SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information & Communication Technologies (ICT)

ICT-213311
Omega

Deliverable *5.5*

Inter-MAC Final Evaluation Report

Contractual Date of Delivery:	*March 31st, 2011*
Actual Date of Delivery:	*April 28th, 2011 for V1, July 5th 2011 for V2*
Editor(s):	*Andreas Foglar*
Author(s):	*See list of authors*
Work package:	*WP5*
Estimated person months:	*60*
Security:	*PU*
Nature:	Report
Version:	**2.0**
Total number of pages:	25

Abstract

This document contains a detailed description of the Inter-MAC software and hardware implementation. In addition it describes the results of system evaluation of the Inter-MAC concept, carried out on real demonstrators.

Keyword list

Inter-MAC, simulation, demonstration, performance, protocol, data plane, control plane, path selection, HW/SW

Executive Summary

WP5 objectives in Omega

The Inter-MAC work package aimed at the definition of a technology independent convergence layer in charge of interfacing the network protocol with a number of heterogeneous physical transmission technologies in a seamless way. This Inter-MAC layer controls the different MAC layers by means of flexible and extensible technology-dependent Inter-MAC adaptors and provides functionalities like path selection, guaranteed quality of service, security, energy efficiency and is backward compatible and has better performance than state-of-the-art approaches.

Deliverable D5.5

D5.5 “Inter-MAC Final Evaluation Report” describes the final MAC functionality and its evaluation along with an outlook about further work.

The specification of protocols and the interfaces among Inter-MAC engines was provided in D5.3 “Inter-MAC Protocol Entities Interfaces Specification” [OMD54] OMEGA Deliverable D5.4, “Inter-MAC Protocols Performance Report”, December 2010, available on http://www.ict-omega.eu/fileadmin/documents/deliverables.html

[OMD53] whereas the functionalities provided by them were described in D5.2 “Inter-MAC architecture design” [OMD52]. D5.5 is also related to D1.1 “Final Usages Scenarios Report” [OMD11] since two of these scenarios have been taken as a reference to evaluate handover (Scenario My media follows me) and performance when a link in usage goes down (Scenario Working from home).

Simulations and measurements

Inter-MAC protocols simulations and preliminary performance evaluation have been described in D5.4 “Inter-MAC Protocols Performance Report” [OMD54]. A summary of the results is provided here.

Document structure and content

Chapter 1 focuses on the detailed description of the simulation environment used to test the whole Inter-MAC and Omega network protocols performances and concepts.

Chapter 2 analyses, by means of different focused demonstrator, different protocol performances dealing both with data plane and control plane functionalities.

Chapter 3 presents a first description of an Inter-MAC conformance test with respect to the OMEGA project requirements.

Chapter 4 highlights the main achievements, results and lessons learnt.

Impact on the other Workpackages

The present document impacts mainly on WP7, because all the considerations derived from the simulation and demonstration evidences provide clear indications on which protocol and architectural solutions are more suitable to achieve the desired results for the different Omega scenarios to be addressed in WP7.

List of Authors

First name	Last name	Beneficiary	Email address
Cristina	Peña	TID	alcega@tid.es
Vincenzo	Suraci	UoR-CRAT	suraci@dis.uniroma1.it
Andi	Palo	UoR-CRAT	andi@dis.uniroma1.it
Philippe	Christin	FT	philippe.christin@orange-ftgroup.com
Stefan	Nowak	UniDo	stefan.nowak@uni-dortmund.de
Falk-Moritz	Schaefer	UniDo	falk.schaefer@tu-dortmund.de
Marcin	Brzozowski	IHP	brzozowski@ihp-microelectronics.com

Document History

First name	Last name	Version	Comments
Andreas	Foglar	0.2	Creation of document, table of content, introduction.
Andi	Palo	0.3	Summary of Simulation Work
Falk-Moritz	Schaefer	0.4	Functional Description
Philippe	Christin	0.5	Standardization
Andreas	Foglar	0.6	Document polishing, section 1.1 text and section 3 input
Jean-Philippe	Javaudin	0.9	Final quality review
Board		1.0	Final version to be submitted to EC.
V. Suraci, A. Palo, G. Oddi, M. Castrucci, F. Delli Priscoli		1.9	Chapter added on Inter-MAC conformance test
JP Javaudin		2.0	Final edition

List of Acronyms

Acronym	Meaning
AC	Access Category (of WLAN)
ADP	Adapter
CBF	Current wireless channel Busy Fraction
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
HAN	Home Area Network
HWMP	Hybrid Wireless Mesh Protocol
ICT	Information and Communication Technologies
IEEE	Institute of Electrical & Electronics Engineers
IFS	Inter Frame Space (in WLAN)
IP	Internet Protocol
IPC	Inter Process Communication
ISO	International Organization of Standardization
MAC	Medium Access Control
MIT	Massachusetts Institute of Technology
OMEGA	hOME Gigabit Access
OSI	Open System Interconnection
PC	Personal Computer
PDF	Probability Density Function
PHY	Physical Layer device (transceiver)
PLC	Power Line Communication
PREP	Path Response (HWMP Message)
PREQ	Path Request (HWMP Message)
PSE	Path Selection Engine
QoS	Quality of Service
QoSE	Quality of Service Engine
RSSI	Received Signal Strength Indicator (in WLAN)
RTS/CTS	Request to Send/Clear to Send handshake
RX/TX	Reception/Transmission
TCP	Transfer Control Protocol
TSPEC	Traffic SPEcification
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
VHDL	VHSIC Hardware Description Language
WLAN	Wireless Local Area Network

Table of contents

1 Inter-MAC Implementation

When specifying the Inter-MAC implementation the separation of data and control/management plane became obvious. The data plane tasks are a few but which must be executed with high performance:

- Inter-MAC header insertion and extraction at termination points
- Inter-MAC header evaluation and translation via lookup table
- Extraction/insertion of control/management packets
- Probe frame generation for monitoring engine.

The complexity of the control/management plane is much higher and could increase more if additional intelligence is added. The functionality of the data plane is completed, a fact which could favor implementation in hardware. During the project, however, hardware implementation was abandoned as the functionality was not stable from the beginning.

Note that here the terms control/management plane are used, influenced by the three planes in classical telecommunication. Further on the shorter term control plane is used, although elements of control and management functionality are contained.

1.1 Overview and functional split

To allow for modular implementation and portability both data and control plane core functionality is provided with adaptors/ interfaces. Figure 1 shows the data plane at the left side, control plane at the right with the adaptors:

- Hardware abstraction layer: allows adaptation to different hardware implementations. Within the project three implementations were explored: C code in Linux user space, C code in Linux kernel mode and C code for embedded PowerPC running on FPGA of a RAPTOR[1] system. As expected Linux user space code was far less performing than the two other solutions (see chapter 2). A gate level implementation in FPGA was planned, but abandoned due to lack of resources.
- Control plane interface and data plane adaptor are two corresponding parts to separate data and control plane in modular way.

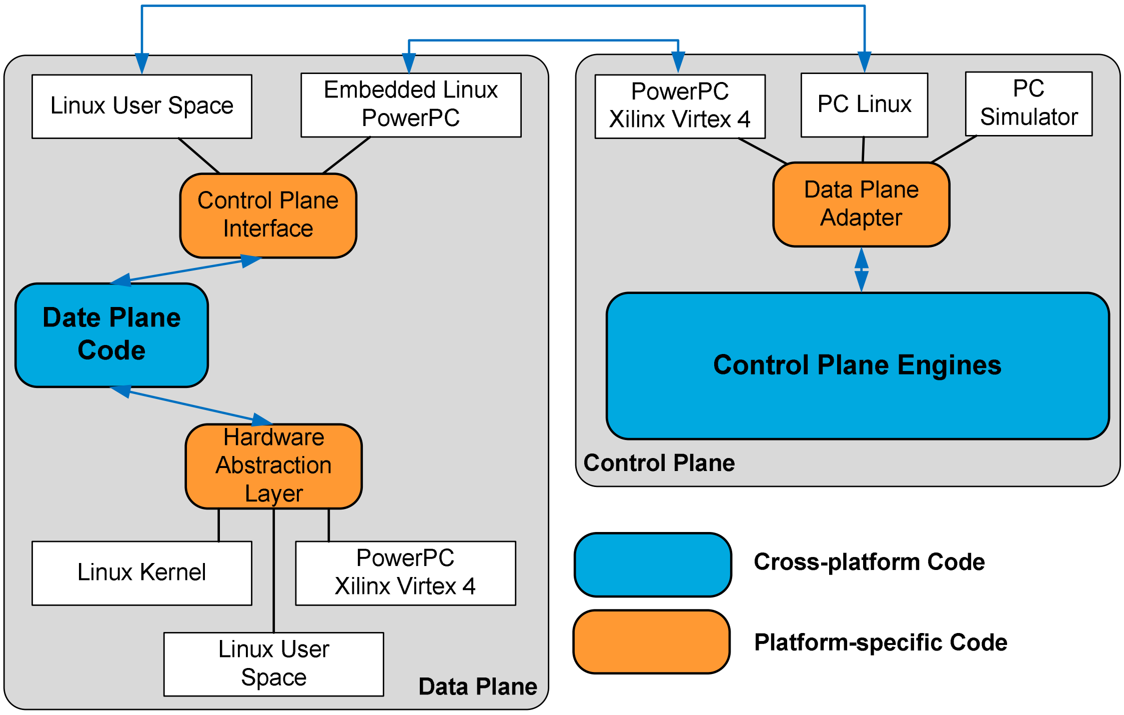

Figure 1: Separation of Data and Control Plane

---

[1] from HNI institute of the University of Paderborn

A more detailed look into the implementation shows Figure 2 for the case of Linux PC implementation.

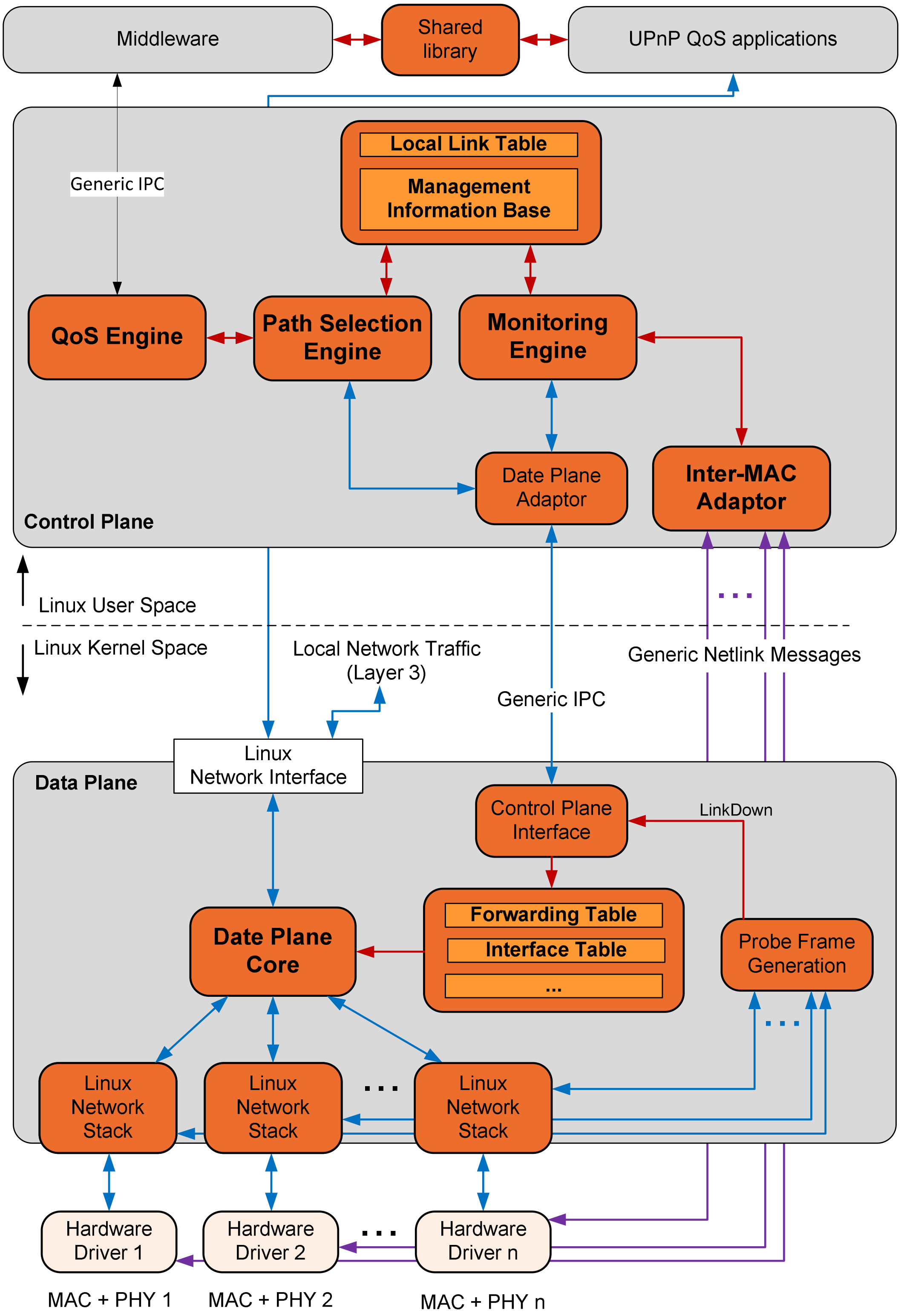

Figure 2: Detailed Functionality of Data and Control Plane

1.2 Data Plane

The data plane core (see Figure 2) receives Inter-MAC frames from the PHYs with the respective technology dependent MAC. Examples for PHY are Ethernet, WLAN, PLC etc., realized with hardware, driver and the Linux network stack. The data plane core forwards the frames according to the respective entry in the forwarding table, either to another PHY or upward to the Linux network interface. For the packets with local destination the Inter-MAC header is removed. For packets coming from the local device (PC) the Inter-MAC header is inserted.

Special hardware is provided to generate probe frames for network inspection. To allow for independent operation the Inter-MAC layer detects itself the attached interfaces and monitors the quality of the links via probe frames. These frames are generated via special hardware for more efficiency.

1.3 Control Plane

1.3.1 Path Selection Engine

The main task of the PSE is to establish end-to-end paths between two Inter-MAC nodes within a heterogeneous meshed HAN over multiple hops. In case the DATA PLANE receives a frame of an unregistered data flow, it triggers the PSE on demand to discover a new path. During transmission, link degradations or link failures are detected which trigger the PSE to identify affected paths within the forwarding table and to search for alternative paths. In both cases, the PSE floods the network with path request messages (PREQ), similarly to the Dynamic Source Routing Protocol for meshed networks. Intermediate nodes receiving a PREQ re-broadcast the PREQ to their remaining adjacent nodes and update the link metric therein. Currently, the available link capacity is used as a metric. Updating this metric along a specific path means to find the minimum link capacity along that path. Finally, after the destination node received the PREQ, it responds with a path reply message (PREP). The originating node collects all incoming PREPs, selects the path with the best metric and activates it by sending a path confirmation message to the nodes along that path to update their forwarding tables accordingly.

To assure fairness in data forwarding, traffic load is distributed across all available links to prevent overloading. Each time the PSE establishes a new path, it selects the path with the maximum available link capacity and thus prevents the use of overloaded links. The PSE periodically checks, whether there are better paths available. If the PSE discovers a better path, data traffic is switched accordingly. In this way, load balancing is realized. To avoid out of order reception, frames belonging to the same flow are always sent across the same path. Hence, destination nodes can deliver incoming frames directly to the network stack.

1.3.2 Quality of Service Engine

The QoSE represents the interface of the Inter-MAC towards higher layers. It communicates QoS requirements delivered by application layers or middleware layers to the Inter-MAC and performs admission control. To ensure a flexible interaction with higher layers, there are adaptors for several applications and middleware layers such as UPnP, UPnP QoS and SIP. Those adaptors are implemented as shared Linux libraries to be included in both C/C++ and Java applications. Higher layers communicate with the QoSE by inter-process communication offered by the operating system such as message queues. The QoSE adaptors parse these messages and provide the QoSE with the extracted information, e.g. the destination address or a traffic specification, in a common semantic language. For any QoS request the QoSE generates a network wide unique flow identifier and triggers the PSE to establish a new end-to-end path across the heterogeneous meshed HAN which meets the corresponding QoS requirements. A flow cannot be admitted in case there is no path that meets the associated QoS requirements and there is no flow with a lower priority that could be dropped (admission control). In case of heavy network congestion or any link failures, flows are dropped starting from the lowest priority. The QoSE communicates these events to higher layers.

1.3.3 Inter-MAC adaptor

1.3.3.1 Available data rate as a link metric

We deployed both, a technology-decoupled approach as well as a technology-dependent one with a proof-of-concept IEEE 802.11n ADP. For technologies without a dedicated ADP, the link capacity can be pre-configured. The Data Plane monitors the current link occupation by counting length and number of sent and received frames. The Data Plane can be extended easily to provide other technology-decoupled metrics like packet error rate, jitter or delay. However, each technology which is attached to the Inter-MAC should provide a dedicated ADP offering more precise measurements and estimations. In contrast to wireline links, wireless links underlie impairments, which cause frequent changes in the available link capacity. Thus, our technology decoupled approach does not provide a reliable basis.

In case of IEEE 802.11n, we modified the wireless driver to periodically report the Current Wireless Channel Busy Fraction (CBF). For this purpose, the <ath9k> driver and the <ath9k_htc> driver for Ubiquity SR71-E with AR9280 chipset and TP-LINK TL-WN721N with AR9271 chipset have been adapted. Hardware registers provide information about the fraction of time the channel was sensed physically busy. Only the latest Linux

<mac80211> stack and <ath9k> use these hardware registers allowing to estimate the current CBF when scanning, while not being connected to an (Independent) Basic Service Set (IBSS/BSS).

AR_CCCNT	Number of cycles
AR_RCCNT	Number of cycles the RX Clear signal was low (not busy)
AR_RFCNT	Number of cycles the station was actively receiving
AR_TFCNT	Number of cycles the station was actively transmitting
If one of the registers overflows, all values are shifted by 1.	

The current configuration of the device must be taken into account. When the extension channel is active, AR_RCCNT must be configured to include the time the extension channel was sensed physically busy.

The virtual carrier sense busy time is not included in these register values and thus has to be added, since the physical CBF can differ significantly from the actual busy fraction (cf. channel load). The modified driver counts type and number of sent and received frames while differentiating for example between RTS/CTS, DATA and QoS DATA per Access Category (AC). Based on this information, the expected duration of the Inter Frame Space (IFS) is added.

To receive all frames the interface have to be configured in promiscuous mode (monitor mode), otherwise packets for other network participants are filtered by the hardware to save energy and computing. All frames received by the hardware triggers the lowest ath9k driver receive function. This function differs with the hardware. Then the packets are forwarded to the Linux networking stack. This causes the <mac80211> Linux wireless functions to be executed. The driver receive and the transmit routines were adapted, so that all packets are handed to new function, which briefly analyzes the packet and keeps track of sent and received frames. Additionally the beacon frames are analyzed and RSSI information are stored (see below). It is important to differentiate the type of frames. Different Access Categories require different, Arbitrary IFS. Additionally a transmission with a previously exchanged RTS/CTS handshake causes additional Short IFS. Furthermore multiple packets in the Linux network stack can be aggregated to a single aggregated MAC Protocol Data Unit in the device hardware. This is why multiple packet transmissions on higher layer can cause only a single packet transmission on lower level and with it only one Short Inter Frame Space (SIFS). If the MPDU was part of an aggregated MAC Protocol Data Unit, the SIFS is not added to the virtual busy time. The same holds for data packets which are part of a Multiple Frame transmission. Here only a SIFS and not the Arbitration Inter Frame Space Number (AIFSN) is added. The CBF monitoring and reporting can be switched on and off. If it is activated, the <mac80211> protocol stack periodically sends CBF reports to the Netlink clients. For example the Linux-wireless “iw” can register to events sent by the <mac80211> stack. In the same way the ADP connects to a configured wireless device and can switch on the reports.

However, the available data rate does not only depend on the current CBF, but also on the properties of the link to a specific station. The ADP offers a service to other CP engines, i.e. to inquire an estimate for the possible throughput to a specific station. The destination MAC address must be handed to the ADP function, which calls via Netlink a <mac80211> function. Here either the rate control scheme of the driver or the <mac80211> rate control is inquired, depending on the configuration of the driver. If the station is part of the IBSS the Modulation Coding Scheme going to be used for a potential packet can be inquired. Because an ongoing exchange of probe frames takes place, this value is quite accurate at least for the next packet. The Modulation and Coding Scheme (MCS) index can be linked to a data rate, which is then handed to the ADP via the Netlink callback.

Through the combination of both Current Wireless Channel Busy Fraction (CBF) and data rate an estimation of the available data rate can be calculated.

1.3.3.2 WiFi Handover

The IEEE 802.11n WLAN ADP also offers technology dependent optimization functionality by enabling an intra-technology handover. The implementation was tested with the following network interface cards: TP-Link TLWN721N, Ubiquity SR71-E and Lantiq Wave 300. Each network interface card operates in promiscuous mode and, thus, receives beacons from other access points (AP), too. The Received Signal Strength Indicator (RSSI) in any beacon with a specific SSID is monitored additionally to the RSSI of the current BSS. All values are periodically reported to the ADP, which checks whether an alternative AP has an average RSSI higher than the current AP by a certain threshold. In this case, the network adaptor is triggered to connect to the AP that reported a higher RSSI. The Inter-MAC detects the LinkDown and LinkUp events accordingly. A new path for the data traffic is found and registered in the forwarding tables of the affected nodes. However, it is important to mention that all APs have to operate on the same channel. Furthermore, processing only the RSSI does not always lead to an optimum configuration. For further enhancement of the handover process additional information on available (I)BBSs, occupied channels and CBFs has to be exchanged between the nodes.

1.3.3.3 Generic Netlink Messages

The communication between mac80211 stack running in the Kernel Space and the ADP running in User Space is realized by Generic Netlink Messages. The linux kernel sources include a library offering Netlink functions. The the latest <mac80211> stack Generic Netlink Messages replace the obsolete IOCTRL functionality and is used by User Space applications like “iw” and other network configuration tools like the Network Manager. It is possible so send information to the driver with or without callback functions. Furthermore the driver itself can send information to the User Space.

2 Inter-MAC Evaluation

2.1 Description of Demonstration Scenarios

This section describes some typical scenarios which occur in a household. The handover example applies if a user walks around in a house with a mobile device. He/she expects for example continuous TV reception. The other important example is a link failure; for example by closing a door a wireless link is interrupted, or by switching a large load the PLC network could fail for a certain node.

2.1.1 WiFi Handover

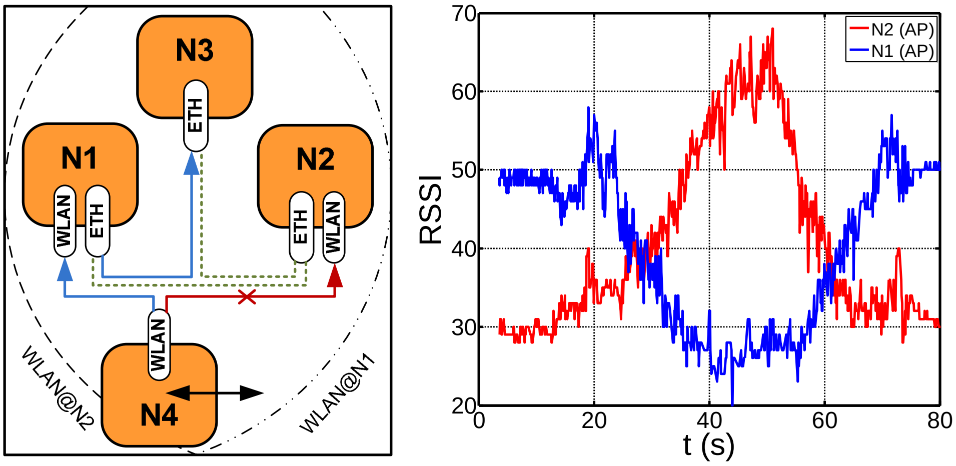

Figure 3: Handover Example

The figure shows an example setup with a multi-hop application data stream and the corresponding RSSI over time. The mobile node (N4) is connected to N1. At second 33 N4 measures a higher RSSI and performs a handover. About 1.1 s are needed for the Inter-MAC to detect all link events, to remove the path (N4->N1->N3), to find an alternative path (N4->N2->N3) and to register it.

2.1.2 Example Home Network Scenario

The behavior of the Inter-MAC in a real HAN scenario is presented to demonstrate a proof of the Inter-MAC concept and its benefits. Figure 4 shows the steps with the self-explaining legend and the technical parameter description following.

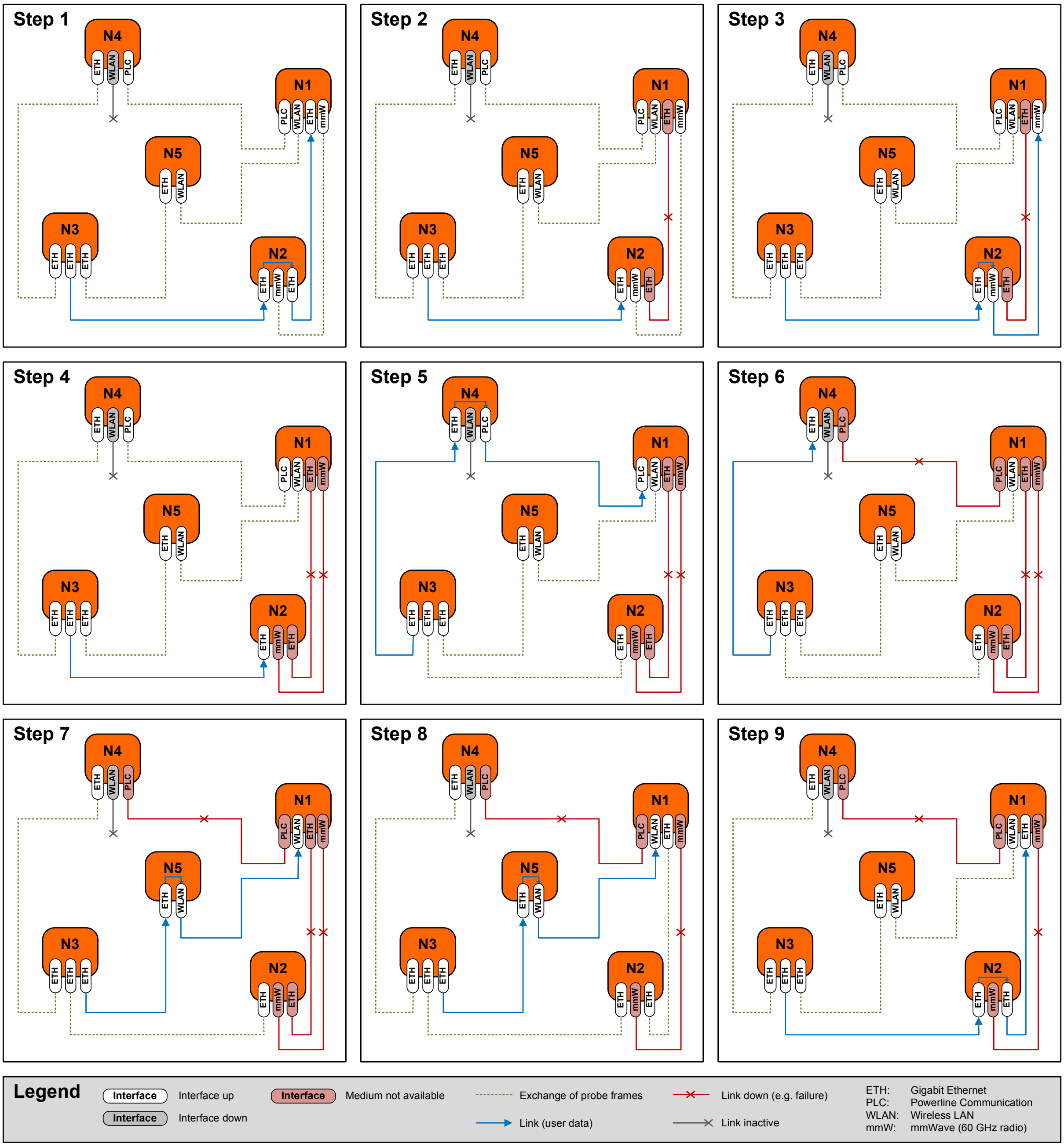

Figure 4: Home Networking Scenario

Linux OS	Ubuntu 10.04
Linux kernel version	2.6.32
N1, N2, N3	PCs (HP compaq 8100) Intel Core i5-660 Processor (3.33 GHz, 4 MB total cache)
N4	Embedded PC Lex Twister (TW2482S-00C) Embedded Intel Core2Duo, 1.66 GHz
N5	Fujitsu Lifebook T730 IntelCore i3-380M, 2.53 GHz
Interfaces	**Network adaptor/chipsets**
ETH @ {N1,N2,N3,N4,N5}	Gigabit Ethernet based on Intel 82571EB (kernel driver e1000e)
mmWave equipment	based on [5]
PLC @ {N1, N4}	Netgear Powerline AV+ 200
WLAN @ N1	D-Link DWA-556 (AR5008 chipset)
WLAN @ N4	TP-LINK TL-WN861N (AR922X)
WLAN @ N5	Embedded WLAN (AR9287)

Interface	Probe Frame Interval (ms)	LinkDown timeout (ms)
Gigabit Ethernet (ETH)	10	50
mmWave 60 GHz radio (mmW)	50	1000
Powerline Comunication (PLC)	200	700
Wireless LAN (WLAN)	10	1000

Table 1: Home network scenario parameters

Figure 4 above shows a meshed network topology of a heterogeneous HAN comprising five Inter-MAC nodes N*i*, *i*=1…5. Each node offers a certain set of interfaces for different networking technologies. For example, N1 has four different interfaces, i.e. PLC, WLAN, Gigabit Ethernet (ETH) and mmWave (mmW). In the very beginning, a HDTV video stream is set up between N3 (source) and N1 (destination) with a data rate of approximately 27 Mb/s. The corresponding path across the HAN is made up by two Gigabit Ethernet links (N3->N2->N1). One possibility for a node to save energy is to power down interfaces which are currently not in use. This is the case for the WLAN interface at N4. All the rest of the interfaces exchange probe frames. Now, the Ethernet cable at N1 is detached and for the corresponding link (N2->N1), N2 fires a link failure event in the Inter-MAC. As the best suited path is provided by the mmWave 60 GHz radio link (N2->N1), the stream is switched accordingly. Subsequently, this link degrades and after the LinkDown timeout (cf. configuration table) the Inter-MAC at N2 recognizes this link as not working. Now the PSE at the source node N3 has to find an alternative path to the destination node N1. The best available path is now provided by an Ethernet connection (N3->N4) and a PLC connection (N4->N1). Assume that an additional electric device is switched on which causes heavy interference in the in-house power grid such that the PLC link (N4□N1) cannot be used anymore. Consequently, the Inter-MAC at N3 again tries to find an alternative path to the destination which turns out to be provided by an Ethernet connection (N3->N5) and a WLAN link (N5->N1), which obviously is the only path available between the source node and the destination node. At the end of the scenario, the Ethernet connection (N2->N1) is re-established. A better path than currently in use becomes available, so that the forwarding tables at N3 and N2 are re-configured such that the original path (N3->N2->N1) is in use again.

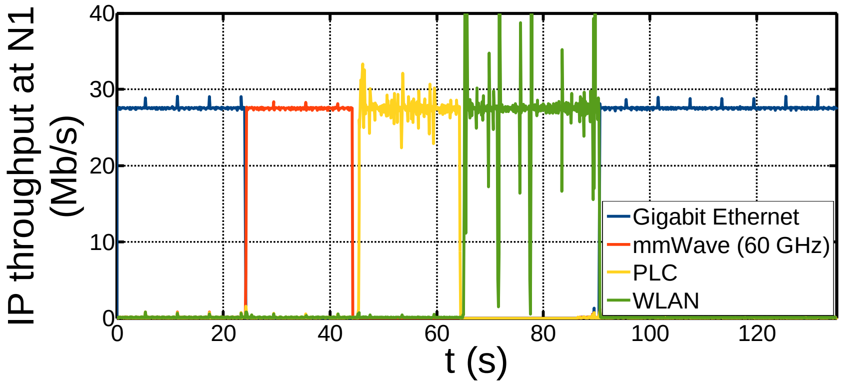

Figure 5: Throughput at Node N1

In the figure above, the IP throughput at N1 is shown over the time labeled with the corresponding technology in use. As can be seen, the HTDV stream can be transmitted at a constant data rate with some very short fades when the technology has been switched. Obviously, WLAN is characterized by a large variance regarding the data rate, whereas Ethernet and the mmWave technology provide links with more constant data rates. Besides, PLC is associated with a certain variation of the data rate as the behavior of the PLC channel can be considered as a mixture of a wireless and wireline channel. Our results reflect that the Inter-MAC concept as well as our implementation is able to transmit a HDTV stream over a heterogeneous meshed HAN without throughput degradation. Only a small delay has to be taken into account when switching the technology.

2.2 Measurement Results from Demonstrations

In this section some selected measurement results are shown as examples. Extensive collection of results is available within internal documents and involved partners.

2.2.1 Packet Error Rate PER

Packet error rate is the most sensitive parameter for experienced QoS. Packet losses request packet retransmissions (in case of TCP) which contribute significantly to end-to-end latency. For streaming applications (RPT/UDP) packet losses directly affect image and voice quality.

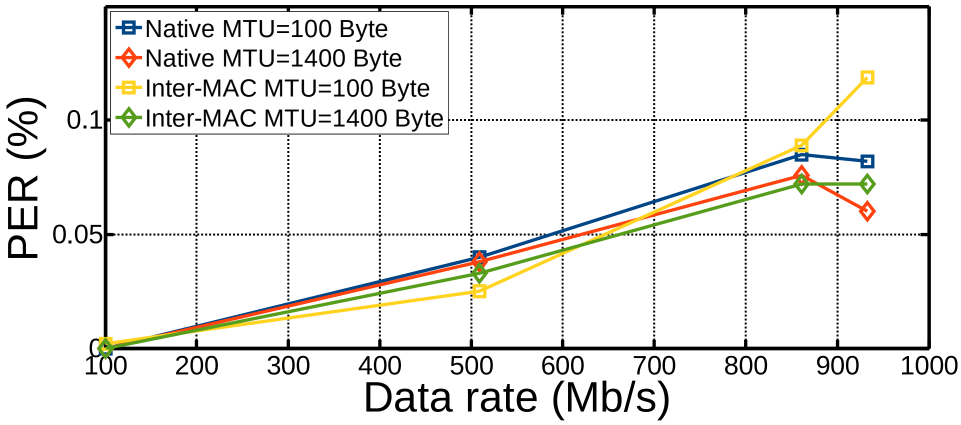

Figure 6: Packet Error Rate PER vs. Throughput

2.2.2 Round Trip Time RTT

Round trip delay is experienced by the user when it exceeds significantly ten milliseconds. It translates for example into echo, non-lip-synchronous video, but also to slow-reactive data applications with "slow-motion" drop-down menus.

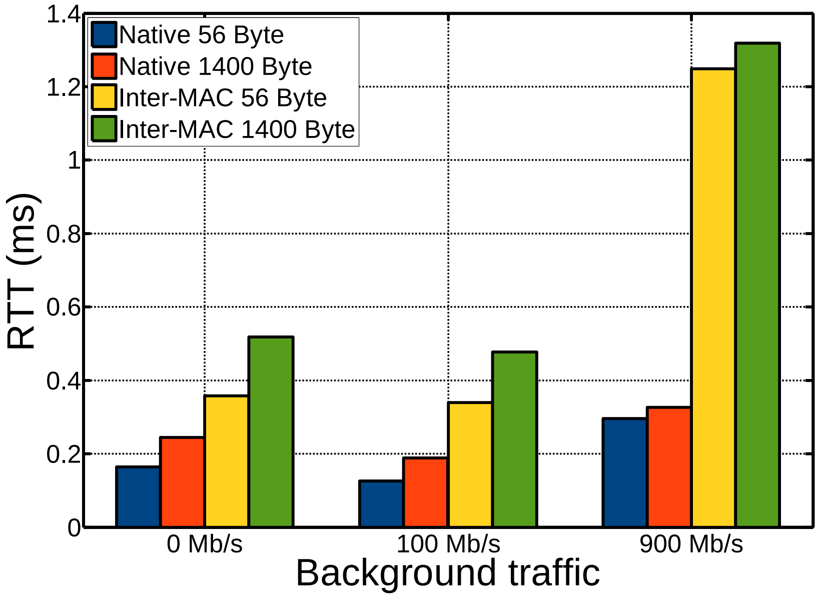

Figure 7: Round Trip Time RTT depending on Background Traffic

2.3 Summary of Simulation Work

The text in this section is taken from D5.4, updated and compared with measured results.

2.3.1. Overview

OPNET is a powerful network simulation tool that can be used to model communication systems and to predict network performance. It was originally developed at MIT (Massachusetts Institute of Technology) and was introduced in 1987 as the first commercial network simulation tool. Actually, it provides a comprehensive development environment supporting the modelling of communication networks and distributed systems. Both, the behaviour and the performance of modelled systems can be analyzed by performing discrete event simulations. The OPNET environment incorporates tools for all the phases of a project, including modelling design, simulations, data collection and data analysis.

In order to provide a very helpful tool, at one hand for the design process of the Inter-MAC layer architecture and at the other hand to evaluate the performances of the Omega Network, a whole simulation framework is developed under OPNET Modeller. This framework consists of new node models that integrate inside their ISO/OSI stack the Inter-MAC layer. In this way a node with one IP address can have more than one physical

interface without the need of bridging these technologies. So the nodes obtained in this way are connected together with each other using heterogeneous technologies.
The implementation of the Inter-MAC layer reflects the exact architecture defined in D5.2 with control and data plane well distinguished between them. Regarding the Control plane the most time-critical actions to be performed are the path selection functionalities. The path selection protocol of the Inter-MAC layer is called Hybrid Wireless Mesh Protocol (HWMP). The simulations aimed to compare the performances of the two modalities. At one hand the reactive only mode without proactive paths, and at the other hand the hybrid mode which creates a proactive set of paths from any Omega node to the Omega Gateway node. The principal pros and drawback between the two modalities emerged from this test.
Regarding the functionalities these tests have been done to evaluate:

- Path setup time
- Convergence time
- Wireless link handover
- Wired link failure
- Overhead measurements

Refer to the D5.4 for more detailed information about the simulations.

2.3.2. Path setup time

The main purpose of this test was to measure convergence times of both Proactive and Reactive modes in HWMP. In order to compare the results, the tests were set up using 5 different scenarios which have 4, 8, 12, 16 and 20 nodes. For each scenario, two separate tests were made: one with the proactive paths disabled (that is, using just the Reactive mode of HWMP) and one with the Hybrid mode enabled (that is, using both Proactive and Reactive modes of HWMP). Before going into the results, it is important to notice 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 in order 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).

2.3.2.1. Reactive Mode Path Set-Up Time

The simulations show that when using the Reactive mode of HWMP only, the initial delay to start the application does not exceed ten milliseconds in the scenarios tested. This time is a function of the number of nodes, the actual traffic of the network, the number of hops from source to destination and the type of the links. The order of delay for the application to start is acceptable and imperceptible.

2.3.2.2. 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 in a linear manner, starting from 0.790ms in the scenario with 4 Nodes and ending at 6.997ms in the scenario with 20 Nodes. The Proactive mode protocol convergence time is a bit smaller than the reactive mode path setup time. This because the Proactive mode starts right after simulation start time, there is no background flow set in the network yet, reducing the link delays and therefore convergence time.

2.3.3. Wireless link handover

In this case a mobile node which was the source of a voice over IP call changed its location in the Omega network causing serious signal degradation due to the increased transmission range from the access point. So the path selection protocol managed to find another access point whose link quality satisfied the flow requirements. In this scenario the path switch time was measured as well as the bit error rate. The simulation showed that in the worst case the gap of time on which the service is not available is nearly 1.5 seconds. The principal contribution to this time is the path degradation detection. Another factor that contributed in this time is the fact that the outgoing transmission queue was full. In such situation the control packets sent by the path selection engine are stuck in the transmission queue and must wait for it to be emptied. For every packet in the queue there are done 3 attempts before it is dropped. These are the packets that had to use the old path and are not available anymore. So the resulting path setup time is 0.7 seconds. During this time the bit error rate has a peak of 10^{-3} while normally it was 10^{-5}.

2.3.4. Wired link failure

This case was a hard failure of a link at a certain time. In the Omega network between two nodes is deployed a video streaming application whose path contains two subsequent links. The first one is Gigabit Ethernet and the second is a PLC bus link. The Gigabit Ethernet link is in common even with another flow, an FTP session flow. At a certain time this link goes down so the path selection engine has to find a new route for the two mentioned flows.

Given the fact that it is a wired link, the detection and the subsequent re-routings happened much faster. In this case the re-routing interval of time for the video streaming flow is about 380 ms. In this scenario the detection of link failure contribute is 250 ms and the rest is the time of the new path discovery.

The FTP session is affected in another manner. Before the link break its download response time is 6 seconds for every identical file. During the new path setup it reaches a peak of 8.5 seconds and after the re-routing it comes back to 6.5 seconds.

2.3.5. Overhead of Hybrid Wireless Mesh Protocol HWMP

The main purpose of this test is to measure the total amount of control bits generated using HWMP as the path selection protocol. We have proceeded as in section 2.3.2, using the same set of scenarios which consisted 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 results can vary significantly depending on the network topology and since each node utilized in the simulation can have up to three technologies and therefore links.

Measuring the amount of control bits received it can be observed from the results that the values follow a non linear trend, incrementing the amount of control bits when the number of nodes in the network is incremented. This result can be explained as follows: increasing the number of nodes N the number of connections between nodes is given by N(N-1)/2 in a full mesh topology. This approximates very well the trend of the graphics taking into account that it is not a quite full mesh topology.

If we had to consider the utilization of all the links of the network the situation would change. Broadcasting a control message to all nodes in the network means that each node makes a broadcast to each of its interfaces but only once. Thus the complexity of this operation is given by O(N*Avg(I)). Where Avg(I) is the average number of interfaces of the nodes and it’s a constant value. So the utilization of the overall network capacity increases in a linear manner with N.

- 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 more PREQs and PREPs are sent throughout the network in broadcast;
- Compared to the Omega network requirements, the obtained values are rather small and should not significantly influence the network performance.

2.3.6. Summary of control plane results

The core of Inter-MAC technology and key for the success of the future home network architecture developed in Omega is the path selection algorithm. Moreover, it is the process which consumes more processing time. That is why this algorithm has been evaluated with simulation and real test-beds. The following conclusions have been achieved:

- Theoretical considerations derived from the simulation results:
    - Given the maximum number of nodes assumed in a home environment, the reactive part of the path selection algorithm can manage easily an Omega Network and any eventual re-routes from failures and congestions.
    - The convergence time of the protocol is less than 1 second. The total re-routing time – adding the detection time – in the worst case is about 1.5 seconds in a path with wireless links.
    - The reactive part can be further modified in order to suit perfectly to the Omega Network. In fact, it is addressed for Wireless Mesh Networks. For example the fundamental metric would be the delay and thus the first Path Requests that arrives at the destination node represents the path with the best metric.
    - The combination of the two modalities improves the initial delay during the path setup by reducing it from a maximum of 10 ms to practically zero. But from our point of view, it does not justify the increased complexity of the Path Selection Engine and the additional overhead

    of the protocol. So, the best solution of compromise in an Omega Network is adopting a reactive approach.

- Empirical considerations derived from measurements performed in “physical” test-bed:
    - The experiments carried out in the real test-bed showed that the actual determination of an alternative path after a link break is actually only a rather small portion of the convergence time and most of the time is used for mechanisms outside of the path selection protocol. Consequently, strong care must be taken for the fast and correct detection of link breakages.

Furthermore, the fast cleanup of the transmission queue is also important for a minimum convergence time.

2.3.7. Summary of data plane performance

Demonstration activities carried out with different versions of data plane and running on different platforms, entail the following conclusions:

- The maximum throughput achieved with an optimized version of data plane running on RAPTOR board is only around 250 Mbit/s. The round-trip latency is usually around 125 µs, with a jitter of 50 µs.
- This throughput is limited by frame rate which, at the same time, is limited by the available processing capacity and the number of cycles per frame. Further analysis show that a throughput of 500 Mbit/s could be achieved with the following improvements:
    - Moving RX and TX related tasks to hardware, by implementing them as finite state machines.
    - A table search engine, implemented as a hardware accelerator.
- Furthermore, with a multi-processor approach, a throughput of 1 Gbit/s could be bordered or even exceeded.
- When looking for a cheaper solution for proof of concept or debugging of Inter-MAC technology:
    - It has been proven that data plane running within the kernel space is able to deal with almost the maximum throughput of 100 Mbit/s interfaces, being expectable to achieve better results when using Gigabit Ethernet cards.
    - Data plane running in user space, as user application, cannot fulfill all the requirements of a gigabit Omega network. However, it is very valuable for testing and debugging.

Summarizing, the simulations helped to identify the feasible concepts, while the demonstrators evidenced that what has been done in a simulated world can be realized in a real prototypal setup. Moreover the demonstrators showed that the prototypal performances are encouraging and further optimization of C-code as well as hardware implementation will lead to even better performances.

3 Inter-MAC Conformance Test

3.1 OMEGA device robustness

Objective: testing the input/output behavior and the error handling capabilities of the OMEGA device, by injecting both well-formed Inter-MAC packets and malformed Inter-MAC packets.

This section describes a qualitative test procedure that one have to carry out in order to test the input/output behavior of an OMEGA device. This approach is based on the idea of injecting both well-formed Inter-MAC packets and malformed Inter-MAC packets and testing the OMEGA’s device error handling.
The test procedure consists of two particular sub-tests:

1. The first sub-test is performed considering the range of acceptable values of each field of the Inter-MAC packet and testing both acceptable and unacceptable values, without regard to the relationships among fields. The OMEGA device behaviour is analyzed.
2. The second sub-test completes the analysis carried out in the first point, by testing the behaviour of the OMEGA device in the case in which several hypothetic inconsistencies exist among Inter-MAC packet fields parameters (both header and payload).

The test procedure is performed using a specific user-space application (Packet Generator) developed to build both well-formed and malformed Inter-MAC packets. The application uses Linux TAP driver to create a virtual Ethernet device through which all frames are injected.

3.1.1 Single packet-field test results

This section shows the test to be carried out on single fields of the Inter-MAC Header and the Control Plane header, testing both acceptable and unacceptable values. The list of fields is reported under the form of a table which describes the single field as well as provides the field-specific acceptable values.

3.1.1.1 Inter-MAC Packet Header

Name	Type	Acceptable values	Description
hop_count	__u8 (int)	All values are acceptable	Hop counter describes the number of hop that the frame is allowed to do
seq_no	__be16 (int)	All values are acceptable	Identifies the frame in case of duplicated frames (flooding, broadcast)
qos_class	__u8 (int)	Unused up to now	Identifies the class of service
Flags	__u8 (int)	Used xxxxxx1x, xxxxxx0x, to represent if the packet is control (1) or data (0) type.	Identifies the type of the frame: Control frame, Data frame and the presence of optional fields (payload encryption, legacy address, timestamp).
imac_dst	unsigned char [3]	All values are acceptable	Inter-MAC Address representing the destination of the frame
Authentication	__u8 (int)	Unused up to now	This field is used for authentication and authorization
imac_src	unsigned char [3]	All values are acceptable	Inter-MAC Address representing the source of the frame
flow_id	__be16	All values are acceptable	The identifier is a hash of some fields of the payload like IP source address, IP destination address, TCP/UDP source ports.
i_proto	__be16	All values are acceptable	Specifies the type of protocol encapsulated in the payload. The values are the same as Ethertype field.
legacy_dst	unsigned char [6]	All values are acceptable	Destination address of a legacy device
legacy_src	unsigned char [6]	All values are acceptable	Source address of a legacy device
timestamp_tx	__be16	Unused up to now	Timestamp when the frame is transmitted by the Inter-MAC layer
timestamp_rx	__be16	Unused up to now	Timestamp when the frame is received by the Inter-MAC layer

Table 2: Inter-MAC packet header fields

Since all values are syntactically correct for each Inter-MAC header field, no errors occur whatever values are present in the fields.

3.1.1.2 Control Packet Header

Name	Type		Description
type	__u8 (int)	Unused up to now	The type of the Control Packet (Data or ACK message)
destinationEngine	__u8 (int)	The acceptable values are: 0, 1, 2, 3	The Destination Engine which manages the control packet (e.g. Path Selection Engine or Monitoring Engine)
sequenceNumber	__be16 (int)	The acceptable values are all the integer values greater than 0	The Sequence Number of the Control Packet
payloadLength	__be16 (int)	The acceptable values are all the integer values greater than 0	The length of the Control Packet payload (excluding Control Header)

Table 3: Control packet header

The following single-field tests have to be performed on the Control Packet Header:

- The destinationEngine field has to be set with values outside the acceptable range. The OMEGA device must correctly discard the packets with such values.

- Regarding the sequenceNumber field, all values are acceptable and no errors occur whatever values are present in this field.
- The payloadLength field has to be tested with bigger and smaller values than the real payload length. In both cases the Control Packet must be discarded by the Control Plane of Inter-MAC.

3.1.2 Complex test results

This section reports a set of tests with the aim of testing the behavior of the OMEGA device in the case in which several hypothetic semantic inconsistencies exist among Inter-MAC packet fields parameters (contained in both header and payload). The following two tests have to be performed:

1. The flags field inside the Inter-MAC Header is set to xxxxxx0x (Data Packet), while the payload does not contain data packet payload (e.g. Inter-MAC Control Packet or random data values).
2. The flags field inside the Inter-MAC Header is set to xxxxxx1x (Control Packet), while the payload contains data packet payload (e.g. IP or ARP payload). All the sequence of checks performed by the Control Plane execution code on the Control Packet header has been tested.

3.1.2.1 Data type inconsistency

This test consists in setting the flags field inside the Inter-MAC Header to xxxxxx0x (Data Packet), while the payload does not contain data packet payload. Two cases are possible:

- If the i_proto field inside the Inter-MAC header is a registered protocol type, then the packet is handled by the proper upper layer protocol procedure (e.g. ARP or IP), which uses its own checks to discard or not the packet.
- If the i_proto field inside the Inter-MAC header is not a registered protocol type, the packet is discarded by the operative system.

3.1.2.2 Control type inconsistency

This section reports the test case to be carried out by setting the flags field inside the Inter-MAC Header to xxxxxx1x (Control Packet), while the payload has to contain data packet payload (e.g. IP or ARP payload). The structure of a Control Packet, as listed in Section 1.2 is:

Control Packet Payload Structure							
Control Header						**Engine Specific Payload**	
type	**destination Engine**	**sequenceNumber**		**payloadLength**		**type**	…
1 Byte	**1 Byte**	**1 Byte**	**1 Byte**	**1 Byte**	**1 Byte**	**1 Byte**	**… Bytes**

Table 4: Control packet payload structure

It is composed by a common Control Header and an Engine Specific Payload. A sequence of checks on the Control Packet header is carried out by the Inter-MAC's Control Plane execution code and it is reported in the following list of activities:

1. A malformed packet arrives to the Inter-MAC Control Plane.
2. The Control Plane dispatches packets based on destinationEngine field, inside the Control Header. If the destination engine does not belong to the supported engines, the packet has to be discarded because no engine can manage it.
3. Each destination engine interprets the message based on the field type, inside the Engine Specific Payload, by calling the proper deserializing method.

3.2 Test environment

Test environment consists of a user space application (Packet Generator) able firstly to generate Inter-MAC packets and then to evaluate OMEGA device behavior for each of these packets.

The application uses the TAP driver provided by Linux kernel. TAP allows packet reception and transmission for user space programs. It can be seen as a simple Ethernet device, which, instead of receiving packets from

physical media, receives them from user space program and instead of sending packets via physical media writes them to the user space program.
This approach has the advantage to testing quickly a large amount of packets on a single machine, avoiding sending them over the network.
More specifically, TAP provides to the user space application two interfaces:

- /dev/net/imac_tap (character device);
- imac_tap0 (virtual Ethernet device).

Packet Generator performs the following actions:

1. opens /dev/net/imac_tap
2. registers the network device imac_tap0 with the kernel
3. generates an Ethernet frame (malformed or well-formed) stored as an array of characters
4. writes the frame to /dev/net/imac_tap

Kernel will receive this frame from imac_tap0 interface. In this way it's possible to test Inter-MAC layer behavior for the received frame.

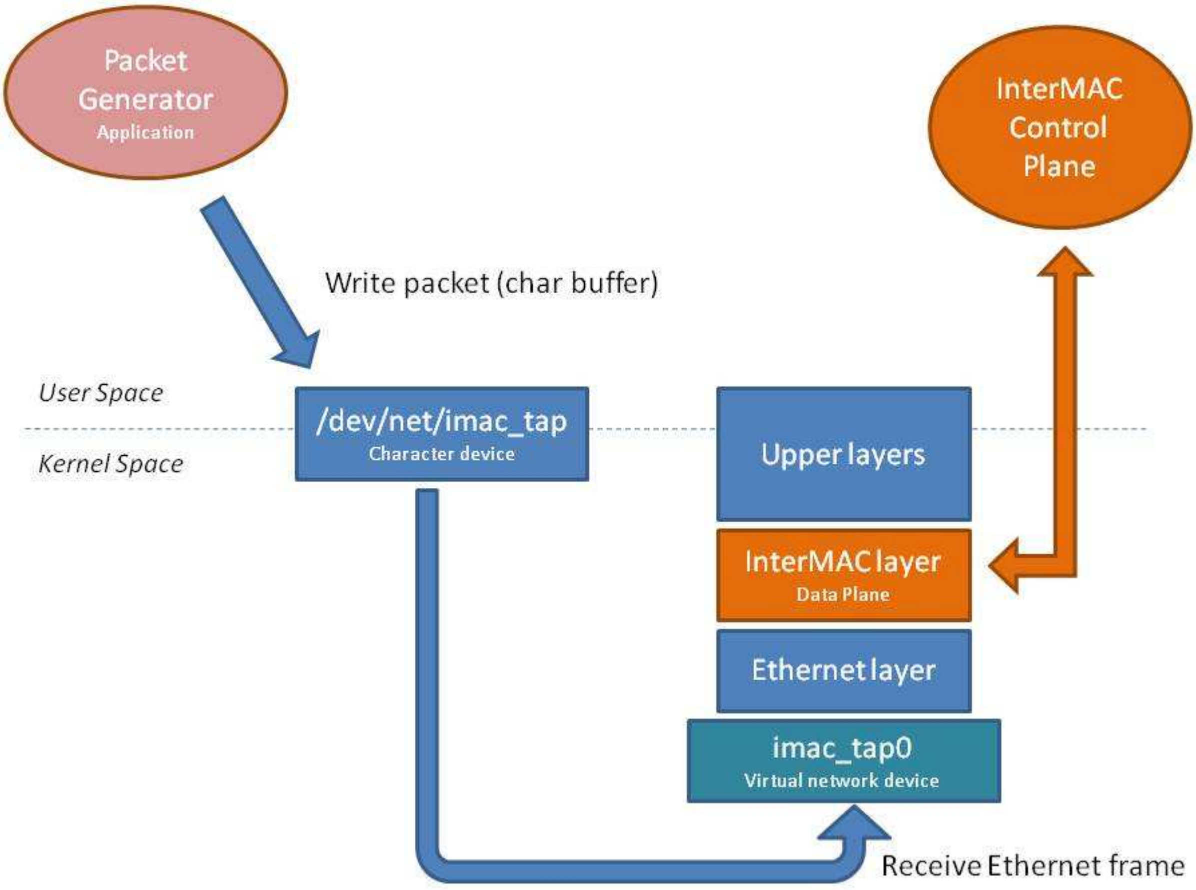

Figure 8: Test environment

3.3 Single Technology performance test

Reference Test Case Scenario:[2]

---

[2] For more information on the proposed test case scenario, refer to the OMEGA deliverable D7.4 "*Omega Platform based on Linux PCs*".

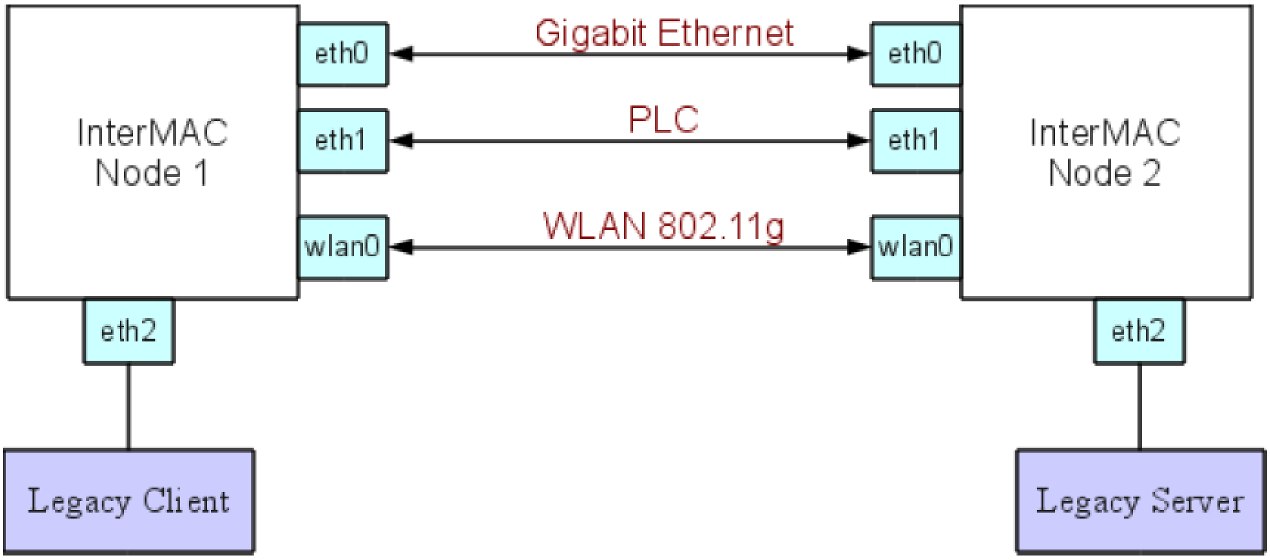

Figure 9: Testing scenario

3.3.1 Inter-MAC overhead delay

Communication Technologies: Wi-Fi 802.11g, Gigabit Ethernet 802.3z, Power Line Communication (PLC).
Traffic flows characteristics: 1000 ping requests/replies with two different MTU values and different amount of background traffic.
Objective: Evaluate the impact of the Inter-MAC on the end to end delay for each packet, caused by the increased processing time. Round Trip Time (RTT) will be measured separately using the three technologies (Wi-Fi 802.11g, PLC, Gigabit Ethernet). A summary of the expected values is reported in the next table. Meanwhile figure

Technology	Background traffic(Mbps)	Max delay(ms)
Gigabit Ethernet	0	0.5
	25	0.5
	100	0.5
	900	1.5
PLC	0	5
	25	5
	100	1000
	900	1000
WLAN	0	5
	25	2500
	100	2500
	900	2500

Table 5: Inter-MAC overhead delay

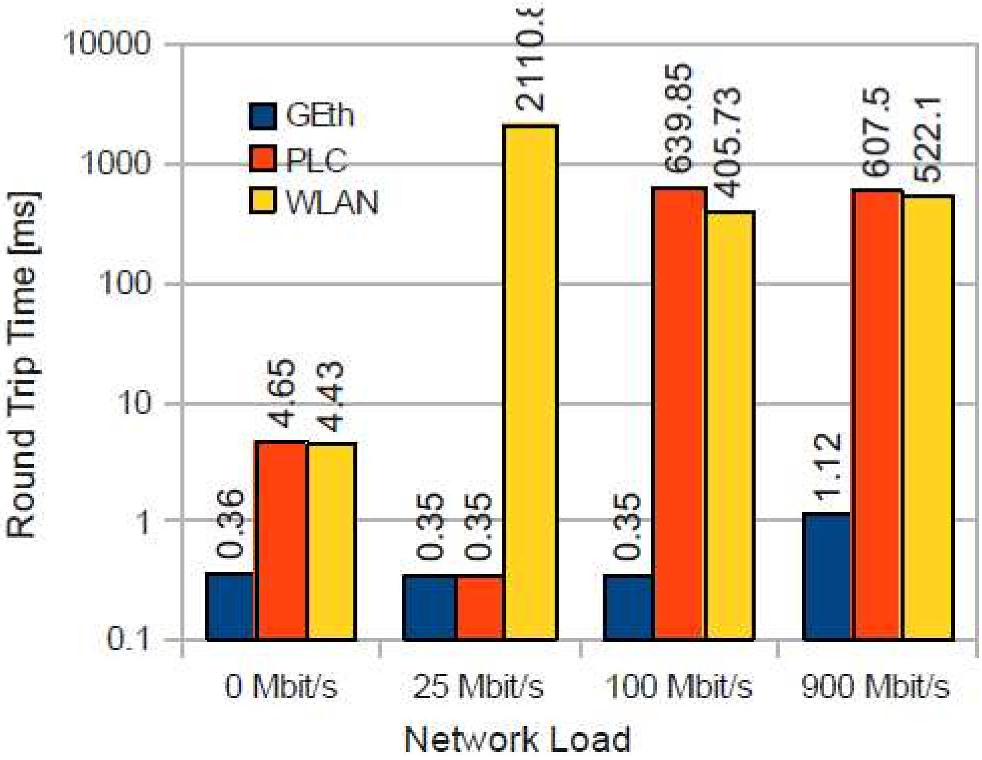

Figure 10: A testing example of the overhead delay

3.3.2 Single technology handover

Communication Technologies: Wi-Fi 802.11g, Gigabit Ethernet 802.3z, Power Line Communication (PLC).
Objective: Test the Inter-MAC reaction when a link breakdown is detected. To estimate the time needed to handover to another technology we consider the network topology presented in Figure 9. In all cases one has to measure the handover time from Gigabit Ethernet to the other links (Gigabit Ethernet, Power Line Communication and Wireless LAN 802.11b). Each time link degradation is detected one has to write a time-stamped message to a log file. The time-stamping precision must be to the microsecond. Similarly, the instants of the updates to the forwarding table must be logged, so one can measure the handover time as the difference between these two values. Thus one has to estimate the handover time as the period from the detection of link degradation to the forwarding table update. The expected values are reported in the table below. Figure 11 depicts the output results of an example test [OMD7.4].

Technology	Handover delay (ms)
Gigabit Ethernet	2
PLC	20
WLAN	80

Table 6: Inter-MAC handover delay

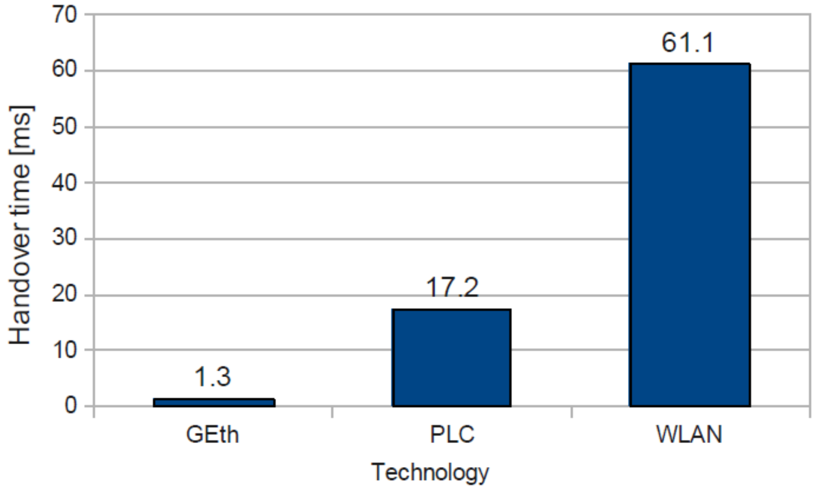

Figure 11: A testing example of handover delay

3.3.3 Load balancing

Communication Technologies: Wi-Fi 802.11g, Gigabit Ethernet 802.3z, Power Line Communication (PLC)
Objective: Test the Inter-MAC load balancing functionality. We expect that Control Plane is able to split (on per flow policy) the video and file transfer flow in order to improve the download response time while maintaining the same bandwidth for the video flow. Especially to avoid congestion on scenarios for which one route can be saturated by the bandwidth of the overall flows.

3.3.4 Inter-MAC path selection protocol performance

Communication Technologies: Wi-Fi 802.11b, Fast Ethernet 802.3u, Gigabit Ethernet 802.3z, Power Line Communication (PLC)
Objective: The main purpose of this test will be to measure a single path set-up time of the path selection protocol between two nodes in an OMEGA network. Given the fact that timing constraints strongly depend on the application type and on the network topology one can refer to the table below [OMD51] to verify that the OMEGA network created fulfills these constraints.

Service class	Average throughput; peak rate	Delay	Jitter	Packet loss
Emergency	Some kbps max	Should be real time		Critical criterion
Conferencing	2 CBR flows; some 10 Kbps	Real time constraint < 10 to 100 ms	< 20 ms	< 10^{-3}
	2 flows from 128 Kbps to 4 Mbps	20 to 30 ms	< 10 ms	< 10^{-5}
Streaming	CBR flow, some kbps for audio applications, from2 to 50 Mbps or 600 Mbps in the future for video applications	Video- audio sync constraints < 400 ms		< 10^{-5}
	< 50Kbps	Audio 3D sync < 300 ms		
Less sensitive traffic	Flow < 100 Kbps	< 50 ms	< 1ms	< 10^{-3}
Best effort	Variable from some kbps to 1,5 Mbps			
	VBR flow up to the Gbps			
other	Mono-directional flow 1Gbps or more	< 400 ms	< 1 ms	

Table 7: Relation between OMEGA service classes and QoS requirement

4 Outlook

This section gives some ideas about further work on Inter-MAC.

4.1 Inter-MAC Commercial Exploitation

Commercial exploitation of innovations often leads to a hen-and-egg problem. Before wide deployment takes place costs for components remains high. Investment in cost optimized solutions is not made. In case of Inter-MAC the need for self-configurating, heterogeneous home networks is already there. OMEGA has proven the feasibility of the concept and triggered standardization. The Inter-MAC specification team has paved the way for implementation by separating data and control plane functions.

Standardization is essential for adaptation by the industry. Chip and box manufacturers today are under heavy cost pressure. Hardware is sold at low price or given away for free together with service subscription. Hence even small investments are not undertaken unless commercial exploitation is assured. In practice for Home Gateway devices the immediate customer request for a function is required. Hence all preconditions for commercialization are in place. Possible hurdles are outlined in the next section.

4.1.1 Performance and Cost Estimation

Today's Home Gateways mainly consist of one main chip with optionally some peripheral devices. The main chip is a complex SoC containing among others a standard processor core such as MIPS or ARM and HW accelerators to speed-up processing intensive functions such as de/encryption, video trans-coding etc. The Inter-MAC data plane functionality typically would map to a HW accelerator, while the control plane would be done by the standard processor core. Additional HW cost is expected to be low, as the Inter-MAC HW accelerator functionality is small. The additional memory needed for code and data of the control plane is no problem for the ever increasing DRAM sizes. But a chip re-design requires large investment for mask and wafer production cost. It must be assured that a sufficient number of consumers will subscribe to a new, “OMEGA” service of self-configuring home network.

In addition, significant effort is expected from the Box Manufacturer to transform the available control plane SW into a commercial SW, which must be stable under all conditions. One approach could be to optimize the SW for one specific target customer group together with an operator. The acceptance of this first user group would encourage further developments. Another approach could be to provide the OMEGA solution as open source SW to allow a large community of developers to improve it.

4.2 Standardization

The Inter-MAC demonstrator is a proof of concept that a 2.5 convergence layer is advantageous for home network and moreover can be achieved without changing the existing technologies. Although it is a first step from research program, it is not sufficient to deploy commercial products.

During the year 2009, we have seen some chipset companies starting initiatives to hybrid wireless and wired technologies for home networking:

- Atheros (WiFi chipmaker) acquired Intellon (Homeplug chipmaker) in Sept 2009 to aggregate the increasing HD video flows.
- Gigle Networks (Power line chipmaker) has shown at CES2009 a prototype integrating multi-PHY (WiFi, Ethernet, Homeplug) for a unified Home Network called “Invisible Home Network” based on a proprietary protocol (Xtendnet).

A presentation of Omega-WP5 has been done in July 2009 during the IEEE 802.1AVB meeting in San Francisco. The feedback was reserved because of the complexity and the cost of a multi-interfaces approach.

In 2010, the industrial ecosystem seems to be ready to work on a standardized program to move forward the home network vision.

As many supported home network technologies are specified in IEEE (802.3, 802.11, P1901), this Standard-Developing Organization was a good candidate to propose this approach. A Project Authorization Request was prepared and sent to ComSoc sponsor. The approval of a new P1905.1 working group was done in Nov 2010 (http://grouper.ieee.org/groups/1905/1/).

5 References

[OMD54] OMEGA Deliverable D5.4, “Inter-MAC Protocols Performance Report”, December 2010, available on http://www.ict-omega.eu/fileadmin/documents/deliverables.html

[OMD53] OMEGA Deliverable D5.3 “Inter-MAC protocol entities interfaces specification”, January 2010, available on http://www.ict-omega.eu/publications/deliverables.html.

[OMD52] OMEGA Deliverable D5.2 “OMEGA Inter-MAC architecture design”, June 2009.

[OMD51] OMEGA Deliverable D5.1 “OMEGA Common Semantic”, November 2008, available on http://www.ict-omega.eu/publications/deliverables.html.

[OMD11] OMEGA Deliverable D1.1 "Final usage scenarios report", September 2008, available on http://www.ict-omega.eu/publications/deliverables.html.

[OMD74] OMEGA Deliverable D7.4 " Omega Platform based on Linux PCs ", October 2010, available on http://www.ict-omega.eu/publications/deliverables.html.