WORTECS

H2020-ICT-2016-2
RIA

Project-ID: 761329
WORTECS
Networking research beyond 5G

D3.3
Hybrid Network architecture for Tbps transmission and associated metrics definition for radio interface selection

Contractual Date of Delivery:	2019, August 31th
Actual Date of Delivery:	2019, Agust 28th
Editor(s):	Marcin Brzozowski (IHP)
Author(s):	Marcin Brzozowski (IHP), Vladica Sark (IHP)
Work package:	WP3
Security:	PU
Nature:	Deliverable
Version:	version 2.1
Total number of pages:	23

Abstract
This deliverable introduces the major solutions to integration heterogeneous technologies into a single network. We show major challenges of future ultra-high data rate networks and discuss various solutions to integrate these technologies.

Keyword list
Heterogeneous networks, high throughput, low delay, quality of service

Executive Summary

This task focuses on the integration of various wireless heterogeneous technologies. In recent years, several research efforts focused on heterogeneous networks. For example, we studied and developed the InterMAC protocol in our previous FP7 EU OMEGA project. The InterMAC integrated various networking technologies (mainly Ethernet, Wireless LAN, and 60 Ghz radio) into a single network, providing huge benefits to the end users. For example, the InterMAC configures the network and carries out seamless handovers in case of network problems. Such integrated heterogeneous networks provide high throughput and reliability, and low delays. However, future ultra-high data rate technologies pose new challenges and therefore past solutions, such as the InterMAC, may not support them. The major challenges include extremely fast frame processing to support data rates of Tbps and huge memory requirements. Therefore, new protocols for future ultra-high data rate networks need hardware-based implementations based on parallel processing.

Based on our previous work, on the InterMAC, we work on a new communication layer between the data link layer (layer2) and the network layer (layer3), and name it Layer2.5. This layer, similarly to the InterMAC, consists of the data and control planes. In this project, we mainly focus on the data plane and its challenges. The Layer2.5 provide many benefits to the end users, such as vertical handovers, load balancing and also splitting data across several connections to increase the throughput, lower latencies and also increase the security.

Since there are usually several paths available from the source device to the destination, the Layer2.5 needs to select one of them based on well-defined criteria, for instance, sending the data flows with the highest priority first. The Layer2.5 can also consider the requirements of the flow, for instance expected throughput and delay, when selecting paths for transmissions. The Layer2.5 monitors also the performance of communication links mainly by sending probe frames.

We implemented the preliminary version of the data plane on the FPGA platform, which includes four 10 Gbps interfaces. Our design support data rates up to 8.7 Mbps of a single lane.

Now we have a working implementation of data plane and in the next steps, we are going to carry out various experiments with our Layer2.5. Later, we will focus also on the control plane and various policies to select the best path to the destination.

List of Authors

First name	Last name	Beneficiary	Email address
Víctor	Guerra	ULP	victor.guerra@fpct.ulpgc.es
José	Rabadán	ULP	jose.rabadan@ulpgc.es
Rafael	Pérez	ULP	rafael.perez@ulpgc.es
Marcin	Brzozowski	IHP	brzozowski@ihp-microelectronics.com
Vladica	Sark	IHP	sark@ihp-microelectronics.com
Bastien	Bechadergue	OLD	bastien.bechadergue@oledcomm.net
Guillaume	Vercasson	BCOM	guillaume.vercasson@b-com.com
Dominic	O'Brien	UOXF	dominic.obrien@eng.ox.ac.uk
Ravinder	Singh	UOXF	ravinder.singh@eng.ox.ac.uk
Olivier	Bouchet	ORA	olivier.bouchet@orange.com

Document History

First name	Last name	Version	Comments
Marcin	Brzozowski	1st of July 2019	First draft
Marcin	Brzozowski	12th of August 2019	V1 revised
Marcin	Brzozowski	23th of August 2019	Some text corrections
Marcin	Brzozowski	28th of August	Final version ready

List of Acronyms

Acronym	Meaning
FPGA	Field programmable gate arrays
Gbps	Gigabits per second
HetNet	Heterogeneous Networks
IP	Internet protocol
LQI	Link quality indicator
LAN	Local area network
MAC	Medium access control
Mbps	Megabits per second
PER	Packet error rate
PHY	Physical layer
QoS	Quality of service
Rx	Receive
Rx	Receiver
RSSI	Receiver signal strength indicator
RAID	Redundant Array of Independent Disks
RSVP	Resource Reservation Protocol
SFP	Small form-factor pluggable
SDN	Software-defined networking
Tbps	Terabits per second
TCP	Transmission control protocol
Tx	Transmit
UDP	User datagram protocol
VR	Virtual reality
WiFi	Wireless fidelity

Table of Contents

List of Figures

1 Introduction

Nowadays, home users benefit from various communication technologies, wired and wireless, such as Wireless LAN, Bluetooth, Ethernet, and Power Line Communication. However, this variety of networking technologies brings also difficult challenges, since home users have to configure and maintain such complex networks. Further, typical communication protocols cannot exploit all benefits of these networks. Therefore, in our previous project (ICT OMEGA, 2008-2011 [1]) we introduced a new InterMAC protocol for home heterogeneous networks, which became the IEEE 1905.1 [2] standard in year 2013.

We envision that future home networks will also include several different technologies. However, they will support much higher data rates than today networks. Further, we also expect new advances in wireless communication, leading to ultra-high-speed wireless networks at home. Obviously, home users will have to cope again with the problems of complex heterogeneous networks, or use the InterMAC protocol. However, these future ultra-high-speed networks will pose new challenges, and therefore we have to adapt the InterMAC.

In this work, we present the new challenges of future wireless home networks and examine various techniques to deal with them. We introduce the architecture of future InterMAC-like protocols and present our preliminary implementation based on the FPGA platform. We call this new implementation the Layer2.5.

To effectively select links for transmission, the Layer2.5 needs to determine how good they are, based on various link metrics. In this deliverable, we define most important metrics and show how to estimate them.

This document starts with the introduction to the InterMAC protocol, its history, major challenges it faces and the architecture. Then, we describe new challenges to the InterMAC for future wireless ultra-high-speed networks. In the next section, we introduce the architecture of Layer2.5, the successor of InterMAC for future networks, and major benefits it provides. This section presents also universal metrics for path selection and the way to estimate these metrics. In the next section, we present our preliminary implementation of the Layer2.5 based on the FPGA platform. Finally, we conclude this work.

2 InterMAC and challenges for wireless Tbps networks

We base our work on heterogeneous networks on our previous research activities, mainly on the OMEGA project. In this project, we developed a new communication sublayer, named InterMAC, which responds to challenges of home heterogeneous networks.
In this section, we briefly introduce the InterMAC protocol and also show major challenges for future, high-speed wireless networks. The next version of InterMAC must meet those challenges to support future heterogeneous networks.

InterMAC motivation

In year 2008, we envisioned future home networks based on several heterogeneous technologies, as depicted in Figure 1. At that time, home networks included mainly Ethernet, Wireless LAN and Powerline Communication technologies. Further, these networks may also integrate home automation sensor networks, for example IEEE 802.15.4 and ZigBee.

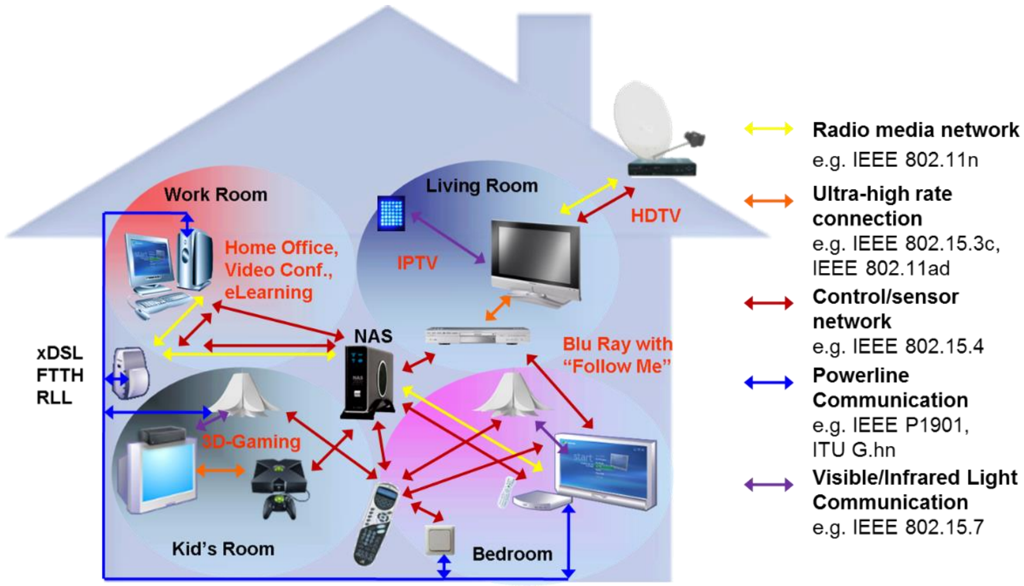

Figure 1 Home networks include several heterogeneous networking technologies connected into a single network

Such technologies working as a single integrated network provide many benefits but also pose new challenges to home users:

1. Complex manual configuration
   With several network technologies at home, the users must configure the network manually. It requires a huge effort for typical home users and they often need the support of experts.

2. Problems with dynamic networks
   Home heterogeneous networks may suffer from various run-time problems, leading to performance degradation or communication network outage. Because of high complexity in home heterogeneous networks, home users cannot easily find and solve network problems, and again they need the support of experts.

3. Inefficient use of network resources
   Usually there are several alternative paths between senders and receivers in heterogeneous home networks. These paths differ in delays, reliability, throughput, etc. However, typical LAN protocols for

home networks cannot select the best path for ongoing transmissions. In addition, home networks cannot split traffic among different paths and support load balancing.

4. Lack of QoS control
   Usual LAN protocols cannot effectively restrict the network access for home users. Although modern gateways and WiFi access points introduce basic QoS features, some connections may still suffer from excessive network load. Therefore, home heterogeneous networks needs techniques to control the network access and guarantee the expected QoS for home users.

Obviously, home users cannot enjoy all benefit of heterogeneous networks because of problems mentioned above. Therefore, we worked on the InterMAC solution, presented it the following paragraph.

InterMAC solution

To deal with the various challenges of home heterogeneous networks, listed in the previous paragraph, we introduced a new communication sub-layer and named it InterMAC.

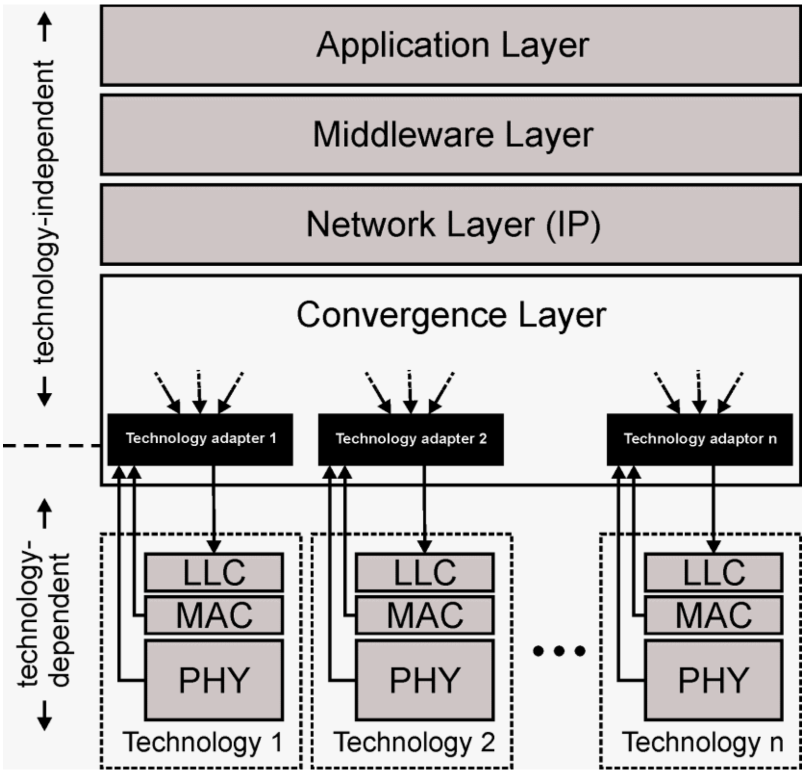

Figure 2 The new convergence Layer (InterMAC) is located between the Network and DataLink Layers

The InterMAC sublayer hides the complexity of all communication technologies, such as Ethernet, Wireless Lan, and Power Line Communication. Therefore, we placed the InterMAC atop of various technologies, just above the data link layer (see Figure 2). The upper layers, mainly the network layer, are not aware of the underlying communication technologies. Therefore, the network layer finds only a single communication interface, the virtual InterMAC interface, although there are several physical technologies available. Hence, the network layer just passes outgoing data to this single interface. Then, the InterMAC decides which underlying communication technology to use for outgoing frames, based mainly on the current link performance. Further, when some communication links suffer from performance degradation, the InterMAC uses other, “better” links. In the following, we explain how the InterMAC deals with the challenges of heterogeneous home networks listed in the previous paragraph:

1. Complex manual configuration
   The InterMAC configures the heterogeneous networks on its own. It discovers all links available and discovers paths from sources to destinations. It simplifies the use of heterogeneous home networks,

since the home users do not have to configure anything.

2. Problems with dynamic networks
The InterMAC monitors continuously all links and in case of network problems it looks for another path to the destination. Since it takes only a few milliseconds to change the path [3], the home users do not notice the handover, that is, switching from one technology to another.
Further, the InterMAC detects also new links added to the network and uses them for transmissions.

3. Inefficient use of network resources
The InterMAC can use several available links at the same time, depending on the forwarding policy. For instance, it supports load balancing by sending various network streams over different paths. Further, the InterMAC examines the QoS requirements of ongoing traffic (e.g. latency or throughput), and then selects the appropriate path to the destination to meet these requirements.

4. Lack of QoS control
Each flow transmitted with the InterMAC have its priority and QoS requirements. The InterMAC considers these both features when discovering paths throughput the network. However, in case of network overload some frames cannot be transmitted. In this case, the InterMAC tries to transmit highest priority stream first, and it sends the remaining data only if there is enough network capacity. In this way, the InterMAC ensures that the transmission of high-priority streams is not affected by low priority data.

Clearly, to exploit the benefits of the InterMAC, each network device must implement this new layer 2.5. However, not all devices can update their software stack to include the InterMAC. Therefore, we introduced also features to allow backward compatibility with so called legacy devices, i.e. devices that do not include the InterMAC. In such cases, the legacy device is connected to the InterMAC Proxy, which resembles a network bridge. The proxy captures all traffic from legacy devices, adds required frame headers and passes data to the network. In this way, also the devices without InterMAC gain benefits of home heterogeneous networks.

Challenges to wireless Tbps networks

We started working on the InterMAC in 2008. During this time, the fastest technology at home was Gigabit Ethernet, whereas Wireless LAN supported a data rate of 300 Mbps, and the InterMAC had to support such data rates.
Each network device that implemented the InterMAC had to carry out extra packet processing, mainly adding and updating of InterMAC headers, and looking up the forwarding table. To reduce the frame processing delays we intended to develop the InterMAC as a hardware implementation, on FPGA platforms. In the end, however, the software implementation of InterMAC supported a data rate of 1 Gbps. Further, based on our calculations [4] the software-version of the InterMAC could support data rates up to 1.5 Gbps. Unfortunately, we were not able to verify it because of missing technologies that support data rates higher than 1 Gbps.

Each year we observe advances in wireless communication, also in home networks. While working on the InterMAC we also investigated and developed our wireless communication technology based on the 60 Ghz frequency band. Nowadays such technology achieves a data rate of multiple Gbps, and in near future we expect even faster wireless communication. For instance, the standard IEEE 802.11ad [2] supports data rates of 7 Gbps, and the upcoming standard IEEE 802.11ay will support the throughput of 100 Gbps [5].

These advances in wireless communication bring new challenges to heterogeneous networks. Therefore, we are investigating the problem of heterogeneous networks again, this time focusing on future, ultra-fast wireless technologies. The major difference between past and future wireless technologies is the data rate. While developing the InterMAC we had to support technologies up to 1 Gbps, whereas we expect data communication with a data rate even 1000 Gbps (1 Tbps). Since our InterMAC supports data rate up to 1.5 Gbps, we need new ideas and implementations:

1. Hardware implementation
Clearly, the current InterMAC software implementation is too slow to support future, ultra-fast networking, unless some hardware accelerators are provided. Therefore, in this project we consider mainly a hardware implementation to deal with such high data rates. However, some parts of the

InterMAC can still be realized as software, mainly the control plane.

2. Parallel processing
   Previously we worked on the MAC layer for 100 Gbps wireless networks [6] and we realized that hardware can also be too slow to support such high data rates. Therefore, we had to split the frame processing into several parallel lanes to deal with so high data rates. Similarly, we expect such parallel processing in the InterMAC for future wireless networks to support such challenging frame processing.

3. Handover and memory constraints
   When a link stops working or its performance degrades, the InterMAC switches traffic to another path (handover) to support service continuity. The total handover time includes the following steps: detection of link problems, discovery of the alternative route, and the switching time. The InterMAC needs about 1.3 ms to discover a new path for 2-hop networks based on Gigabit Ethernet, and 61 ms on wireless LAN [4]. During this time, the data plane must store all frames in the local memory and send them after discovering a new path. However, future wireless technologies will support very high data rates, and therefore the data plane needs a huge amount of memory to store packet while waiting for a new path. Therefore, we have to find new ways to deal with packet buffering.

Although we had already implemented an extra communication layer – the InterMAC – to integrate heterogeneous technologies into a home network, the future, ultra-high-speed wireless networks pose new challenges. Therefore, we have to examine these new challenges and based on our previous experience gained with the InterMAC provide novel approaches to support future wireless heterogeneous networks.

3 Layer2.5 Architecture

This section gives a brief overview on the Layer2.5 architecture based primarily on the InterMAC, our protocol for heterogeneous home networks. Apart from the architecture, we present also link metrics used by the Layer2.5 to effectively select transmission links.

Heterogeneous Networks

A heterogeneous home network includes several various technologies integrated into a single network. Figure 3 shows the WORTES demonstrator, which is an example of heterogeneous networks. In this scenario, the Virtual Reality (VR) server sends data to the VR headset using two wireless communication technologies: optical and radio. However, the VR server is not aware of underlying technologies; it just passes data to the HetNet switch, which implements the new Layer2.5. This layer selects the best links or paths to the destination, based on current link occupation, QoS requirements of data streams, etc. Further, if any of these previously selected links suffers from network problems, the Layer2.5 diverts the traffic to another link.

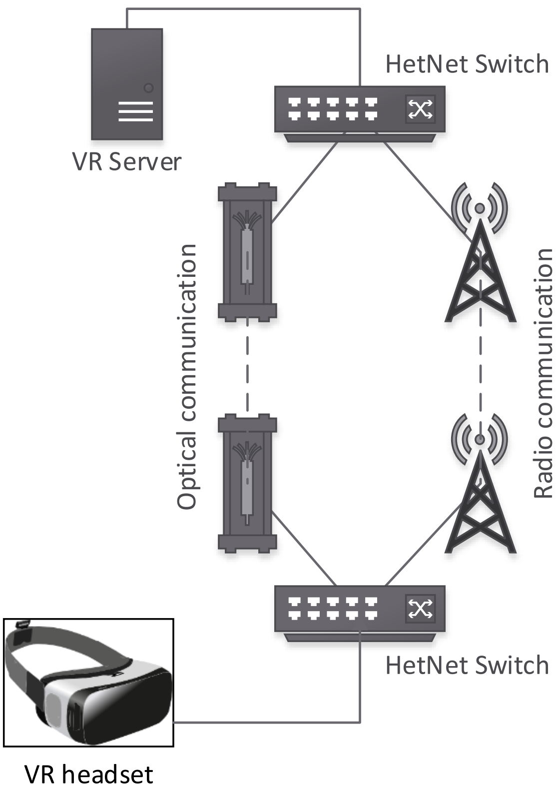

Figure 3 The WORTECS Demonstrator represents a simple heterogeneous network

Complex heterogeneous networks include more communication technologies, switches, and also multi-hop paths between sources and destination. The Layer2.5 supports also such complex scenarios in a similar way like the InterMAC does. However, in this project we mainly focus on challenges of ultra-high-speed wireless networks and multi-hop communication is beyond the scope of this work.

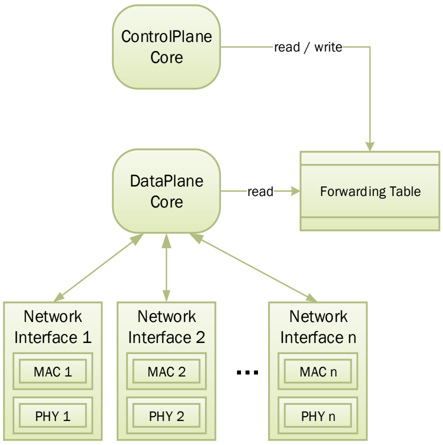

Figure 4 Separation of DataPlane and ControlPlane allows flexible implementation, for instance, in hardware and software or on different devices

Data Plane and Control Plane

The Layer2.5 performs two major tasks: fast data forwarding and path discovery to the destination. Based on these tasks we split the Layer2.5 into two entities, similarly to our InterMAC or Software Defined Networking [7]: the data plane and the control plane, explained below.

Although each network device, such as servers, laptops or smartphones, may include the Layer2.5 in their protocol stack, in this document we consider only the Layer2.5 implementation in HetNet switches, as depicted in Figure 3. In this case, any network device connected to the HetNet switch benefits from the Layer2.5.

Data plane

The data plane works like a network switch. After receiving a frame on a certain network interface, the data plane looks up the forwarding table to find the outgoing network interface for this frame. Then, the frame is forwarded using the selected interface.
Apparently, the data plane must process each frame quickly enough to support high data rates. Therefore, we minimized the number of operations in forwarding and moved all other (not time-critical) operations to the control plane. However, there is still a risk the data plane would be too slow for Tbps networks. Hence, the data plane must processes many frames in parallel, as we did in our data link layer for 100 Gbps wireless communication [8].
The data plane creates also probe frames to determine the link quality. It periodically sends probe frames over each available link, and waits for responses. After getting a response from the neighbor, the data plane estimates the round trip time to this neighbor and updates the link table. To determine if links are still working, the data plane stores the last time when a frame was received on each interface. Then, it marks some links as not working, when no frame arrived within a predefined time.

Control plane

The control plane mainly maintains the forwarding table, which the data planes uses when forwarding frames. To update the forwarding table, the control plane needs to discover the path to the destination. In our previous project, we developed and implemented a path selection engine [9] for the InterMAC and it resembles common routing protocols. Further, the control plane may also monitor underlying links for potential problem and triggers then handover. As stated before, in this project we primarily focus on the data plane and a detailed study of control plane protocols is beyond the scope of this project.

Major benefits of Layer2.5

In this section, we introduce the major benefits of using the Layer2.5 in heterogeneous network and the protocols used to provide those benefits.

Automatic selection of links/paths

Although there are many links and technologies available, the end user does not need to bother with the selection of network links. The Layer2.5, in our case the HetNet switch, receives frames from the end user and selects the best link for transmission, based on current link occupation, data stream requirements, etc.

Vertical handover

The Layer2.5 allows a seamless vertical handover, that is, the HetNet switch diverts the network traffic to another link (see Figure 5) and the end user does not notice it. For instance, in case of problems with wireless optical transmission, the switch start sending data using the radio. The end user keeps receiving frames and does not notice the change in the transmission technology. Further, the Layer2.5 carries out handover even when the current link is still working, for instance, when it discovers a better link.

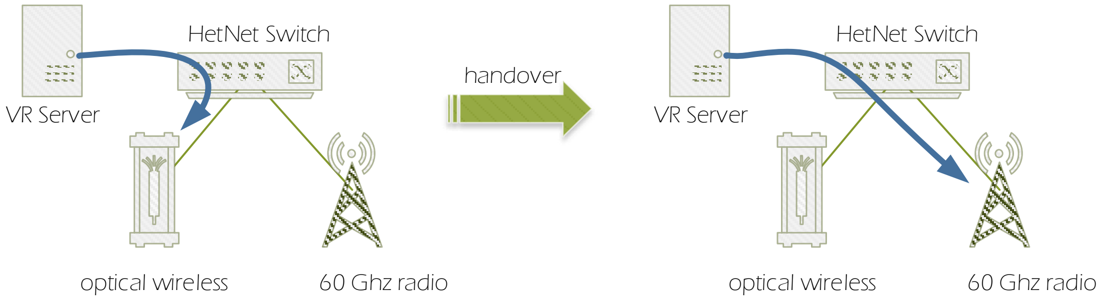

Figure 5 The Layer2.5 may decide to switch the current link to another (handover), for instance, on problems with the current connection

Each handover process includes two major steps:

1. Event discovery
   The Layer2.5 detects an event that triggers the handover, for example, the current link suffers from performance degradation and cannot support the data flow requirements.

2. Link change
   As soon as the Layer2.5 detects an event and triggers the handover, it looks for an alternative link and finally updates the forwarding table of the switch (see Figure 5). In case of multi-hop paths to the destination, the Layer2.5 needs to discover the complete path and update forwarding tables of the corresponding switches.

As previously mentioned, to support seamless handover the data plane needs to store frames in the local memory while discovering a new path but it requires a huge amount of memory, for example, more than 12 Gigabyte for 1-second buffering at the data rate of 100 Gbps.
When the data plane does not buffer frames during the handover, it leads to packet losses. Depending on the scenario and network protocols, the sender transmits lost frames again, leading to extra delays and huge memory requirement at the end user.

Some applications accept packet losses and extra delays caused by retransmission, but critical data streams do not. For such challenging applications, the Layer2.5 uses more complex policies when performing handover. For instance, the Layer2.5 introduces an intermediate step, apart from the handover decision. In this case, the Layer2.5 groups link problems into two categories, minor and major, just like the yellow and red lights.

1. Minor problems (yellow light)
   These link problems do not affect ongoing transmissions and the Layer2.5 delivers all packets without delays. However, when the current link performance gets worse, there is a risk of connection problems in near future. To cope with potential future connection problems, the Layer2.5 starts sending frame copies over another alternative link or more links. In case the current links suffers from more severe problems that lead to packet losses, the receiver still gets packet copies transmitted over alternative

links.

2. Major problems (red light)
Here the performance of the current link is below the acceptance level. For example, the link is overloaded or packets are not received at all. Clearly, it is the case when the Layer2.5 usually performs handover and so does it now as well. However, if this major problem was preceded by a minor problem, the receiver is already getting frames via alternative links. In this case, the Layer2.5 only removes the non-working link from the forwarding table. Unfortunately, when the major problems occurs immediately, without minor problems before, the Layer2.5 runs into the same problems as before: packet losses and delays.

Clearly, this extra policy with minor errors works only with gradual performance degradation. In case of sudden link problems, the application still suffers from the problems previously mentioned. In this case, the Layer2.5 should always send copies over two or more alternative links for critical applications.

Load balancing

Typical home heterogeneous networks, without the Layer2.5, do not monitor the performance of available links and do not select specific paths for transmissions. It may overload some links, although some other network connections are not used at all. To deal with this problem, the Layer2.5 introduce load balancing to home networks, as depicted in Figure 6.
In short, the Layer2.5 selects links for transmissions based on current link occupation, delays, etc., and also on user-defined policies. In the default policy, the HetNet switch distributes traffic among several links to avoid overload problems (see Figure 6).

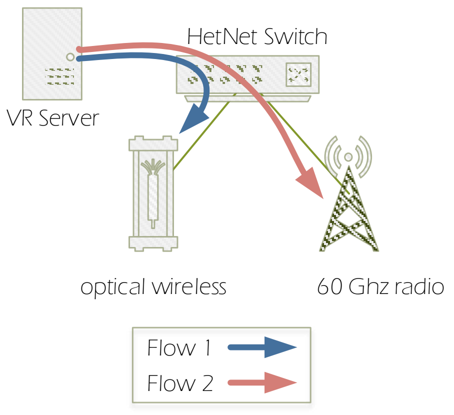

Figure 6 Load Balancing distributes network traffic among many links

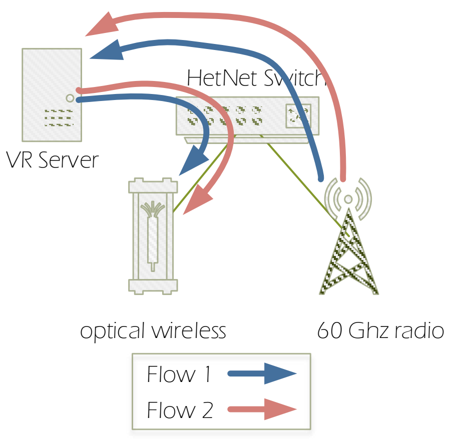

Figure 7 Another scenario with Load Balancing: uplink and downlink traffic use different links

In general, HetNet switches send packets belonging to the same data stream (e.g. a movie stream) using the same connection. It avoids extra work needed at the receiver to put all frames in order in case they are transmitted over various links. Nonetheless, some scenarios may benefit from splitting packets across several links, explained in the next paragraph.
The Layer2.5 allows also to transmit data asymmetrically (see in Figure 7). For instance, downlink packets traverse some specific links (optical wireless in Figure 7), whereas uplink frames follow another path (60 Ghz radio in Figure 7). This approach allows to use asymmetric connections in home heterogeneous networks. However, it requires extra care since the sender does not get immediate feedback from the receiver, making retransmission and link maintenance more complex.

Split/Merge

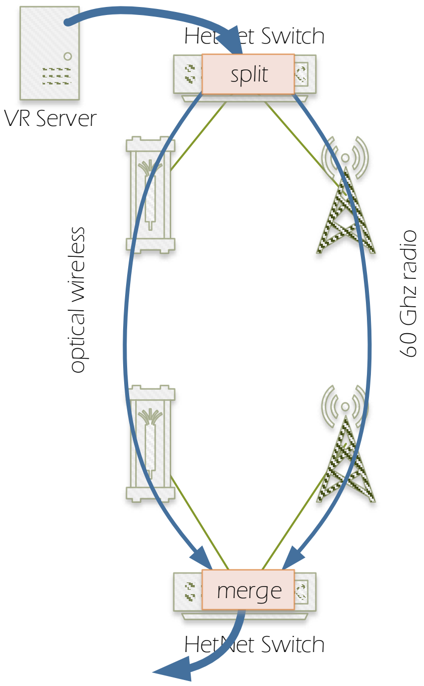

Figure 8 The Layer2.5 may split ongoing transmissions to several links and merges them at the receiver

Then Layer2.5 can send a single data stream over several links, as depicted in Figure 8, providing the following benefits:

1. Higher throughput
   Since some frames traverse one link, and the remaining frames the other one, the total throughput sums the capacity of both links. In this way, the Layer2.5 supports also applications that require data rates higher than any available technology. Clearly, the Layer2.5 can use more than just two links and achieve even higher data rates.

2. Lower delay and PER
   When frames traverse only a single path to the destination and some packets are lost, the sender can transmits them again but they arrive delayed. The Layer2.5 may reduce these delays by sending frame copies over two or more links in parallel. In case some packets are lost or delayed, their copies may still reach the destination on time. Further, sending extra frame copies may results also in a lower overall PER, provided at least a single frame copy reaches the receiver. Clearly, the price for lower delays and PER are extra frame copies transmitted on other links.
   Instead of sending frame copies over several links, the Layer2.5 may use so called Network Coding [10] when transmitting data. It resembles the RAID approach [11] for hard disk backup strategy. In short, even when one or more hard disks are broken, the user can still access all data, since it is replicated across all disks. Replication in terms of networking means appropriate data coding. That is, before transmitting data, the sender learns about the current replication policy (e.g. required PER or delay). Then, it puts extra redundancy in data stream (extra packets or extra data in each frame) based on the current link performance. In case some packets are lost or corrupted, the receiver can still recover data from received frames.

3. Higher security
   In general, an attacker can easily capture wireless transmission without being noticed, provided he has a suitable transceiver. To reduce the risk of data capture, the Layer2.5 can send frames over several links, just like in the scenario for higher data dates. In this case, the attacker must capture several wireless transmission, which makes it harder to capture all data. Further, appropriate data coding can make it

harder to read even a small part of data when capturing frames from a single link only.

The end user can use together all mentioned features and tailor solutions for his purposes. For example, if the user requires both higher throughput and low delays or PER, it splits data across several links (higher throughput). Further, to support lower delay the Layer2.5 adds redundant data to those packets (low delays/PER), presumably using network coding.

After the sender splits data into several sub-streams, transmitted over various links, the receiver must merge them to form a single data flow. However, since these links have different properties (e.g. delay, packet error rate), the receiver probably gets frames not in the right order and must rearrange them (put in the right order).
To allow packet reordering, the receiver need to figure out the right order. Therefore, the sender adds a sequence number to each packet. After getting a frame with a higher sequence number than expected, the receiver waits until the expected packet arrives. In general, it puts the frame with the higher sequence number into the local memory. The receiver handles this frame later, after it gets all missing frames.
Clearly, with ultra-high-speed data rates the receiver needs a huge amount of memory to store disordered frames. Therefore, there is a need for novel frame reordering schemes for such fast networks. For instance, the sender might split the frames according to the current delay of all links used for transmissions. In short, links with lower delays send more frames than the links suffering from higher delays, provided they still have enough capacity. Clearly, this complex problem must be addressed in future research activities.

OpenFlow and Software-Defined Networking

In software-defined networking (SDN) all network devices, for example switches, are flexible and allow changes in the way they work. The SDN approach make also use from data and control planes. There are also standards for SDN, for example the OpenFlow standard [12], which defines, among other things, the protocol to adapt the forwarding tables of network switches. Since there are already OpenFlow switches commercially available, i.e. switches that implement the OpenFlow standard, we can build our Layer2.5 idea atop of them. In this case, each HetNet switch (Figure 3) is an OpenFlow switch, and we can adapt its Forwarding Table with the OpenFlow standard. In this way, we do not have to develop the data plane, as the OpenFlow switches include it.
However, we decided not to use the OpenFlow standard in this project for the following reasons. We expect that frame forwarding with such high data rates requires several techniques and novel ideas, also in the forwarding core, and they may not be supported by the OpenFlow standard. For example, for some communication technologies we might need to aggregate several frames into large ones, and also adapt protocol headers, to improve the overall performance, especially for split/merge operations. Further, we also need to discover the link quality by sending probe frames and possibly by adapting data frames. However, we cannot do it easily with the current version of the OpenFlow standard.
Nonetheless, we expect that future ultra-fast heterogeneous network can benefit from next versions of the OpenFlow standard. However, these new version must include improvements that make such fast packet processing possible. The results of our project may serve as a starting point for new versions of the OpenFlow standard.

Metrics for radio selection

Link metrics provide information on how good certain connections, for example, the link delay or the maximal amount of data than can be transmitted (throughput). As the connection quality varies over time, for instance when a user is moving, link metrics need updating.
The Layer2.5 needs link metrics to select presumably the best link or path for transmissions. Further, the present link metrics allow also the Layer2.5 to learn about connection problems and trigger a handover.

Universal metrics

To select the link for transmission, the Layer2.5 compares the link metrics of all available connections. In this project, it examines the underlying radio technologies. Clearly, the links metrics of all available links must be of the same type so that the Layer2.5 can effectively compare them. However, various communication technologies may provide different metrics. For example, the radio estimates the Link Quality Indicator (LQI) or Receiver Signal Strength Indicator (RSSI) and these metrics are not available on other technologies. Therefore, the radio driver needs to convert these radio specific links to universal ones.
We looked at various applications for wireless technologies and studied their requirements for the network performance. Although these applications impose various number and type of requirements, the most common

are the following three: data rate, delay and jitter. Therefore, we use also these three metrics as universal metrics, which every underlying technology must provide.
Two out of three universal metrics, delay and jitter, the data plane measures on its own by sending probe frames. Therefore, the corresponding technologies need to provide only the data rate metric, derived usually from the channel modulation and the link quality.

Radio metrics
The radio can usually exactly provide the current data rate by examining the applied modulation scheme and channel coding. Further, by measuring the error vector magnitude it determines the link quality. In case the quality degrades, the radio may predict that in near future it will change the modulation and channel coding to cope with communication problems. Since it will results in a lower data rate, the radio may inform the Layer2.5 in advance about it. In this case, the data plane can also switch to another link in advance, if the predicted reduction in data rate would affect ongoing transmissions.

Link metrics estimation
As mentioned above, the Layer2.5 estimates some link metrics by sending probe frames. Therefore, the Layer2.5 keeps sending probe frames to neighboring devices and they reply with an acknowledgment. Then, the sender estimates the Round-Trip-Time, and the link delay to the next hop. To achieve high precision in the link delay estimation, the transmitter and receiver include hardware timestamp in probe frames. Apart from the link delay the Layer2.5 estimates also the jitter.
Further, the Layer2.5 detects also when some links stop working. If no frames were received within a predefined time, the Layer2.5 marks the corresponding link as not working and triggers others modules to find another path to the destination.
The Layer2.5 can also consider the Packet Error Rate (PER) of various links when selecting the best path. However, it is not trivial to estimate the link PER, since there are at least two causes that affect it. First, wireless communication is prone to the errors due to various problems, such as fading, interferences, etc. Second, when devices tries to send data with a higher data rates than the physical link capacity, it results in packet losses and a high PER. To make it more complex, some communication technologies include approaches to deal with packet losses and bit errors (such as retransmissions, various FEC schemes, and different modulations) so that it is not always clear at the Layer2.5 level what happens in the communication link. Clearly, the most accurate way to determine the PER and data rate is to get this information directly from the underlying technology. However, not every technology provides this information and the Layer2.5 needs to estimate these metrics on its own. We still have not found the best way to estimate the PER but consider the following idea, depending whether links are occupied or not:

1. For not occupied links, that is, not too many frames are transmitted on those links, the Layer2.5 keeps sending huge number of probe frames, without gaps between them. Since each frame includes another sequence number, the receiver can detects lost frames and estimate the PER.
2. If the link already transmits huge amount of data, the Layer2.5 cannot send extra probe frames to determine PER, as it may affect the ongoing transmission. Therefore, the Layer2.5 adds extra data at the end of the frame, mainly the sequence number. Since the receiver – the next hop towards the destination – implements also the Layer2.5, it reads the extra data appended to frames. Then, based on the sequence number included in this data it estimates the number of lost frames and the PER.

Since the Layer2.5 estimates several link metrics on its own (link delay and jitter, link availability and Packet Error Rate), these metrics are available for any network connection, regardless of the underlying communication technology. However, each technology must still provide the current throughput to allow the Layer2.5 to select a path to the destination.

Technology selection policies

Based on the metrics of all available links the Layer2.5 selects outgoing connection for each frame received. We defined the following policies for the selection of outgoing links.

1. Basic priority
   Each wireless technology has a predefined priority number. Then, the Layer2.5 select the technology with the highest priority number for transmissions, for example the 60 Ghz radio. Other links, the optical wireless for example, are only backups used when the primary link is not working.

2. Load balancing
   This policy distributes the network traffic among several available technologies. Similarly to our

InterMAC protocol, the Layer2.5 requires the remaining capacity metric. It tells how much extra data can be forwarded over the specific link without causing overload. It suffices for the technologies to provide information about the total link capacity and the Layer2.5 estimates the remaining capacity. With this policy the Layer2.5 does not split single flows. Therefore, packets belong to the same flow traverse the same link or path to the destination.

3. QoS driven
   This policy requires the knowledge of QoS parameters of network flows, such as delay or data rate. The Layer2.5 can learn these parameter, for instance, by using the Resource Reservation Protocol (RSVP) [13]. When the Layer2.5 knows both the QoS requirements of the flows and the current link metrics, it can distribute all flows among available links to satisfy the QoS needs, and also split some flows among several links if needed.

4 Implementation

As previously mentioned, our software-based InterMAC implementation cannot support data rates higher than 1.5 Gbps and therefore cannot be used in our demonstrator, and also in future high-speed wireless networks. Therefore, we intended to port the InterMAC to a hardware platform and decided to use an FPGA board, which provides fast packet processing and flexibility (we can update or add new features and upload a new image file). As the WORTECS project does not focus on the implementation, but rather on analysis of challenges in future networking technologies, we implement only the basic features of the Layer2.5, also a subset of ideas presented in the previous chapter. Our FPGA implementation includes primarily the data plane and the interface to the control plane, mainly to update the forwarding table. In the first phase, we update the forwarding table by sending frames to the data plane from our computer.

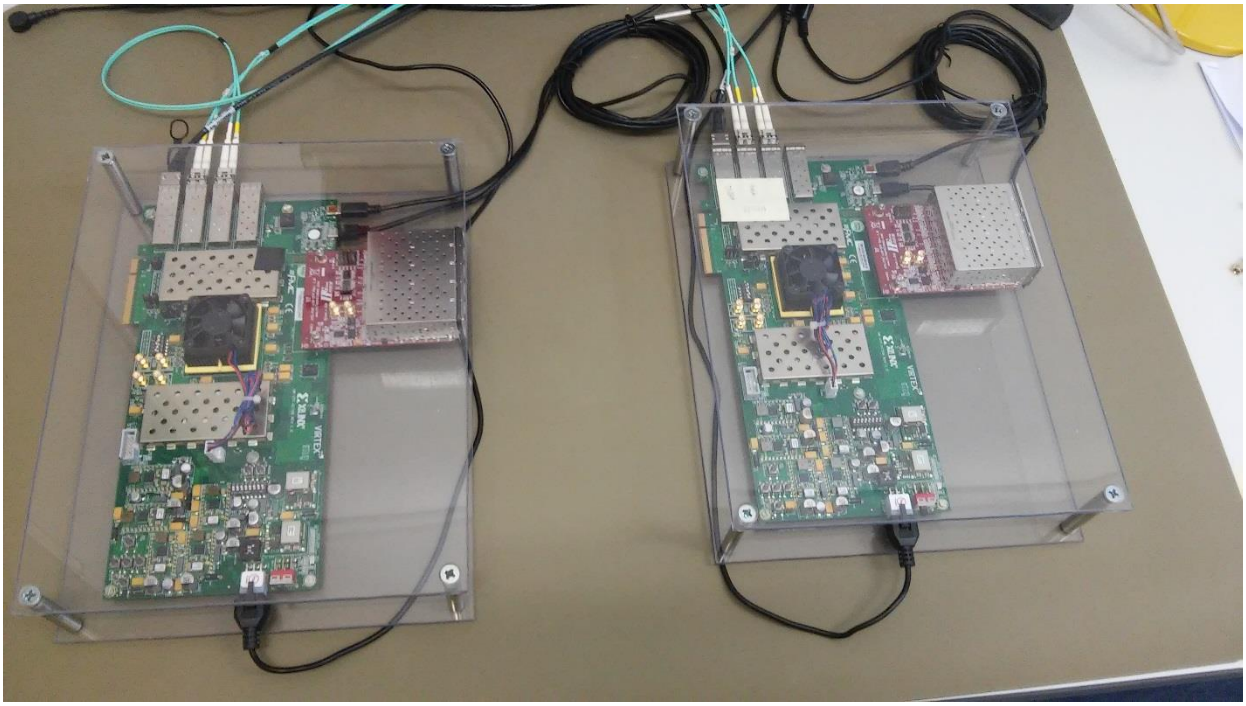

Figure 9 Two FPGA VC709 platforms used the implementation of Layer2.5 data plane

In this project we based our implementation on the Xilinx VC709 FPGA development board, which includes four 10 Gbps SFP+/SFP ports and to enable Ethernet-based communication we used the appropriate Ethernet PHY IP core of Xilinx.

Forwarding path
The complete Rx and Tx paths depicts Figure 10. To enable parallel processing of frame received on different interfaces, we created a separate path for each interface. Clearly, all these paths works in the same way.
Upon receiving a frame, the Ethernet IP core stores in in the Rx buffer. The frame handler checks this buffer continuously, and as soon as a new frame appears in the Rx buffer, the frame handler starts processing it. First, the handler gets frame headers (Ethernet, IP and TCP/UDP headers) and traverses the forwarding table to find the outgoing interface. Then, it passes the complete frame to the switch module, and finally to the outgoing port.

Control frames
We defined also some frames used to control the behavior of the FPGA-based data plane. Although we did not implement the control plane, the FPGA provides an interface to manage the forwarding table. For example, to add or delete entries to/from the forwarding table, we just send well-defined Ethernet frames from an external computer to the FPGA. In a similar way, we obtain various statistics from the data plane, for instance, the values of data rate or delay for each FPGA Ethernet port.
Probe frames are another type of control frames used in our protocol. As previously mentioned, the data plane uses them to estimate the delay of each link to the neighbor.

Preliminary results
We tested the performance of our FPGA-based data plane implementation using the iperf3 [14] application. We executed the application on a standalone computer equipped with a 10 Gbps Ethernet interface. The computer generated and passed TCP traffic to the data plane running on the FPGA platform. Upon receiving frames, the data planes looked up the forwarding table and forwarded frame to outgoing interface, connected to another standalone computer. The iperf3 estimated the throughput, delays and packet error rates of the the TCP stream transmitted between both computers via the FPGA platform.
After adapting our design, the data plane runs stable throughout several hours tests without major problems. We measured the throughput of a single Rx/Tx lane (see Figure 10) to be 8.7 Gbps, although it should support almost 10 Gbps. We examined the problem to find the cause of a lower throughput than expected. We figure out that our data plane implementation makes about 160 ns long pauses (25 clock cycles) between sending two consecutive frames, due to our design constraints. To achieve data rates of 10 Gbps there must be no gaps between frames. It requires adapting our design by adding intermediate frame handlers and buffers, but the estimated effort goes beyond the budget available in the WORTECS project.

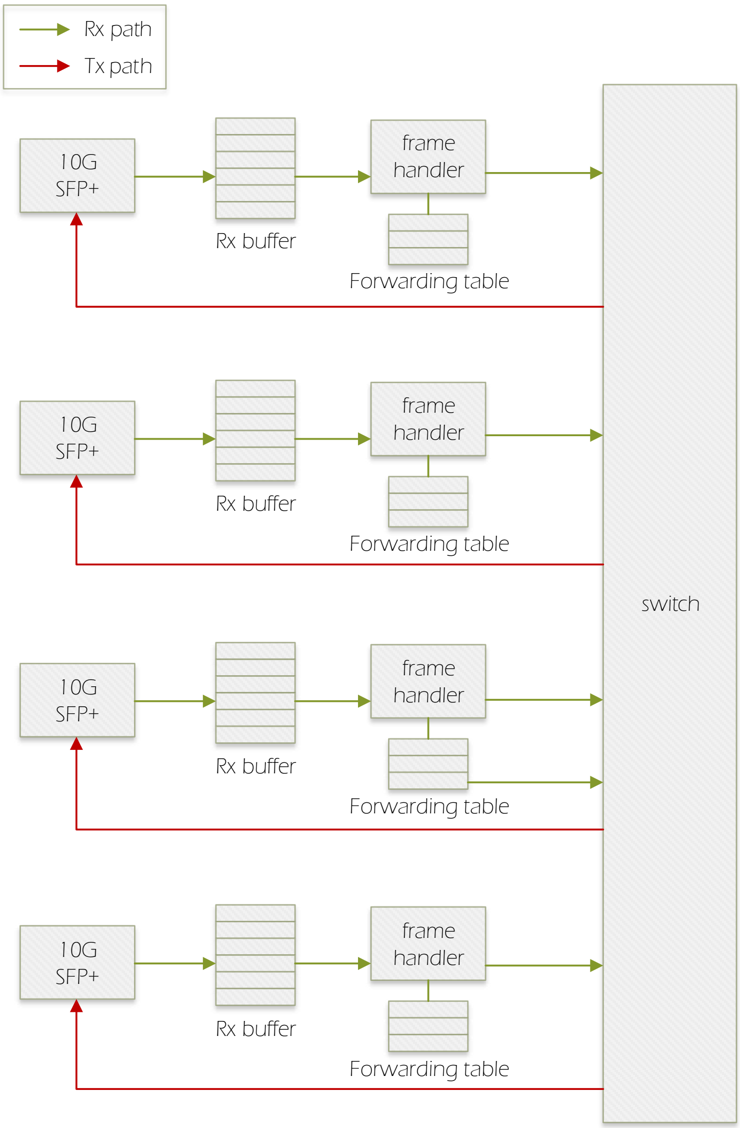

Figure 10 The data plane implementation on the VC709 FPGA platform. There are four 10 Gbps SFP/SFP+ interface available and therefore we created a separate lane for frame processing for each interface.

5 Conclusion and perspectives

In this work, we presented challenges of future wireless heterogeneous networks and reasons why previous solutions, mainly our InterMAC protocol, cannot be directly applied for these networks. We introduced new architecture and solutions that support future wireless networks, and named it the Layer2.5.

Our implementation, presented in this work, is the first step towards protocols for ultra-high-speed wireless heterogeneous networks. Soon we will start performing experiments with various novel wireless technologies and evaluate the performance of our implementation. We also have to verify the estimation of link metrics when using various wireless technologies with the Layer2.5.

We mainly focused on the data plane in our current work. In next step, we will also examine the control plane and various challenges of path and link selection, for example, various policies to choose the best link for transmissions. Then, we will have the complete Layer2.5: the data plane and the control plane.

Finally, we are going to examine whether OpenFlow switches can execute the data plane of Layer2.5. In case the current OpenFlow standard does not support it, we will study missing OpenFlow features. Then, this study may influence future versions of this standard for ultra-high-speed wireless networks.

6 References

References

[1] J.-P. Javaudin, M. Bellec, D. Varoutas, and V. Suraci, “2008 IEEE OMEGA ICT project: Towards convergent Gigabit home networks,” in *19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–5.
[2] *IEEE 1905.1-2013 - IEEE Standard for a Convergent Digital Home Network for Heterogeneous Technologies*: IEEE SA, 2013.
[3] M. Brzozowski, R. Jennen, S. Nowak, F.-M. Schaefer, and A. Palo, “Inter-MAC—From vision to demonstration: Enabling heterogeneous meshed home area networks,” in *2011 14th ITG Conference on Electronic Media Technology*, 2011, pp. 1–6.
[4] R. Kraemer, M. Brzozowski, and S. Nowak, “Reliable architecture for heterogeneous home-networks: The OMEGA I-MAC approach,” in *2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS)*, 2011, pp. 279–284.
[5] Y. Ghasempour, C. R. da Silva, C. Cordeiro, and E. W. Knightly, “IEEE 802.11 ay: Next-generation 60 GHz communication for 100 Gb/s Wi-Fi,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 186–192, 2017.
[6] L. Lopacinski, J. Nolte, S. Buechner, M. Brzozowski, and R. Kraemer, “100 Gbps wireless–data link layer VHDL implementation,” *Measurement Automation Monitoring*, vol. 61, 2015.
[7] D. Kreutz *et al.*, “Software-defined networking: A comprehensive survey,” *arXiv preprint arXiv:1406.0440*, 2014.
[8] L. Lopacinski, M. Brzozowski, R. Kraemer, and J. Nolte, “100 Gbps Wireless-Challenges to the data link layer,” 2014.
[9] O. Bouchet, A. Kortebi, and M. Boucher, Eds., *Inter-MAC green path selection for heterogeneous networks*: IEEE, 2012.
[10] T. Matsuda, T. Noguchi, and T. Takine, “Survey of network coding and its applications,” *IEICE transactions on communications*, vol. 94, no. 3, pp. 698–717, 2011.
[11] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, “RAID: High-performance, reliable secondary storage,” *ACM Computing Surveys (CSUR)*, vol. 26, no. 2, pp. 145–185, 1994.
[12] A. Lara, A. Kolasani, and B. Ramamurthy, “Network innovation using openflow: A survey,” *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 493–512, 2013.
[13] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, “RSVP: A new resource reservation protocol,” *IEEE network*, vol. 7, no. 5, pp. 8–18, 1993.
[14] https://iperf.fr/, *iPerf - The ultimate speed test tool for TCP, UDP and SCTP.* Accessed on: Aug. 16 2019.