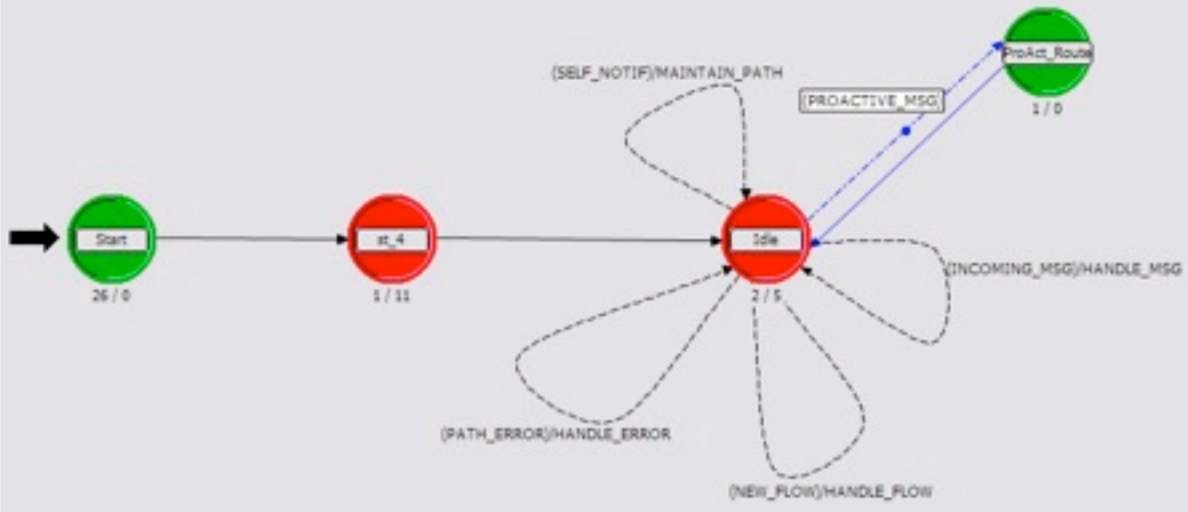


Figure 32 - Process Model of the Path Selection Engine.

IV.3.4.2 <u>HWMP Reactive Mode</u>

As seen in the previous chapter about Path Selection in Mesh Networks, the Reactive mode in HWMP is basically based on AODV, meaning that it uses PREQs and PREPs packets in order to find a path to the destination. The Reactive mode implemented in OPNET Modeler works as follows: First, the QoS engine receives a request to start a flow. The QoS engine verifies if there is a Proactive route set; if not, it starts a Reactive path request, sending to the Path Selection engine the required information (Destination Imac, Target Metric, Flow Id and Flow Descriptor).

Once the Path Selection process model is invoked, it starts the creation of a PREQ packet and sends it in broadcast. After a node receives a PREQ, it updates the accumulated metric and checks if whether this metric satisfies the Target Metric sent by the QoS engine - if not, the packet is discarded. Then, after a PREQ has been accepted and only if the node is not the destination node, it retransmits this PREQ through all the interfaces except the one was received from. When forwarding a PREQ, the Forwarding engine is invoked passing to it a data structure that contains the required information for the packet to be properly addressed. Further, the TTL value is reduced by a unit, while the Hop Count is incremented by the same amount. In case the PREQ arrives to its destination; first, the accumulated metric is checked to see if it satisfies the target metric; if it does, the PREQ is accepted, the forwarding and path selection tables are updated and a timer to send a PREP is set to NetDiameterTraversalTime (100ms by default). In the situation where another PREQ arrives after the timer has already started, the accumulated metric that it carries is compared to that stored in the path selection table, if better the entry is updated along with the forwarding table entry.

Once the timer expires, a PREP packet is sent in unicast back to the source node using the forwarding tables entries set by the incoming PREQs. This PREP follows basically the same rules applied to the PREQs and also carries an accumulated metric that is updated every time a PREP transits through a link. In addition, every time a PREP transits a link, it notifies the monitoring engine that a path for a flow has been accepted so that it can start monitoring the flow. When a PREP arrives to its destination (the source node of the flow), the retry_preq timer is cancelled and the application that was previously stored is unlocked in order to start data transmission. In case the Proactive Tree was already set and the application was already running, the latter is re-routed using the updated path that the Reactive mode created.

The retry_preq timer is a timer set every time a PREQ is sent. The purpose of this timer is to resend a PREQ if a PREP is not received after 2*NetDiameterTraversalTime (2*100ms by default). If the retry_preq timer expires, a new PREQ is sent; this can be done up to MAX_RETRIES times (3 by default). The Reactive mode updates its forwarding information by sending a PREQ every MAINTAIN_TIME (every 2 seconds by default) using the hwmp_path_maintenance() function.

IV.3.4.3 HWMP Proactive Mode

As previously said the Proactive mode starts as soon as the device is turned on and as long as the OMEGA Gateway node is set. In order to set a node as Gateway in the OMEGA network, when adding the node commodities in the project editor (e.g. imac addresses, target metric) one must set the node to be a Gateway by enabling this commodity. In the process model that was implemented, the Proactive mode starts 100ms after the Path Selection engine is invoked. This delay was necessary to allow the engine to be correctly initialised. As soon as the interrupt invoking the Proactive mode is called, the node that was previously set as a Gateway sets a boolean variable to true to identify itself in the code and then calls the hwmp_proactive_mode() function. As with the Reactive mode, the Proactive mode updates its forwarding information by sending Proactive PREQs (from now on called PPREQs) periodically (every 2 seconds) using the same hwmp_path_maintenance() function with the only difference that an interrupt is called in order to enter the ProAct_Route state and execute the corresponding code. In order to avoid a situation in which an undeliverable proactive packet keeps circulating the OMEGA network, a TTL (time-to-live) field with a value of 10 was used; this value can be modified for larger network scenarios but for the OMEGA network it was proven to be sufficient.

The Proactive mode in HWMP is very similar to its Reactive counterpart with some differences:

- First, the Gateway sends Proactive PPREQs packets to every node in the network but instead of them being addressed to a single destination they are sent in a broadcast manner (imac destination set to BROADCAST while imac source is set to the Gateway's imac address), meaning that every intermediate node has to respond with a Proactive PREP (from now on called PPREP) packet back to the Gateway.
- Second, every PPREQ packet carries only the Accumulated Metric (that is, no Target Metric). This is obvious since no flow has yet commenced and no requirements have been set on the path. Instead, as mentioned before, PPREQ find the best available path back to the Gateway node.
- Third, the Proactive mode has no Wait Window when receiving a PPREQ since, given the fact that the packet is sent in broadcast and every node has to reply with a PPREP, a wait window as used in the Reactive mode would cause a significantly incremented delay when creating the tree to the Gateway node.
- Fourth, the function to resend a PPREQ was not implemented in the Proactive mode; instead if a PPREP packet fails to arrive to the destination, the network waits the Path Maintenance time (2 seconds) in order to update the Path Tree.
- Lastly, when a PPREQ or a PPREP arrives at any given node it adds (or updates) an entry to the Forwarding Table. This entry differs from the entries created in the Reactive mode in that they

contain a further attribute indicating that the entry belongs to the Proactive made Tree. It was important to distinguish these entries in the Forwarding table so that if a link failed and the Proactive tree was not influenced, the flow could continue transmitting using the Proactive path as a backup.

In summary, excluding the differences mentioned with the Reactive mode, the Proactive mode follows basically the same algorithm, using to do this the proactive set of functions and a set of instructions that is shared between the Proactive and Reactive mode (See Annex A - A.1 Header Block).

IV.3.4.4 Handling Of Link Failures

Link failures is something that can occur constantly, especially in networks with wireless links where devices can change their position from time to time such as in the OMEGA network. When a link fails, the idea is to realise that this has happened and notify the affected entities as soon as possible. In order to do this, HWMP implements another type of packet called Path Error (or PERR from now on) that notifies all affected entities that a path used to transmit data has gone down (or it's no longer available) and that a new path to the destination has to be found. This functionality was implemented in the Path Selection engine in conjunction with the monitoring engine which is the one in charge of monitoring and notifying the Path Selection engine that a link is no longer available. The Path Selection engine receives from the monitoring engine:

- ➡ The list of flows that have been affected.
- ➡ The link interface that has been affected at the node of interest.

With this information, the Path Selection engine creates and sends a PERR for each affected flow, being the destination of the PERR the originator of the flow. Once the PERR arrives at its destination, the forwarding entry to the affected destination of the flow is cancelled and a Reactive PREQ is created. As with PREQs and PREPs, PERRs are handled in much the same way, with the exception that they neither carry a path metric nor do they update any tables.

IV.3.4.5 Considerations Taken During Implementation

During the implementation of HWMP in OPNET Modeler, certain considerations for different aspects had to be taken, some due to the fact that the proposed draft for the IEEE 802.11s standard which uses HWMP leaves some aspects undefined. Others, due to the fact that HWMP was being implemented with the OMEGA network in mind. Detailed next are some of the considerations taken:

- *Decision About Path Metrics:* The Metrics taken into consideration were the available band, the maximum delay and the hop count, where in reality only the first two were considered when comparing two metrics. In order to decide for the best path available, PREQ and PREP carry a value called accumulated metric. This value allows the destination to decide for the best available path out of all PREQ it might have received by choosing either the larger available bandwidth with the smallest delay or the path with the least used percentage of band. The implementation created chooses the larger available band with the smallest delay. However, this choice can be easily changed on an update of the implementation.

- *Propagation of PREQs:* In order to further reduce the number of PREQs propagated throughout the network, in addition to the sequence numbers and PREQ_ID, a PREQ that arrived to the node that generated it was automatically discarded.

- *Cooperation between Proactive and Reactive modes:* The main reason to use HWMP over other protocols was to gain the benefits of both reactive and proactive modes at the same time. In order to do this both modes had to work in cooperation so the following actions were taken. First, if the proactive tree was set the flow can start immediately using this already set path but, if the destination was past the Gateway, it notifies the originator of the flow to start a PREQ in order to find the best possible path. Second, if the destination of the flow is outside the OMEGA network (e.g. Internet Traffic) the Reactive mode is not used. And third, in order to distinguish paths created by the Proactive and Reactive modes, a boolean value Root node was added to every entry in the Forwarding table.

- *Obtaining Global Data:* Since information about the total amount of control packets sent throughout the OMEGA network was needed to estimate the traffic generated by the protocol, a global variable was defined that sums all bits belonging to PREQ, PREP and PERR packets.

IV.3.5 Deploying Applications In OPNET

In order to deploy an application in OPNET two specific nodes are needed in the network, these are: Application Config and Profile Config. Application Config lets the user create or choose from existing applications to be used as a data flow for the network. This applications can be FTP, Video Conferencing, Email, HTTP, Voice call, etc. Once chosen the application, Profile Config lets the user configure settings like the start time, end time, offset and others for the applications chosen.

After choosing the applications and choosing the time settings for them, one proceeds to effectively deploy the applications by choosing the Source and Destination node for each one. To do this, it's important to first add the commodities to each node involved that is either transmitting or receiving. This commodities must be added in both source and destination nodes and include: the Destination and Source imac address, Target metric (minimum bandwidth and maximum delay required) and Destination node name.

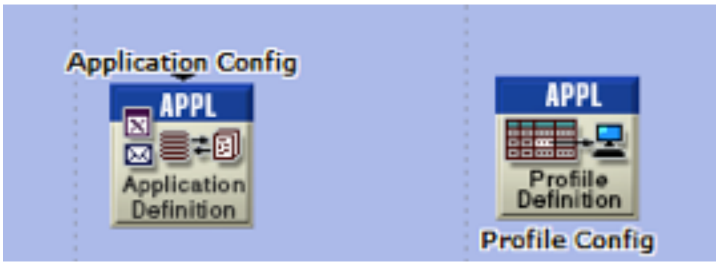


Figure 33 - Application Deployment Nodes.

Chapter Five: Simulations And Results

V.1 Introduction

The work of this thesis is centred around the implementation of the HWMP Path Selection protocol with the OMEGA network in mind. In order to test the performance of the protocol, several scenarios were created that focused in gathering specific performance parameters such as protocol convergence time, overhead and link failure response time. These scenarios do not represent an actual home network but instead aim to represent the specific parameter that is intended to be collected. In addition, since the OMEGA network is an actual HOME network (HAN) environment, a scenario representing this was created in order to see more real-life simulations and results. The results obtained from now on aim to not only measure the performance of HWMP on a meshed network such as OMEGA, but to also help optimise its performance according to the actual network environment where the protocol is proposed to be implemented.

V.2 The Scenarios

The scenarios that follow were used during the simulation stage of this thesis, these were created taking the performance data that was to be obtained into consideration. The next paragraphs give only a brief introduction to each scenario and the information that was gathered from it. Further information about specific settings, comments and conclusion regarding each scenario are given ahead in section V.3 Simulations And Results.

V.2.1 Family Home Test Network

The Family Home scenario (See Figure 34) aims to reproduce the “typical” OMEGA network infrastructure. It consists of 13 network devices connected using at least one technology between the three technologies that are available in OPNET (WiFi 802.11g at 54Mbps, Ethernet at 1Gbps and PLC at 200Mbps). The following table presents a summary of the nodes and the technologies they implement:

Node	WiFi	PLC	Ethernet	Node	WiFi	PLC	Ethernet
OMEGA Gateway	X	X	X	*Desktop Computer*		X	
Home TV	X		X	*Smartphone 2*	X		
NAS			X	*Printer*		X	X
Game Console	X		X	*Laptop*	X		X
Node 13	X	X		*Room TV*	X		X
Smartphone	X			*Smartphone 3*	X		
Node 4		X	X				

Table 1 - Node Technologies in the Family Home network.

The long Bus line (in black colour) in the scenario represents the electrical wiring and thus the PLC technology. The Gigabit Ethernet connections are represented by a maroon colour while most of the nodes that implement WiFi have an icon representing this. Since this scenario aims to reproduce a "typical" Home network, it was used to measure not only the performance of HWMP, but also the performance of the Inter-MAC and common everyday use applications.

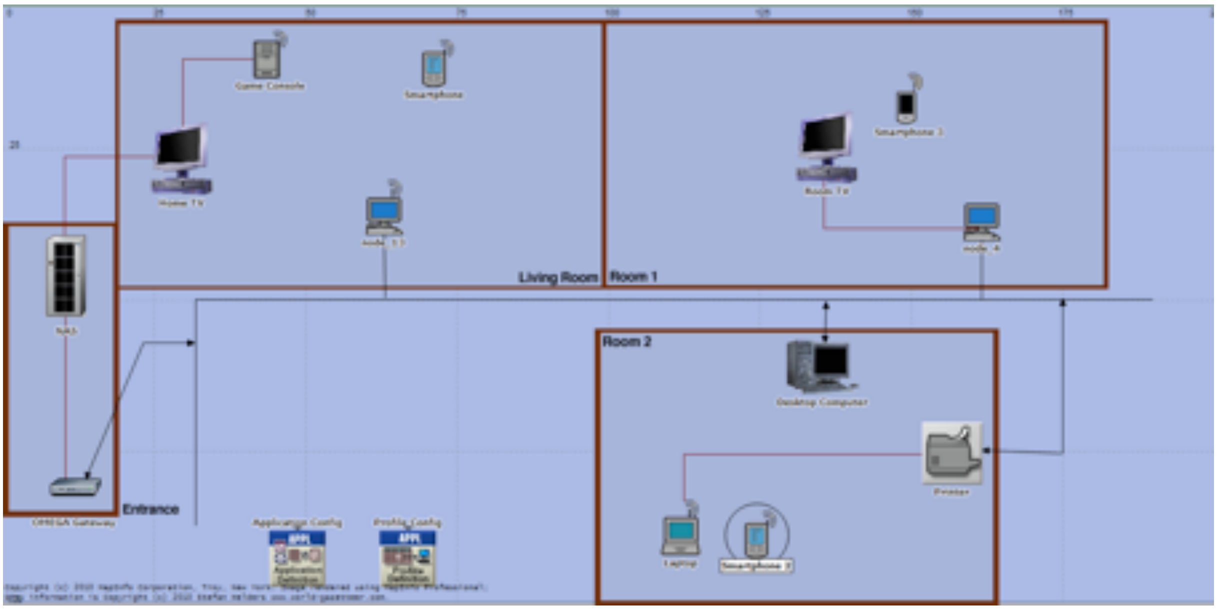


Figure 34 - Family Home Scenario.

V.2.2 HWMP Performance Test Networks

These scenarios aim to measure HWMP performance in terms of convergence time, protocol overhead and link failure response time. For each HWMP Performance Test Network the nodes were set into position in a random manner with distances between them varying from a couple of meters to a maximum of around 200m in straight line for the 20 Node network. Even though the nodes were placed randomly, two considerations were taken:

- At least half the nodes would use a minimum of two technologies in order to increase the number of possible paths (except for the 4 Nodes scenario).
- The Gateway was to be chosen depending on its network location, choosing the most network-centred node possible.

In following, each test network is represented by a table that summarises the list of nodes and technologies involved and a figure showing the topology that was implemented.

V.2.2.1 Performance Test Network With 4 Nodes

Node	WiFi	PLC	Ethernet	Node	WiFi	PLC	Ethernet
Node 1	X		X	Node 3	X		
Node 2			X	Node 4 - Gateway			X

Table 2 - Node Technologies in the 4 Node test network.

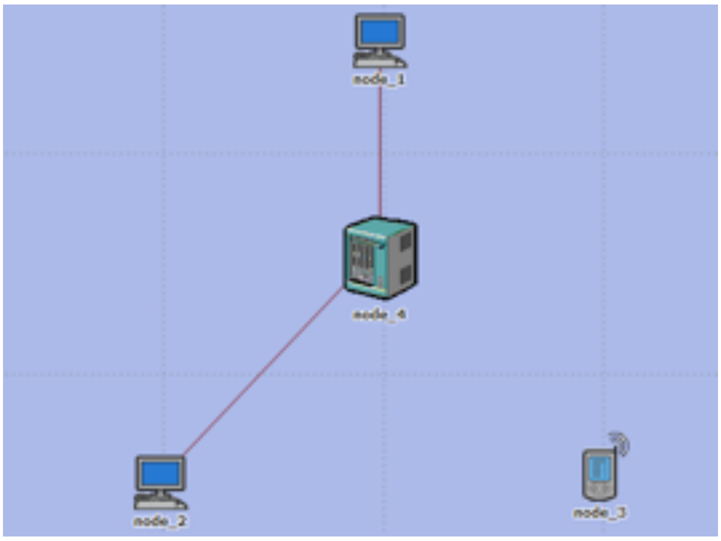


Figure 35 - Scenario with 4 Nodes.

V.2.2.2 Performance Test Network With 8 Nodes

Node	WiFi	PLC	Ethernet	Node	WiFi	PLC	Ethernet
Node 1		X	X	Node 5 - Gateway		X	X
Node 2	X		X	Node 6			X
Node 3	X			Node 7	X		X
Node 4			X	Node 8	X	X	

Table 3 - Node Technologies in the 8 Node test network.

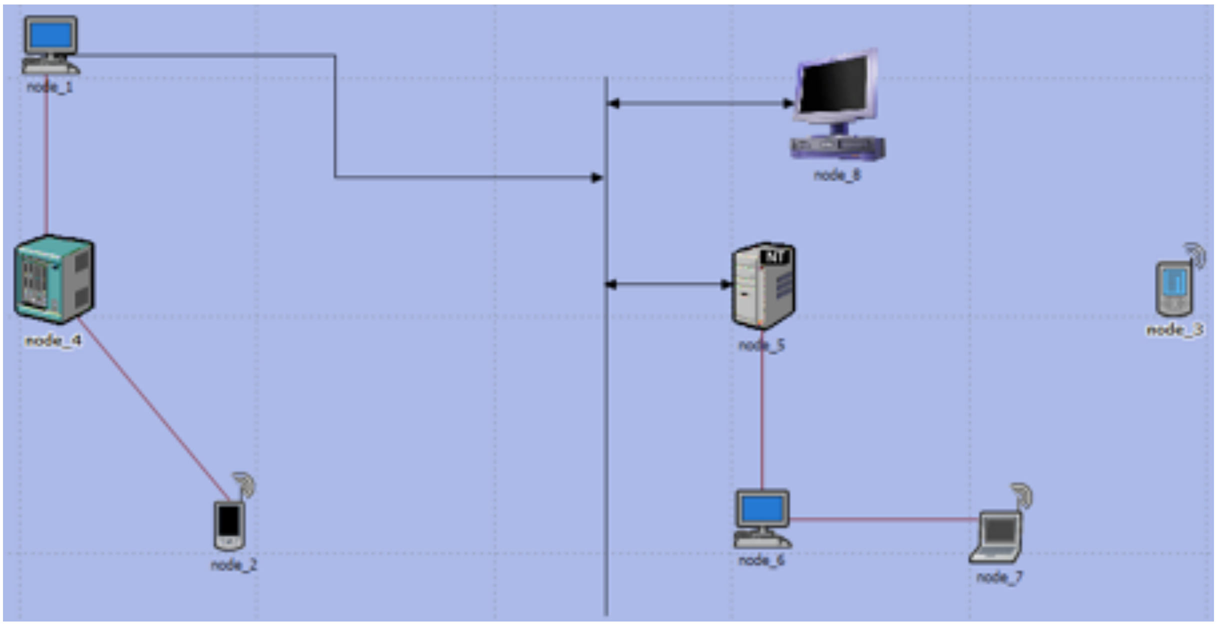


Figure 36 - Scenario with 8 Nodes.

V.2.2.3 Performance Test Network With 12 Nodes

Node	WiFi	PLC	Ethernet	Node	WiFi	PLC	Ethernet
Node 1		X	X	Node 7	X		X
Node 2	X			Node 8	X	X	
Node 3	X			Node 9		X	X
Node 4			X	Node 10	X		X
Node 5		X	X	Node 11	X	X	
Node 6			X	Node 12 - Gateway		X	X

Table 4 - Node Technologies in the 12 Node test network.

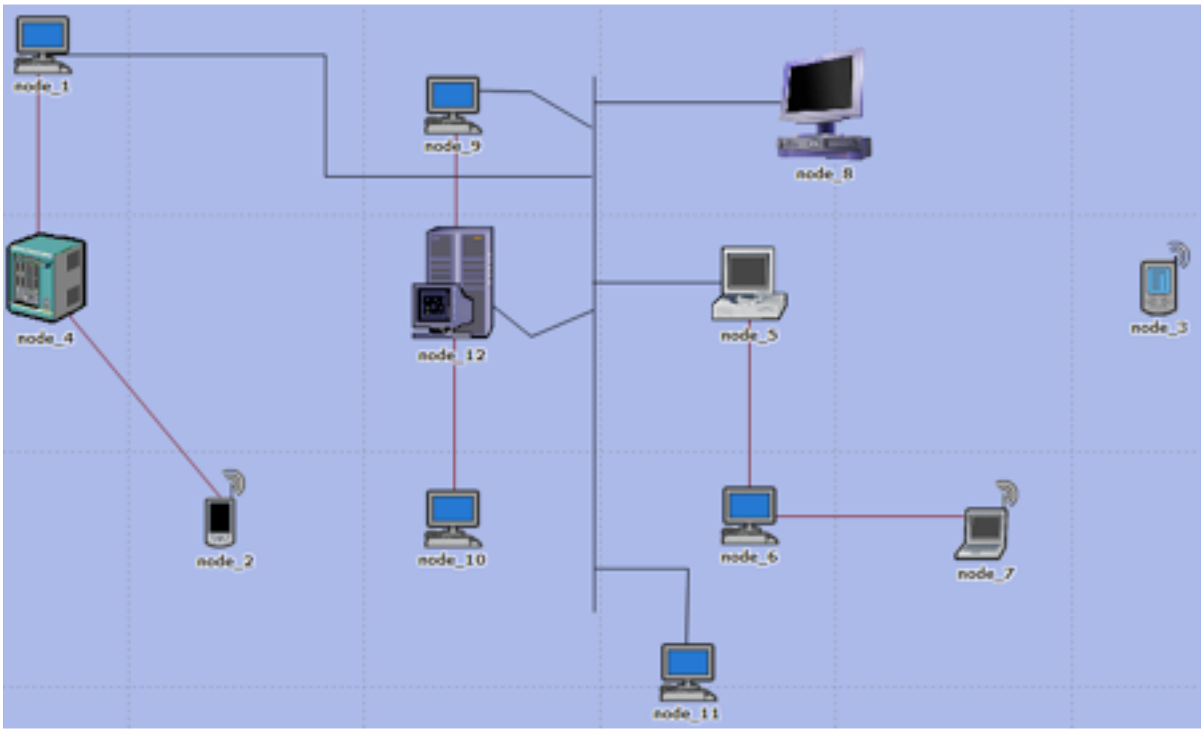


Figure 37 - Scenario with 12 Nodes.

V.2.2.4 Performance Test Network With 16 Nodes

Node	WiFi	PLC	Ethernet	Node	WiFi	PLC	Ethernet
Node 1		X	X	Node 9		X	X
Node 2	X		X	Node 10	X		X
Node 3	X			Node 11	X	X	X
Node 4			X	Node 12 - Gateway	X		X
Node 5		X	X	Node 13		X	X

Node 6			X	Node 14			X
Node 7		X	X	Node 15	X	X	
Node 8	X	X		Node 16	X	X	

Table 5 - Node Technologies in the 16 Node test network.

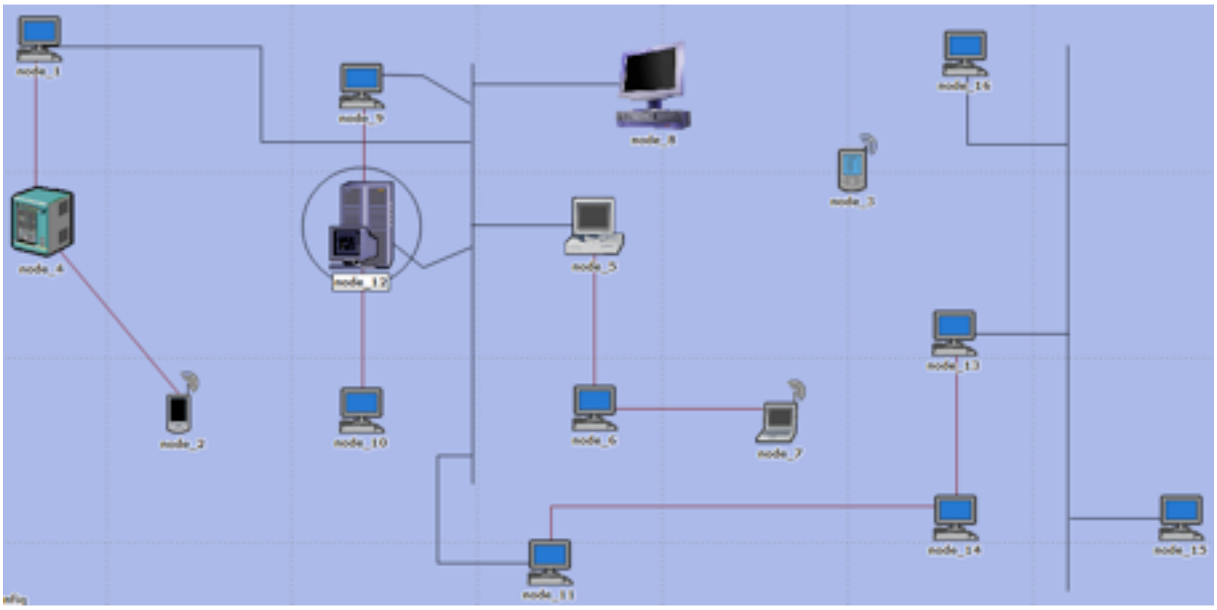


Figure 38 - Scenario with 16 Nodes.

V.2.2.5 Performance Test Network With 20 Nodes

Node	WiFi	PLC	Ethernet	Node	WiFi	PLC	Ethernet
Node 1		X	X	Node 11	X	X	X
Node 2	X		X	Node 12		X	X
Node 3	X			Node 13		X	X
Node 4			X	Node 14			X
Node 5		X	X	Node 15	X	X	
Node 6 - Gateway			X	Node 16	X	X	X
Node 7	X			Node 17	X	X	X
Node 8	X	X		Node 18	X		X
Node 9		X	X	Node 19		X	X
Node 10	X		X	Node 20	X		X

Table 6 - Node Technologies in the 20 Node test network.

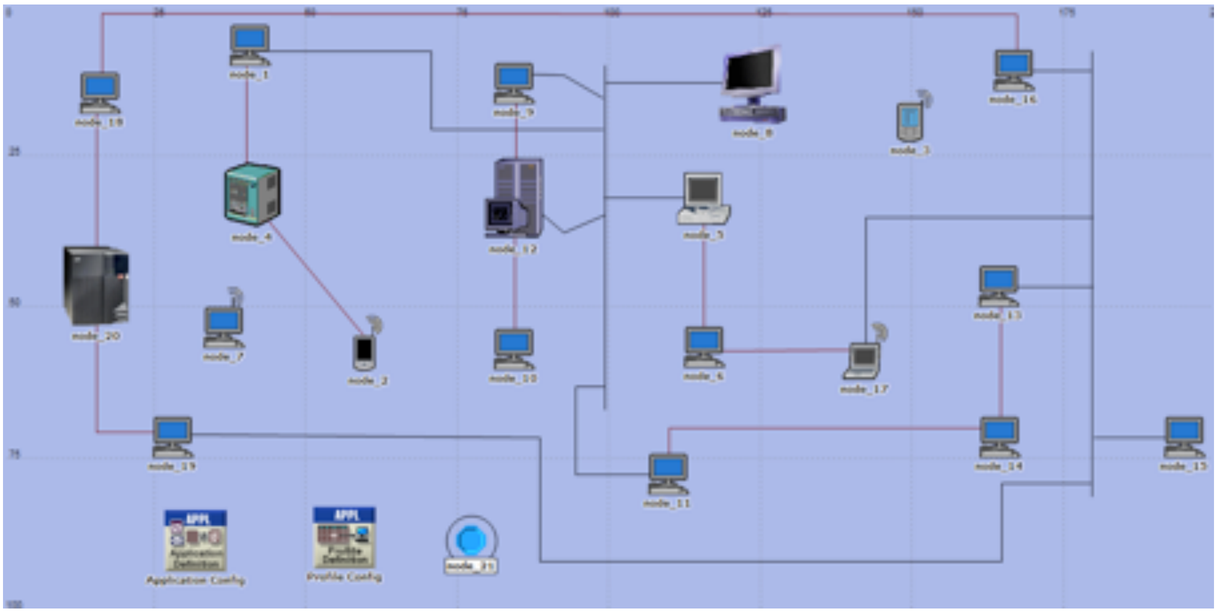


Figure 39 - Scenario with 20 Nodes.

V.2.3 Link Failure Test Network

Although the Link Failure response time was tested using the HWMP Performance Test Networks, the Link Failure Test Network was created with the idea to make it visually easier for the reader to see the outcome of the network topology when a link fails. The idea is to easily visualise the path initially taken by a flow and the path that the flow would have to take once a link is unavailable. Figure 40 shows the network topology implemented for this purpose.

Node	WiFi	PLC	Ethernet	Node	WiFi	PLC	Ethernet
Node 1	X		X	*Node 4*	X		X
Node 2			X	*Node 5*	X		X
Node 3			X	*Node 7 - Gateway*			X

Table 7 - Node Technologies in the Link Failure test network.

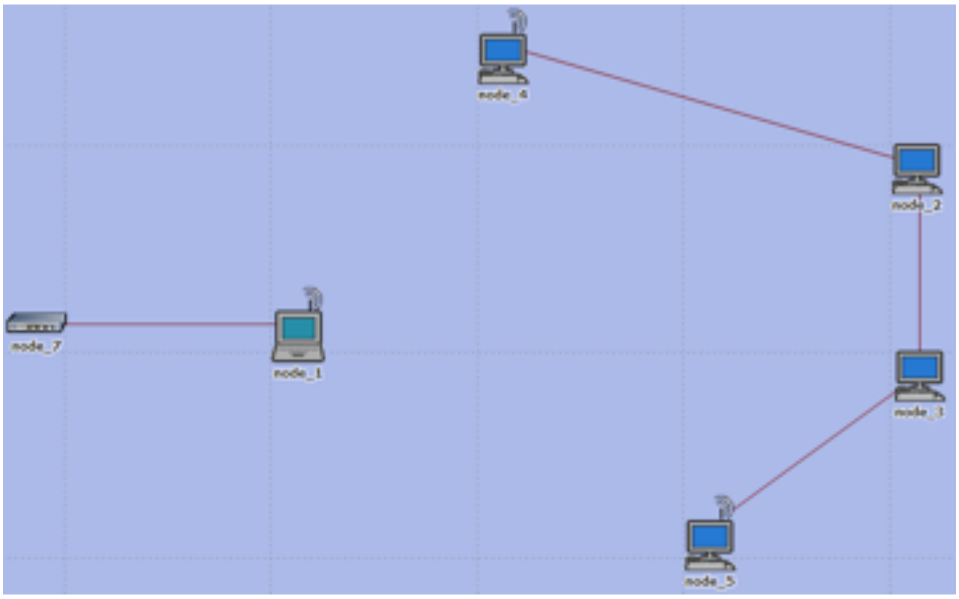


Figure 40 - Link Failure Scenario.

V.3 Simulations And Results

V.3.1 Path Set-Up And Convergence Time Test

The main purpose of this test was to measure convergence and path set-up times of both Proactive and Reactive modes in HWMP. In order to compare the results, the test was to be made using all five scenarios shown previously as HWMP Performance Test Networks which have 4, 8, 12, 16 and 20 nodes. For each scenario, two separate test were made, one with the Gateway node disabled (that is, using just the Reactive mode of HWMP) and one with the Gateway node enabled (that is, using both Proactive and Reactive modes of HWMP in order to measure the Proactive mode convergence time).

It is important to notice before going into the results that the times to be presented represent two different situations. In the first case, when using just the Reactive mode in HWMP, the time shown represents the time required by the protocol to find a path and notify the source node to start transmission of data, that is, it represents the Path Set-Up Time. In the second case, the time depicted represents the time required by the Proactive mode in HWMP to find a path to all nodes involved in the topology, that is, the Proactive Mode Protocol Convergence Time. In a Real-Life scenario, the Proactive tree would have been already set-up by the time the application starts, so the initial delay for the application is practically zero (taking only into consideration the delay caused by path set-up). Considering a hypothetic scenario in which the data flow wants to starts as soon as the device is turned on, the initial delay for the application would be that of the Proactive mode in HWMP.

The simulations considered one or two background flows that started around 5 seconds after simulation start. The main flow to which it was measured the Path Set-Up Time started 7 seconds after simulation start. The background flows were created in order to use around 60% of link capacity of a common link between background flows and main flow. The background flows were FTP traffic, requiring the transfer of files every 10ms with a file size dependent on the channel capacity. The main flow was a Video Conference requiring 10 Mbps of band.

V.3.1.1 Reactive Mode Path Set-Up Time

The following figure shows the results obtained when measuring the time required to establish a path using the Reactive mode in HWMP.

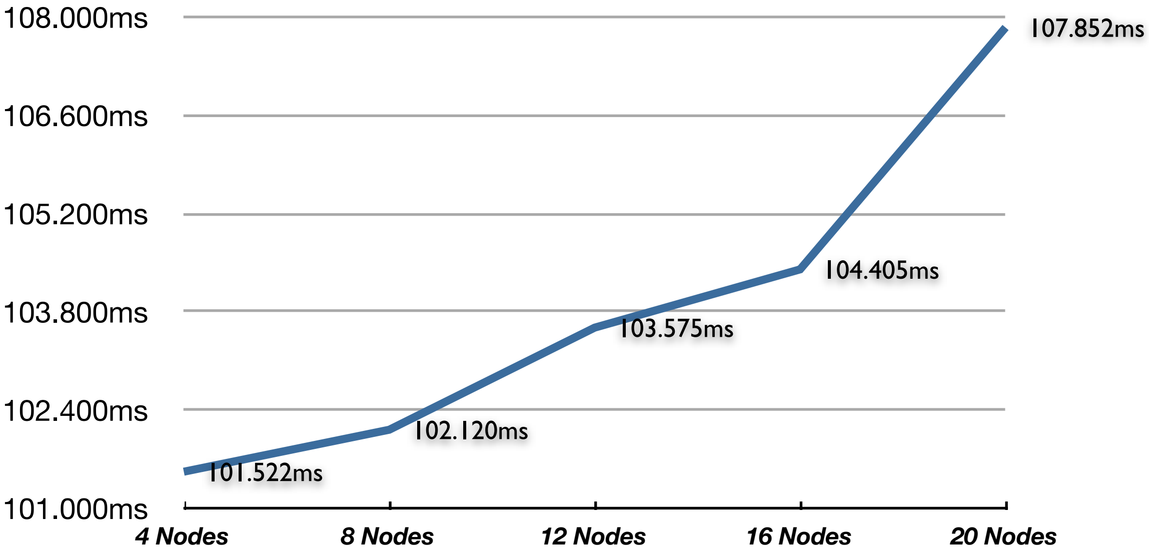


Figure 41 - Path Set-Up Time using the Reactive mode of HWMP.

The figure above shows that when using the Reactive mode of HWMP only, the initial delay to start the application would be just over 100ms. It is important to consider that this time is due to the fact that the official document on HWMP establishes a 100ms waiting time at the destination node for other PREQ packets to arrive, delaying the transmission of the PREP by that time. If the initial delay is to be reduced, it would be convenient to adapt the Wait Window to the network taken into consideration, in this case the OMEGA network. A possible solution would be to establish a relation between the number of nodes/number of effective links and the Wait Window time, so that smaller networks can have smaller delays.

V.3.1.2 Proactive Mode Protocol Convergence Time

After simulating the convergence time of the Proactive mode of HWMP, it was found that the necessary time to establish a path to every node grew on a linear manner, starting from 0.790ms in the scenario with 4 Nodes and ending at 6.997ms in the scenario with 20 Nodes. There are two facts that must be taken into account when analysing the result:

- The lower time required to find paths to every node in the network is due to the fact that in the Proactive mode of HWMP there is no Wait Window at the destination, meaning that when a PREQ arrives the node forwards it and immediately replies with a PREP packet back to the Gateway node.
- Since the Proactive mode starts right after simulation start time, there is no background flow set yet in the network, reducing the link delays and therefore convergence time.

Figure 42 shows the results obtained when measuring the time required by the Proactive mode in HWMP to converge.

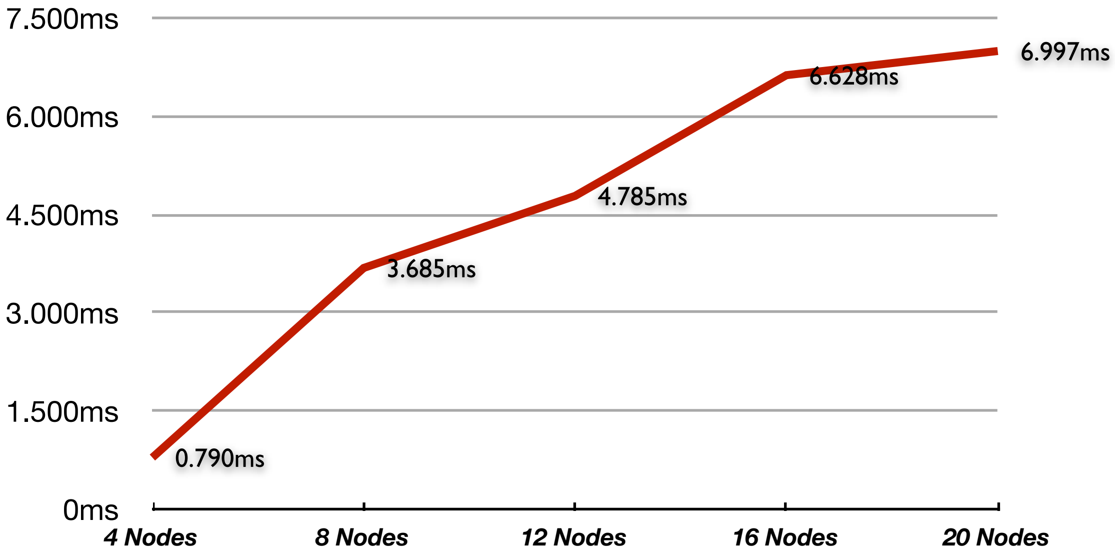


Figure 42 - Convergence Time using the Proactive mode of HWMP.

V.3.2 Protocol Overhead Of HWMP

The main purpose of this test was to measure the total control bits sent because of the use of HWMP. As done in the previous test, the scenarios used were those described as HWMP Performance Test Networks, which consist in five scenarios with 4, 8,12, 16 and 20 nodes. The test was separated into two main parts: The first considers only the Reactive mode in HWMP, while the second considers both Reactive and Proactive modes in HWMP.

The network was set-up to have no background flows and to have the main flow (Video Conferencing at 10 Mbps) start 7 seconds after simulation start time. The simulation was set to run for 60 seconds. Given the fact that the default path maintenance time is 2 seconds, the number of reactive attempts was 27, while the total number of hybrid attempts was 57 (30 for the proactive mode and 27 for the reactive mode).

The results that are about to be shown are only a representation of the performance of HWMP when used as the Path Selection protocol for the OMEGA network. These results can vary significantly depending on the network topology where it's implemented and since each node utilised in the simulation can have up to three technologies and therefore links, it is very hard to obtain a relation between the data. Instead, the idea is to judge if whether the results obtained satisfy or not the requirements set by the OMEGA network.

The first result presented in figure 43 is the Control Overhead. Here we can see the total number of bits sent because of PREQs and PREPs packets whether the Gateway was enabled (Hybrid mode) or not (Reactive mode only). It can be observed from the graphic that the values follow a non linear trend, incrementing the amount of control bits sent every time the number of nodes in the network are incremented. This is an obvious result since having more nodes in a network also means having more links where PREQs packets need to be forwarded. It is important to observe that this result was obtained considering only one flow in the network and no link failures. The inclusion of more flows would certainly increment the number of control packets since the total kbit sent in Reactive mode is multiplied times the number of flows in the network, thus incrementing the total overhead created.

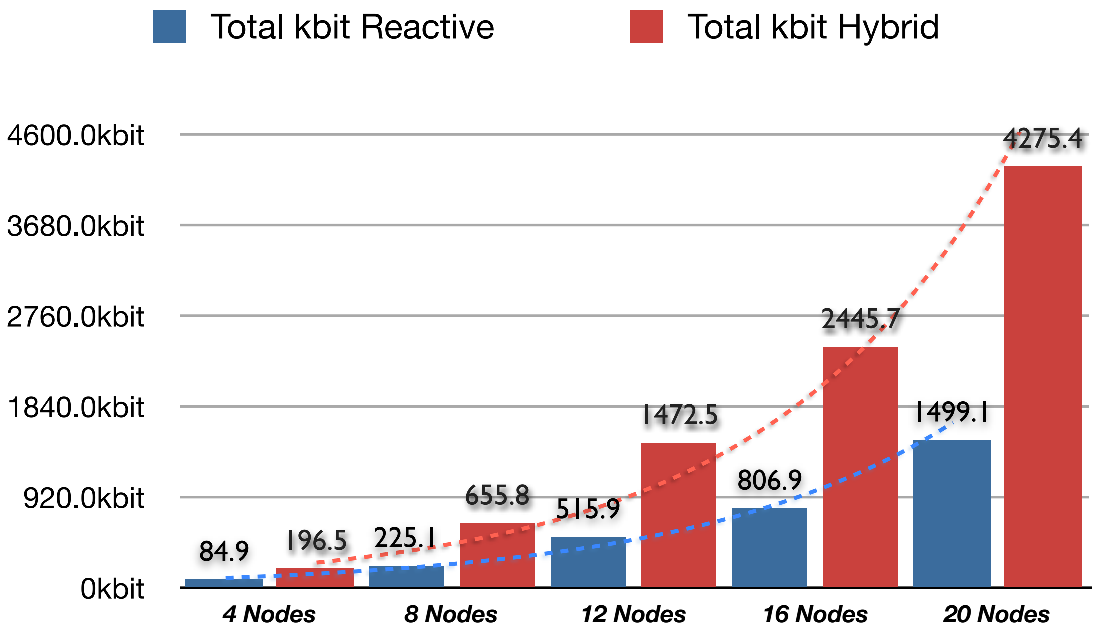


Figure 43 - Control overhead in kbit for the Reactive and Hybrid modes of HWMP.

The following results shown in figures 44, 45 and 46 were obtained from the Total Control Overhead values of figure 43. In figure 44 is shown the overall link utilisation in kbit/s for each scenario simulated. In blue is shown the overall overhead generated by a single flow requiring to set-up an optimum path. Being a reactive path request, the link load generated is carried in its majority by the links involved in the path that is being set-up. Instead, shown in red, is the overall link load generated when using both proactive and reactive modes of HWMP which contains the additional load caused by the proactive mode. Because the proactive mode control packets are sent in broadcast, the link load caused is "spread" throughout the network instead of influencing a particular set of links. The results shown put in evidence two important facts:

- First, the average overhead generated by HWMP (Reactive or Hybrid) increments as the number of nodes in the network increments. Moreover, the increment is more pronounced when considering the Hybrid mode since in this mode PPREQs and PPREPs are sent in broadcast.
- And second, the values obtained when put in relation to the OMEGA network are rather small and should not significantly influence the networks performance.

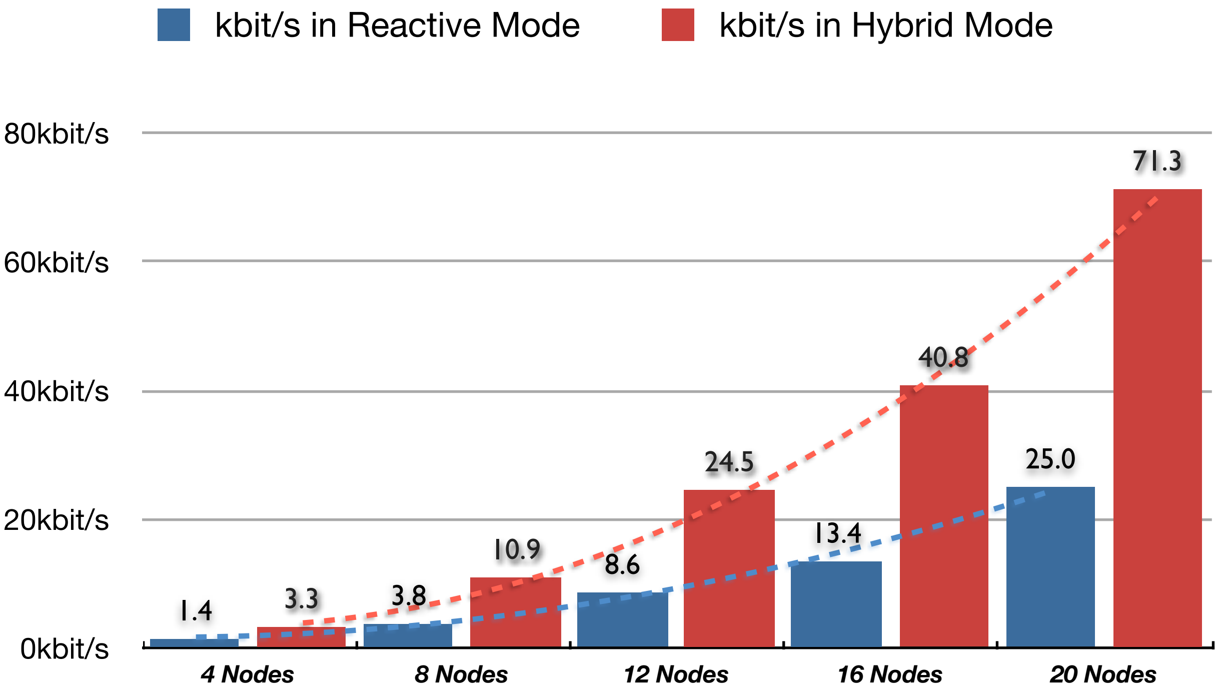


Figure 44 - Control overhead in kbit/s for the Reactive and Hybrid modes of HWMP.

The next result expresses the number of kbit sent due to control overhead for each time a path needs to be found. One attempt is defined as the initiation of a request to find a path in the network, this can happen either because a new flow needs to be established or because the flow has already been established and the path has to be maintained. Moreover, also considered as an attempt is the initial set-up and subsequent path maintenance of the Proactive tree in HWMP. As said before, the total attempts were 57 (30 for the proactive mode and 27 for the reactive mode). In figure 45 is shown the average per attempt control overhead of HWMP.

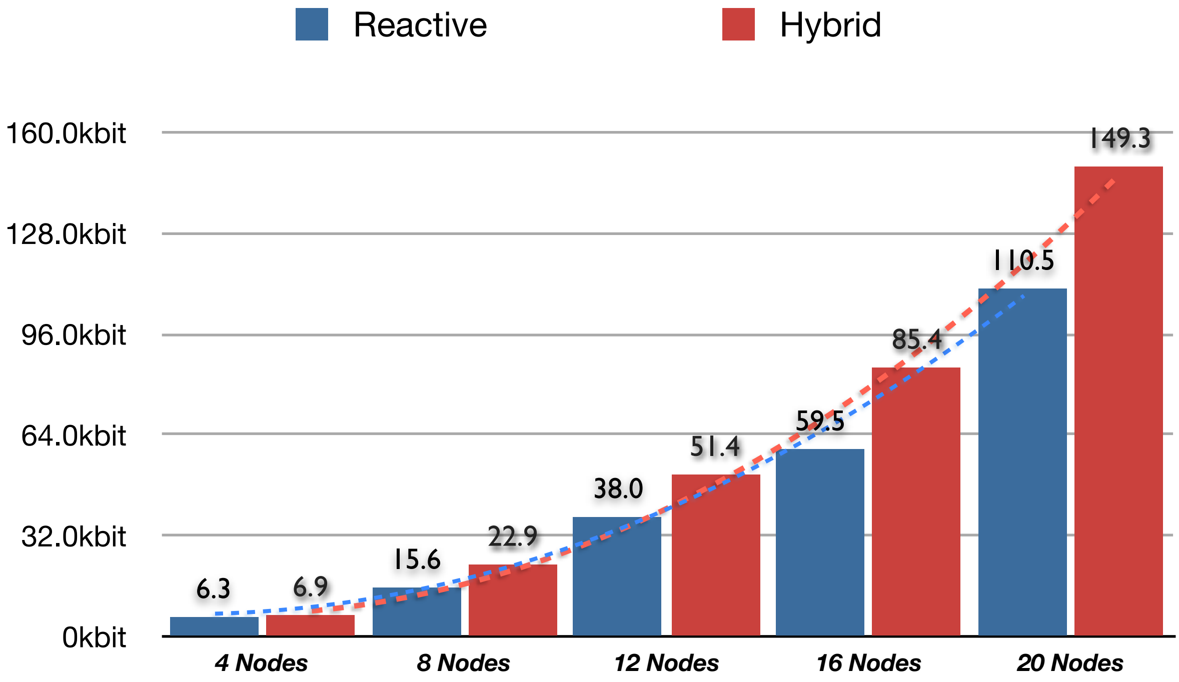


Figure 45 - Average per Attempt Control overhead in kbit of HWMP.

Last, a graph containing the number of control bits sent in average by each node after 60 seconds of simulation is presented. There are two aspects that are important to notice:

- The first is that the values follow a more linear trend. This is contrary to the trend seen when measuring the total bits sent throughout the network.
- And second, it can be observed that the majority of the overhead traffic is caused by the proactive mode in HWMP since PPREQs are sent in broadcast and influence every link in the network.

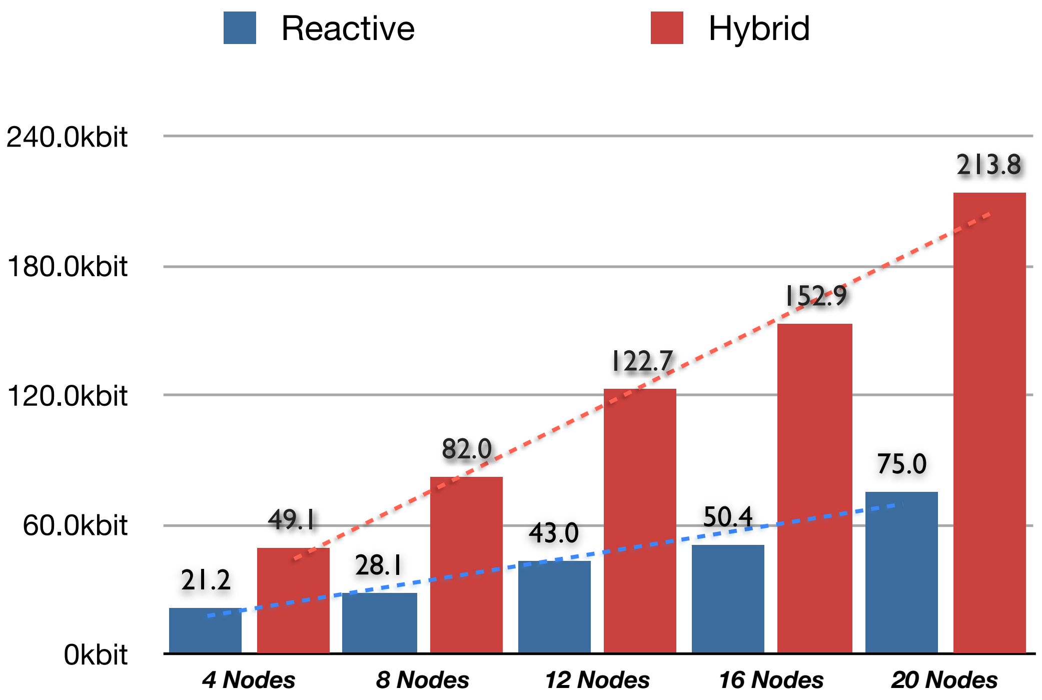


Figure 46 - Average per Node Control overhead in kbit of HWMP.

V.3.3 Link Failure Test

The Link Failure Response Time represents the time since a link goes down until the time a new path is been found where to readdress the data packets. Since link failures are common, especially in networks with wireless devices that can change their position constantly, it was important to measure the time needed by HWMP to respond to this event. The Link Failure Test was performed using a wide variety of scenarios:

- First, the test was conducted using all five HWMP Performance Test Networks. The networks were set-up to have no background flow and a main flow that started 7 seconds after simulation start time.
- Second, the Link Failure scenario was used configuring two flows that ran through the same Ethernet link that after failing had to readdressed using a wireless link.
- And third, the Family Home scenario was set-up to have 5 flows all running through the PLC link. This was done in order to be able to determine the link failure response time in a more realistic network scenario where packets suffer of additional delays.

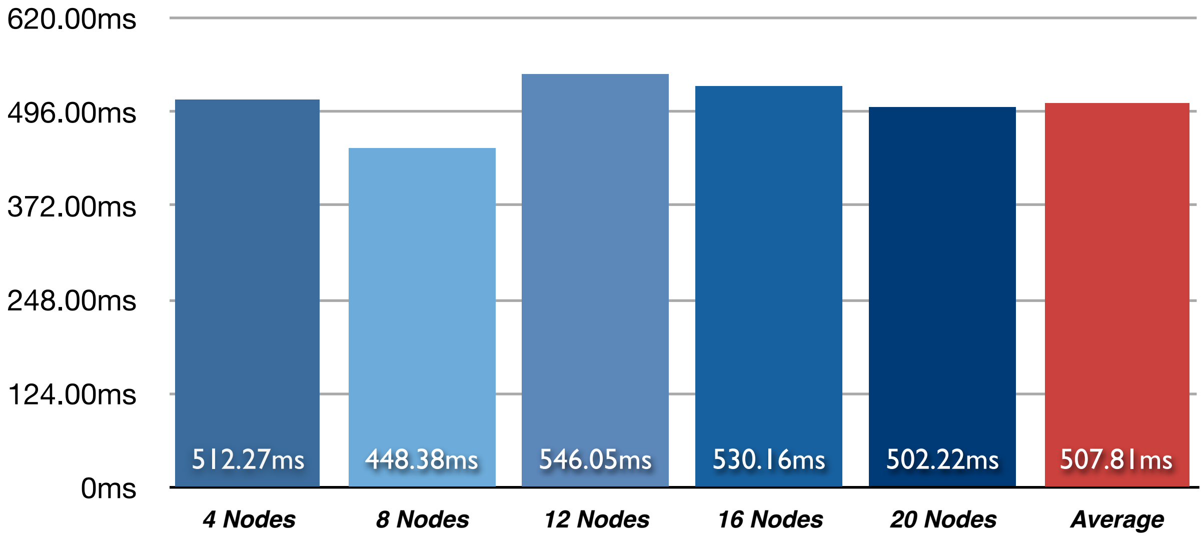


Figure 47 - Link Failure Response Time without Background Flows.

In the figure above, the Link Failure Response Times considering the HWMP Performance Test networks are shown. As can be seen the time needed to readdress data packets after a link has failed had an average time of 507ms when considering no background flows. Moreover, there seemed to be no relation between the number of nodes in the network and the response time. Instead, the difference seen can be attributed to the choice of the Source node, Destination node and the positioning in the network of the failed link.

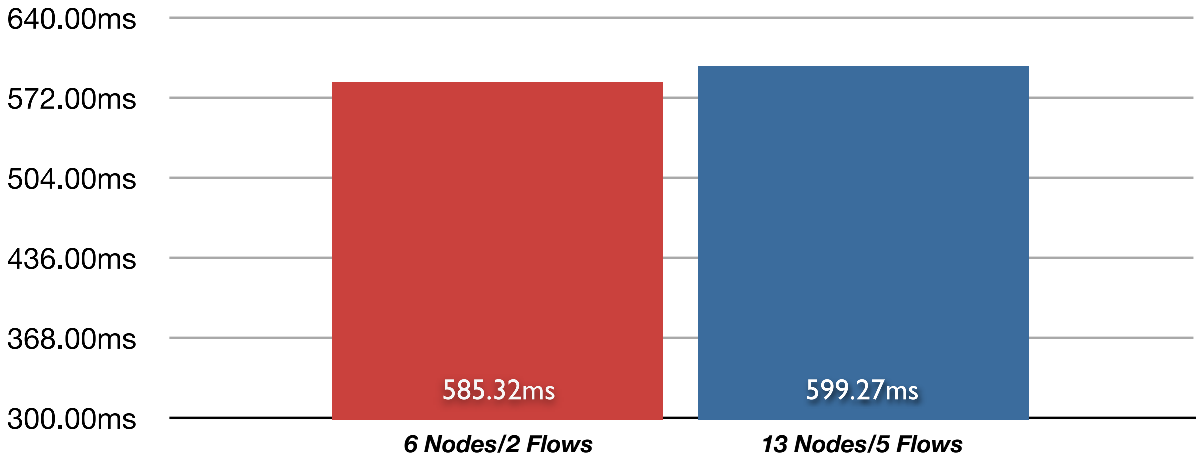


Figure 48 - Link Failure Response Time with Background Flows.

After measuring the response time with networks that carried no background load, further testing was done on two more networks which had background flows on them. First, the Family Home scenario was set-up with five flows utilising overall 50% of the PLC link. After 60 seconds of simulation time, the link between the Game Console and the Home TV failed, causing the flow that was running through it to be readdressed using the wireless link of node 13 with destination Room TV. Even after incrementing channel utilisation and therefore queuing delay, the response time remained under 600ms. This is an important result since it provides an approximate upper limit on the Link Failure Response Time given the fact that this scenario is intended to represent more accurately a Real-Life OMEGA network. Furthermore, the response time found during simulations was always in the range varying from 500ms to 600ms, the top range belonging to that found in the 13 node network with 5 background flows.

Second, the result for the Link Failure Test network showed a similar result with a Link Failure Response Time of 585ms when considering 2 background flows. The scenario was set with a flow going from source node 7 to destination node 2 and another flow going from source node 5 to destination node 4. Figure 49 shows how data is readdress in the Link Failure Test network after the Ethernet link between the nodes 4 and node 2 failed.

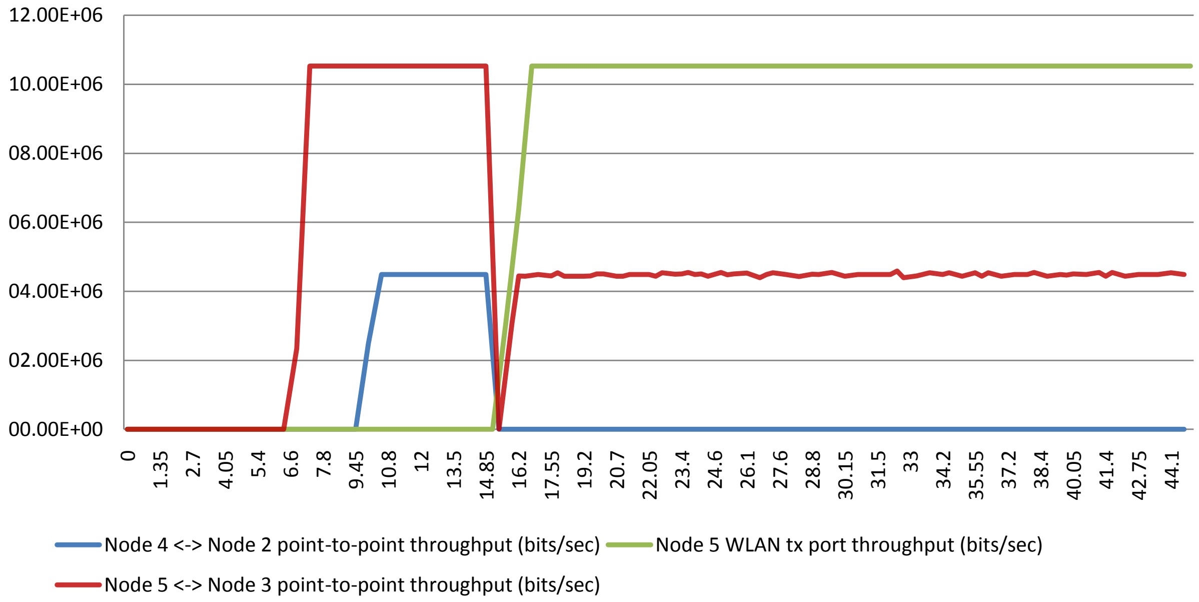


Figure 49 - Link throughput in bits/s before and after the link has failed.

Before the link failed, one flow followed this path: node 7 → 1 → 4 → 2 while the other follows: Node 5 → 3 → 2 → 4. As it can be seen, once the link between node 4 and node 2 fails, the first flow has to to be readdress using the bottom part of the network following this path: Node 7 → 5 → 3 → 2 while the second flow has to take by force the direct wireless link path Node 5 → Node 4. Figure 49 shows the paths pre and post link failure. Below in figure 50 are shown the queuing delays for two of the links involved in the test.

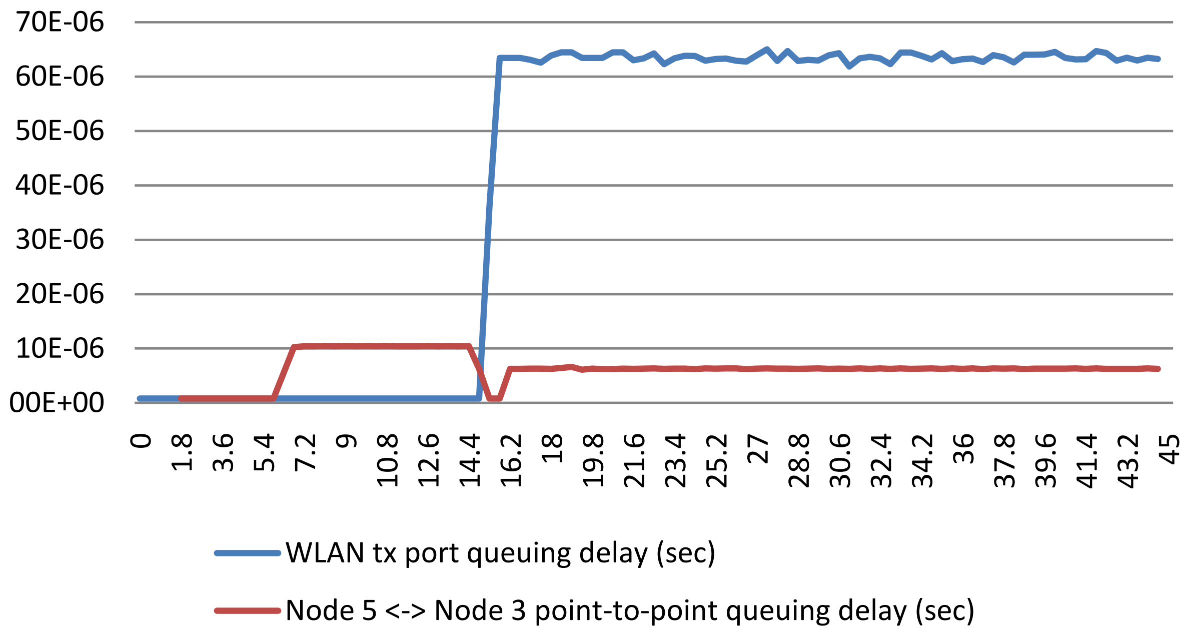


Figure 50 - Link delays before and after the link failure.

V.3.4 Family Home Network Test

The purpose of the previous tests was to obtain general data about the behaviour of HWMP if implemented on an heterogeneous network such as OMEGA. Now, to end this series of simulations and to prove the value of the results obtained in all previous tests, further simulations were required to be performed on a more realistic scenario. To this end, several tests were made on the Family Home scenario. This tests included measuring the protocol overhead, link failure response time and a new test, the time required by a node to find a new path once it has changed its position from one room to another.

V.3.4.1 Protocol Overhead On The Family Home Network

The Family Home network was set-up to have five flows. All these utilised the PLC line inside the Family Home scenario and comprehended Video Conferencing (2), Video Streaming, Printing and an FTP file download. The simulation time was set to 60 seconds in both Reactive only and Hybrid modes of HWMP. The flows are described in the following table:

	Source	Destination	Start time (sec)	Type
Flow 1	Home TV	Room TV	7	Video Streaming
Flow 2	Smartphone	Smartphone3	15	Video Conference
Flow 3	Smartphone2	Smartphone3	25	Video Conference
Flow 4	Home TV	Printer	45	File Print
Flow 5	Node 4	Game Console	55	File Download

Table 8 - List of flows in the Family Home scenario.

The values seen in figure 51 are higher than those observed in the HWMP Performance Test networks due to the fact that this network carried five flows instead of one. Moreover, since the intention of the Family Home scenario is to represent a typical Home Network, it is composed of fewer links where control packets need to be retransmitted, reducing the link load caused by HWMP.

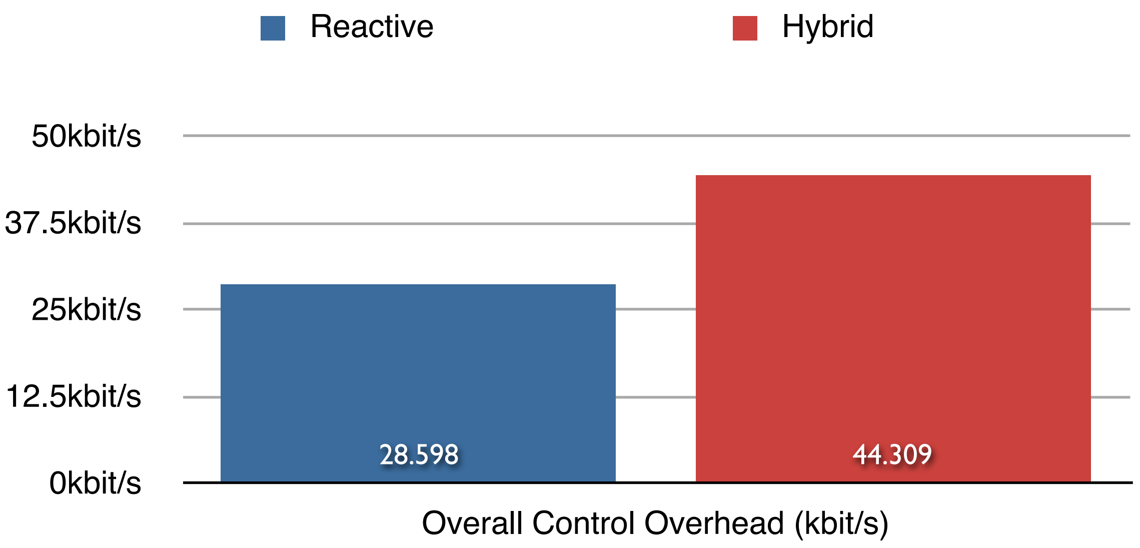


Figure 51 - Control overhead in kbit/s for the Reactive and Hybrid modes for the Family Home Network.

Although this network presented an overall higher control overhead per second than in previous tests with a similar scenario, the average per attempt control packet kbit was lower, meaning that every time a path needed to be found (or maintained) fewer retransmissions took place. This fact indicates that the Family Home network despite the fact that it has 13 nodes, contained fewer interfaces where to forward packets. Hence, in order to be able to establish a relation between the number of bits sent and the number of nodes, one must also consider the number of interfaces each node has since in the OMEGA network simulated every node can have up to N interfaces (N being the number of technologies implemented), further incrementing the complexity. Figure 52 shows the average per attempt control overhead.

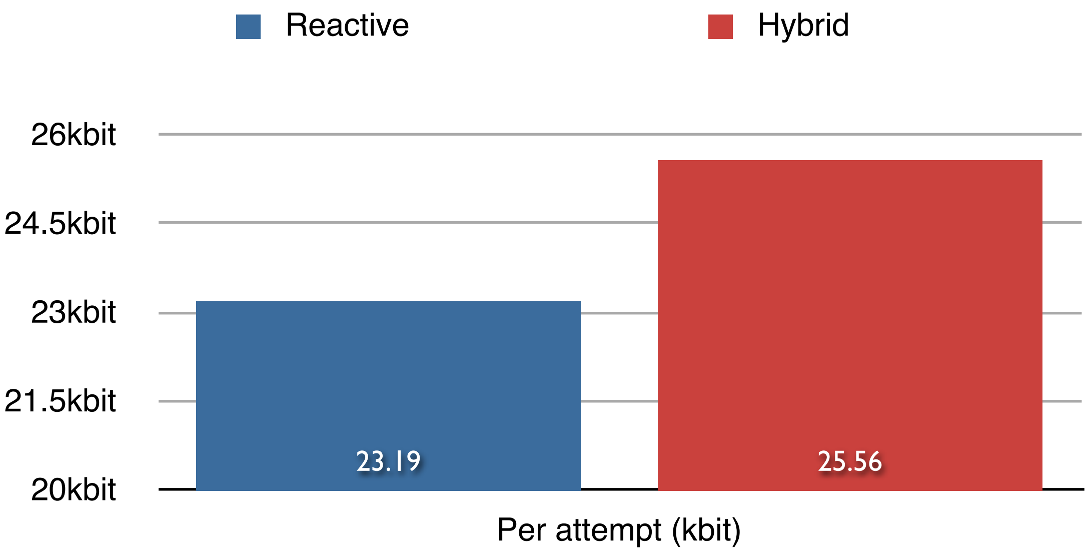


Figure 52 - Average per Attempt Control overhead in kbit for the Family Home Network.

The following figures show some of the data gathered from the Home Family network:

- Figure 53: Video called party packet delay variation.
- Figure 54: Video called party packet end-to-end delay.
- Figure 55: Video conferencing traffic received by smartphone3.
- Figure 56: PLC throughput.
- Figure 57: PLC utilisation percentage.

Figure 53 shows how smartphone3 starts receiving data from smartphone at second 15. At second 25 the second smartphone (smartphone2) joins the video conference, incrementing the total traffic received to about 1 MBps at smartphone3.

Figures 56 and 57 show how the PLC line starts receiving traffic after each flow has begun transmission. In the end, PLC utilisation resulted at around 50%.

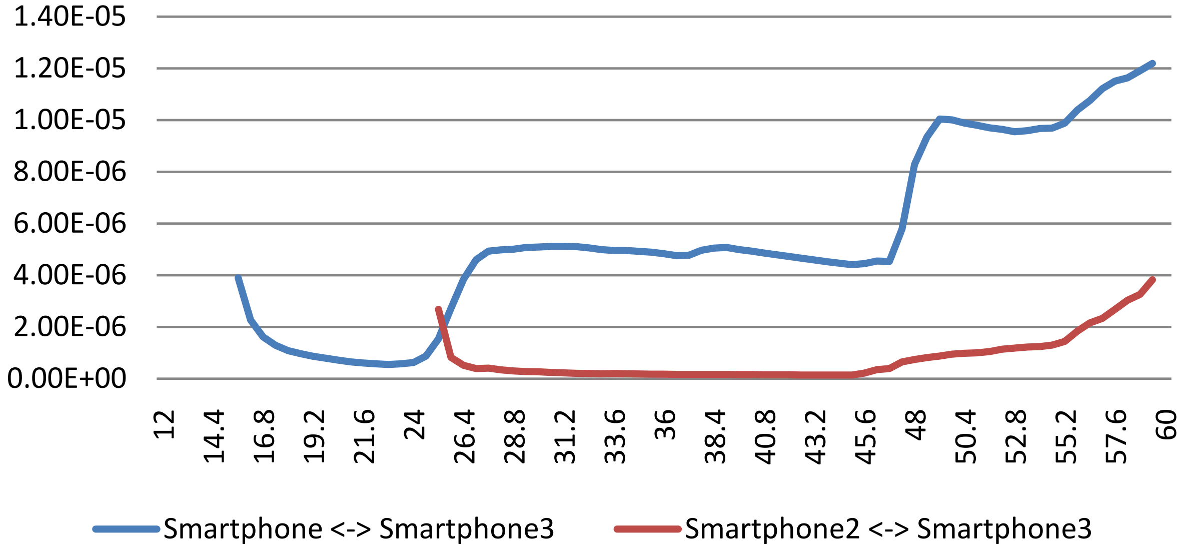


Figure 53 - Video Called Party Packet Delay Variation.

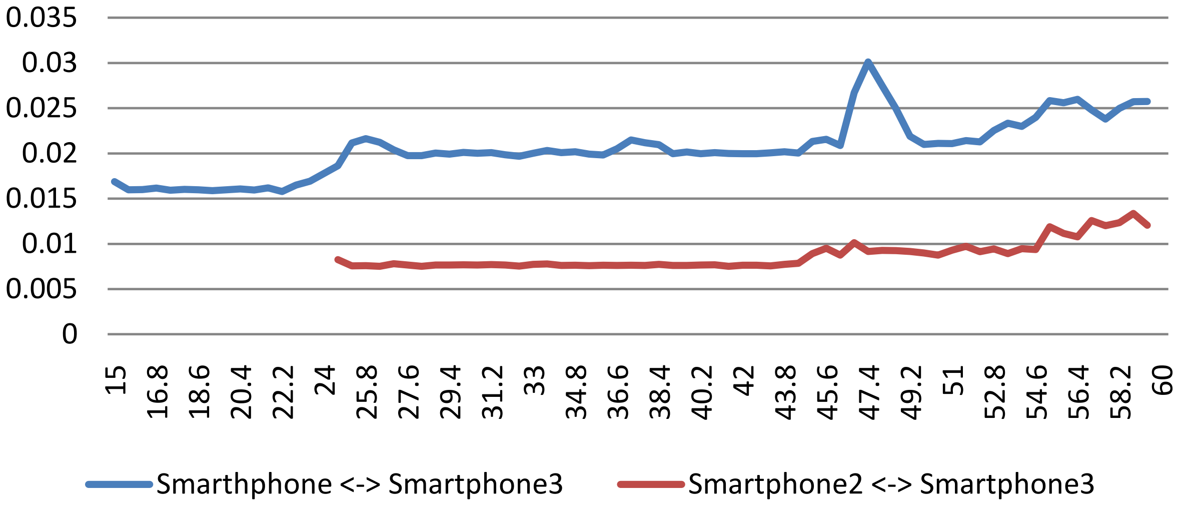


Figure 54 - Video Called Party Packet End-To-End Delay.

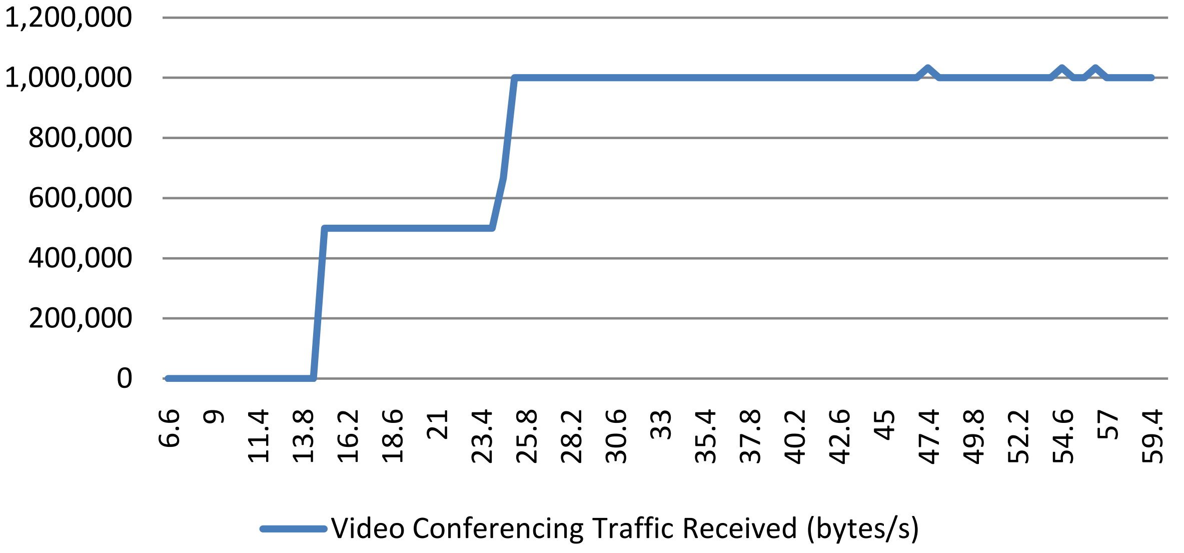


Figure 55 - Video Conferencing Traffic Received by Smartphone3.

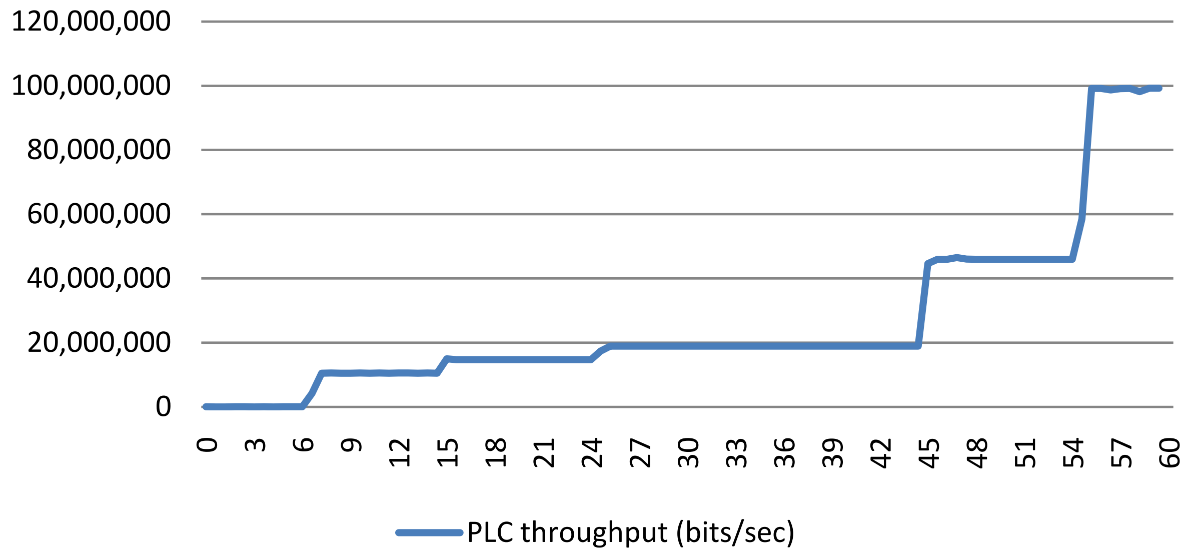


Figure 56 - PLC throughput.

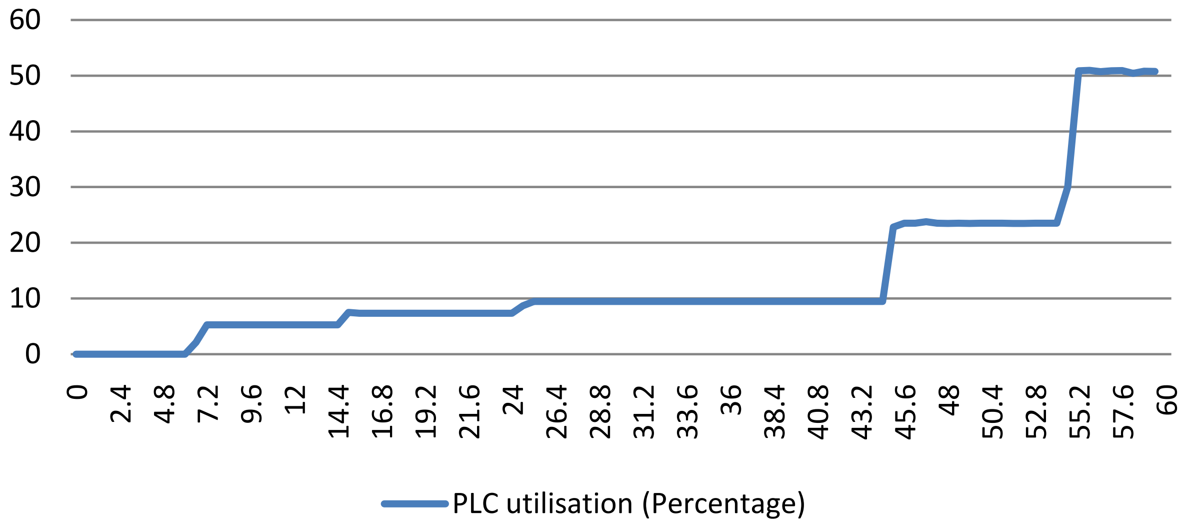


Figure 57 - PLC utilisation percentage.

V.3.4.2 Link Failure Response Time On The Family Home Network

The Link Failure Response Time was measured in this network failing the link between the Game Console and the Home TV. This result was already presented in section V.3.3, resulting in a Response Time of just under 600ms with a loaded network. For further referencing see section V.3.3.

V.3.4.3 Path Recovery Time After Node Changes Position

Since in the OMEGA network mobility is a key issue, it was necessary to measure the time required by HWMP to find a new path for a flow once the node involved had changed its position. To enlighten the situation here is what happens: A user (Named Steve) is sitting in his room (Room 2) and requests another family member (Named Lisa) who is in Room 1 to start a video conference. The conversation starts almost immediately as the initial delay in HWMP is practically zero. Steve and Lisa continue talking normally and suddenly Steve is called by his father from the Living Room. He goes there while maintaining a conversation with Lisa. After talking to his dad, he goes back to his room (Room 2) and continues talking to Lisa on the phone. This situation is presented in figure 58, where the trajectory followed by Smartphone 2 can be seen (the smartphone belonging to Steve), while Lisa's position remains unchanged (Smartphone 3).

The initial path taken by the Video Conference was: Smartphone 2 → Laptop → Printer → Node 4 → Room TV → Smartphone 3. After Steve goes to the Living Room, his smartphone no longer has covering range of the Laptop to which he was connected, so in order to maintain the communication a new path had to be found. While in the Living Room, the path taken by the Video Conference was: Smartphone 2 → Home TV → NAS → OMEGA Gateway → Node 4 → Room TV → Smartphone 3. One might wonder why HWMP chose this path instead of going directly to the PLC line by using Node 13. This was done on purpose in order to create a more complex path back to the destination by assigning the WiFi nodes in the Living Room different BSS numbers than that for Smartphone 2 except for Home TV which shared the same BBS number with Steve's smartphone.

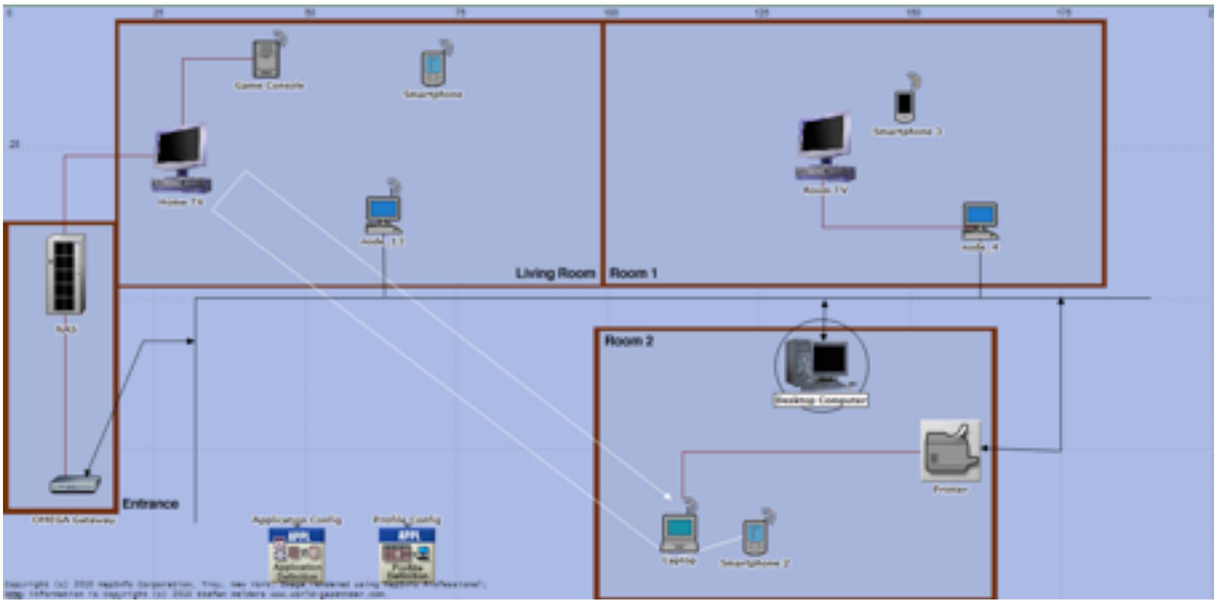


Figure 58 - Family Home scenario with node changing its position.

Since Steve changes its position two times, two Link Failures and therefore two PREQs are generated in order to find a new path back to Lisa's smartphone. The Response Times measured for this kind of Link Failure is shown on figure 59.

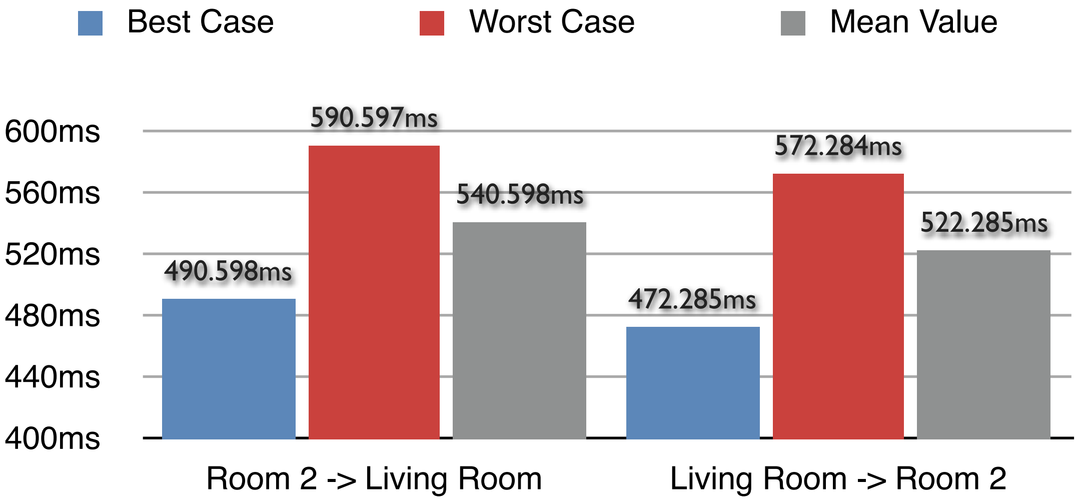


Figure 59 - Best and Worst Case response times of HWMP once a node changes position.

As can be seen from the figure above, the response times for a node's displacement are similar to those found when a link failed in the network. Taking the mean value as a reference and given the relatively small times where there is no path between source and destination, these should cause a minimum perception to the users involved in the Video Conference. Nevertheless, the times obtained from the simulation could be reduced by 100ms if the Wait Window at the destination node is removed in case of a link error (with the possible effect of not choosing the optimum path until the next path maintenance), and by another 50ms is the probes frequency is reduced by half (incrementing the overhead as a downside). Notice that in this simulation the exact time when the link effectively failed was unavailable (contrary to generated link failures where the OPNET user can specify the exact time of failure). In order to estimate the Link Failure Response Time the arrival time of the last probe had to be registered, meaning that either the link failed immediately after (Worst Case) or the link failed exactly 100ms after (Best Case).

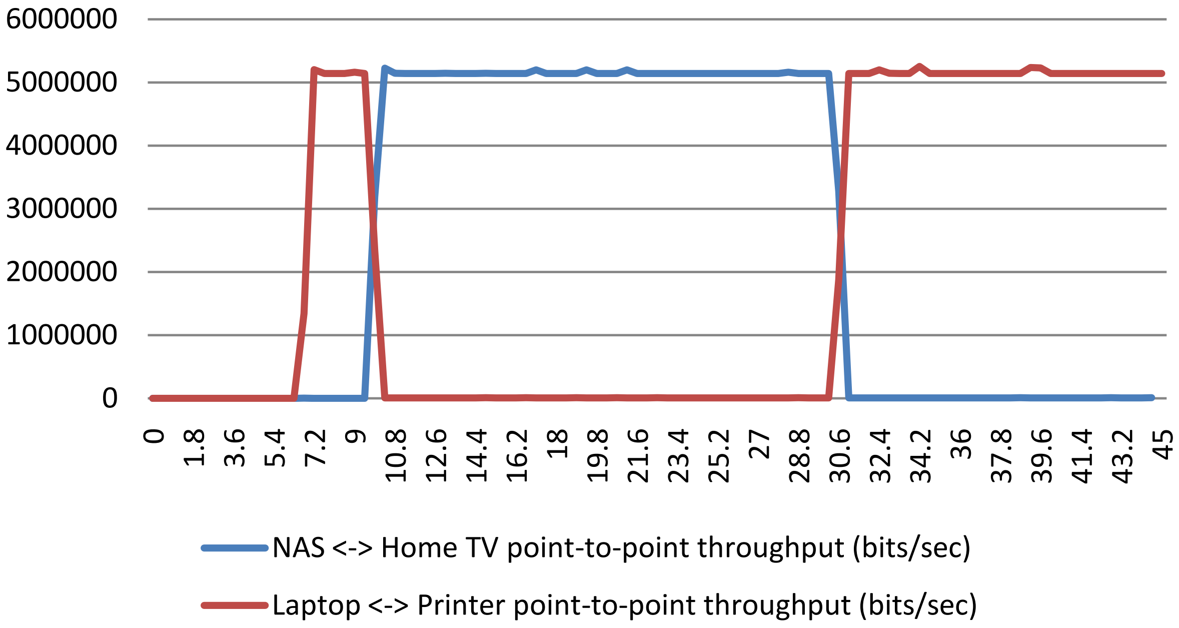


Figure 60 - Link throughput when a node changes position.

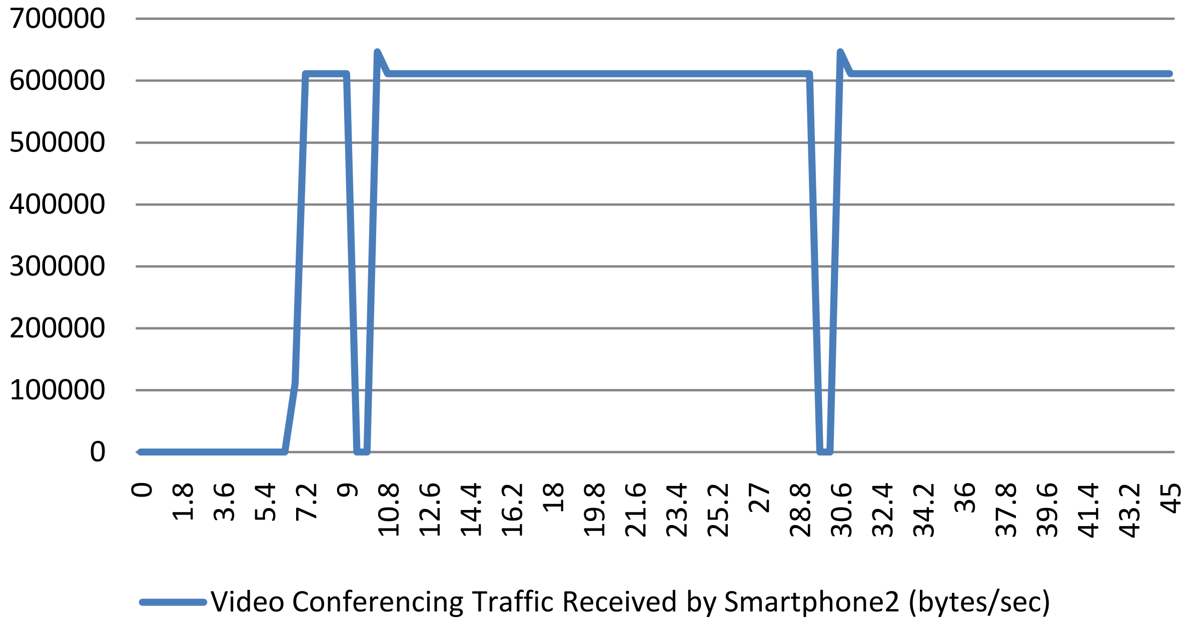


Figure 61 - Video Conferencing Traffic Received by Smartphone2.

In figure 60 shown above, the readdressing of the flow is evident as initially the flow utilises the Laptop → Printer link, then after Steve has gone to the Living Room data is readdressed using the Home TV → NAS link. Once he comes back to his room (Room 2) the path taken by the flow returns to that it originally had. Further, figure 61 shows how Video Conferencing Traffic is not received for a small amount of time coinciding with Steve going to and coming back from the Living Room.

Conclusions

Nowadays, people live in a fully connected environment from a technological point of view. It is hard to imagine a day in which we don't access the internet to check our emails, our most popular social network, to read the latest news, to search for the address or to many other basic services offered by the internet that have become part of our life. Years ago cellphones were just that, a mobile equipment used to make phone calls while outside our homes. Today, it has become standard the implementation of some sort of network connection in our phones like WiFi and/or Bluetooth. Also, nowadays we tend to use our phones for many other uses such as working with documents, videoconferencing, email, gaming, internet browsing, etc. What's more, many other devices have started to integrate either wireless or wired connections, devices such as our music players, the HDTV sets, printers and the list goes on. Hence, it is not hard to imagine a near future in which there are so many network capable devices in our hands that a home with a network connection that was accessed before by a single element, is now being accessed by a much larger number of devices. As user requirements increase, so do the speeds that internet service provider offers. In the last few years, we have seen the increased reach of broadband services to our homes with the use of DSL and Cable connections, and lately with the use of FTTH. With connection speeds that continue to rise it's reasonable to ask the following question: Can Home Networks keep up with the increasing bandwidth provided by the internet companies? This question was answered as the Project OMEGA. OMEGA's solution was to allow heterogeneous devices to connect using Gbps capacity, creating a user-friendly, easy to set-up and maintain fully connected Home Network without the need for major interventions to our home's infrastructure. To allow the use multiple heterogeneous technologies OMEGA presented the Inter-MAC layer, a layer created with the purpose in mind of allowing different technologies to coexist in the same network, incrementing both access speeds and robustness.

This solution presented as OMEGA allows the bandwidth inside our home network to match or even surpass that offered by internet service providers, ending the bottleneck created by home networks as known today. The next step that must be brought to our attention relates to the fact that having such a network as OMEGA and considering the always increasing number of network capable devices in our homes, there is a high chance that, contrary to the use of today's home networks, a high percentage of traffic will stay inside OMEGA. So by now we have an heterogeneous home meshed network with Gbps capacity but considering the increasing number of devices that will use it connecting between them independently, and considering the new services that will become available for this network it was necessary to ask a second question: how will OMEGA manage the increasing number of devices that connect to it and how will it manage and satisfy their QoS requirements? This thesis proposed as an answer to these questions the use of an adaptation of the *Hybrid Wireless Mesh Protocol* as the Path Selection protocol inside OMEGA. The work of this thesis consisted in implementing HWMP in OPNET Modeler, adapting it to work together with the Inter-MAC layer and previously created modules, and establishing the link metrics used and the path selection criteria based on these metrics.

Implementing HWMP in OPNET Modeler presented several challenges both from a programming point of view and from a design point of view. It was necessary to adapt and work together with the already implemented engines in order to complete the Inter-MAC architecture. Moreover, design issues such as the cooperation between reactive and proactive modes in HWMP, the metrics considered, their propagation and the ultimate decision when selecting a path was defined. The code that implements HWMP in OPNET Modeler required serious debugging with different scenarios in order to find specific scenarios in which the protocol did not work as expected.

After this debugging pre-tests, several scenarios were created in order to obtain the maximum amount of data in relation to the performance of HWMP when implemented inside an OMEGA network. These results established key aspects for any Path Selection protocol such as convergence time and path set-up time, protocol overhead generated, response time in case of link failures and in case of devices moving. Also, these results demonstrated the capability of HWMP to reduce to a minimum the initial delay for a flow while still maintaining an optimum path which is one of the main benefits of HWMP. Regarding the overhead introduced by HWMP it can be said that, although it was shown how it increases when increasing the number of nodes, it did not show values that would significantly affect the OMEGA network's performance. On the other hand, it was shown how the response time in case of a link failure was always around 500ms, independent on the number of nodes in the network and only slightly dependent on the link capacity used by background flows. Since OMEGA will certainly need to handle several mobile devices moving from room to room, a very important result was to measure the time needed to readdress a flow once a node changed its position. This time, as with the link failure response time, presented a mean value of just over 500ms. Further, depending on the path decision criteria chosen, it could be unnecessary to have a wait window for more PREQs at the destination (which is by default 100ms). Eliminating completely the wait window especially after a link failure was detected could help reduce Path Error Response Time by 100ms, reducing the amount of packets lost. Moreover, in case of low mobility it could be wise to increase the number of probe frames sent by the monitoring engine (to every 50ms for example). This fact would allow nodes to regain a path once they move from a room to another in just over 400ms, with the disadvantage of increasing control overhead and detecting false link failures.

One of the most important aspects of this implementation is that it allows us to measure the performance of HWMP on heterogeneous networks, independently of whether it's a home network or not. Moreover, since the measure of protocol overhead did not take into account the encapsulation created by the Inter-MAC layer, one could easily use this implementation to gather information about the behaviour of this protocol in non heterogeneous networks. Implementing this protocol with the OMEGA network in mind allowed also to have a better idea on how to adjust its parameters so that it could work in the most efficient way. Luckily, there is so much information that can be gathered from this implementation, leaving a lot of room for future investigations. For example, it would be interesting to obtain a relation between the number of nodes and effective links (due to the heterogeneous nature of OMEGA) and the performance obtained so that the network could learn about itself and optimise its parameters depending on whether OMEGA is installed in an apartment with one room or in a house with two floors and 6 rooms. Lastly, future works could implement differently how the ultimate decision about selecting a path is taken. This is because while the implementation created for this thesis only considered one option, there are many that could eventually be implemented or improved such as selecting the path that uses the least percentage band in order to reduce queuing delays.

References

[1] Suraci, V., Delli Priscoli, F., Castrucci, M., Tamea, G., De Vecchis, W. and Di Pilla, G., *"Inter-MAC: Convergence at MAC layer in Home Gigabit Network"*, ICT Mobile Summit, Stockholm, 2008.

[2] Suraci, V., Tamea, G., Castrucci, M., Meyer, T., Pablos, R., Purón, D., Novak, S. and Jennen, R., *"Inter-MAC architecture design"* available at http://www.ict-omega.eu/publications/deliverables.html.

[3] *"OMEGA Project Flyer"*, available at www.ict-omega.eu, January 2008.

[4] Bellec, M., Javaudin, J., Foglar, A., Hoffmann, O., Jaffré, P. and Isson, O., White Paper: "*Inter-MAC concept for Gbps Home Network*", April 2009.

[5] Hiebel, S., Maaser, M., Nowak, S. and Meddour, D., *"OMEGA Common Semantic"*, available at the URL www.ict-omega.eu/publications/deliverables.html, November 2008.

[6] Castrucci, M., Jaffré, P., Tamea, G., Lucidi, D., Liberatore, C., Bahr, M., King, M.J., Rebering, S., Ortner, A., Kortebi, A., Meyer, T., Drakul, S. and Mijic, G., *"OMEGA Architecture Model"*, available at the URL www.ict-omega.eu/publications/deliverables.html, December 2008.

[7] Siaud, I., Roblot, S., Hamon, M., Caldera, P., Max, S., Choi, C., Grass, E., Helard, J., Stephan, A., Tonello, A., Bellin, M. and Hoffman, O., *"OMEGA Technical requirements"*, available at the URL: http://www.ict-omega.eu/fileadmin/documents/deliverables/OMEGA_D2.1_v1.2.pdf.

[8] OMEGA Press release, *"Gigabit speed in all rooms without cable clutter"*, January 2008.

[9] Bahr, M., *"Proposed Routing for IEEE 802.11s WLAN Mesh Networks"*, Siemens Corporate Technology, Information & Communications, 2006.

[10] Bahr, M., *"Update on the Hybrid Wireless Mesh Protocol of IEEE 802.11s"*, Siemens Corporate Technology, Information & Communications, 2007.

[11] 802.11 Working Group, *"IEEE P802.11s/D3.04 Draft STANDARD for Information Technology, Telecommunications and information exchange between systems, Local and metropolitan area networks Specific requirements"*, October 2009.

[12] Yoon, W., Chung, S., Lee, S., and Lee, Y., *"An Efficient Cooperation of On-demand and Proactive Modes in Hybrid Wireless Mesh Protocol"*, 33rd IEEE Conference on Local Computer Networks, 2008. LCN 2008. October, 2008.

[13] Hossain, E., Leung, K., (2008) *"Wireless Mesh Networks: Architectures and Protocols"*, Springer.

[14] Yang, K., Ma, J. and Miao, Z., *"Hybrid Routing Protocol for Wireless Mesh Network"*, International Conference on Computational Intelligence and Security, 2009. CIS '09, December 2009.

[15] Kazunori Ueda, and Ken–ichi Baba, *"Proposal of an Initial Route Establishment Method in Wireless Mesh Networks"*, International Symposium on Applications and the Internet, 2009. SAINT '09. Ninth Annual. July 2009.

[16] Clausen, T., Jacquet, P., *"RFC 3626 - Optimized Link State Routing Protocol (OLSR)"*, October 2003.

[17] Mobile Ad Hoc Networking Working Group of the Internet Engineering Task Force (IETF) *"RFC2026 - Ad hoc On-Demand Distance Vector (AODV) Routing*", Internet Draft, February 2003.

[18] Holter, K., *"Comparing AODV and OLSR"*, Essay, April 2005.

[19] Huhtonen, A., *"Comparing AODV and OLSR Routing Protocols"*, HUT T-110.551 Seminar on Internetworking, Sjökulla, April 2004.

[20] Huawei, Z. and Yun, Z., *"Comparison and Analysis AODV and OLSR Routing Protocols in Ad Hoc Network"*, Henan Normal University, Xinxiang, China, 2008.

[21] Chen, J., Lee, Y., Maniezzo, D. and Gerla, M., *"Performance Comparison of AODV and OFLSR in Wireless Mesh Networks"*, MedHocNet'06, Lipari, Italy, June 2006.

[22] Ksentini, A. and Abassi, O., *"A Comparison of VoIP Performance Over Three Routing Protocols for IEEE 802.11s-based Wireless Mesh Networks (WLAN Mesh)"*, Proceedings of the 6th ACM international symposium on Mobility management and wireless access, Vancouver, Canada, 2008.

[23] Tung, Y., Shahriar Rahman, Q., Al-Khatib, M. and Alsharif, S., *"Performance Evaluation of Ad-Hoc On-Demand Distance Vector (AODV) in Mobile Ad-Hoc Networks"*, International Conference on Wireless and Optical Communications Networks, August, 2006.

[24] Boukerche, A., Guardalben, L., Sobral, J.B.M. and Notare, M.S.M.A., *"A performance evaluation of OLSR and AODV routing protocols using a self-configuration mechanism for heterogeneous wireless mesh networks"*, 33rd IEEE Conference on Local Computer Networks, 2008. LCN 2008. October, 2008.

[25] Meyer, S., Javaudin, J., Varoutas, D., Rebernig, S., Goni, G., Karathanos, E. and Fernández, E.: *"OMEGA final usage scenarios report"* available at the URL: http://www.ict-omega.eu/fileadmin/documents/deliverables/Omega_D1.1.pdf

[26] BBC News, *"Where Next for the Web"*, March 9th, 2010. Available at http://news.bbc.co.uk/go/pr/fr/-/2/hi/technology/8555987.stm [Accessed on March 9th, 2010].

[27] Quitney Anderson, J. and Rainie, L., *"The Future of the Internet 2010"*, Pew Research Center's Internet & American Life Project, February 2010. Available at http://pewinternet.org/topics/Future-of-the-internet.aspx [Accessed on March 9th, 2010].

[28] *"OPNET Modeler 14.5A"*, OPNET Technologies, Inc. 7255 Woodmont Ave., Bethesda, MD 20814, USA, www.opnet.com.

[29] *"OPNET Modeler Product Documentation"*, integrated with OPNET Modeler 14.5A software component.

[30] Mohammad M. Siddique, Andreas Könsgen, *"Advanced Communication Lab Opnet Tutorial"*, available at http://www.comnets.uni-bremen.de/lehre/opnet-tutorial.pdf, SS 2006.

[31] Jarmo Prokkola, *"OPNET - Network Simulator"*, Tietotalo, University of Oulu, Finland, April 2006.

Terminada la tesis… pero no sin antes
agradecer a todas las personas que estuvieron de
alguna forma presentes, con su apoyo y motivación,
con palabras de aliento, con ayuda en muchos sentidos…

Primero lo primero.. a mi mama, Julia Maillo,
quien ha trabajado durante toda su vida con el único
fin de que yo llegase hasta este punto… la base me la diste tu
ahora me toca seguir a mi pero siempre
contare contigo… Gracias ma!

A mi papa, Luis Daniel Montiel, quien siempre
me daba esa visión de padre, siempre estuvo pendiente con
sus llamadas y quien me ayudo a aprender muchas cosas
de la vida. Gracias por tu apoyo..

A mis hermanos, Dani y Desi,
casi tres años sin verlos, pero siempre me recordé de ustedes.
Desi gracias por esas tardes de HALO :-) y
Dani, en muchos sentidos eres el ejemplo que seguimos..

A mi mama, mi papa y mis hermanos…
Los quiero mucho!

Un sentito ringraziamento a Vincenzo Suraci,
chi mi ha dimostrato che ogni impegno si deve fare
con passione e così lo farò in futuro.

Ad Andi, senza di chi non sarei
a questo punto. Falemnderit!

A mis compañeros de casa y amigos, Rich, Jose y Cyn…
Ya pasamos lo fácil! ahora nos toca lo difícil… Se les quiere!

To Catherine and Ruby, you've been excellent flatmates!
Thanks for the English corrections!
Hope we'll meet again..

Ad Iunia, grazie per essere il supporto
che avevo bisogno… y Ariana, gracias
por ser un apoyo durante este año y medio!.

A Kiko, mi mejor amigo, aunque no hayas
estado aquí todo este tiempo.. se que estarás cuando te necesite.

Finalmente, a mis familiares, los amigos de la comunidad y por supuesto, a
Colmillo y Loba (aunque no sepan leer), se les quiere a todos!