

UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXVI Ciclo

**Integration of Heterogeneous Networks:
Protocols, Technologies, and Applications**

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Gianluigi Ferrari

Dottorando: *Giovanni Spigoni*

Gennaio 2014

Alla nonna Renata

Contents

Introduction	1
1 Heterogeneous Networks: Integration and Interaction	3
1.1 Introduction	3
1.2 Motivations	5
1.2.1 ABC Concept	5
1.2.2 Cellular Networks' Offload	6
1.3 Protocols and Technologies	8
1.3.1 Multipath TCP	8
1.3.2 IEEE 802.21	10
1.4 An Application Scenario: VANET	13
1.4.1 IEEE 802.11p	13
1.4.2 The X-NETAD Project	14
2 Vertical Handover: Preliminaries	19
2.1 Introduction	19
2.2 VHO Classification	20
2.3 Preliminaries on VHO Algorithms	21
2.3.1 Bandwidth Estimation Techniques	21
2.3.2 A Starting Hybrid RSSI/Goodput VHO Algorithm	22
2.4 A Low-complexity RSSI-based VHO Algorithm	26
2.5 A Simplified Hybrid RSSI/Goodput VHO Algorithm	30

3	Vertical Handover: Performance Analysis in Cellular Systems	33
3.1	Introduction	33
3.2	Windows Early Implementation	34
3.2.1	Experimental Setup	34
3.2.2	Experimental Results	37
3.3	Simulation-based Investigation	46
3.3.1	The OPNET Simulator	47
3.3.2	Performance Results	50
3.4	Android Implementation	59
3.4.1	Application Development and Trial	59
3.4.2	Experimental Results	64
3.5	Discussion on the VHO in HetNets	70
3.5.1	Role of the (Internal) Decision Algorithm	70
3.5.2	Role of the Authentication Procedures	70
3.5.3	Impact of VHO on HetNet Coverage	70
3.5.4	Impact of VHO on HetNet Capacity	71
3.6	Conclusions	71
4	A Vehicular HetNet Platform Implementation	73
4.1	Introduction	73
4.2	WiCar System Design	74
4.2.1	High-level Architecture	75
4.2.2	WiCar Protocol	77
4.3	WiCar Implementation	83
4.3.1	The Hardware Platform	83
4.3.2	The Software Implementation	86
4.4	Conclusions	89
	Conclusions and future directions	91
	Bibliography	93

Contents	iii
Ringraziamenti	99

List of Figures

1.1	Comparison of Standard TCP and MPTCP Protocol Stacks.	9
1.2	Example MPTCP Usage Scenario.	11
1.3	Illustrative X-NETAD traffic alerts dissemination scheme.	16
1.4	A sample map retrieved from a X-NETAD application server: the green color is associated with light traffic, the yellow indicates an intermediate traffic intensity, and red color is dedicated to highly congested roads.	17
2.1	Flowchart of the original VHO algorithm proposed in [1, Fig. 1] (courtesy of T. Inzerilli and A. M. Vegni).	23
2.2	Two sample realization of the VHO algorithm obtained with Matlab simulation results.	25
2.3	Dataflow of the low-complexity RSSI-based VHO algorithm.	27
2.4	Dataflow of the low-complexity hybrid RSSI/goodput-based VHO algorithm.	32
3.1	Sketch of the proprietary AAA mechanism used in the Guglielmo IEEE 802.11 network.	34
3.2	Sketch of the common AAA mechanism used in the UMTS networks.	35
3.3	Instantaneous handover time of the VHO algorithm presented in Section 2.4.	39

3.4	PMFs of the handover time of the VHO algorithm presented in Section 2.4: from UMTS to WiFi (upper) and from WiFi to UMTS (lower).	40
3.5	Goodput improvement using the low-complexity RSSI-based algorithm presented in Section 2.4.	41
3.6	RSSI and goodput experienced by the MT following a sample path using the low-complexity RSSI-based algorithm.	43
3.7	Goodput improvement using the low-complexity RSSI-based algorithm presented in Section 2.5.	44
3.8	RSSI and goodput experienced by the MT following a sample path using the hybrid RSSI/goodput VHO algorithm.	45
3.9	Protocol stack of a WiFi node and a UMTS node in the OPNET simulator.	48
3.10	Protocol stack of the new node with WiFi and Lte interfaces developed with OPNET modeler. The module <i>bw_test</i> is responsible for the computation of the available bandwidth estimation, while the <i>dg_change</i> module manipulates the IP routing table in order to change the default gateway and to redirect specific traffic over a given network.	49
3.11	First simulation scenario with a single MT following a deterministic path.	52
3.12	(a) Received power on the WiFi interface and (b) goodput on both the interfaces for the single node scenario.	53
3.13	Second simulation scenario with multiple MTs moving according to the Random Waypoint mobility model.	54
3.14	(a) Goodput and (b) number of the transmitting MTs in each network for the multiple node scenario.	56
3.15	Third simulation scenario with multiple MTs moving according to the Random Waypoint mobility model. Every node has both WiFi and Lte interfaces.	57

3.16	Simulation results for the third scenario. Here are shown the bandwidth estimation and the related selected network, of a single node during a 15 minutes long simulation.	58
3.17	Simulation results for the third scenario. Total aggregated goodput of the networks. The data load increases till to saturate both WiFi and Lte networks. Results for VHO hybrid and RSSI-based algorithms are shown.	60
3.18	BabelTen User Interface: (a) main view, with application menu; (b) map view; (c) Guglielmo hotspot coverage; (d) single hotspot details.	61
3.19	(a) Estimated delay, as a function of the Day Index; (b) Estimated goodput, as a function of the Day Index; (c) Amount of user transmitted and received data, as a function of the Day Index.	65
3.20	Cumulative transmitted and received data, as a function of the Day Index.	67
3.21	Mobile Network Analysis: (a) HSDPA - Delay PMF; (b) HSDPA - Goodput PMF; (c) UMTS - Delay PMF; (d) UMTS - Goodput PMF; (e) All mobile networks - Delay PMF; (f) All mobile networks - Goodput PMF.	69
4.1	A vehicular scenario with three different wireless networks.	75
4.2	High-level representation of the WiCar system.	78
4.3	Start-up procedure, services publishing stage.	80
4.4	Messages flow of request-response service and notify service.	81
4.5	Message flow of stream service and direct stream service.	82
4.6	The prototype of the main module of the WiCar system.	87

List of Tables

3.1	Parameters of the VHO algorithm.	36
3.2	Handover time for the VHO algorithm presented in Section 2.4 (mean, standard deviation, minimum, and maximum).	38
3.3	Average values of goodput, delay, downlink and uplink traffic, in WiFi and 3G networks, respectively.	68
4.1	Base message structure of the WiCar protocol	83
4.2	Types of message of the WiCar protocol and their meanings	84
4.3	WiCar protocol: Registration message structure	84
4.4	WiCar protocol: Service Descriptor message structure	85

Introduction

Today, the possibility of being connected to the Internet at every time and without interruption is almost a reality. The great capabilities of new generation cellular networks and their wide coverage enable people to use the innumerable resources of the Internet, almost everywhere and in any mobility scenario. All modern mobile devices have multiple interfaces to get connected to the Internet, and (almost) all smartphone users think to know which interface is the best one to use in a specific situation. In particular, despite the great improvement of cellular networks, in certain situations, the use of an alternative network—for instance, WiFi, is to be preferred. Therefore, the selection of the best network is not straightforward. If we change perspective and we do not talk about people and their smartphones, rather about mobile machines—say vehicles—that have to stay connected in order to provide or to receive a certain service, then the matter of finding, at every time, the best network to connect to, appears a little more urgent. Furthermore, since in some situations it could be very important to have a performing connection, for example with very low delay, then it is evident that the selection of the best network is not trivial. The characteristics of the networks to use, in order to choose the best network, are different according to the application at hand. A world where machines move automatically and use the Internet just like humans seems at the moment far away, but it is rapidly approaching. Besides the problem of network selection, one could wonder why one should just use the best network, instead of using all networks available in order to get the best “sides” of all? The development of efficient methods for the integration of multiple networks is an interesting but still open research area.

This thesis focuses on the interaction and integration of heterogeneous networks. Several innovative protocols, technologies, and applications developed, in order to make network integration easier for humans and automatic for machines, will be presented.

We will start explaining, in Chapter 1, the motivations supporting our interest on the integration of heterogeneous networks, focusing on possible technologies enabling this integration. In Chapter 2, we will start talking about *Vertical Handover* (VHO). This procedure is a key element for possible integration of wireless networks: it enables the transition of a connection between two different networks. We propose a simple algorithm for the implementation of VHO. In Chapter 3, we deeply analyse the performance of the proposed VHO algorithm, in terms of simulation and experimental results. The first test was performed on a desktop computer running Windows 7 operating system, and the second one on Android mobile device. In Chapter 4, the design and development of a platform for the integration of networks in vehicular environment is then presented.

Chapter 1

Heterogeneous Networks: Integration and Interaction

1.1 Introduction

Every different communication technology was designed to satisfy specific needs in a specific environment, every technology has its advantages and its drawback that can be useful or harmful for a specific purpose. Modern computers (fixed or mobile) integrate in a single device many heterogeneous communication interfaces. Almost every user is skilled enough to know which interface to use in a specific situation, in order to get connected—to the Internet or to a another local device—as fast as possible, as simple as possible and with the lowest possible costs. Nevertheless, there are some situation where it could be very useful if computers could decide automatically which interface to use in every situation without interfere other ongoing communications. And by the way, it is always a good thing if computers do autonomously what people have to do manually. Furthermore it would be very interesting if a computer could offer all the advantages of every communication technology at the same time.

This chapter tries to give a summary of the motivations behind the integration of heterogeneous networks. The *Always Best Connected* (ABC) concept and the *offload* procedure are described: the former is a general concept based on the integration

of networks, the latter could leverage the interaction between different networks—mostly cellular networks and Wireless Local Area Networks (WLAN).

Integration and interaction of different technologies need themselves new technologies and protocols in order to be performed. In this chapter are briefly described two protocols:

- **Multipath TCP:** not yet a standard, the experimental RFCs are published by the Internet Engineering Task Force (IETF). This protocol aim is to integrate different networks (i.e. different connections) into a single one in order to obtain a higher throughput and a more reliable connection.
- **IEEE 802.21:** this standard defines a Media Independent Handover (MIH)-layer that provides functions and messages that can be used by higher level application to execute seamless handover between different connections. Heterogeneous technologies are aware of this layer and provide it with information about their capabilities.

While Multipath TCP is a promising protocol enabling the integration of heterogeneous networks, IEEE 802.21 is a technology that provides tools for the interaction between networks. Since Multipath TCP doesn't need to modify existing networks and protocols, while IEEE 802.21 needs additional feature to be implemented for every communication technology, we believe that Multipath TCP will have greater diffusion than IEEE 802.21, even though a comparison between these protocols is not possible as they have different purposes and applications.

Eventually, an evolving application scenario for integration and interaction of heterogeneous networks is presented. A Vehicular Ad-hoc Network (VANET) is a very challenging scenario where the connectivity between vehicles (i.e. vehicles equipped with on-board computers) has to deal with high difference of speed between nodes, high interferences and a continuously changing network topology. Although on-board connectivity is not a concern today, since cellular networks can provide good connectivity also to high-speed moving vehicles, it is essential, for some applications, to keep the delay of the network very low. For example in vehicular safety communications applications it is important that urgent messages are sent through ad-hoc con-

nection between vehicles instead of using the cellular network. In this case the IEEE 802.11p standard could be a valid solution providing a fast and low-delay connection between vehicles. The X-NETAD project, described in Section 1.4.2, leverages the high availability of the cellular network and at the same time the low-delay capabilities of an ad-hoc connectivity between vehicles with the purpose to deliver traffic information messages.

1.2 Motivations

1.2.1 ABC Concept

The concept of *always connected*—the possibility of being connected everywhere at any time—was born with the advent of cellular networks, which can be considered as the first example of really pervasive wireless communications, at least in terms of voice connections. Since its origin, the meaning of always connected has been continuously changing. Nowadays, it coincides with the capability of surfing the web, taking advantage of its plethora of services through a high-speed Internet access. This condition, at least in technologically advanced countries, has been achieved. In fact, the frontier of today is represented by a new objective, namely, that of being *Always Best Connected* (ABC), as introduced several years ago [2]. According to this concept, the user should be able of taking advantage of the best available access network at any point in time, choosing among the large array of solutions offered by the market, including the various generations of cellular networks (e.g., GSM/GPRS, UMTS, and LTE) [3], metropolitan area networks (e.g., IEEE 802.16 [4] and HiperLAN [5]), wireless local area networks (e.g., IEEE 802.11a/b/g/n [6]), and also personal area networks (e.g., Bluetooth [7]).

An always best connected scenario, where a person is allowed to choose the best available access networks and devices at any point in time, generates great complexity and a number of requirements, not only for the technical solutions, but also in terms of business relationships between operators and service providers, and in subscription handling.

The definition of best depends on a number of different aspects such as personal

preferences, size and capabilities of the device, application requirements, security, operator or corporate policies, available network resources, and network coverage. Depending on the applications and user preferences, a user can be connected over one access at a time or over multiple accesses in parallel. The ABC concept includes virtually all types of access technologies; fixed and wireless, and existing technologies as well as those that are yet to come.

The idea of connecting to the best access point of the network is not novel, since it is used by the so-called Horizontal HandOver (HHO) mechanism [8], which is crucial, in cellular networks, to offer continuous connectivity to the customers. Unfortunately, the HHO procedure works only with access points belonging to the same access network. Therefore, in order to achieve a real ABC connectivity, it is necessary to extend the functionalities of the HHO, in order to make the transition of a mobile terminal (MT) between access points belonging to heterogeneous networks possible. A mechanism able to perform this task, possibly in a transparent way from upper-layer applications and “painlessly” from the user perspective, is commonly known as Vertical HandOver (VHO).

1.2.2 Cellular Networks’ Offload

The continuously growing traffic generated by Mobile Terminals (MTs)-such as smartphones, tablets, phablet, netbooks, and other mobile Internet devices-is nowadays one of the biggest challenges for mobile network operators, especially because this process is not supposed to vanish, at least from a short-time perspective. Therefore, in order to prevent network saturation phenomena, the operators are forced to increase their network capacity more quickly than the customers’ demand increase. This goal would be probably achieved by a combination of methods: (i) increasing the available bandwidth in cooperation with public communications agency, e.g., by exploiting the spectrum holes; (ii) increasing the cell spectral efficiency through technology upgrades, e.g., by switching from 3G to 4G [3] or WiMAX [4] technologies; (iii) reducing the number of users per macro-cell by either reducing the cell size or offloading data traffic through WiFi access points or femtocells [9, 10]. An example of the latter solution can be found in 3G and 4G networks at customers’

homes, where the deployment of the so-called, respectively, Home Node-B (HNB) and Home-Evolved Node-B (HeNB) allows to overlap reduced size cells (femto-cells) on top of the macro-cell of the corresponding Node-B/eNode-B base station [11]. The femtocell configuration is an example of Heterogeneous Network (Het-Net), since the involved devices (HNB/HeNB and Node-B/eNode-B) have different capabilities (e.g., the coverage range), even if they share the same technology [11]. This configuration is viable if both HNB/HeNB and Node-B/eNode-B belong to the same network operator and this could be a serious limitation for its commercial deployment, since the customer has to agree to keep a HNB/HeNB in his/her apartment, feeding it with its self-funded Internet connection. Considering the diffusion of WiFi access points (e.g., IEEE 802.11a/b/g/n [6]), from the customer viewpoint a more attractive solution would consist in jointly using his/her 3G cellular and WiFi connections. In this case, the devices are different not only in terms of capabilities, as in the femtocell case, but also in terms of technology. Moreover, the UMTS and WiFi networks may belong to two different non-related operators. It is interesting to observe that unlike in a femtocell configuration, the WiFi connection can be used both to replace the 3G connection (the so-called WiFi offloading [12]) and to increase the bandwidth of the 3G connection [13]. From the point of view of a mobile network operator, the hybrid UMTS-WiFi solution is more appealing, since it helps reducing the traffic load on the 3G network. On the other hand, the user has a real advantage only if the WiFi connection can offer an economical saving or a throughput benefit, with respect to the 3G connection. For this reason, the choice cannot be taken by the operator alone, but the customer has to be, at some extent, involved in the decision process. This is particularly true when the UMTS and WiFi networks belong to two different operators without a specific commercial agreement. In a classic cellular network, the switch between two different network base stations is governed by a relatively simple Horizontal HandOver (HHO) mechanism [8] and, therefore, it is seamless from the user perspective. In a HetNet, with a single involved technology (e.g., UMTS network with femtocell) HHO is still possible but it is more complex, especially in the transition from a macro-cell to a femtocell. In the case of a hybrid HetNet with two involved technologies, as considered in the current chapter, the

complexity is even higher, since it is necessary to use the so-called Vertical HandOver (VHO) mechanisms.

1.3 Protocols and Technologies

1.3.1 Multipath TCP

As the Internet evolves, demands on Internet resources are ever-increasing, but often these resources (in particular, bandwidth) cannot be fully utilized due to protocol constraints both on the end-systems and within the network. If these resources could be used concurrently, end user experience could be greatly improved. Such enhancements would also reduce the necessary expenditure on network infrastructure that would otherwise be needed to create an equivalent improvement in user experience. By the application of resource pooling [14], these available resources can be *pooled* such that they appear as a single logical resource to the user.

Multipath transport aims to realize some of the goals of resource pooling by simultaneously making use of multiple disjoint (or partially disjoint) paths across a network. The two key benefits of multipath transport are the following:

- To increase the resilience of the connectivity by providing multiple paths, protecting end hosts from the failure of one.
- To increase the efficiency of the resource usage, and thus increase the network capacity available to end hosts.

Multipath TCP is a modified version of TCP [15] that implements a multipath transport and achieves these goals by pooling multiple paths within a transport connection, transparently to the application. Multipath TCP is primarily concerned with utilizing multiple paths end-to-end, where one or both of the end hosts are multi-homed. It may also have applications where multiple paths exist within the network and can be manipulated by an end host, such as using different port numbers with Equal Cost MultiPath (ECMP) [16].

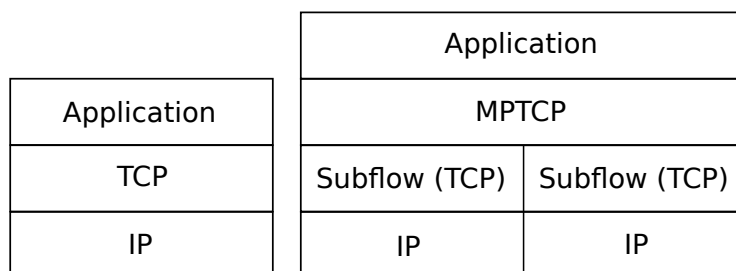


Figure 1.1: Comparison of Standard TCP and MPTCP Protocol Stacks.

MPTCP, defined in [17], is a specific protocol that instantiates the Multipath TCP concept.

Although multihoming and multipath functions are not new to transport protocols (Stream Control Transmission Protocol (SCTP) [18] being a notable example), MPTCP aims to gain wide-scale deployment by recognizing the importance of application and network compatibility goals. These goals relate to the appearance of MPTCP to the network (so non-MPTCP-aware entities see it as TCP) and to the application (through providing a service equivalent to TCP for non-MPTCP-aware applications).

It is not the goal of this thesis to explain in deep the architecture of the MPTCP protocol, anyway, to get some insight about this protocol, in Figure 1.1 is illustrated the layered architecture of MPTCP.

Following is provided a high-level summary of normal operation of MPTCP, and is illustrated by the scenario shown in Figure 1.2.

- To a non-MPTCP-aware application, MPTCP will behave the same as normal TCP. Extended APIs could provide additional control to MPTCP-aware applications [19]. An application begins by opening a TCP socket in the normal way. MPTCP signaling and operation are handled by the MPTCP implementation.
- An MPTCP connection begins similarly to a regular TCP connection. This is illustrated in Figure 1.2 where an MPTCP connection is established between addresses A1 and B1 on Hosts A and B, respectively.

- If extra paths are available, additional TCP sessions (termed MPTCP “subflows”) are created on these paths, and are combined with the existing session, which continues to appear as a single connection to the applications at both ends. The creation of the additional TCP session is illustrated between Address A2 on Host A and Address B1 on Host B.
- MPTCP identifies multiple paths by the presence of multiple addresses at hosts. Combinations of these multiple addresses equate to the additional paths. In the example, other potential paths that could be set up are A1 ↔ B2 and A2 ↔ B2. Although this additional session is shown as being initiated from A2, it could equally have been initiated from B1.
- The discovery and setup of additional subflows will be achieved through a path management method; this document describes a mechanism by which a host can initiate new subflows by using its own additional addresses, or by signaling its available addresses to the other host.
- MPTCP adds connection-level sequence numbers to allow the reassembly of segments arriving on multiple subflows with differing network delays.
- Subflows are terminated as regular TCP connections, with a four-way FIN handshake. The MPTCP connection is terminated by a connection-level FIN.

1.3.2 IEEE 802.21

As explained in [20], the main purpose of IEEE 802.21 is to enable handovers between heterogeneous technologies (including IEEE 802 and cellular technologies) without service interruption hence improving the user experience of mobile terminals. Many functionalities required to provide session continuity depend on complex interactions that are specific to each particular technology. IEEE 802.21 provides a framework that allows higher levels to interact with lower layers to provide session continuity without dealing with the specifics of each technology. That is, this protocol can be seen as the “glue” between the IP-centric world developed in the Internet

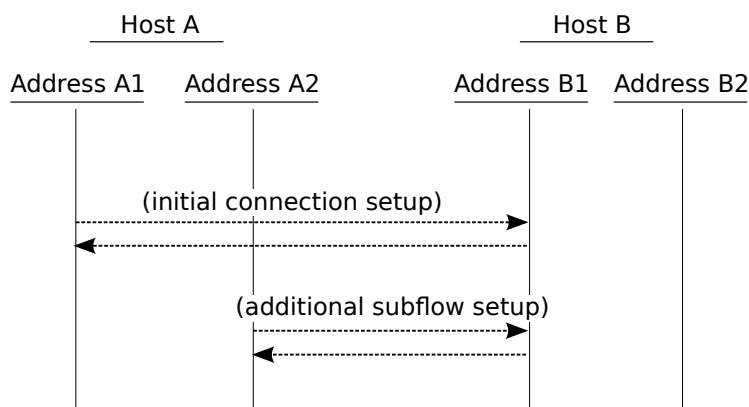


Figure 1.2: Example MPTCP Usage Scenario.

Engineering Task Force (IETF), and the reference scenarios for future mobile networks currently being designed in the Third Generation Partnership Project (3GPP) and 3GPP2 or other technology specific solutions. While the IETF does not address specific layer 2 technologies, the interest of 3GPP/3GPP2 in non-cellular layer 2 technologies, such as WLAN, is restricted to its integration into cellular environments. IEEE 802.21 provides the missing, technology-independent, abstraction layer able to offer a common interface to upper layers, thus hiding technology-specific primitives. This abstraction can be exploited by the IP stack (or any other upper layer) to better interact with the underlying technologies, ultimately leading to improved handover performance. We deepen the aims and objectives of 802.21. To achieve these goals, IEEE 802.21 defines a media-independent entity that provides a generic interface between the different link layer technologies and the upper layers. To handle the particularities of each technology, 802.21 maps this generic interface to a set of media-dependent service access points (SAPs) whose aim is to collect information and control link behavior during handovers. In addition, a set of remote interfaces, terminal-network and network-network, are defined to convey the information stored at the operator's network to the appropriate locations (e.g., to assist the terminal in handover decisions). All the functionality of 802.21 is provided to the users by a set of services: Event, Command, and Information. These services are the core of the

specification, and define the semantic model of the communication with the lower layers and the network.

IEEE 802.21 Objectives

Following the lines presented above, the contribution of the 802.21 standard is centered around the following three main elements:

- A framework that enables seamless handover between heterogeneous technologies. This framework is based on a protocol stack implemented in all the devices involved in the handover. The defined protocol stack aims to provide the necessary interactions among devices for optimizing handover decisions.
- The definition of a new link layer SAP that offers a common interface for link layer functions and is independent of the technology specifics. For each of the technologies considered in 802.21, this SAP is mapped to the corresponding technology-specific primitives. The standard draft includes some of these mappings.
- The definition of a set of handover enabling functions that provide the upper layers (e.g., mobility management protocols such as Mobile IP [21]), with the required functionality to perform enhanced handovers. These functions trigger, via the 802.21 framework, the corresponding local or remote link layer primitives.

Although the main purpose of IEEE 802.21 is to enable handover between heterogeneous technologies, a set of secondary goals have also been defined. These secondary goals are:

- Service continuity, defined as the continuation of the service during and after the handover procedure. One of the main goals of 802.21 is to avoid the need to restart a session after a handover.
- Handover-aware applications. The 802.21 framework provides applications with functions for participating in handover decisions. For instance, a voice appli-

ation may decide to execute a handover during a silence period in order to minimize service disruption.

- Quality of service (QoS)-aware handovers. The 802.21 framework provides the necessary functions in order to make handover decisions based on QoS criteria. For instance, we may decide to hand over to a new network that guarantees the desired QoS.
- Network discovery. This is an 802.21 feature that allows users to be provided with information on candidate neighbors for a handover.
- Network selection assistance. Network selection is the process of making a handover decision based on several factors (e.g., QoS, throughput, policies, and billing). In line with the above, the 802.21 framework only provides the necessary functions to assist network selection, but does not make handover decisions, which are left to the higher layers.
- Power management can also benefit from the information provided by 802.21. For instance, power consumption can be minimized if the user is informed of network coverage maps, optimal link parameters, or sleep or idle modes.

1.4 An Application Scenario: VANET

1.4.1 IEEE 802.11p

Vehicular environments impose a set of new requirements on today's wireless communication systems. Vehicular safety communications applications cannot tolerate long connection establishment delays before being enabled to communicate with other vehicles encountered on the road. Similarly, non-safety applications also demand efficient connection setup with roadside stations providing services (e.g. digital map update) because of the limited time it takes for a car to drive through the coverage area. Additionally, the rapidly moving vehicles and complex roadway environment present challenges at the PHY level.

The IEEE 802.11p WAVE [22] standardization process originates from the allocation of the Dedicated Short Range Communications (DSRC) spectrum band in the United States and the effort to define the technology for usage in the DSRC band.

The primary goal of IEEE 802.11p is to enable public safety applications that can save lives and improve traffic flow. Private services are also permitted in order to spread the deployment costs and to encourage the quick development and adoption of DSRC technologies and applications.

The IEEE 802.11p standard is limited by the scope of IEEE 802.11, which is strictly a MAC and PHY level standard that is meant to work within a single logical channel.

In an overly simplified manner, the IEEE 802.11 MAC is about how to arrange for a set of radios in order to establish and maintain a cooperating group. Radios can freely communicate among themselves within the group but all transmissions from outside are filtered out. Such a group is a Basic Service Set (BSS) and there are many protocol mechanisms designed to provide secure and robust communications within a BSS. The key purpose of the IEEE 802.11p amendment at the MAC level is to enable very efficient communication group setup without much of the overhead typically needed in the current IEEE 802.11 MAC. In other words, the focus is on simplifying the BSS operations in a truly ad hoc manner for vehicular usage.

The IEEE 802.11p standard is meant to:

- Describe the functions and services required by WAVE-conformant stations to operate in a rapidly varying environment and exchange messages without having to join a Basic Service Set (BSS), as in the traditional IEEE 802.11 use case.
- Define the WAVE signaling technique and interface functions that are controlled by the IEEE 802.11 MAC.

1.4.2 The X-NETAD Project

In this section, we will provide an overview of an innovative approach for effective cross-network information dissemination, with applications to VANETs. In particu-

lar, we describe the main approach followed in the on-going bilateral Italy-Israel X-NETAD project (Eureka Label E! 6252 [23]), sponsored by the Ministry of Foreign Affairs (Italy) and the Israeli Industry Center for R&D (Israel), under the “Israel-Italy Joint Innovation Program for Industrial, Scientific and Technological Cooperation in R&D”. The X-NETAD project is carried out by Guglielmo Srl (Italy), Cellint Traffic Solutions Ltd (Israel) and the *Wireless Ad-hoc and Sensor Networks Lab (WASNLab)*, Department of Information Engineering of the University of Parma (Italy).

The X-NETAD project leverages on the spontaneous formation of WiFi-based local VANETs, with direct connections between neighboring vehicles, in order to disseminate, very quickly and inexpensively, traffic alerts received from a pre-existent cellular network. Cellint Traffic Solutions Ltd has developed an efficient and accurate real time traffic detection systems (commercially denoted as TrafficSense), mainly based on data extraction from cellular networks [24]. TrafficSense relies on the existence of a cellular network over the road areas to be monitored and is based on the fact that, with an extremely high probability, there is (at least) a cellular phone inside each vehicle. Cellular phones are also a good way to disseminate traffic information and supply drivers with real time traffic alerts in their neighborhood. The neighborhood of a vehicle can be defined as a circle of 20-50 km around the current vehicle location. However, traffic alerts dissemination using cellular phones have some inherent limitations. In particular, the continuous use of cellular phones for real time traffic alerts on the cellular network has a relatively high cost and it drains a large amount of energy from the mobile phone’s battery.

Approximate real-time location estimation is needed in order to receive only relevant and localized information. This information can be provided by GPS receivers or by means of cellular cell-base triangulation. Currently, less than 20% of phones have a built-in GPS which can give an accurate position information. Cellular signal triangulation requires no additional hardware, because it exploits the cellular base station towers signal, which is triangulated to get an approximate location. This technique gives a more loose position estimation than GPS, and its accuracy depends on the number of cellular base station seen by phone.

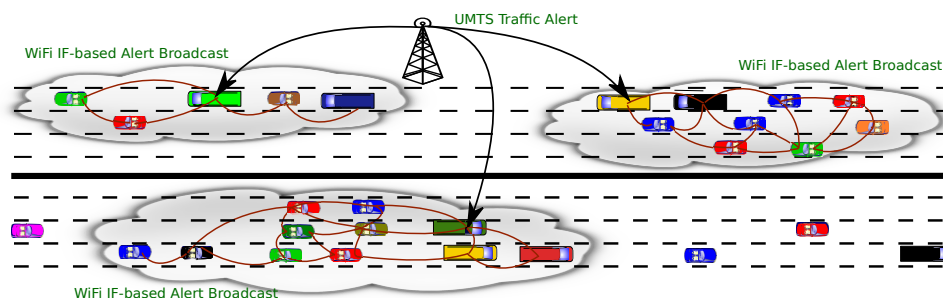


Figure 1.3: Illustrative X-NETAD traffic alerts dissemination scheme.

The goal of the X-NETAD project is the design and implementation of a low cost, efficient, and ubiquitous traffic alerts dissemination system to drivers. In particular, it supposes the formation of ephemeral local VANETs, created by one *primary* vehicle surrounded by other *secondary* vehicles. The former vehicle acts as a gateway for traffic alerts coming from the cellular network, which are disseminated to the latter vehicles by means of WiFi *ad-hoc* broadcast communications. An illustrative representation of the cross-network traffic alert dissemination system made by the X-NETAD project is shown in Figure 1.3.

This is done by using Android-based smartphones (or tablets) as the target development platforms. These devices are typically equipped with two interfaces, for cellular and WiFi networks, respectively. The key idea of the X-NETAD approach is that of leveraging on the spontaneous formation of WiFi local VANETs, with direct connections (*ad-hoc* WiFi or other) between neighboring vehicles, in order to disseminate traffic alerts in the VANETs very quickly and inexpensively. More precisely, the X-NETAD system is designed in such a way that, at any given time, only a small percentage (i.e., 10%) of the vehicles (i.e., the Android smartphones inside the vehicles) will be directly connected to the traffic alerts dissemination system using cellular Internet. These vehicles, called *primary*, will then propagate the received traffic alerts through an *ad-hoc* WiFi network to the others vehicles which cannot receive directly UMTS traffic alerts. These latter vehicles will be called *secondary*.

The traffic alerts' dissemination is carried out through a very efficient broad-

ically broadcast (through the WiFi network) their location to vehicles without GPS, enabling them to coarsely estimate their location, even without having a functional GPS.

A traffic alert message contains an image, an audio file and some XML-formatted meta-data. At the moment of writing this thesis, the service is active in Israel only (as it can be seen from the map presented in Figure 1.4), but it will be quickly extended to others geographic areas like Italy and other European countries. Accordingly to the user preferences, the map can be displayed on the screen immediately after its reception, or can be stored on the local memory. Using the information gathered from the GPS or from the received messages position field, it is possible to draw the actual vehicle's position and direction within the map, showing at the same time, additional information like speed and acceleration. The audio message is always played, and it contains a localized traffic announcement, with valuable information concerning traffic jams, accidents and on-going roadworks. We underline that, thanks to the Cellint traffic detection system, the information provided to the users is always up to date.

Chapter 2

Vertical Handover: Preliminaries

2.1 Introduction

A Vertical Handover (VHO) procedure is composed by three main phases: initiation, decision, and execution [26]. During the initiation phase, the mobile terminal (MT) or the network controller trigger the handover procedure, according to the specific networks' conditions. In the second phase, the VHO algorithm chooses the new access point according to a pre-determined set of metrics, such as the Received Signal Strength Indicator (RSSI), the network connection time, the available bandwidth, the power consumption, the monetary cost, the security level, and, obviously, the user preferences [27]. During the final execution phase, all signaling operations for communication re-establishment and data transfer are carried out. The most relevant international standardization effort regarding VHO and continuous communications, the IEEE 802.21 standard, only refers to the first two phases (initiation and decision) that are relatively technology-independent, but it deliberately ignores the execution phase [28]. The latter is considered by other standardization bodies, like the 3GPP consortium [3] or the Internet Engineering Task Force (IETF) [29]. However, there is no universal and definitive solution, since the mobility management problem is affected by too many factors (technological, commercial, and even social). Most of the VHO approaches, for example that considered in [30], leverages on Mobile IP [31], a

level-3 solution that is based on the idea of maintaining the same IP address in every network visited by the MT. In the last years, several works have been based on the application-level Session Initiation Protocol (SIP) [32], mostly because it can better support Voice over IP (VoIP) applications [33]. A promising approach for the third phase of VHO (execution) is represented by the Multipath TCP protocol described in Section 1.3.1. This approach has the advantage, compared with the SIP protocol, to be general purpose. That is, it is suitable for every session based on TCP connection, since it simply extends the TCP protocol.

2.2 VHO Classification

There are several possible classifications of the VHO algorithms. In particular, they can be distinguished between *no-coupling* and *coupling*. The first group of VHO algorithms refers to scenarios without any form of cooperation between the involved players (users and network operators) [26]. This situation offers the highest degree of freedom to the user, at the price of an increased complexity of the whole handover procedure and of a degraded performance. Clearly, with a higher level of coupling (namely, loose or tight [34]) a better performance can be achieved.

In a no-coupling scenario handover times are typically long. Therefore, in order to avoid any lack of connectivity during the handover *execution* phase, it is necessary to adopt a *make-before-break* approach. In other words, the old connection is torn down only after the new connection has been established, thus yielding to a period of coexistence of the two connections, during which the MT becomes a temporary multi-homed host. The management of a multi-homed host during the execution phase is an open problem, without a universal solution, and, currently, every Operating System (OS) has its own solution for this problem [35].

2.3 Preliminaries on VHO Algorithms

2.3.1 Bandwidth Estimation Techniques

A bandwidth estimation technique aims at estimating, as accurately as possible, the bandwidth offered by a certain network.¹ In our case, the network could be split in two portions: the *local*, between the MT and the access point; and the *remote*, that coincides with the backbone feeding the access point. The end-to-end bandwidth is obviously determined by the minimum of the bandwidths offered by the individual networks. In the case of the UMTS network, the local system portion (i.e., the cell) is certainly the bottleneck, as it is reasonable to expect that the backbone bandwidth will be larger than that offered to the customers. On the other hand, in WiFi networks the quality of the backbone connection is unpredictable by the MT, which, therefore, cannot predict if the bottleneck is the local or the remote portion.

There are two main categories of bandwidth estimation techniques: direct and indirect. The techniques of the first group actively estimate the available bandwidth by sending a train of probe packets across the network, towards a known (either remote or local) destination. Conversely, the indirect methods try to passively estimate the available bandwidth without introducing network overhead and, therefore, they are more appealing.

In a UMTS network, it is possible, in principle, to indirectly estimate the available bandwidth, by knowing the modulation format and the channel coding technique currently adopted by the Adaptive Coding Modulation (ACM) mechanism. Disappointingly, this information rarely coincides with the truly available bandwidth, since the network operator dynamically assigns (on-demand) the resources to the customers. However, due to the MCHO assumption, the MT has no means to know what the available bandwidth will be in the future.

On the other hand, several studies have shown that the local bandwidth could be indirectly estimated also in IEEE 802.11 networks, by observing the physical and medium access control (MAC) parameters, such as the modulation format, the RSSI, or the Network Allocation Vector (NAV) [6]. The indirect estimation techniques have

¹In this work, the terms goodput and bandwidth are used interchangeably.

shown to be sufficiently accurate [36] and they have also the advantage of being non-intrusive. Unfortunately, when the bottleneck is given by the remote network, local information is useless.

It can then be concluded that, in both IEEE 802.11 and UMTS networks, passive estimation of the available bandwidth, through indirect estimation techniques, is not feasible. Conversely, direct techniques can be certainly employed in these networks, since they are independent of the type of network and of the characteristics of the MT. Moreover, these techniques, such as Wbest [37], have shown to be able to obtain an accurate estimate of the end-to-end bandwidth. Despite its attractive features, direct bandwidth estimation techniques have still two critical drawbacks: (i) they are always characterized by a certain degree of intrusiveness; (ii) in order to offer valuable information, they require that the MT is already authenticated to the network of interest. The latter is the major drawback for VHO, since it implies that, for the purpose of proactive estimation of the end-to-end bandwidth (as required by the VHO algorithm described in Subsection 2.3.2), the MT has to keep alive both previous and new connections, leading to energetic and economical inefficiencies.

2.3.2 A Starting Hybrid RSSI/Goodput VHO Algorithm

The VHO algorithm proposed in [1] is based on two performance metrics: the received Power (denoted with P) and the GoodPut² (denoted with GP), which coincides with the effective bandwidth independently available in each link. This VHO algorithm, illustrated in Figure 2.1, requires a fresh and reliable estimation of the instantaneous goodput and of the instantaneous received power. These are periodically estimated, considering a constant sampling interval T_s (dimension: [s]). While an estimation of the received power can be easily obtained in most cases, the estimation of the bandwidth can be challenging, as preliminary discussed in Subsection 2.3.1.

We observe that the algorithm illustrated in Figure 2.1 uses the *instantaneous*

²In this paper, we consider the goodput as the transmit rate (dimension: [b/s]) at which *data* information is transmitted. Alternatively, the goodput could be defined as the long-term fraction of time during which transmitted *data* packets are successfully received. The throughput should instead be defined as the long-term fraction of time during which (*data* and *control*) packets are successfully received.

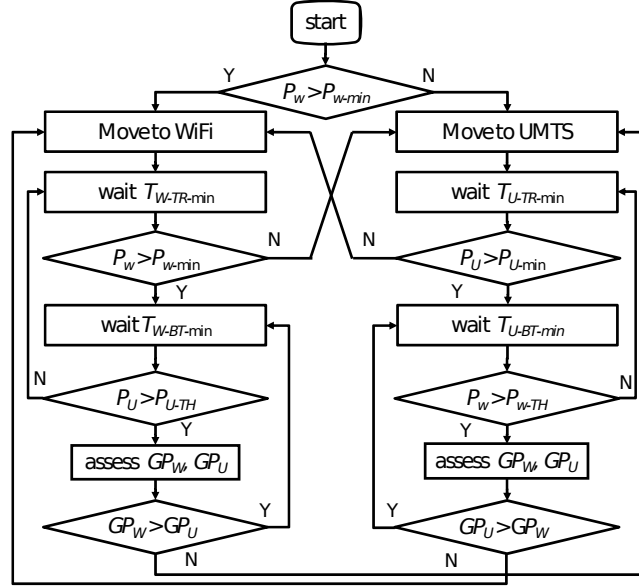


Figure 2.1: Flowchart of the original VHO algorithm proposed in [1, Fig. 1] (courtesy of T. Inzerilli and A. M. Vegni).

received power and an *average* goodput, obtained with proper filtering techniques. More specifically, in [1] the average goodput is computed using two different filtering techniques: (i) weighed Moving Average (MA) and (ii) Exponential Smoothing Average (ESA). In the case of MA filtering, the length of the moving window (in terms of samples), denoted as $K \in \mathbb{N}$, is the only design parameter. In particular, given the sequence of $K + 1$ instantaneous goodput samples $\{g_i\}_{i=n-K}^n$ till the (discrete-scale) estimation epoch $n \in \mathbb{N}$, the average goodput $GP(n)$ can be expressed as

$$GP(n) = \sum_{i=n-K}^n \frac{g_i}{K}.$$

Assuming that $g_i = 0, i < 0$, it follows that $GP(n) = 0, n < 0$. If ESA filtering is

considered, then the average goodput can be computed as follows:

$$\begin{cases} \text{GP}(n) = w_1 g_n + w_2 \text{GP}(n-1) + w_3 \text{GP}(n-2) \\ w_1, w_2, w_3 \in [0, 1] \\ \sum_{i=1}^3 w_i = 1 \end{cases} \quad (2.1)$$

where $\{w_1, w_2, w_3\}$ are real weighing coefficients that can be tuned to obtain the desired filter behavior. Under the assumption that $g_i = 0, i < 0$, also with the ESA filter it follows that $\text{GP}(n) = 0, n < 0$.

The received power P_x ($x \in \{U, W\}$) is used twice. First, it is compared with a threshold P_x^{MIN} to decide whether the corresponding network is available. Then, it is compared with the threshold P_x^{TH} to decide if this network can be a potential handover candidate. If both networks have a sufficient level of received power (i.e., $P_x > P_x^{\text{TH}}$ for both $x = U$ and $x = W$), the decision algorithm elects the WiFi network if its filtered goodput (namely, GP_W) is higher than that of the 3G network (namely, GP_U). On the other hand, the algorithm selects the UMTS network when $\text{GP}_U > \text{GP}_W$. The waiting times inserted between the various decision blocks are expedient to reduce the ping-pong effect between the two networks. One should observe that, with the exception of the first decision block ($P_x > P_x^{\text{MIN}}$), the algorithm outlined in Figure 2.1 is perfectly symmetric.

The performance of the VHO algorithm proposed in [1] is analyzed using a custom-made Matlab simulator [38]. In particular, we consider a single WiFi cell (with “optimistic” radius equal to 125 m) and a single UMTS cell (with radius equal 500 m—this limited radius is considered only for the purpose of simulation). The simulator considers only the variations of the received power and of the bandwidth, ignoring any physical and MAC layer details. In particular, the received power is a function of the distance between the MT and the access points (according to the Friis formula), while the available bandwidth is a function of both distance and network congestion. In order to simulate different levels of traffic loads at UMTS and WiFi cells, a random number of users (on average, 20), generating traffic according to three different classes of services (asynchronous data, voice, and video streaming), is considered. The users are moving following a pre-determined random direction with a

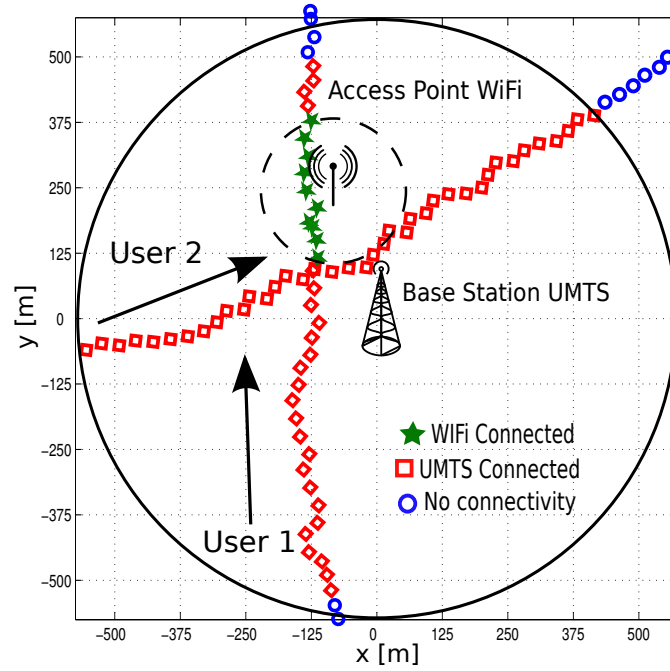


Figure 2.2: Two sample realization of the VHO algorithm obtained with Matlab simulation results.

limited speed (between 0.5 m/s and 1.5 m/s).

In Figure 2.2, simulation results, relative to two significant realizations (paths), are shown. In particular, they refer, respectively, to a first user crossing vertically the simulation area, passing close to the WiFi access point, and to a second user, who follows a path almost tangent to the WiFi cell. The first user, after entering the WiFi cell, can successfully complete the handover from the UMTS network to the WiFi network, since the received power P_W is much higher than the considered threshold. On the other hand, the WiFi signal power received by the second user, in passing by the WiFi cell, is close to the threshold: this could lead to a ping-pong phenomenon, but the considered VHO algorithm, owing to the considering filtering approach, prevents it.

2.4 A Low-complexity RSSI-based VHO Algorithm

The first simplified novel VHO algorithm is derived from the algorithm presented in Subsection 2.3.2 by applying the following modifications.

- The goodput is no longer considered to make a handover decision. In fact, as explained in Subsection 2.3.1, indirect estimation does not provide sufficient information, and direct estimation cannot be a feasible solution, because of its economical and efficiency drawbacks.
- The VHO algorithm is asymmetric, as it assigns an intrinsic preference to IEEE 802.11 networks. This is motivated by two reasons. First, WiFi connectivity is currently less expensive (at least in Italy) than 3G connectivity. Moreover, our experimental results have shown that the bandwidth offered by IEEE 802.11 networks is typically larger than that offered by 3G networks.
- The received power is replaced with the RSSI, as the latter can be measured more easily from the received packets.
- The waiting times between consecutive operations are removed, due to their inefficiency.

We now describe the operations of the novel VHO algorithm, with reference to the dataflow shown in Figure 2.3. As in the Subsection 2.3.2, the symbol x is used to represent both types of interface. The algorithm is entirely based on the RSSI measurements. Note that in both WiFi and UMTS cases the instantaneous RSSI values (denoted as $RSSI_x$) are considered. In the WiFi case, the filtered values of the RSSI (denoted as $RSSI_W^{ESA}$) are also considered.

The instantaneous RSSI value $RSSI_x$ (at each interface) is compared with two thresholds, denoted as TH_x^U and TH_x^L . The *lower* threshold TH_x^L is used to determine when the RSSI is not sufficient to guarantee a stable connectivity: therefore, it is slightly higher than the corresponding interface sensitivity. Clearly, when $RSSI_x < TH_x^L$ the connection on the interface x is torn down. On the other hand, the *upper* threshold TH_x^U is used to determine if the measured RSSI is sufficient to establish a

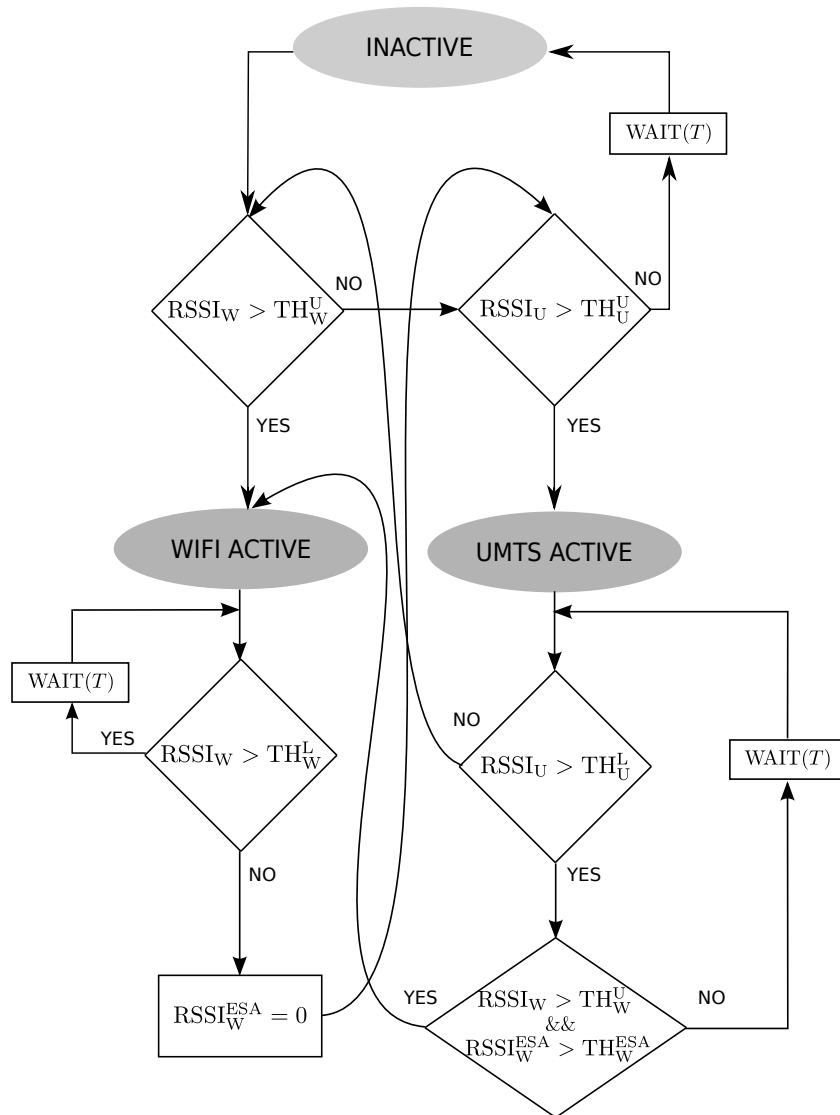


Figure 2.3: Dataflow of the low-complexity RSSI-based VHO algorithm.

stable connection. To this end, we assume that $TH_x^U > TH_x^L$. The use of two thresholds (per network interface) is the first countermeasure against the ping-pong effect and was historically introduced in the context of cellular networks for managing horizontal handovers [39].

The WiFi RSSI values obtained with ESA filtering can be expressed using (2.1) and simply replacing the goodput with the RSSI:

$$\begin{cases} \text{RSSI}_W^{\text{ESA}}(n) = v_1 \text{RSSI}_W(n) + v_2 \text{RSSI}_W^{\text{ESA}}(n-1) + v_3 \text{RSSI}_W^{\text{ESA}}(n-2) \\ v_1, v_2, v_3 \in [0, 1] \\ \sum_{i=1}^3 v_i = 1 \end{cases} \quad (2.2)$$

where $\{v_1, v_2, v_3\}$ are proper weighing coefficients and n is the time epoch. We have chosen the ESA filter because it offers performance similar to the MA filter, but with a more compact structure. The filtering operation is only performed on the WiFi interface, in order to avoid instantaneous peaks on the RSSI of the WiFi network. The $\text{RSSI}_W^{\text{ESA}}$ values are compared to another threshold, denoted as TH_W^{ESA} . Unlike the instantaneous RSSI, that is used to take forced and quick decisions, the filtered measures of the RSSI are used to make “effective” ABC decisions. Moreover, the use of average measurements is expedient to further mitigate the ping-pong phenomena.

According to Figure 2.3, the MT can be in three different self-explanatory states: INACTIVE, WiFi ACTIVE, and UMTS ACTIVE. When in the INACTIVE state, the MT measures, with period T_s (dimension: [s]), the RSSI level at each network interface. As soon as the first (of the two) RSSI level overcomes its upper threshold, the corresponding interface notifies the event to the VHO manager, triggering the execution of the Authentication, Authorization, and Accounting (AAA) procedure to join the selected network. We observe that if both networks are available, the priority is always given to the WiFi network. If the AAA procedure in the selected network x succeeds, the state of the MT switches from INACTIVE to “ x ACTIVE.”

Due to the asymmetric nature of the algorithm, the WiFi ACTIVE and the UMTS ACTIVE states have to be treated separately.

When in the UMTS ACTIVE state, the MT periodically (with period T_s) compares RSSI_U with the lower threshold TH_U^L . If $\text{RSSI}_U < TH_U^L$, the handover man-

ager immediately starts the authentication of the WiFi network, after verifying that $\text{RSSI}_W > \text{TH}_U^U$. If the latter condition is not satisfied, the VHO manager is forced to torn down the UMTS connection and the MT switches to the INACTIVE state. On the other hand, if RSSI_U remains higher than the threshold TH_U^L , the manager has the opportunity of “quietly” evaluating the condition of the WiFi network, in order to assess the possibility of performing a handover. In particular, the algorithm performs a double check, verifying that $\text{RSSI}_W > \text{TH}_W^U$ and that $\text{RSSI}_W^{\text{ESA}} > \text{TH}_W^{\text{ESA}}$. In case of success, the VHO manager starts to re-route the user traffic on the IEEE 802.11 interface and begins the log-off procedure on the UMTS network.³ Obviously, in the case of a failure of the double check, the MT is forced to maintain the current UMTS connection returning to the UMTS ACTIVE state.

When the MT is in the WiFi ACTIVE state, the behavior of the VHO algorithm is different from that observed when the MT is the UMTS ACTIVE state. In fact, in this case, until RSSI_W remains higher than the threshold TH_W^L , the MT is forced to remain in the WiFi ACTIVE state, ignoring the conditions on the UMTS interface. Only when the WiFi connectivity is lost ($\text{RSSI}_W < \text{TH}_W^L$), the VHO manager compares RSSI_U with the threshold TH_U^U , in order to initiate the AAA procedure in the UMTS network—as already explained, the intrinsic preference for the WiFi network is only motivated by real-life experience. Before the AAA operation is started, the filtered value of the RSSI is set to zero in order to prevent rapid re-connections to the WiFi network.⁴ Finally, due to the long duration of the WiFi AAA procedure, during the UMTS ACTIVE \rightarrow WiFi ACTIVE transition, there are some hidden transitional states, not shown in Figure 2.3 for the sake of simplicity. However, one should take into account the possibility of a failure of the AAA procedure: this will produce a back transition towards the UMTS ACTIVE state or towards the INACTIVE state.

³The re-routing of existent connections needed to have seamless connectivity after a handover is a problem not addressed in this work. Therefore, after a handover the pre-existent user connections will be likely interrupted.

⁴Note that the same result can be obtained by inserting a delay with fixed duration $T \gg T_s$. However, our solution allows to continuously check the RSSI_U which cannot be done with the fixed delay.

2.5 A Simplified Hybrid RSSI/Goodput VHO Algorithm

While the VHO algorithm presented in Section 2.4 is based on the implicit assumption that, when available, an IEEE 802.11 network always offers a better service than a 3G network, we now propose another VHO algorithm that builds on the previous one, but makes also use of goodput information. This extension is motivated to avoid switching from the UMTS network to the WiFi network when the latter offers a smaller effective bandwidth. Although this extension goes back to the approach proposed in [1], the complexity will be kept lower.

The dataflow of the new hybrid algorithm is shown in Figure 2.4. By comparing this dataflow with the dataflow of the RSSI-based algorithm (Figure 2.3) there is an additional state, the WiFi CONNECTED/UMTS ACTIVE state (highlighted at the bottom), where the MT is authorized in both networks. The presence of this state is expedient to estimate the bandwidths of both networks. The bandwidth is estimated by measuring the time necessary for downloading a 400 kByte size file from a remote host (for the ease of simplicity, the file is hosted by a Google server). The bandwidth estimation method is explained in more detail in Section 3.2.1.

Due to the asymmetric nature of the algorithm, the MT can move towards this new state only from the UMTS ACTIVE state. In particular, during this transition the MT performs the AAA procedure towards the IEEE 802.11 network. Then, the MT remains in the WiFi CONNECTED/UMTS ACTIVE state for all the time needed to estimate the goodput of both networks. As soon as the new measurements, denoted, respectively, as GP_W and GP_U , are available, the VHO algorithm decides to switch to the WiFi ACTIVE or to come back to the UMTS connected state. In the latter, the MT disconnects from the IEEE 802.11 network and resets its filtered RSSI, in order to reduce the waste of resources, as previously explained in Section 2.4. From a practical point of view, when $RSSI_W^{ESA} > TH_W^{ESA}$ the goodput is periodically estimated with a variable (but short) period, given by the sum of the time necessary to complete the AAA procedure and the time necessary to fill again $RSSI_W^{ESA}$.

Finally, due to the typically long times needed by the WiFi AAA procedure, during the UMTS ACTIVE \rightarrow WiFi CONNECTED/UMTS ACTIVE transition there

are some hidden transitional states, not shown in Figure 2.4 for the ease of simplicity. In particular, when the AAA procedure fails, the transition to the WiFi CONNECTED/UMTS ACTIVE state cannot happen and it is necessary to come back to the UMTS ACTIVE state.

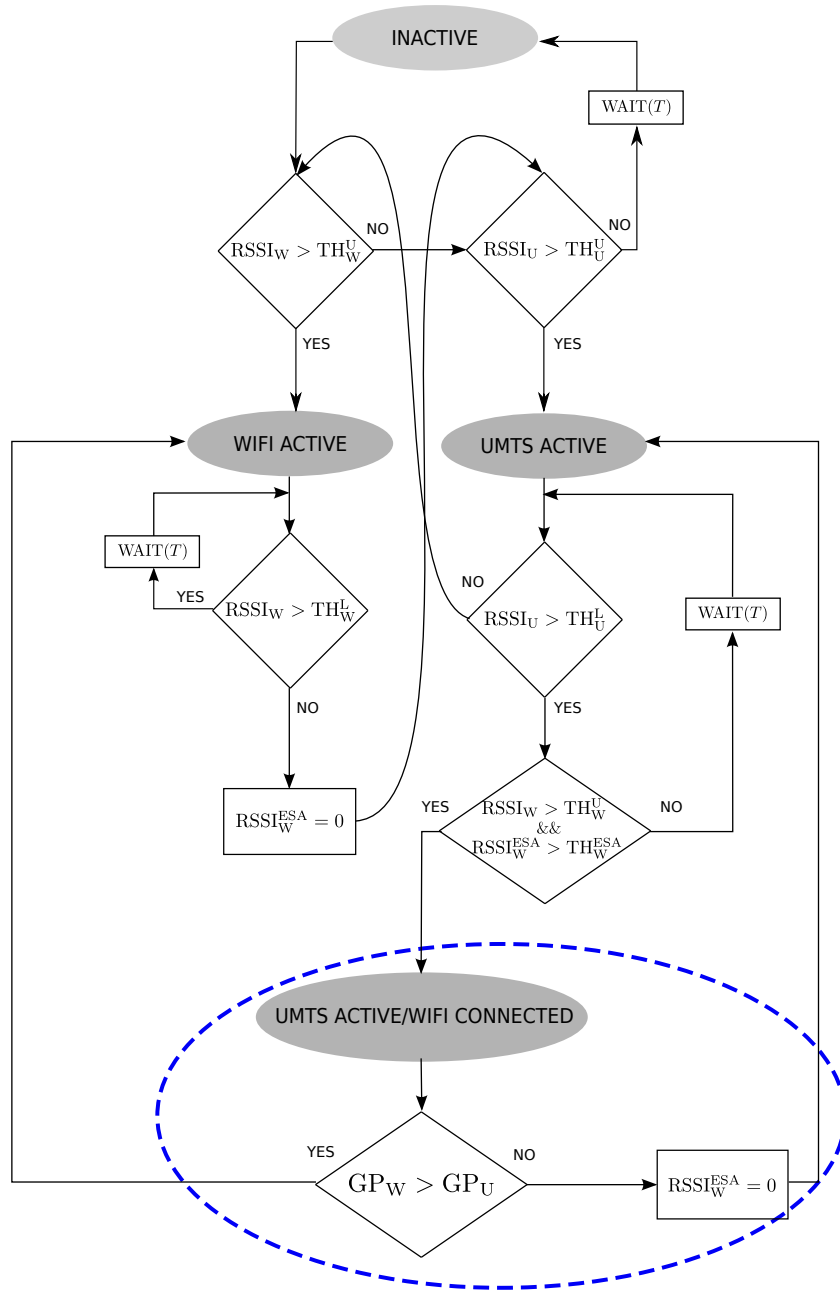


Figure 2.4: Dataflow of the low-complexity hybrid RSSI/goodput-based VHO algorithm.

Chapter 3

Vertical Handover: Performance Analysis in Cellular Systems

3.1 Introduction

The algorithms presented in the previous chapter are a simple but effective implementation of the three phases of the vertical handover. The last phase (execution) is certainly the most challenging and operative system-dependent. In this chapter we are going to describe the investigation of these algorithm both with simulation methods and with experimental implementation. The goal is to verify the feasibility of these algorithms and to study the performance. Regarding the feasibility, it is to highlight that the same implementation can not be extended to all operative systems, since the execution of the VHO needs to interact with the low level functions provided by the operative system. Every system has different methods to provide network control to the applications and sometimes, mostly on mobile devices, it doesn't provide control at all. It must be pointed out that our experimental implementations are based on common operative systems, without any sort of modification of the system itself. Therefore the applications for VHO developed are totally reusable.

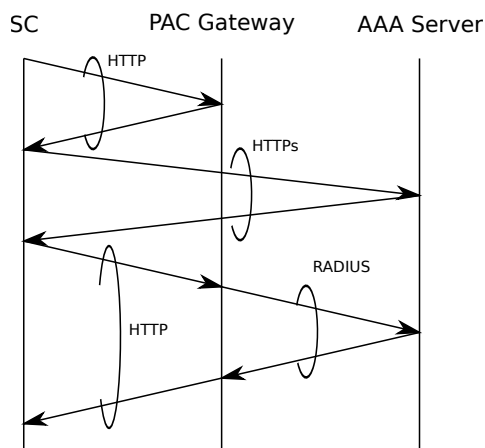


Figure 3.1: Sketch of the proprietary AAA mechanism used in the Guglielmo IEEE 802.11 network.

3.2 Windows Early Implementation

3.2.1 Experimental Setup

The main goal of this work is the test of our VHO algorithm in a realistic environment, leveraging on commercially available connectivity service providers and using standard MT devices. In particular, we perform our test using a notebook running Windows 7 OS, equipped with a Broadcom IEEE 802.11g compliant network interface, and integrated by a UMTS compliant USB Huawei dongle.

IEEE 802.11 connectivity is provided by a hot-spot of one of the biggest Italian Wireless Internet Service Provider (WISP), namely Guglielmo S.r.l. [40]. The hotspot is composed by a Brown IEEE 802.11 access point, integrated with a captive portal, while the Authentication Server (AS) is remotely located, as in the standard WISP Roaming (WISPR) configuration [41]. The proprietary AAA procedure, sketched in Figure 3.1, foresees two additional message exchanges with respect to the WISPr directives [41], thus increasing the time needed to complete the AAA procedure. The UMTS connectivity was instead offered by the Public Land Mobile Network (PLMN) of Telecom Italia, one of the most important Italian mobile operator. The sequence of

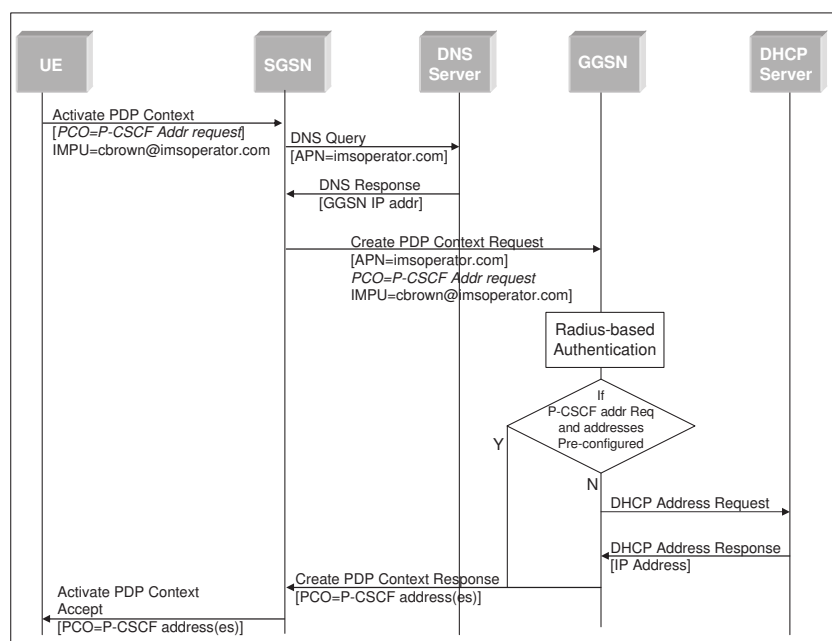


Figure 3.2: Sketch of the common DNS AAA mechanism used in the UMTS networks.

messages needed to complete the AAA procedure in a typical 3G network is sketched in Figure 3.2.¹ We observe that in both cases we have not a direct control on the traffic generated by other users.

We have implemented the VHO mechanism described in Section 2.3 and its novel low-complexity modifications (presented in Section 2.4 and Section 2.5) on top of a so-called Smart Client (SC) software. According to the WISPR directives [41], a SC is an application studied for enhancing the user experience, making automatic the AAA procedure. The current implementation of the SC runs on a Windows 7 platform,² but porting on other development platforms (e.g., Android, iPhone, etc.) is the subject of current work. Basically, the goal of the SC is that of constantly monitoring the status of the available connections and executing the VHO algorithm described

¹The AAA sequence adopted by Telecom Italia is not publicly available, but we are confident of its compliance with the 3GPP recommendations.

²The SC also supports the Microsoft Vista OS.

Table 3.1: Parameters of the VHO algorithm.

Parameter	Value
v_1	0.08
v_2	0.15
v_3	0.77
T_s	0.5 s

in Section 2.3. Additionally, once the MT initiates a VHO, the SC automatically has to take care of the proper AAA procedure. The SC controls both network interfaces, working with every IEEE 802.11 device able to provide real-time RSSI information, and with every 3G device (e.g., modem 3G, dongle USB) that supports the Microsoft Remote Access Service (RAS) API [42]. Due to the *make-before-break* approach, the SC has also to manage the routing functionalities of the OS, in order to make non-critical the multi-homed situation that appears after the authentication with the second network interface [43, 35].

The parameters of the VHO algorithm used in the experiments are summarized in Table 3.1, where $\{w_i\}$ are the weighing coefficients used to compute $\text{RSSI}_W^{\text{ESA}}$ according to (2.2). Our experiments were conducted in a building within the Department of Information Engineering of the University of Parma. The nearest UMTS base station is placed roughly at 1 Km from the building, and it guarantees a 2 Mbit/s downlink (384 kbit/s uplink) bandwidth, being compliant with the UMTS specifications. We have placed the hotspot in the WASN Lab (a room within the Department of Information Engineering), at 1 m above the ground. The hotspot is fed by an optical fiber network with 100 Mbit/s of symmetric bandwidth, but the hotspot imposes a symmetric limit on the available bandwidth equal to 2 Mbit/s, to replicate the conditions guaranteed to typical customers—despite the identical nominal downlink UMTS bandwidth (2 Mbit/s), the WiFi network has often outperformed, in our tests, the UMTS network. The test were performed by walking through the building, keeping the notebook on our hands and measuring (i) the time needed to perform

the handover and (ii) the goodput variations. The tests were always performed during working hours, in order to obtain results associated with realistic daytime situations. Hence, we have measured the classical following metrics.

- *Handover time*, which refers to the duration between initiation and completion times of the handover process. The initiation corresponds to the instant when the VHO manager begins the AAA procedure to connect to a given network. In particular, we consider the instant at which the first data packet routed via the new connection is successfully acknowledged by the remote destination.
- *Goodput*, which refers to the data rate delivered to the mobile terminals on the network [27]. In order to measure the end-to-end goodput, we periodically download a 400 kByte file from a remote server (hosted by Google), using the HTTP over TCP protocol. Since the goodput test is performed during a walk, there is a tradeoff between the duration of the download and the accuracy of the estimate. In fact, the distance covered during a single test is inversely proportional to the effective data rate. In order to limit this effect, we impose a double timeout over the download test: (i) a timeout of 1 s over the establishment of the HTTP connection; (ii) a timeout of 1 s over the data reception from the remote server.

3.2.2 Experimental Results

Low-complexity RSSI-based VHO Algorithm

In this subsection, we present the experimental results obtained while testing the VHO algorithm, introduced in Section 2.4, in the scenario described in Section 3.2.1. The handover time is automatically measured by the SC, and we average over 20 different runs, where in each run the path in the building and the corresponding handover instant have changed. In practice, we have collected the handover time values relative to 88 UMTS \rightarrow WiFi and 88 WiFi \rightarrow UMTS transitions.

In Table 3.2, the average handover time is shown (with corresponding standard deviation) for the WiFi \rightarrow UMTS and the UMTS \rightarrow WiFi transitions, respectively.

Table 3.2: Handover time for the VHO algorithm presented in Section 2.4 (mean, standard deviation, minimum, and maximum).

	WiFi \rightarrow UMTS	UMTS \rightarrow WiFi
Mean [s]	4.13	5.43
Std. Deviation [s]	1.76	3.30
Max [s]	10.72	15.69
Min [s]	2.41	1.22

In Figure 3.3, the instantaneous values of the handover time are shown as functions of the considered transitions. From Figure 3.3, it emerges that WiFi and UMTS networks have very different behaviors. In particular, the UMTS network exhibits an almost constant handover time, around its average of 4.13 s (Table 3.2). However, there is a relevant number of samples also in the region between 5 s and 10 s, while the few values above 10 s can be considered as outliers. On the other hand, the WiFi network is definitively worse than the UMTS network, since it presents a higher average value of 5.43 s (Table 3.2) and a much greater standard deviation. At the same time, one should observe that the minimum value is very small.

For the sake of completeness, we have also estimated, upon time discretization in 0.25 s bins, the Probability Mass Function (PMF) of the handover time. The obtained PMFs, for the two VHO operations, are shown in Figure 3.4. Observing the upper sub-figure in Figure 3.4, it seems that the handover time from the UMTS network to the WiFi network spreads between 1 s and 10 s. This relatively high variability has several motivations. First of all, in order to save energy, the MT is supposed to logout from a given network, once the VHO manager has selected the other network. Sometimes (more often in the WiFi network) the logout fails, and the remote authentication server keeps the authentication state for a certain timeout (roughly 60 s), before to automatically logout the user. In these cases, the UMTS \rightarrow WiFi transitions can experience a small handover time since the MT is de-facto already authenticated to the network. Moreover, while the authentication procedure at MAC layer has, practi-

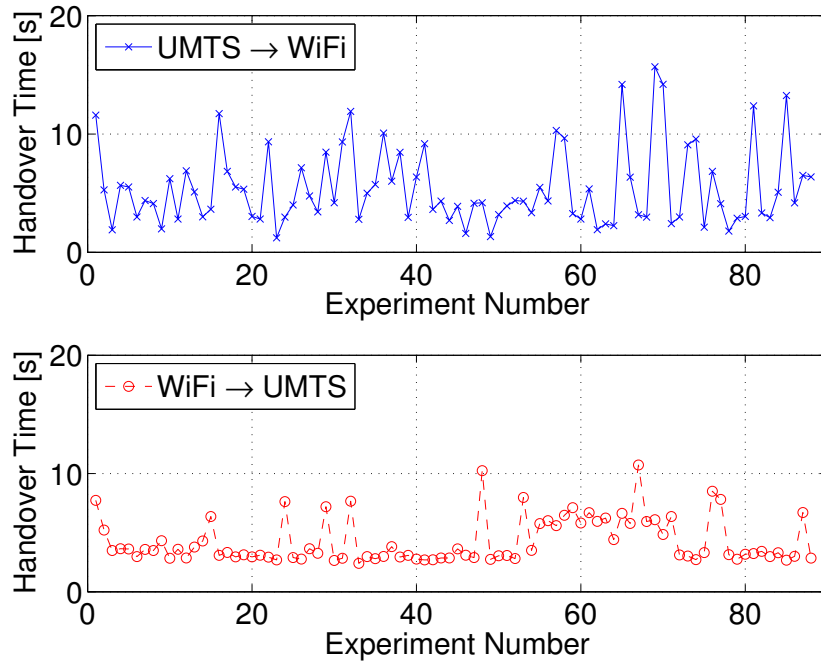


Figure 3.3: Instantaneous handover time of the VHO algorithm presented in Section 2.4.

cally, no impact, the DHCP release of an IP address might be source of randomness. In fact, the RFC recommends that the demanding host has to wait a random time in the interval (0 s,10 s), determining a strong unpredictability. Moreover, the DHCP mechanism is managed by the OS itself, making it difficult to understand its impact. Finally, when the WiFi signal is received close to (metal) furniture, the RSSI experiences large oscillations that can slow down the AAA procedure. Conversely, the RSSI of the UMTS network is more stable, and it has a small probability of experiencing such large variations.

From the results shown in the lower subfigure of Figure 3.4, it can be observed that the handover time from the WiFi network to the UMTS network is generally shorter and more predictable (i.e., its PMF is more concentrated) than that in the opposite direction. However, due to the no-coupling and the lack of any optimization,

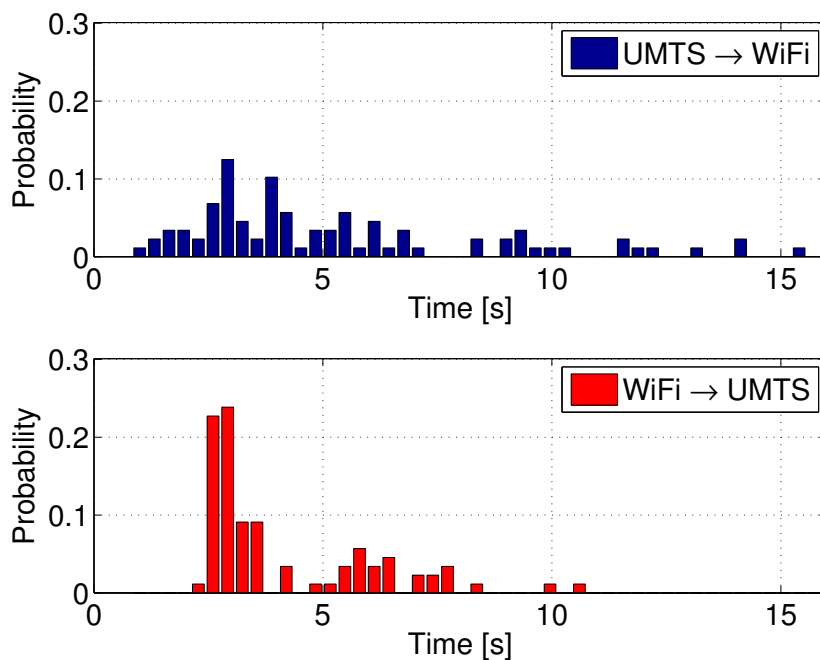


Figure 3.4: PMFs of the handover time of the VHO algorithm presented in Section 2.4: from UMTS to WiFi (upper) and from WiFi to UMTS (lower).

the handover times are long, also in the case of transition to the UMTS network. This result has somehow to be expected, since the proposed VHO algorithm is designed to be used for slowly mobile MTs, e.g., people moving from a place to another, but not in a highly mobile scenario.

In order to measure the goodput, we focus on a single walking path, chosen among the experimental data set. The selected path is shown in Figure 3.5, along with the layout of the environment where the tests were performed (a portion of the Department of Information Engineering of the University of Parma). The bold solid lines represent reinforced concrete walls, that are source of a strong signal attenuation. In correspondence to a glass window or a door (where the bold solid lines are interrupted), the signal attenuation is clearly much weaker. The path followed by the user is represented by a dashed line and is delimited by the words “start” and “end.”

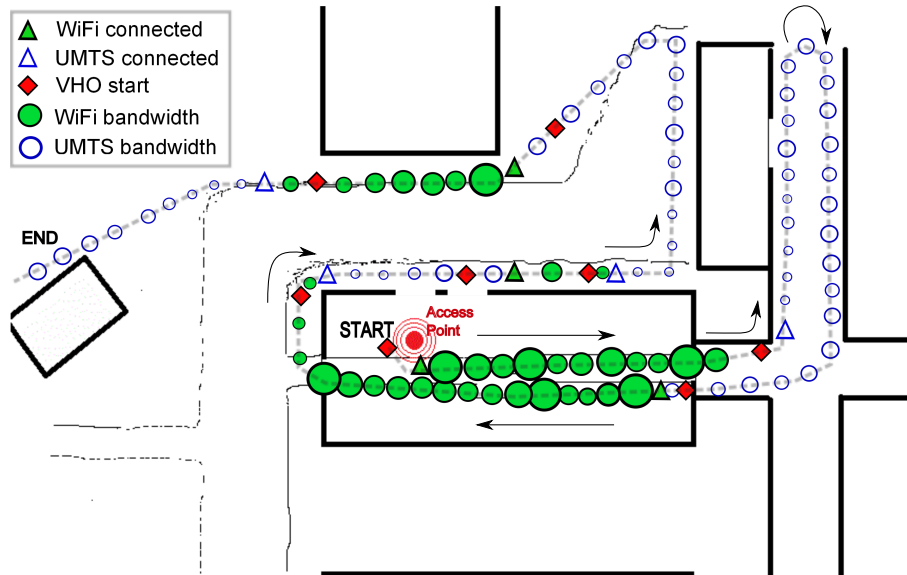


Figure 3.5: Goodput improvement using the low-complexity RSSI-based algorithm presented in Section 2.4.

The circles drawn along the path represents the measured available goodput, which is proportional to the diameter of the circle. Filled circles indicate connection to the IEEE 802.11 network, while in correspondence to empty circles the MT is connected to the UMTS network. A (filled) diamond denotes the beginning of a VHO procedure, while the triangles indicate when the procedure has been successfully completed. The filled triangles indicates that the VHO procedure has established a WiFi connection, while empty triangles denotes the establishment of a UMTS connection. We stress the fact that between diamonds and triangles the MT is still connected with the old network, in order to avoid loss of connectivity before finalizing the VHO. Finally, the distance between the circles is directly proportional to the duration of the bandwidth test and, hence, it is inversely proportional to the available bandwidth.

Observing Figure 3.5, it can be concluded that the proposed VHO algorithm works well also in a complicated hybrid indoor/outdoor scenario, while walking between reinforced concrete walls. In particular, the dimension of the circles show that

the goodput offered by the WiFi network, despite its high variability, is generally higher than that offered by the UMTS network. The UMTS network, instead, has shown a very homogeneous behavior along the entire path (both indoor and outdoor). From Figure 3.5, one can also observe that the presence of a window just in front of the WiFi access point is detrimental for the VHO algorithm. In fact, while walking in front of the windows, the MT receives a very strong WiFi signal for several meters, and then the VHO quickly invokes the handover procedure. Unfortunately, because of the long duration of the AAA to the WiFi network, the MT becomes connected to the WiFi network too late, when the user has already lost its “window” of good connectivity. We remark that this particular situation could not be counter-acted by our VHO algorithm. In fact, in order to prevent the two consecutive handovers that happen in front of the window, it would be necessary to change the coefficients of the ESA filter, thus yielding a higher handover latency. In other words, coping with this very particular case would degrade the performance in other circumstances.

In order to give more insights on the algorithm behavior, in Figure 3.6 we show³ $RSSI_W$ (in the upper subfigure) and the goodput (in the lower subfigure) experienced by the MT as functions of time. Note that one discrete time corresponds to a position along the path shown in Figure 3.5. The goodput has a bimodal behavior, since it has a floor at roughly 800 Kbit/s, corresponding to the 3G connection, while it increases to a higher level as soon as the MT can use the WiFi connection. More precisely, the IEEE 802.11 goodput has a very irregular shape, but it is higher than that of the 3G network almost always. Looking at both Figure 3.5 and Figure 3.6, one can observe that, inside the building where the WiFi access point is placed, the RSSI of the WiFi signal varies slowly and the goodput is more regular and higher. On the other hand, in correspondence to the windows, when the handover procedure starts frequently, the goodput is more irregular and it is difficult to appreciate the benefits offered by a VHO solution.

³We omit $RSSI_U$, since it is always higher than TH_W^L and practically constant.

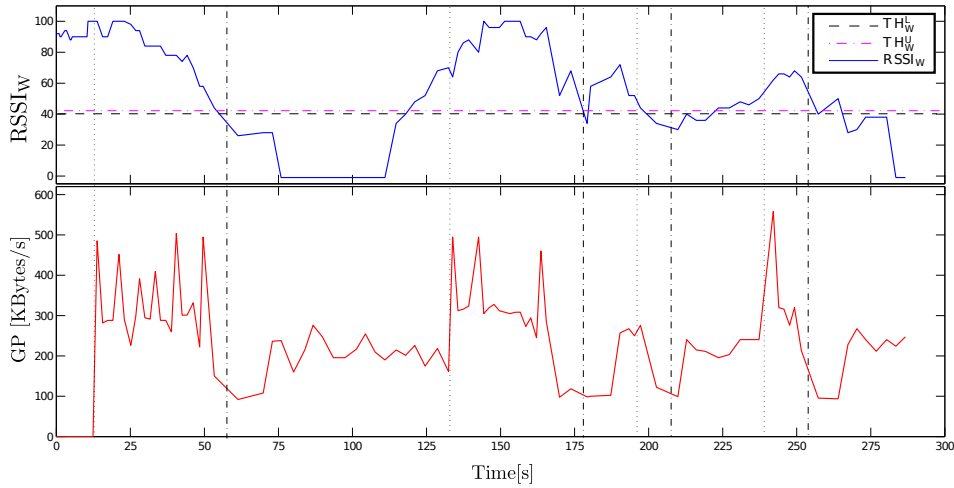


Figure 3.6: RSSI and goodput experienced by the MT following a sample path using the low-complexity RSSI-based algorithm.

Low-complexity Hybrid RSSI/Goodput VHO Algorithm

The performance of the hybrid VHO algorithm has been analyzed considering a slightly different path in the experimental scenario, shown in Figure 3.7. In this particular scenario, the assumption of a larger WiFi network bandwidth with respect to that of the UMTS network, is not satisfied. As in Figure 3.5, the diamonds in Figure 3.7 indicate the beginning of the handover procedure. In this case, the handover may fail with a higher probability, due to the additional goodput check. This motivates the presence, in Figure 3.7, of several diamonds not followed by triangles.

In Figure 3.8, the RSSI and goodput, relative to the VHO algorithm introduced in Section 2.5, are shown as functions of time. For the sake of clarity, a direct comparison with the RSSI-based VHO algorithm is also considered. In the upmost subfigure, $RSSI_W$ is shown together with the corresponding upper and lower threshold; in the lowest subfigure, the estimated goodputs GP_W and GP_U are directly compared; finally, in the middle subfigure the overall goodput guaranteed by the RSSI-based VHO algorithm (denoted as GP) and the hybrid VHO algorithm (denoted as GP^H) are directly compared. According to the results in Figure 3.8, in the initial phase the

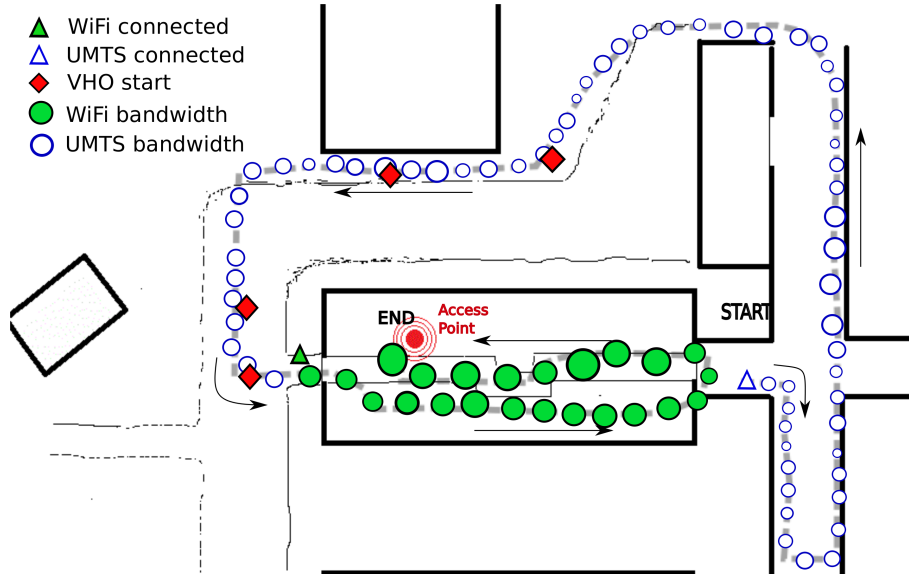


Figure 3.7: Goodput improvement using the low-complexity RSSI-based algorithm presented in Section 2.5.

MT is disconnected from the WiFi network, because the received power is too low. At a given point of the path, $RSSI_W$ starts to quickly increase, and then it soon overcomes the threshold TH_W^U . A few seconds later, therefore, the filtered RSSI also goes over its threshold ($RSSI_W^{ESA} > TH_W^{ESA}$). At this moment, the RSSI-based VHO algorithm begins the handover to the WiFi network, ignoring the fact that the effective goodput available in the WiFi network is lower. On the other hand, in the case of the hybrid VHO algorithm, the MT starts the bandwidth estimation process, after which it decides to keep the UMTS connection because it becomes aware of the guaranteed higher goodput. In other words, the hybrid VHO algorithm has shown to have better goodput performance than that of the RSSI-based algorithm, at the cost of a slightly longer handover time and a slightly higher complexity. These prices to be paid are due to the presence of a double connectivity situation, which requires to properly configure the OS routing table, in order to perform the bandwidth test in both networks, without penalizing the user.

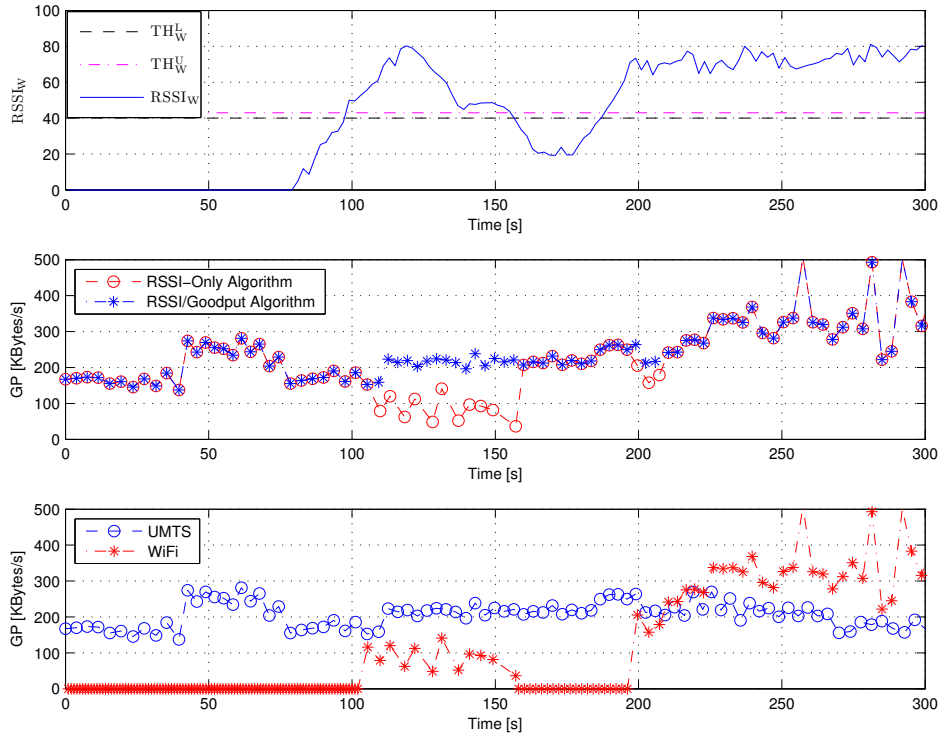


Figure 3.8: RSSI and goodput experienced by the MT following a sample path using the hybrid RSSI/goodput VHO algorithm.

In terms of handover time, the hybrid VHO algorithm has shown similar performance to the RSSI-based one, the only difference being a longer handover time in the UMTS \rightarrow WiFi transition because of the time needed to perform the bandwidth test. The additional delay is upper bounded by the sum of the two timeouts introduced in Section 3.2.1 (HTTP connection and data reception), which is approximately equal to 2 s in the standard configuration.

3.3 Simulation-based Investigation

In this section, we present a simulation-based investigation of the proposed VHO algorithms in HetNet scenarios by relying on the Opnet simulator [44]. We first examine a scenario where a single node is moving across the network, in order to reproduce and validate the experimental results presented in Subsection 3.2.1. However, as it will be described in more detail in the next subsection, the current version of the simulated VHO algorithm between WiFi and UMTS networks is only based on the received power of the WiFi interface. The UMTS signal, in fact, is assumed to be always present, since in realistic urban scenarios one can assume that a 3G connection is available at every time. Then, we will extend the analyzed scenario, considering a large number of nodes randomly moving in the HetNet and carrying out, possibly, VHO procedures. Note also that the realistic AAA procedures for 3G and WiFi networks are not exactly replicated in the simulator, but their effects are reproduced by adding, during the VHO process, a random delay drawn from the PMF shown in Figure 3.3. In order to investigate also the hybrid VHO algorithm in the last simulated scenario, we used, as cellular network, a Lte network. This choice permits us to bypass the lack of the HSDPA and HSPA+ modules among the technologies implemented by OPNET. The Lte system, in fact, provides a high data-rate downlink connection that can be compared with that of a HSPA+ equipped 3G system⁴

⁴Downlink connection data-rates for HSPA+ and Lte are not the same, although both technologies provide several tens of Mbps depending on the physical configuration (multiple-antenna techniques, spectrum bandwidth, multi-cell technique, terminal category etc.). We point out that a comparison between WiFi and UMTS—without HSPA—available bandwidths wouldn't make sense, since WiFi data-rate is always much higher than UMTS data-rate. Furthermore, investigating the Lte technology instead of a 3G system, can be interesting as Lte system, at present, is still being deployed in many countries and therefore it is a very relevant topic. For VHO between WiFi and Lte we created a scenario where 40 mobile nodes move randomly across the coverage area of a Lte base station, within this area is present a WiFi access point. All the mobile nodes implement the hybrid VHO algorithm. For this scenario we show, first, the actual available bandwidth against the time seen by a single node, this result is similar to the result shown in Figure 3.8, then we show the total goodput of the networks averaged over the time against the load of the network.

3.3.1 The OPNET Simulator

OPNET is a modular discret-event simulator providing support for several technologies, among many others, WiFi, UMTS and Lte networks. The WiFi implementation adheres to the IEEE 802.11g standard, whereas the UMTS implementation, supposed to be compliant with the 3GPP Release 5, does not support HSDPA and it offers a maximum downlink throughput equal to 384 kbps (the HSDPA maximal nominal throughput is instead equal to 14.4 Mbit). The Lte implementation is compliant with the 3GPP Release 8 and 9, therefore it is possible to set the number of antennas of the devices (MIMO, multiple-input and multiple-output) and the spectrum bandwidth.

In all the networks of interest, the MTs implement the entire protocol stack, also including the application layer on which desired applications can be run. In particular, we consider three downlink scenarios, the first two with a FTP application running. In these scenarios, each node downloads, every second, from an FTP server a file of a given size, which depends on the scenario of interest. In the second scenario the MTs launch their application one at a time spaced by 10 s. The first MT wait 100 s before to run its application, in order to let every MT to associate to the UMTS network.

In the last scenario, each node executes a HTTP application, namely, every 60 seconds on average, each node downloads a web page of a given size. Lte MIMO configuration is set to 2x1 (two transmitter antennas and one receiver antenna) for the downlink and 1x2 for the uplink, spectrum bandwidth is set to 20 MHz. Wireless communications between MTs and APs (or base stations) follow a signal propagation model where fading with a Ricean distribution is also included. However, in the considered settings, the impairments due to fading are relatively small and, therefore, the signal attenuation is very similar to a free-space model.

The main challenge in simulating a VHO algorithm is the implementation of a node module able to jointly control both a WiFi and a UMTS (or Lte) radio interface. In our simulator, for the couple WiFi/UMTS this goal has been pragmatically achieved by coupling two independents nodes, equipped, respectively, with a WiFi and an UMTS network interface and the corresponding protocol stack, as shown in Figure 3.9. However, since the nodes are forced to move together through the network at a fixed distance of 10 cm from each other, they appear, from a network perspective,

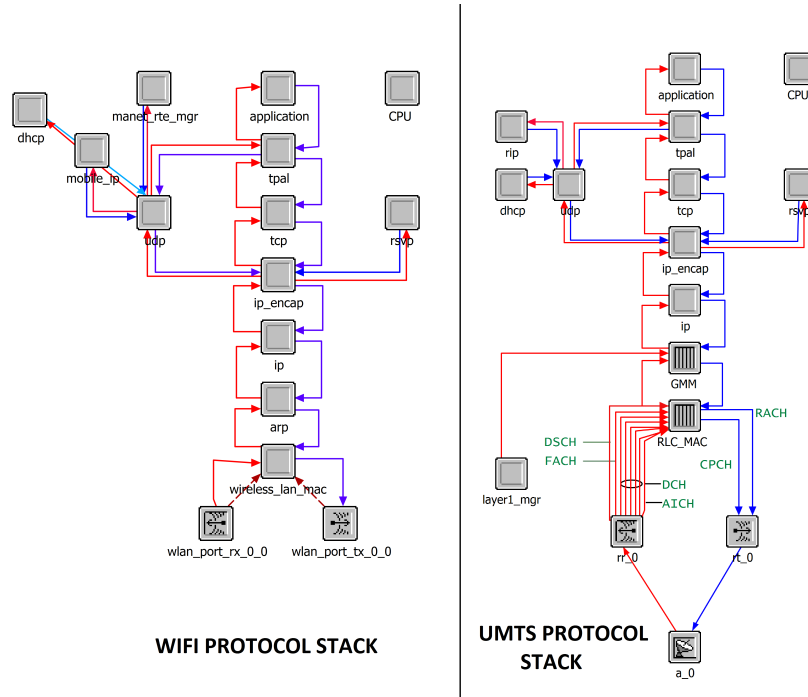


Figure 3.9: Protocol stack of a WiFi node and a UMTS node in the OPNET simulator.

as a single node.

The VHO algorithm is implemented at the MAC layer of the WiFi node, i.e., in the module denoted as *wireless_lan_mac* in Figure 3.9. As already anticipated at the beginning of this section, the power level of the UMTS signal is assumed to be always greater than TH_U^U . Moreover, the power of a beacon frame sent by the WiFi access point is computed and filtered as described in Subsection 2.4. If the conditions triggering the VHO procedure represented in Figure 2.3 are verified, the MAC layer module sends an interrupt to the application layer of both WiFi and UMTS nodes. In our simulator, the threshold is set to -83 dBm. If the VHO is towards the WiFi network (i.e., if $RSSI_W > TH_W^U$ and $RSSI_W^{ESA} > TH_W^{ESA}$), the FTP application on the WiFi node starts receiving data. On the other hand, the application on the UMTS node stops the data generation, in order to simulate a hard disconnection at this interface.

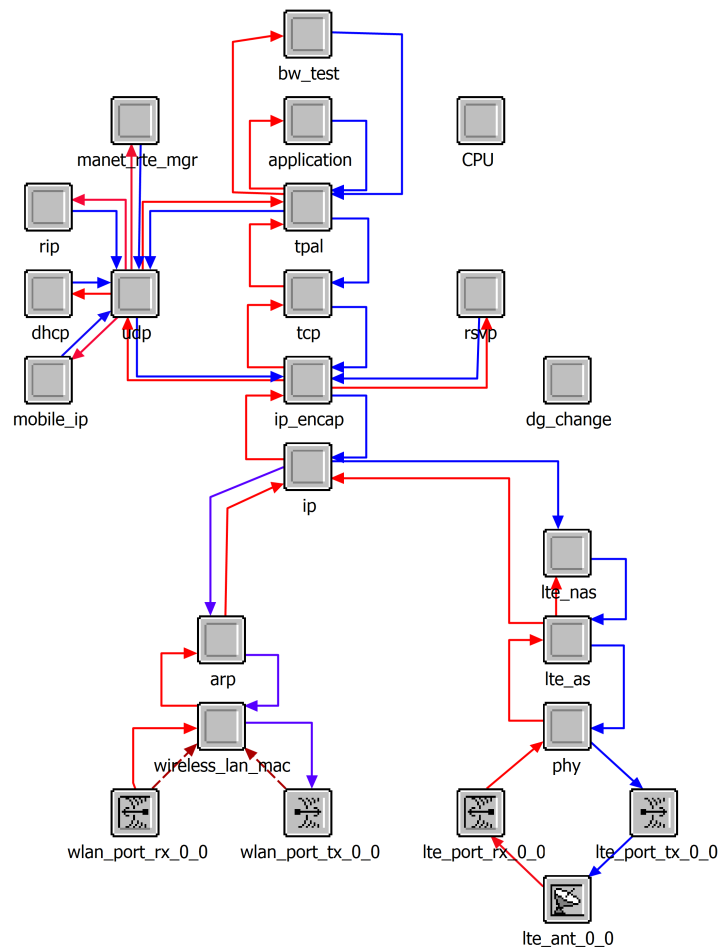


Figure 3.10: Protocol stack of the new node with WiFi and Lte interfaces developed with OPNET modeler. The module *bw_test* is responsible for the computation of the available bandwidth estimation, while the *dg_change* module manipulates the IP routing table in order to change the default gateway and to redirect specific traffic over a given network.

The opposite operations are performed if the VHO is towards the UMTS network.

In the case of WiFi/Lte VHO, we developed a new node having both the WiFi and Lte interfaces as shown in Figure 3.10. In this node the *dg_change* module is responsible for the commutation of the traffic between WiFi and Lte interface. The *wireless_lan_mac* module computes the WiFi power level as explained in the previous paragraph, while the *bw_test* module executes the available bandwidth estimation. Both the *bw_test* and the *wireless_lan_mac* modules communicate their metrics to the *dg_change* module that, if necessary, triggers a VHO.

Unlike what explained in 2.4, the bandwidth estimation is executed not by downloading a simple file from a remote server, rather, a more complex algorithm, called *WBest* [37], is implemented. The advantage of this algorithm is the lower quantity of data downloaded from the server, and thus the lower cost of the bandwidth test. This algorithm, in fact, uses a train of packets, sent by the server, to calculate the average delay between the packets, introduced by the multi-hop network path. On the basis of this delay and of the size of the packets it is possible to estimate with sufficient reliability the available bandwidth of a network path with much less bytes—approximately 100 kB against 400 kB of the algorithm discussed in 2.4.

3.3.2 Performance Results

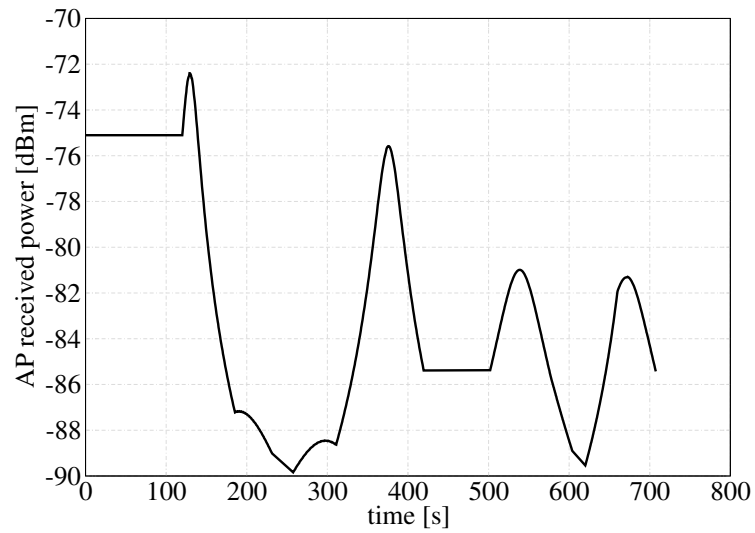
In this subsection, we analyze the performance predicted by the OPNET simulator for three possible scenarios of interest. In the first case, we consider the same scenario used to experimentally validate the VHO algorithm and shown in Figure 3.5: a single node moves, for several times, from the WiFi coverage area to the UMTS cell and vice-versa. This scenario is expedient to verify, trend-wise, the experimental results presented in Section 3.2.2 and, therefore, to obtain a sanity-check of our simulator. In the second case, 40 nodes randomly move inside the same area of interest. A unique WiFi access point is placed in this area, whereas four UMTS antennas are placed to ensure total cellular coverage over the entire area. This is representative of a realistic scenario where many users may be in the same city area close to an access point (e.g., a crowded square) and may want to connect to the Internet (e.g., through their smartphones). In all simulations, the metrics are measured starting from the end of

the initial transient, of duration equal to 100 s, in order to allow the UMTS MTs to perform the network association procedure. The third scenario aims to validate the experimental results of the hybrid VHO algorithm, in this scenario are placed a WiFi access point and a Lte eNodeB (Lte base station), the coverage area of the Lte eNodeB has a radius 1 km long, while the WiFi range is about 400 meters long. In every scenario, the performance metric of interest is the goodput, defined as the total received traffic at application layer.

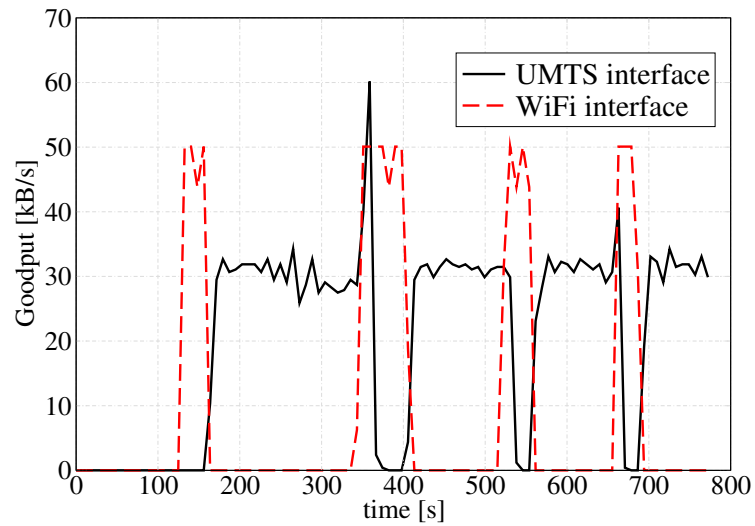
Single Node Scenario, UMTS and WiFi networks

The first scenario is shown in Figure 3.11. In this case, both applications try to periodically (each second) download from the FTP server a file whose dimension is interface-dependent: 35 kB from the UMTS interface and 50 kB from the WiFi interface. This difference in file size has the sole purpose to differentiate in the graph the UMTS traffic from WiFi traffic. The node moves on the deterministic path, highlighted in Figure 3.11 and representative of the realistic one considered in Figure 3.5, at a constant speed $v = 2$ m/s. The total length of the path is approximately 1.4 Km.

In Figure 3.12 (a), the received power (in dBm) of the beacon frames is shown. Beacon frames are sent by the WiFi AP every 0.5 seconds. In this simulated scenario, the curve is smoother than the corresponding curve in Figure 3.8. This is due to the fact that, unlike the experimental scenario where the signal is impaired by the reflections of the buildings, in the simulated scenario only reflections of the ground are considered, therefore the effects on the signal are very similar to that of a free-space scenario. In Figure 3.12 (b), the goodput, due to either WiFi or UMTS connections, of the node moving along the path described in Figure 3.11 is shown as a function of time. As one can see, every time a beacon frame is received from the WiFi access point, as shown in Figure 3.12 (a), the VHO algorithm is triggered and the goodput on the WiFi interface increases, whereas the UMTS interface is triggered down. We recall that the threshold on the received power is set, in our simulator, to -83 dBm. This confirms the experimentally acquired results shown in Section 3.2.2.



(a)



(b)

Figure 3.12: (a) Received power on the WiFi interface and (b) goodput on both the interfaces for the single node scenario.

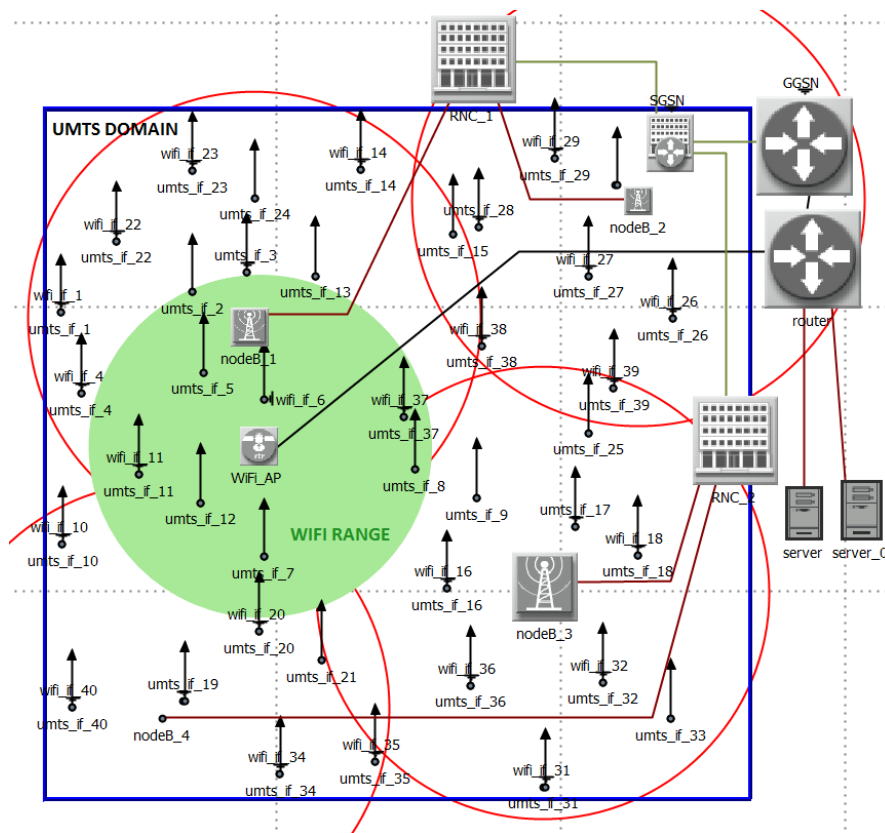


Figure 3.13: Second simulation scenario with multiple MTs moving according to the Random Waypoint mobility model.

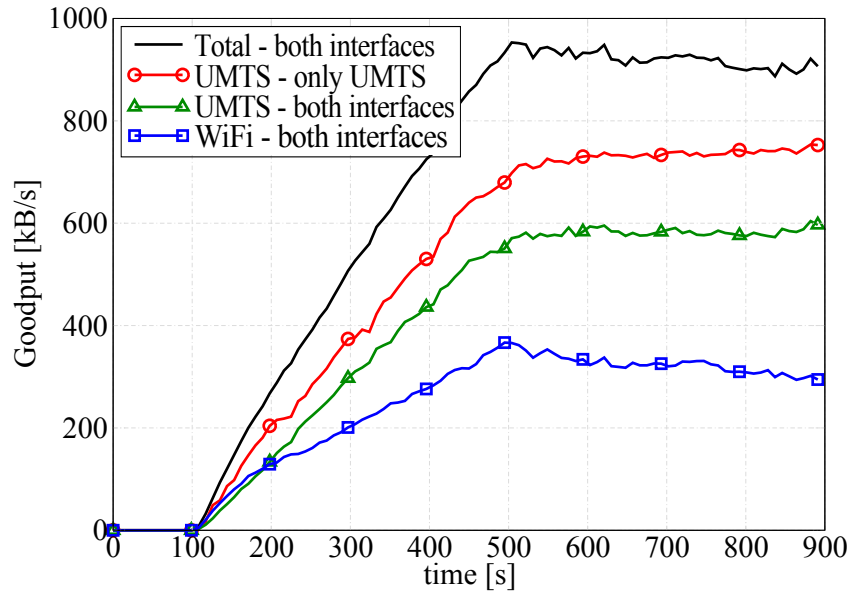
nected MTs as functions of time. Sixty independent simulation runs were performed to eliminate statistical fluctuations of the results. In all simulations, the metrics are measured starting from the end of the initial transient, of duration equal to 100 s, in order to allow the UMTS MTs to perform the network association procedure. One can first observe that the goodput is an increasing function of the time. This is due to the fact that, as time passes, the number of MTs in the network increases and, therefore, the total received traffic on each (or both) the interfaces becomes higher. Moreover, from Figure 3.14 (a), one can see that when both the MTs' interfaces are active, the total received goodput approaches 1000 kB/s, which is close to the maximum possible value of 1200 kB/s. On the other hand, when only the UMTS interface is active, the traffic saturates to approximately 800 kB/s. A similar result holds for the scenario where both interfaces are active. Therefore, the presence of a WiFi connection supplies further connectivity needed to reach the (theoretical) highest possible goodput.

Multiple Node Scenario, Lte and WiFi networks

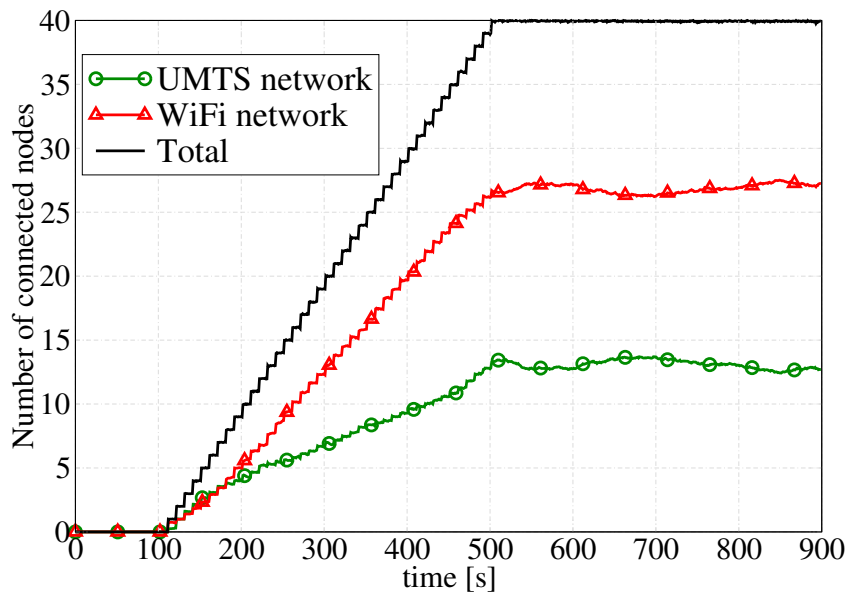
In Figure 3.15, the third simulated scenario is shown. Also in this case, there are 40 MTs moving across the area depicted by a blue quadrangle, the filled circle is the coverage area of the WiFi access point. The mobility model and the speed of the nodes are the same as the ones in the second scenario. Both WiFi access point and Lte system (eNodeB and EPC—Evolved Packet Core) are connected to an IP cloud, called backbone. To the IP cloud are also connected four HTTP servers.

In Figure 3.16 one can see the values of the bandwidth estimation computed by a single node. For the first part of the simulation, both WiFi and Lte network have no traffic to deliver, therefore the estimated bandwidth is the maximal capacity offered by the networks. After approximately nine minutes some traffic is generated on the Lte network, after a few seconds the mobile node detect the strong decrease of available bandwidth over the Lte network, and thus it execute a VHO procedure towards the WiFi network.

In Figure 3.17 is shown the performance of the VHO hybrid algorithm compared with the RSSI-based algorithm. The results are obtained executing 16 simulation



(a)



(b)

Figure 3.14: (a) Goodput and (b) number of the transmitting MTs in each network for the multiple node scenario.

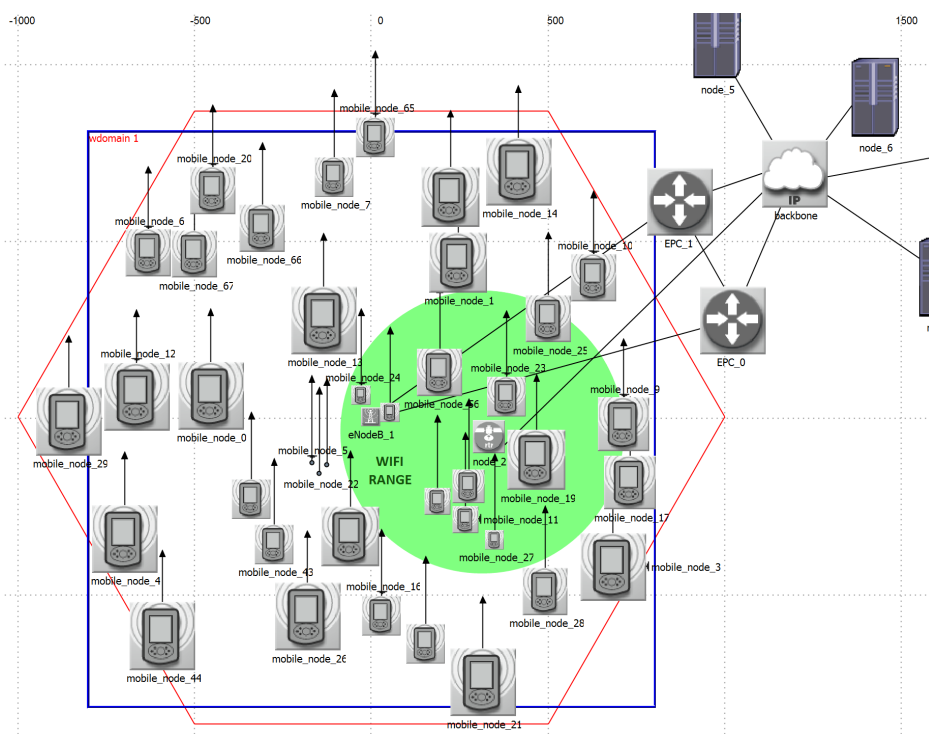


Figure 3.15: Third simulation scenario with multiple MTs moving according to the Random Waypoint mobility model. Every node has both WiFi and Lte interfaces.

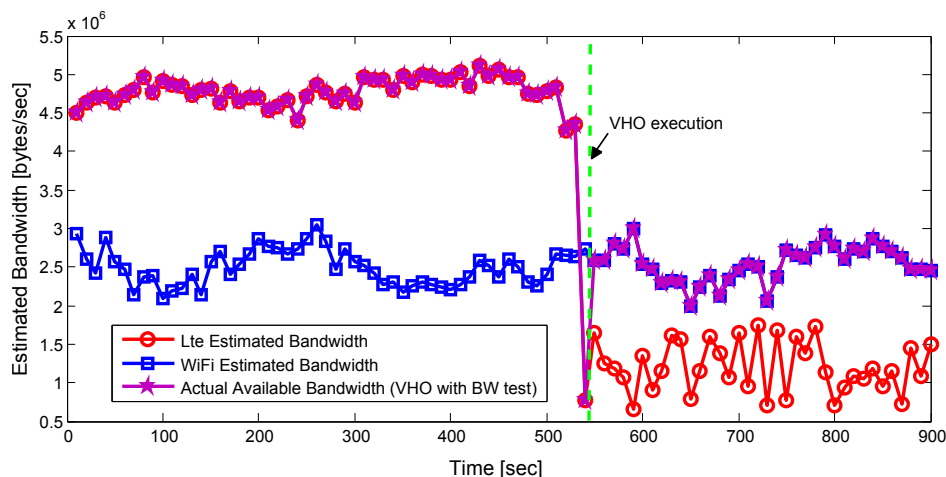


Figure 3.16: Simulation results for the third scenario. Here are shown the bandwidth estimation and the related selected network, of a single node during a 15 minutes long simulation.

runs—eight runs for the RSSI-based algorithm and eight runs for the VHO hybrid algorithm—of the third scenario for a simulated time of 15 minutes. At every run the load of the network increases, this is obtained by augmenting the average size of the HTTP pages downloaded from the server. For every run we computed the average aggregated goodput of the networks. The graph shows, with different curves, the goodput delivered to the mobile nodes through the WiFi interface, through the Lte interface and their sum. The total data load is increased in order to saturate both the WiFi and the Lte networks. We set the WiFi range size so wide, that the number of nodes, within the WiFi coverage, are enough to saturate the WiFi band before the Lte network is overloaded. That is the WiFi network saturates before the Lte network.

One can see that the goodput delivered through WiFi interface doesn't change considerably, whether we use the hybrid or the RSSI-based algorithm. A slight decrease is perceptible in the case of RSSI-based algorithm due to the increasing number of packets collision at WiFi physical layer. The most noticeable difference between the two algorithms is represented by the goodput through the Lte interface.

Obviously, the goodput reached by the Lte network when it is saturated, doesn't change, rather, when the data load equals approximately 7 MBps, one can notice a faster growth of the goodput over the Lte network. This is due to the fact that, when the WiFi cannot transport further data anymore, and the Lte band isn't overloaded yet, the nodes under WiFi coverage implementing VHO hybrid algorithm, switch their connection over Lte. This fact could be seen as a load balancing, between Lte and WiFi, executed by the VHO hybrid algorithm. In fact, thanks to the bandwidth estimation metric, the mobile nodes are somehow aware of the saturation of the WiFi network and can select the Lte connection.

For the sake of completeness, we point out that the total goodput is never equal to the total load requested by the mobile nodes. This is due to the implementation of the HTTP application: the time interval between two HTTP requests is not constant at 60 seconds, rather it has an exponential distribution with 60 seconds average, if it happens that a HTTP request arrives when the previous request (for the same mobile node) isn't carried out yet, the HTTP client interrupt the download in progress to start the new one. The load displayed at the x-axis doesn't consider this reduction of requested data, and therefore the aggregated goodput never equals the theoretical load.

3.4 Android Implementation

3.4.1 Application Development and Trial

The main goal of this work is to develop a novel VHO-enabled mobile application and to evaluate its performance on the field for a long period with real users and heterogeneous conditions in terms of devices, networks, and mobility behaviors. In the remainder of this section we first summarize the used VHO algorithm, and, then we describe the designed application and the experimental setup.

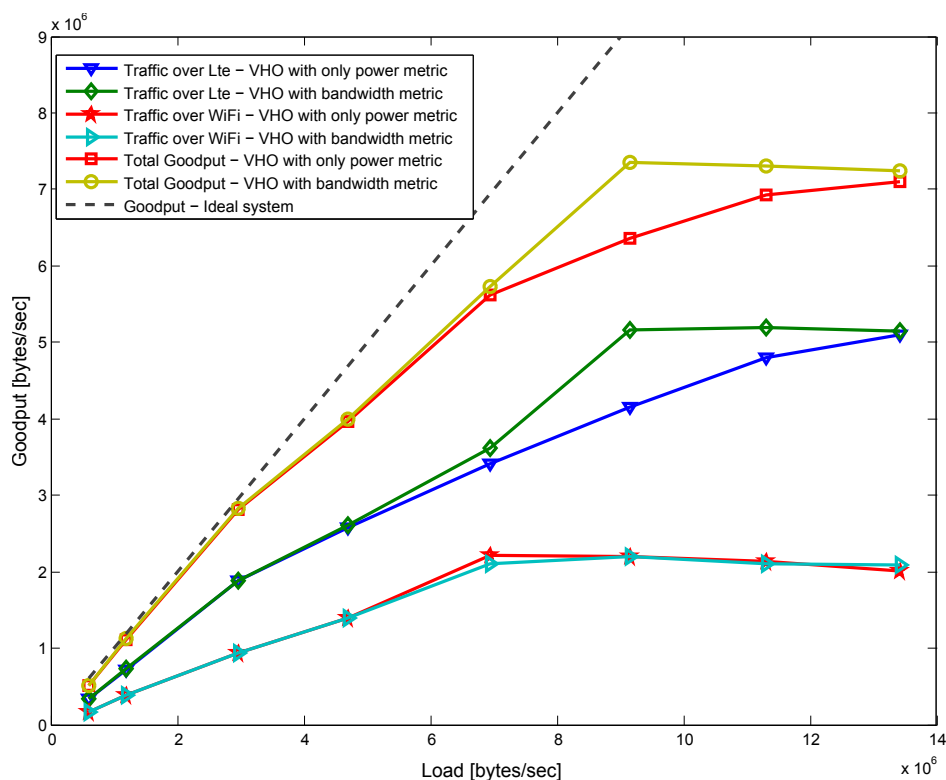


Figure 3.17: Simulation results for the third scenario. Total aggregated goodput of the networks. The data load increases till to saturate both WiFi and Lte networks. Results for VHO hybrid and RSSI-based algorithms are shown.

VHO Algorithm on Mobile Devices

In Section 3.2 is described and analysed the Windows implementation of the make-before-break VHO algorithm presented in Section 2.5. Even though this implementation of the execution algorithm does not provide a (strictly-speaking) seamless handover, it guarantees continuous connectivity by means of proper IP routing table modifications. As we tried to port this algorithm to a mobile device operating system (e.g., Android or iOS), the routing table and the network interfaces could not be properly controlled. Therefore, we developed a derived efficient VHO algorithm

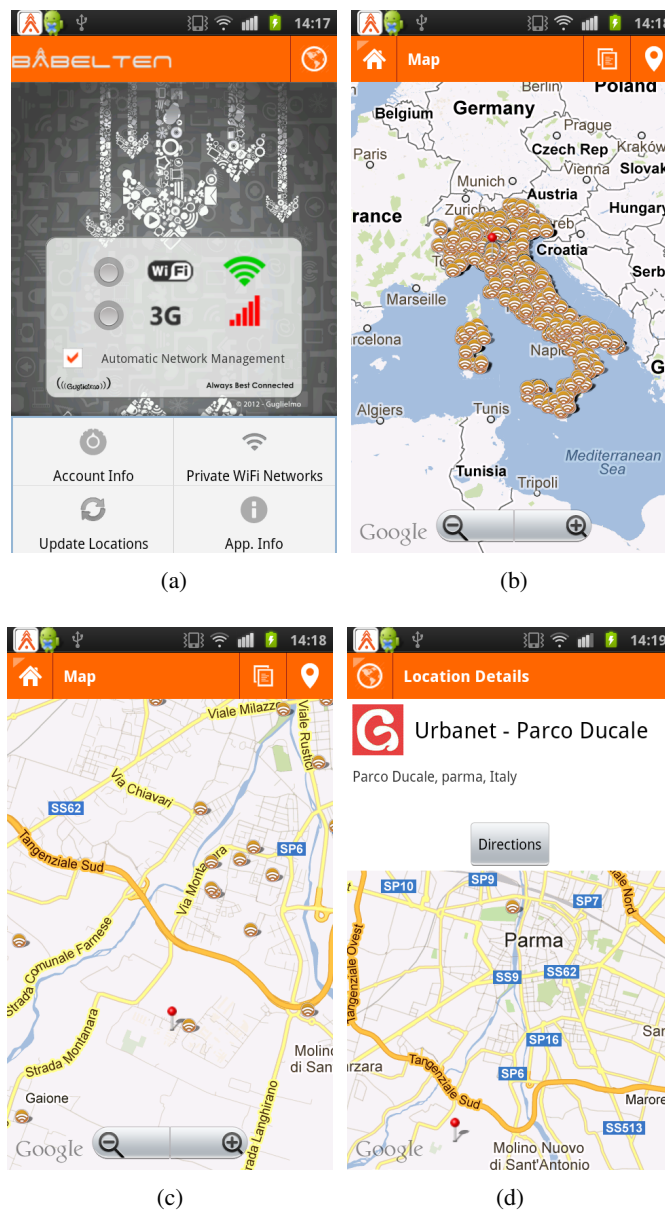


Figure 3.18: BabelTen User Interface: (a) main view, with application menu; (b) map view; (c) Guglielmo hotspot coverage; (d) single hotspot details.

for mobile platform based only on Received Signal Strength (RSS) and achievable goodput metrics.

Mobile Application

The designed application called BabelTen has been developed on the Android platform. The reasons behind this choice reside, on one hand, on the ever increasing use of the Google mobile operating system and on the other hand, on the possibility to have a sufficient control of device network interfaces (without jail-breaking the device) that at the time of writing is absolutely impossible (without jail-breaking the device) on other common platforms such as iOS and Windows Phone. Typically (and, most of the time, by default) many mobile Operating Systems (OSs) (including Android) automatically manage network connectivity preferring a WiFi connection with respect to regarding other available connections without any kind of additional evaluation (typical of a VHO algorithm). This brings the practical risks as keeping the user connected (also for a long period) to a low performance network or, in the worst case, to a network without connectivity (for example if a WiFi hot post is protected by a captive portal and the user has not completed the login procedure yet).

BabelTen's aim is to overcome these OS related limitations to give the user a better experience, selecting the available network with the best performance. Screenshots in Figure 3.18 illustrate the application design and main User Interface (UI) characteristics. BabelTen's UI allows to keep the connection status monitored and to select the operational mode (automatic network management by default or manual connection, if needed). It also allows to visualize, on a map available hotspot locations and, if needed, obtain detailed information and navigate to a target location through the native navigation system integrated in the OS.

Notwithstanding the detailed and easy-to-use UI, the application has been naturally designed to remain active in background, managing automatically user connectivity to guarantee to be always best connected in any situation. To reach this purpose BabelTen implements the VHO algorithm presented in previous section, involving at the same time in the decision process private and commercial WiFi access points. User networks are automatically imported through Android settings without

the need for the user to give private credentials and delegating to the mobile platform the storage and management of this sensible information. Commercial hotspots are provided by one of the biggest Italian Wireless Internet Service Provider (WISP), namely Guglielmo Srl. The list of available locations is maintained and updated by the application and the access to the network is based on a proprietary authentication algorithm. This procedure is absolutely transparent to the user since it is based on the SIM card IMSI number and does not require any additional configuration.

The decision algorithm has to get around the native characteristics of Android OS for WLAN connections. Since Android does not allow to keep multiple connections—over different interfaces—active, we had to choose a "break-before-make" approach, minimizing the off-line period peculiar of this approach. Despite this platform limitation and in order to allow the user to be ABC, we have implemented a periodic bandwidth estimation to evaluate the state of the current network. Therefore, BabelTen periodically performs a bandwidth and delay estimation on WiFi (B_w) and/or mobile (B_m) to evaluate the real throughput on both interfaces. After the bandwidth estimation, if B_w is significantly greater than B_m on commercial hot posts (WiFi networks are in any case the preferred choice if the difference is small) the application switches to mobile and put the WiFi SSID in a temporary blacklist to avoid a ping pong effect between available networks. Beyond this estimation, the application schedules periodically an additional connectivity check to verify if the device is connected Internet or if for some reasons there is a lack of connectivity (for example due to a logout on commercial hotspots). If this is the case, the application performs an handover to mobile putting the SSID in the blacklist. When the blacklist period has expired and if the network is still in the device range, it is automatically re-evaluated in the VHO selection process as candidate network.

In order to estimate not only the real bandwidth but also the amount of amount of data traffic on both interfaces and evaluate the offloading percentage during the trial, BabelTen uses a dedicated Android API to retrieve the amount of transmitted and received in KBytes on WiFi and mobile interfaces. After each sampling period harvested data are transmitted to a remote server to be collected in a dedicated database and analyzed. In order to be evaluated, grouped and geographically visualized on a

map, mobile logs are associated to the cell-ID and WiFi data to the MAC address of the access point.

Experimental Setup

The aim of our trial is to collect as much data as possible in a order to properly evaluate BabelTen's performance and offloading behaviors during a long term real experiment, with final users in a deeply heterogeneous context in terms of devices, networks and mobility behaviors. The trial has been conducted between July and September 2012 with more than 70 active users from all over Italy. As previously mentioned the application is able to manage and connect to configured users private WiFi network and without additional configurations to more than 2000 commercial hotspots distributed on the national territory by Guglielmo s.r.l.

BabelTen has been initially tested in a preliminary alpha version with University of Parma researchers and with Guglielmo employees to collect initial feedbacks and verify data collection. After that the application has been made available for free through Google Play [46]. This approach allowed us to collect more than 150.000 distinct logs related to top four Italian mobile operators (TIM, Vodafone, Wind and 3 ITA), 1467 different GSM cells and 137 access points. Logs came from heterogeneous Android handsets from different producers (Samsung, LG and HTC, etc ...) and equipped with commonly distributed OS versions such as 2.2 (8%) 2.3.x (54%), 4.0.x (30%) and 4.1 (8%).

3.4.2 Experimental Results

Considered Metrics

As mentioned in Section 3.4.1, the goal of our trial is to analyze the behavior of BabelTen in a realistic environment. Since the performance of the VHO algorithm has been already analyzed in Section 3.2, we focus on its impact on mobile data offloading.

From one hand, we characterize the average behavior of the users of BabelTen by measuring the amount of downlink (and uplink) traffic generated by each smartphone

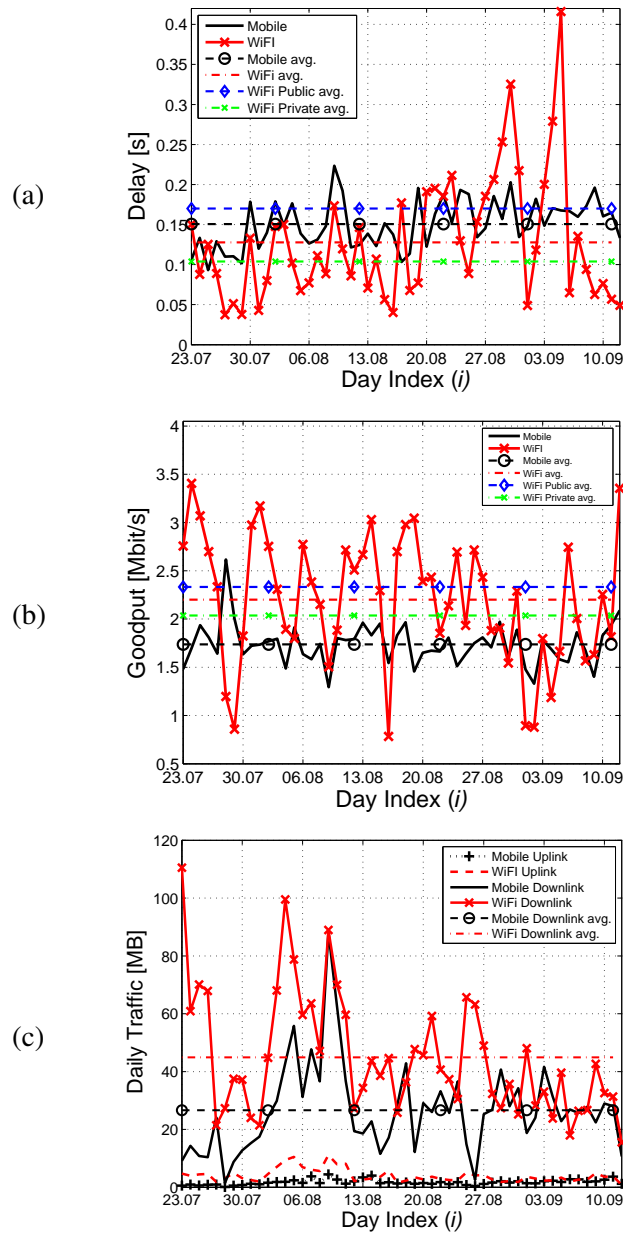


Figure 3.19: (a) Estimated delay, as a function of the Day Index; (b) Estimated goodput, as a function of the Day Index; (c) Amount of user transmitted and received data, as a function of the Day Index.

in every single day of the trial. In particular, we denote as $T_u^{\text{dl}}(i)$ (dimension: [Mbit/s]) and $T_u^{\text{ul}}(i)$ (dimension: [Mbit/s]), the downlink and uplink traffic generated in the i -th day by the user u . By defining \mathbb{U} as the set of all the test users, the average downlink and uplink traffic in the i -day can be expressed as:

$$T^{\text{dl}}(i) = \frac{1}{|\mathbb{U}|} \sum_{u \in \mathbb{U}} T_u^{\text{dl}}(i) \quad T^{\text{ul}}(i) = \frac{1}{|\mathbb{U}|} \sum_{u \in \mathbb{U}} T_u^{\text{ul}}(i).$$

From the other hand, we assess the quality of the network connectivity experienced by the users, by considering two metrics: (i) the goodput; (ii) the end-to-end delay. The former is defined as the effective throughput at HTTP level and it is estimated by dividing the dimension of a fixed-size file by the time required to download it from an Internet web server. The latter is defined as the round-trip time towards an Internet server, and it is estimated by sending a suitable ping probe packet.

We denote as $G(i)$ and $D(i)$, respectively, the average goodput and delay on the i -day of the trial. More precisely, $G(i)$ can be derived as follows:

$$G(i) = \frac{1}{\sum_{u \in \mathbb{U}} N_u(i)} \sum_{u \in \mathbb{U}} \sum_{n=1}^{N_u(i)} g_u(i, n) \quad (3.1)$$

where $N_u(i)$ denotes the number of samples collected by the user u in the i -th day, $g_u(i, n)$ is the n -th goodput estimate carried out by the user u in the i -th day, while $d_u(i, n)$ is the n -th delay estimate. $D(i)$ can be obtained by replacing $g_u(i, n)$ with $d_u(i, n)$ in equation (3.1), where $d_u(i, n)$ is the n -th delay estimate.

Analysis of the Experimental Results

In Fig. 3.19(a) and Fig. 3.19(b), $D(i)$ and $G(i)$, respectively, are shown as functions of the day index, estimated for both WiFi and 3G connections. From these figures, some key properties of the considered networks clearly emerge: (i) WiFi networks generally offer better performance than 3G networks, especially in terms of goodput; (ii) the performance exhibited by 3G networks has a smaller variance than WiFi networks. This makes it easier to predict the network conditions on 3G networks, thus reducing the need for frequent goodput estimations.

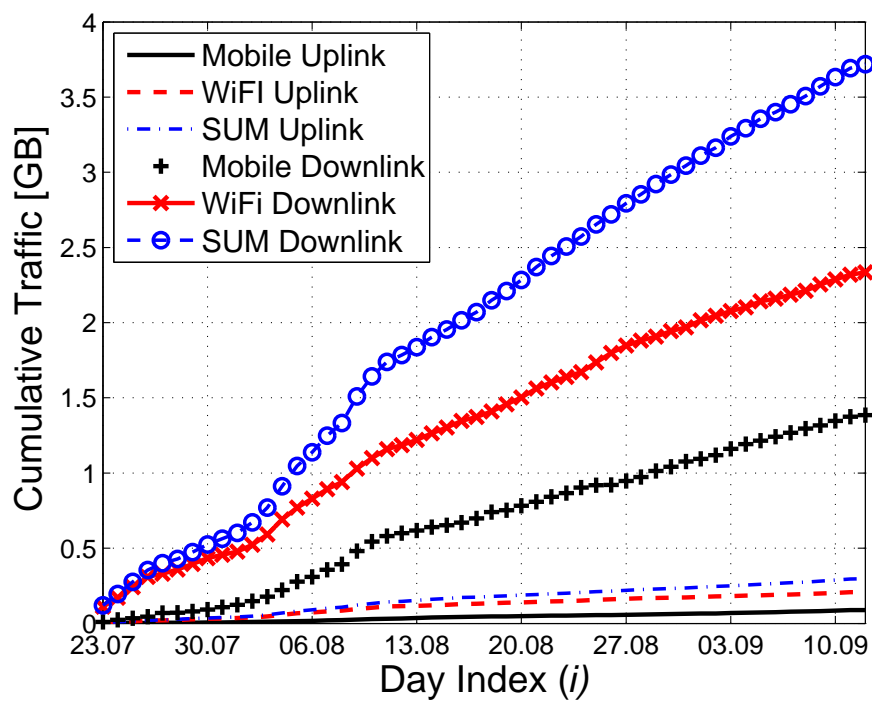


Figure 3.20: Cumulative transmitted and received data, as a function of the Day Index.

Table 3.3: Average values of goodput, delay, downlink and uplink traffic, in WiFi and 3G networks, respectively.

	G [Mbit/s]	D [s]	T^{dl} [MB]	T^{ul} [MB]
WiFi	2.20	0.13	44.89	4.00
WiFi (Public)	2.33	0.38	46.8	4.1
WiFi (Private)	2.03	0.10	44.4	3.8
3G	1.74	0.15	26.64	1.72

Similar considerations can be carried out from the results in Table I, where, for the sake of clarity, we have summarized the average values obtained in our experiments.

In particular, from Table I one can observe that both WiFi and 3G networks exhibit satisfying average goodput performance (higher than 1.5 Mbit/s), but not very satisfying delay performance (in the order of 100 ms). Finally, from Table I it emerges that WiFi networks exhibit a significant advantage in terms of generated traffic with respect to the 3G networks. This means that, despite of the wider coverage of 3G networks, WiFi hot spots offer (on average) better performance than 3G networks and, for this reason they are frequently selected by the VHO algorithm.

From Fig. 3.19(c), it emerges that there is a great variance and in many days it happens that the traffic is symmetrically split among the two networks. Finally, in Fig. 3.20 the cumulative amount of traffic generated by the test users (i.e., obtained by integrating the results in Fig.3.19(c)) is shown.

These results shed light on the "habits" of the test users equipped with Android smartphones. In particular, it emerges that WiFi networks bear two thirds of the monthly generated traffic, while roughly one third is routed in 3G networks. The overall generated monthly traffic is less than 2 GB. These data are in agreement with recent market analysis [47].

Last results are related to the analysis of mobile networks. In particular the graphs in Fig.3.21 illustrate the distribution of delay and goodput for UMTS and HSDPA networks. Data trends show that HSDPA has a smaller delay respect to UMTS, but

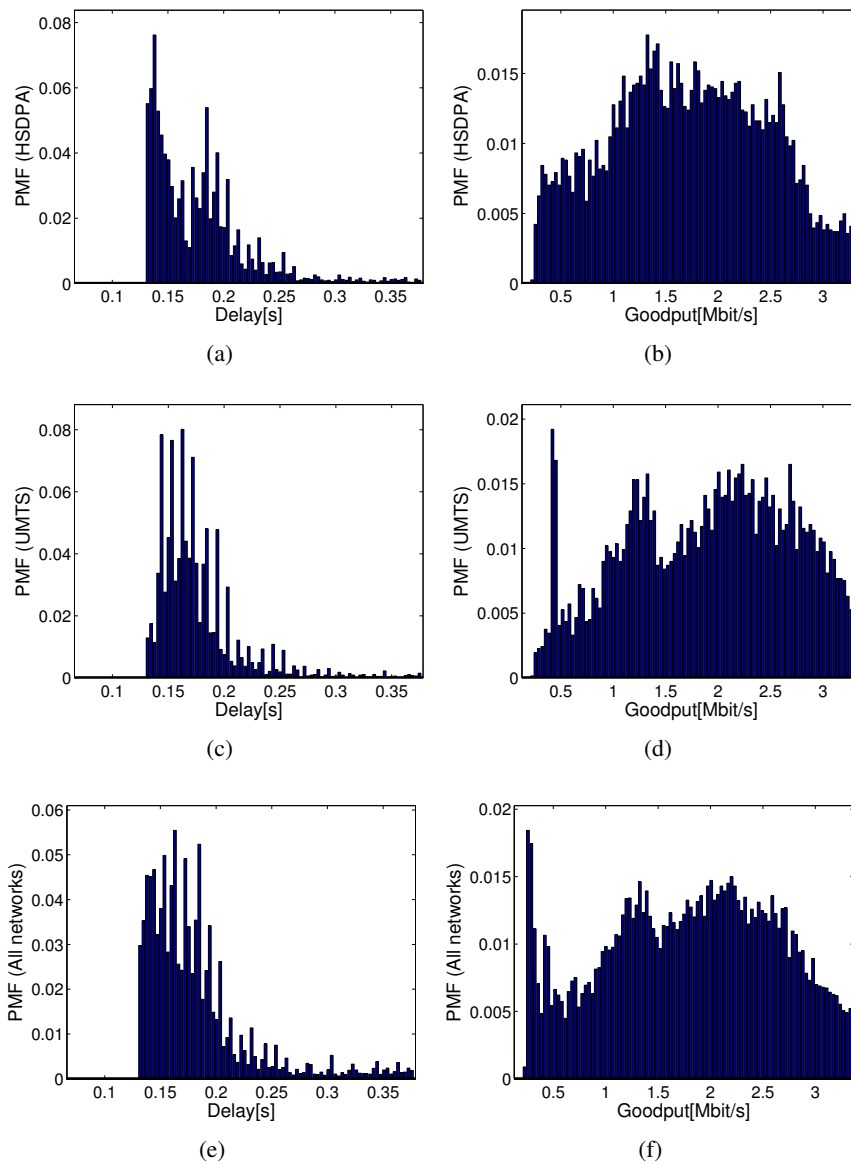


Figure 3.21: Mobile Network Analysis: (a) HSDPA - Delay PMF; (b) HSDPA - Goodput PMF; (c) UMTS - Delay PMF; (d) UMTS - Goodput PMF; (e) All mobile networks - Delay PMF; (f) All mobile networks - Goodput PMF.

that the distributions of goodput are comparable without a significant tendency in favor of HSDPA.

3.5 Discussion on the VHO in HetNets

On the basis of the experimental and simulation results presented and discussed in Section 3.2.2 and Section 3.3, respectively, the following reflections on the role of VHO in HetNets can be carried out.

3.5.1 Role of the (Internal) Decision Algorithm

From the obtained experimental results, it turns out that the VHO decision algorithm is relevant to mitigate the ping-pong effect. However, the RSSI/goodput-based decision algorithms guarantee a very good performance, provided that the received QoS indicator (either the RSSI or a running estimate of the available goodput) is properly filtered, in order to avoid sudden (and temporary) handover decisions.

3.5.2 Role of the Authentication Procedures

The authentication procedure seems to be the real bottleneck of the VHO procedure, especially in a no-coupling scenario, where the MT needs to switch between different operators. In this case, the only solution is a radical simplification of this procedure, possibly demanding some security mechanism to lower levels of the protocol stack. On the other hand, in the case of tight coupling, where the two networks involved in the VHO are operated by the same provider, it might be possible to significantly simplify the authentication procedure, thus reducing the VHO time.

3.5.3 Impact of VHO on HetNet Coverage

The impact of the VHO on network coverage depends on the networks between which it is carried out. More precisely, in the case of two heterogeneous networks with radically different coverage ranges (e.g., UMTS and WiFi), the VHO procedure has basically no impact on the coverage extension, but it is mainly expedient to select,

among two simultaneously available networks, the one which guarantees the highest QoS—according to the chosen performance indicator. On the other hand, VHO might play a key role to efficiently extend the coverage when carried out between networks with more similar coverage ranges, e.g., UMTS and WiMAX networks. The same experimental investigation described in this chapter can be applied to different pairs of networks as well, provided that proper connection procedures and VHO algorithms are implemented in the SC.

3.5.4 Impact of VHO on HetNet Capacity

The impact of VHO on the capacity of a HetNet is not easy to evaluate. In fact, an efficient VHO mechanism allows each MT to connect to the best currently available network. In particular, if the chosen QoS indicator is the available goodput, the selection, by each node, of the network which guarantees the highest goodput implies, from a single-user perspective, maximization of the capacity. Roughly speaking, Figure 3.8 shows that the choice of the network with the best available goodput allows to double the bandwidth experienced by a single MT. In general, VHO will likely be a key ingredient to perform efficient cellular offloading [13] and will thus have a crucial role in 4G systems [27].

However, as known in the realm of game theory, the maximization of each user's utility does not necessarily imply the maximization of the entire HetNet utility [48]. In fact, there may be a large number of MTs which would like to connect to the network with the best goodput (e.g., the WiFi network), thus leading to a violation of the minimum QoS on this network and, therefore, to a congestion. This opens several interesting research perspective, as it is expected that a centralized control of a HetNet will really allow to exploit its potential. On the other hand, efficient decentralized control strategies will likely play a key role in future systems.

3.6 Conclusions

In this chapter, we have proposed two simplified novel VHO algorithms (the first relying on RSSI measures and the second on RSSI and goodput measures) and ana-

lyzed their performance. The simulation-based investigation has shown that the use of VHO has the potential to perform cellular offloading, thus increasing network capacity.

The experimental results carried out using a Windows operating system have shown that the VHO procedure in loosely-coupled heterogeneous networks experiences a long handover time, mostly due to the latency induced by the AAA procedures currently under use in IEEE 802.11 and 3G networks. This problem is exacerbated in the handover from UMTS to WiFi networks. Therefore, the design of effective VHO mechanisms requires to consider a top-down interaction from the high layers of the protocol stack to the bottom layers of the same. The proposed VHO algorithms, nevertheless, are effective in the presence of pedestrian mobility, e.g., when a user walks keeping his/her notebook/tablet in his/her hands. Although the goodput is difficult to accurately estimate in real environments, the second proposed “hybrid” VHO algorithm leverages on infrequent periodic goodput estimation to guarantee ABC conditions to the user.

Finally, we have presented an experimental analysis of a long term trial of our VHO-enabled Android mobile denoted as BabelTen. The aim of the experimentation was to collect data to evaluate the application’s performance and offloading behavior during a long-term real experiment, characterized by heterogeneity in terms of devices, networks, and mobility behaviors. BabelTen allowed the users to efficiently manage mobile and WiFi connectivity (Public and Private) and be ABC, despite the mobile platform (Android) limitations in the network interfaces management. Collected data and evaluated results have shown that WiFi exhibits better performance than average cellular networks (running from 2G to 3.5G) and that two thirds of the traffic load is carried by WiFi networks.

Chapter 4

A Vehicular HetNet Platform Implementation

4.1 Introduction

As stated in Chapter 1, the development of a Vehicular Ad-hoc Network (VANET) is a challenging task, due to the high speed of the vehicles and to the constantly changing network topology. In Section 1.4.1 the IEEE 802.11p standard is described; this technology allows to establish VANETs with low delay and fast association and thus to implement vehicular safety communications applications (e.g. crash prevention with automatic brake activation). The X-NETAD project, presented in Section 1.4.2, integrates a cellular connectivity with an ad-hoc network interface. In X-NETAD the exchange of messages is performed by an application executed by smartphone. The goal of the work presented in this chapter is to develop a system that extends the X-NETAD concept, adding capabilities and functionalities. This new system wants to exploit the information provided by the vehicle through the CAN bus [49] interface, in order to give to the users (i.e. vehicle passengers) information about the status of the vehicle. Furthermore, the information provided by the vehicle can be exchanged with other vehicles in the surrounding area. This capability opens new application scenarios, e.g. cooperative traffic-flow management, cooperative safety applications.

However, the goal of this project is not to implement final applications or to decide what information has to be delivered to the users (other vehicles or passengers), instead, the aim is to design and implement a platform, able to provide services and information to the users, independently from the content of the information and from the type of provided service. These services could be Internet connectivity (i.e. the platform acts as Internet access point), information about the vehicle, or any kind of data provided by a secondary module connected to the main platform.

This chapter describes the WiCar project, a service-providing vehicular platform, in Section 4.2 are presented the designed architecture and protocol, while the details about the implementation of the platform and of the software applications are illustrated in Section 4.3.

The WiCar project is a collaboration between the University of Parma, Guglielmo S.r.l [40], DQuid S.r.l.u [50], and Henesis S.r.l [51].

4.2 WiCar System Design

The system we designed has to be connected to the vehicle through a cabled CAN bus interface, therefore this system cannot be smartphone-based like the X-NETAD system. We decided, instead, to compose our system of a main module connected to the vehicle, the clients of the system can be the vehicle passengers with their smartphones or other main modules installed on board of other vehicles. Additionally, other secondary hardware modules can be installed in the vehicle and be connected to the main board. The secondary modules can provide further features to the system; these features will be forwarded to the clients by the main module. For example, a use case could be a secondary module sending a video streaming to the main board (e.g. video captured by a camera placed in the rear of the vehicle). The passengers could receive and display the video on their smartphones, by connecting their smartphones to the main board and demanding the specific streaming service. With this architecture we can keep the technical specifications and, therefore, the costs of the main module low. The design of the architecture and of the protocol has been developed in collaboration with Guglielmo S.r.l. The system depicted in Figure 4.1 entails the management

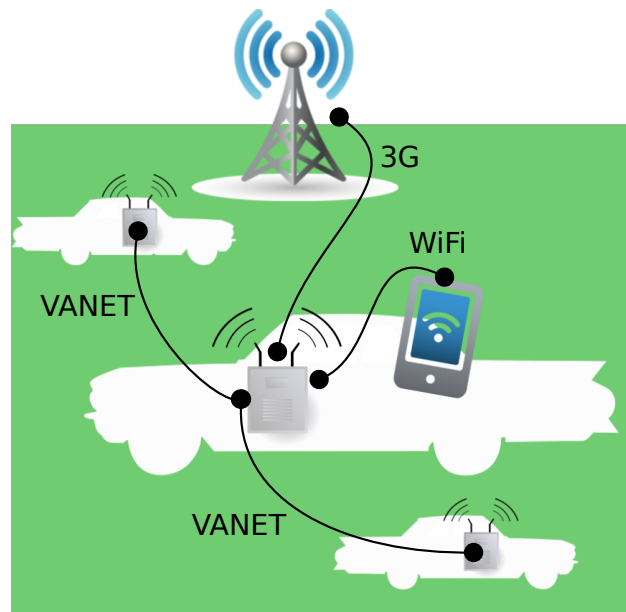


Figure 4.1: A vehicular scenario with three different wireless networks.

of three heterogeneous wireless networks. An efficient and modular management of the networks and of the resources they can provide is one of the goals of the WiCar project.

4.2.1 High-level Architecture

In Figure 4.2 is illustrated the high-level architecture of the WiCar system. One can see that there is a main mobile device, this device could be the administrator of the system, with extra services compared with other mobile devices. The property that distinguish the main device is that it can connect to the main module via USB cable or Bluetooth technology. The secondary modules, called in the Figure *Sender Module*, can also connect to the main module via USB or Bluetooth. It is important to notice that the WiCar project doesn't define how the sender modules communicate with

the main board, the communication with these modules is performed through their *Brokers*. The Brokers are software applications running on the main module, they communicate with the *Engine* using a set of messages defined by the WiCar project. The *Engine* is the main software application, running on the main module as well, its task is to handle all requests coming from the clients and to forward to the clients all information coming from the Brokers.

A key element of the WiCar system are the services. The services are essentially the purpose of the whole system: the Brokers, possibly using a sender module, publish the services they can provide and the clients subscribe to the services they want to enjoy. At the start-up of the system, the Brokers, after checking whether their sender module (if they need it) is running and connected, tell to the Engine all the services they want to publish. When a client, from now on called *Consumer*, wants to use a service, logs on the Engine and asks for a list of available services. Then the Consumer tells the Engine all the services it wants to subscribe to. After that, the Engine acts as an intermediary between the Brokers and the Consumers: if the Engine receives from a Broker some data, related to a service, it forwards that data to all Consumers subscribed to that specific service. There are four types of service:

- Request-Response
- Notify
- Stream
- Direct Stream

The first one represent a simple request-response conversation. The Consumer asks for a specific information (e.g. outdoor temperature) and the Broker send the related response to the Consumer. The notify service represents a sequence of information sent every time a specific event occurs (e.g. vehicle speed beyond a certain threshold). Alternatively a notify could be a simple information, such as vehicle speed, sent periodically at a specific interval of time. The stream and direct stream service are very similar services. They both sent a stream to the Consumer, such as a video or audio stream. The stream service sends the data stream to the Engine, while the

direct stream sends directly to the Consumer. In the first case the same stream could be shared by multiple Consumers, if more than one Consumer is subscribed to the same stream service.

4.2.2 WiCar Protocol

The communication between entities (Engine, Broker, and Consumer) in WiCar system is performed by means of messages. The base structure of these messages is defined in Table 4.1. The *sourceID* field identifies the sender of the message, the *transactionID* field identifies the transaction. Since every message sent should receive a response message, the same value must be used for the *transactionID* field of both the request and the answer messages. The *messageType* field specifies the type of the message, and therefore the content of the *payload* field. In Table 4.2 are explained the meanings of the types of message. The payload of the base structure can be a *Registration Message* or a *Service Descriptor Message*. The former is used only during the registration procedure, that is with REG or UNREG message types, the latter is used with all other type of messages (excluding PING messages). In Table 4.3 it is described the structure of the Registration Message. This type of message is used both by Brokers and by Consumers. Indeed both Brokers and Consumers have to register to the Engine before they are allowed to communicate with other entities. This procedure is necessary in order to prevent unauthorized Brokers from publishing unsafe services and unauthorized Consumers from accessing protected resources. The Engine, by accessing the WiCar server through the Internet connection, can retrieve the database of the authorized entities. The WiCar server is also responsible for the definition of the content provided by services: every service has an ID and this ID is mapped univocally with a type of content. This way, when a Consumer is searching for a specific content, all services that can provide it are found easily by the Engine by means of the ID mapped with that specific content.

In Table 4.4 the fields of the Service Descriptor Message are listed. The *serviceID* field is the identifier of the content of the service. The *serviceName* is a mnemonic name for the service, this name may or may not recall the content of the service. The field *brokerID* specifies the Broker that provides a service. The field *serviceParams* is

Serial Bus = Bluetooth (RFCOMM) || USB || IEEE 802.15.4

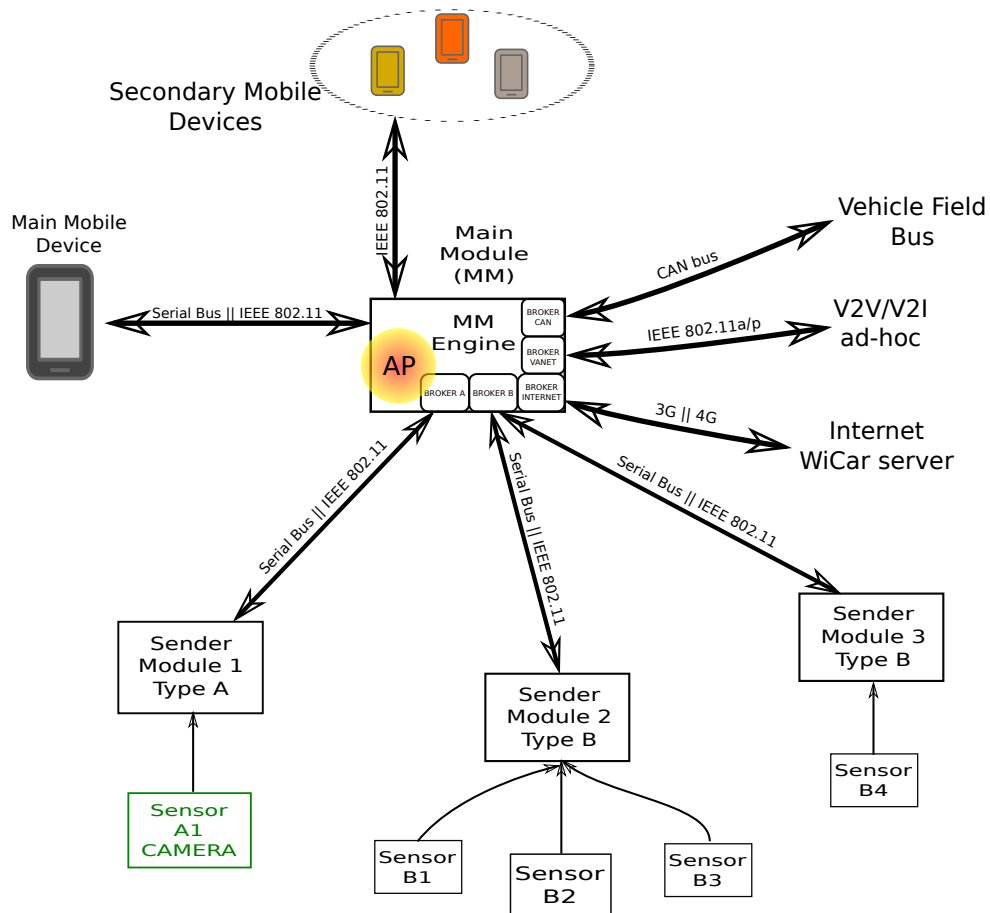


Figure 4.2: High-level representation of the WiCar system.

filled by the Consumer, and it contains the parameters for the service requested by the Consumer. This field is filled only when the Consumer is subscribing to a service. On the contrary, the field *returnData* is filled only by a Broker when it returns to the Engine the requested content, or when it has to notify an event. The field *type* specifies the type of service described by the message. The *streamSinkIP* and *streamSinkPort* fields define the url of the receiver of a stream service. These two fields are set by the Consumer only if the type of the service is STREAM or DIRSTREAM. The Broker of a stream service will be send the data stream to this url. Notice that this url is different from the url specified in the registration message. The latter is used for every control message: it is the main url to use to contact an entity. The former is just the url of the handler of a data stream.

In Figure 4.3 it is depicted the flow of messages exchanged by the entities during the start-up stage. At first the Consumers and the Brokers have to register to the Engine. Thereafter the Brokers publish their services. Some services, in order to be provided, could need some information from other services. In Figure 4.3 the case A and the case C represent this situation. In these cases the Brokers, before they publish their services, ask the Engine for a list of available services (discovery request). If they find the type of service (i.e. type of content provided by the service) they need, they subscribe to it. If the content they need is not available, they wait some time and then try again. This procedure has been thought for that services that provide some information by processing data provided by other services. For instance, a service could process the data stream coming from a video camera and then notify the occurrence of a specific event. In this case the Broker acts as a Consumer since it subscribes a service provided by another service.

In Figure 4.4 and 4.5 the messages flows of the four different services are described. It is important to notice the employment of the START and STOP message. The START message is sent by the Engine to the Broker to tell him it must start to send messages related to a specific service. The START message contains all parameters requested by the Consumer. The STOP message is used by the Engine to inform the Broker that no Consumer is subscribed to a specific service any more. This way the Broker can stop to send to the Engine the messages related to that service.

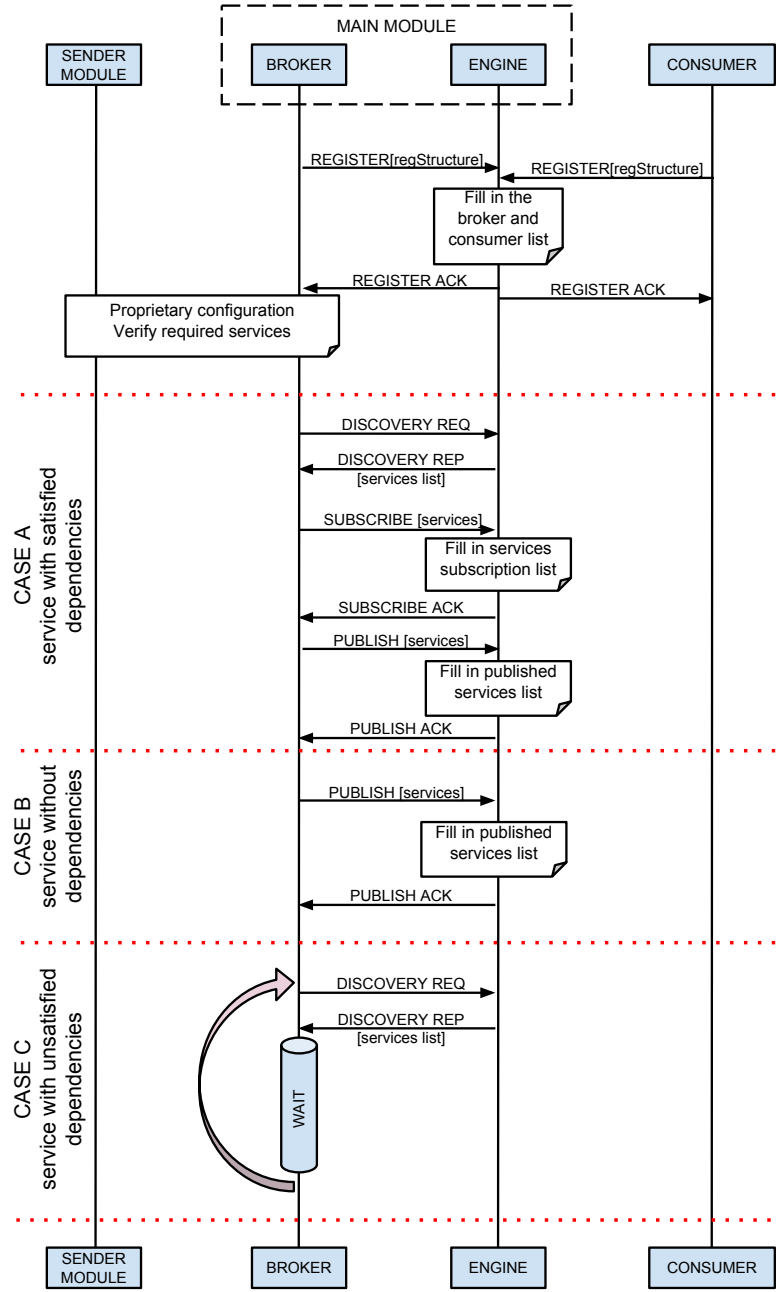


Figure 4.3: Start-up procedure, services publishing stage.

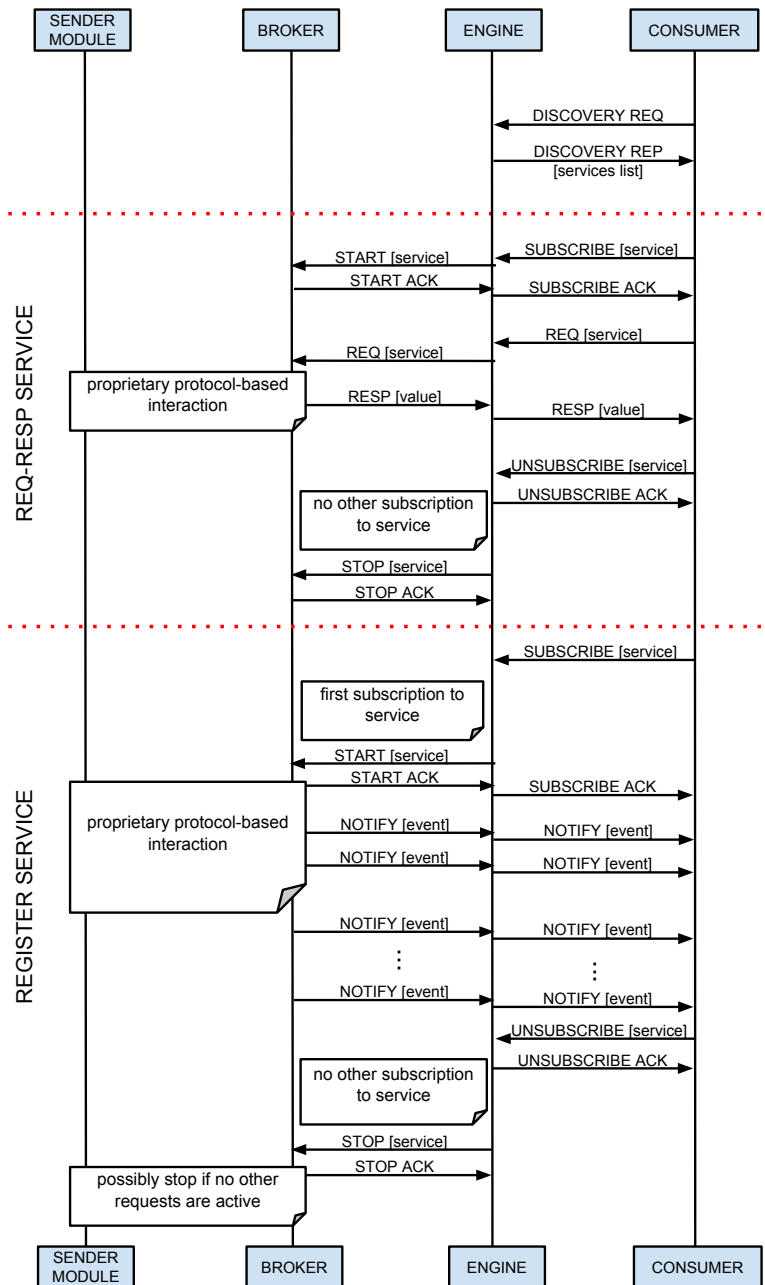


Figure 4.4: Messages flow of request-response service and notify service.

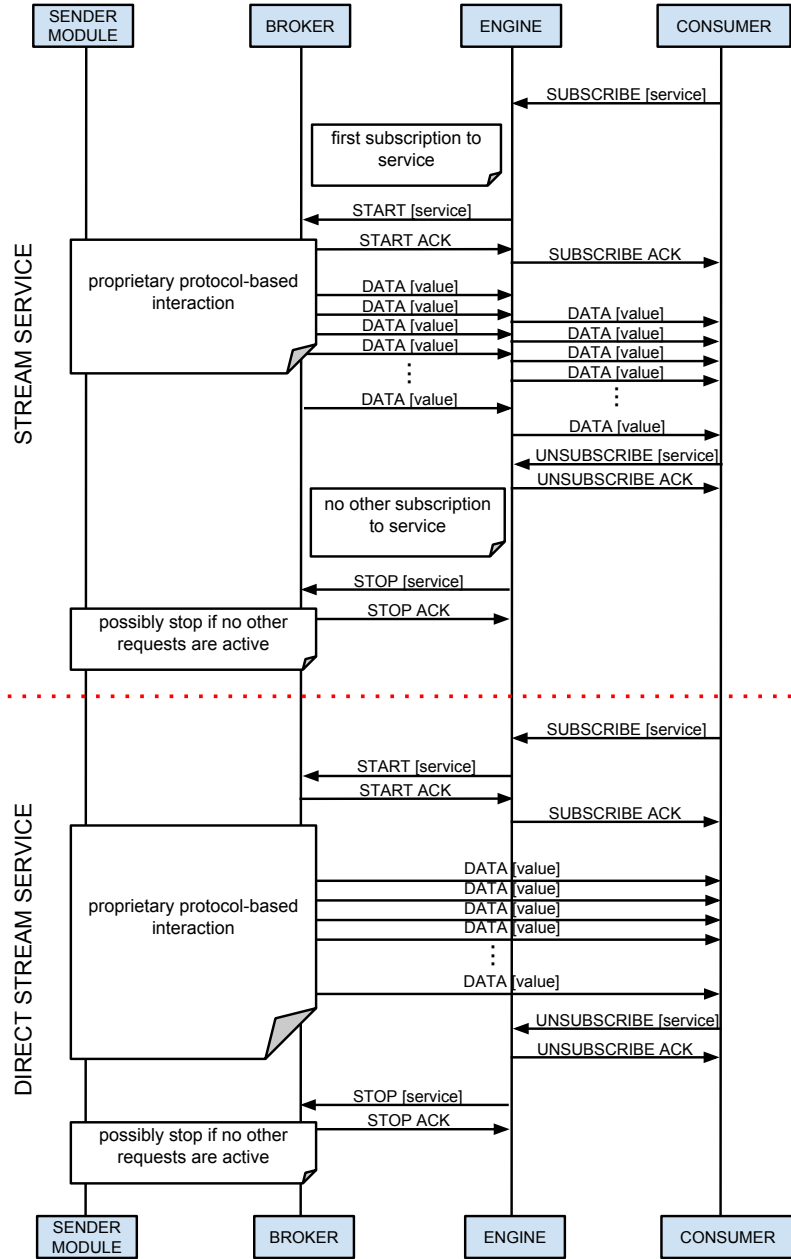


Figure 4.5: Message flow of stream service and direct stream service.

Field Name	Field Type	Field Values
sourceID	String	
timestamp	long	
transactionID	long	
messageType	Enum	ACK, PING, REG, PUBLISH, SUBSCRIBE, UNREG, UNPUBLISH, UNSUBSCRIBE, DISCOVERY, START, STOP, REQ, RESP, NOTIFY, NACK
payload	bite[]	

Table 4.1: Base message structure of the WiCar protocol

4.3 WiCar Implementation

After the design of the architecture and of the protocol of the WiCar system, we started to develop the test bed of this system. For the development of the hardware we relied on the collaboration of DQuid S.r.l.u, one of the partners of the WiCar project. The software for testing WiCar has been developed in collaboration with Henesis S.r.l.

4.3.1 The Hardware Platform

The main module of the WiCar system needs, in addition to the CAN bus interface, a stable connection to the Internet, therefore we decided to equip the module with a cellular network interface. To provide connection between the main module and other devices we thought the best solution is an IEEE 802.11 interface. As the clients connected to the platform can be both vehicles (i.e. vehicles equipped with another main module) and passengers (i.e. their smartphones) we decided to provide our platform

Message type	Meaning of the message
ACK	positive acknowledgment
NACK	negative acknowledgment
PING	ping message, must be answered with an ACK
REG	request to register to the Engine
PUBLISH	request to publish a service
SUBSCRIBE	request to subscribe to a service
UNREG	request to unregister from Engine
UNPUBLISH	request to unpublish a service
UNSUBSCRIBE	request to unsubscribe from a service
DISCOVERY	request the list of available services
START	tell a Broker to start a service (sent by Engine)
STOP	tell a Broker to stop a service (sent by Engine)
REQ	specific request of a request-response service
RESP	specific response of a request-response service
NOTIFY	event notify service (sent to Consumer)

Table 4.2: Types of message of the WiCar protocol and their meanings

Field Name	Field Type	Description
version	String	the version of the Broker
url	String	the url of the entity in the format <ip addr>:<port>
id	String	the global unique identifier of this entity
hash	String	cryptographic function of secret and nonce (for authentication)

Table 4.3: WiCar protocol: Registration message structure

Field Name	Field Type	Description
serviceName	String	name of the service
brokerID	String	global unique identifier of the Broker
serviceID	String	identifier of the content of the service
type	Enum: REQRESP, NOTIFY, STREAM, DIRSTREAM	type of service.
serviceParams	Map (key, value)	requested service parameters
returnData	Map (key, value)	returned data (to the Consumer)
streamSinkIP	String	ip address to which to send the stream
streamSinkPort	integer	port to which to send the stream

Table 4.4: WiCar protocol: Service Descriptor message structure

with two IEEE 802.11 interfaces, a traditional infrastructure-mode IEEE 802.11b/g/n interface for the passengers and an IEEE 802.11p interface for the connection between vehicles. However, since the 802.11p interface and its related driver are not easy to retrieve, we decided to adopt an 802.11a interface; this kind of interface operates almost at the same frequency band of 802.11p standard (5 GHz), though it has no vehicular and fast association capabilities. Although this tradeoff prevents the implementation of applications for real vehicular scenarios (i.e. with vehicles moving at high speed), this solution allows to test the interferences between the two interfaces and to test applications in a semi-realistic scenario—i.e. vehicular environment, but mobility reduced due to the long association time of IEEE 802.11a technology. The main module is equipped also with a Bluetooth and a USB interface for the connection with the main mobile device. Other cabled interfaces, such as Ethernet, are needed for the development and debug stages.

Besides the interfaces, the hardware platform on which to run the WiCar system (i.e. the Engine and the Brokers) is composed of a System-On-Module (SOM) produced by Bluegiga, the APx4 [52]. This SOM is equipped with the Freescale ARM9

processor i.MX28 [53] and it has embedded many interfaces or interface controllers. The core operating frequency is 454 MHz and the RAM memory is 128 KB. The SOM is carried by a board that extends the connectors of the interfaces. The IEEE 802.11a interface is directly mounted on the carrier board. In Figure 4.6 is shown the carrier board and the SOM mounted on it.

4.3.2 The Software Implementation

The operative system running on the APx4 SOM is a customized Linux distribution developed with the toolchain Yocto [54]. The operative system is distributed by Bluegiga together with the APx4 SOM. The software application developed for the main module implements the Engine and a general purpose Broker. The software was developed in C++ language. The entities of the WiCar system were developed in hierarchical manner, that is at first was developed a general class called *Entity*. The class Entity implements all communication functions, handles the received messages and, according to the type of the messages, calls the related callbacks. The Engine and Broker classes have just to implement these callbacks, that is a callback for every type of message.

The main tasks of the Engine are:

- to check the credentials of the entities during the registration,
- to maintain a list of the registered entities,
- to handle the publish requests
- to maintain a list of the published services,
- to handle the subscribe requests
- to maintain a list of the subscribed Consumers,
- to send the START and STOP commands to the Brokers,
- to forward the NOTIFY and RESP messages to the Consumers,

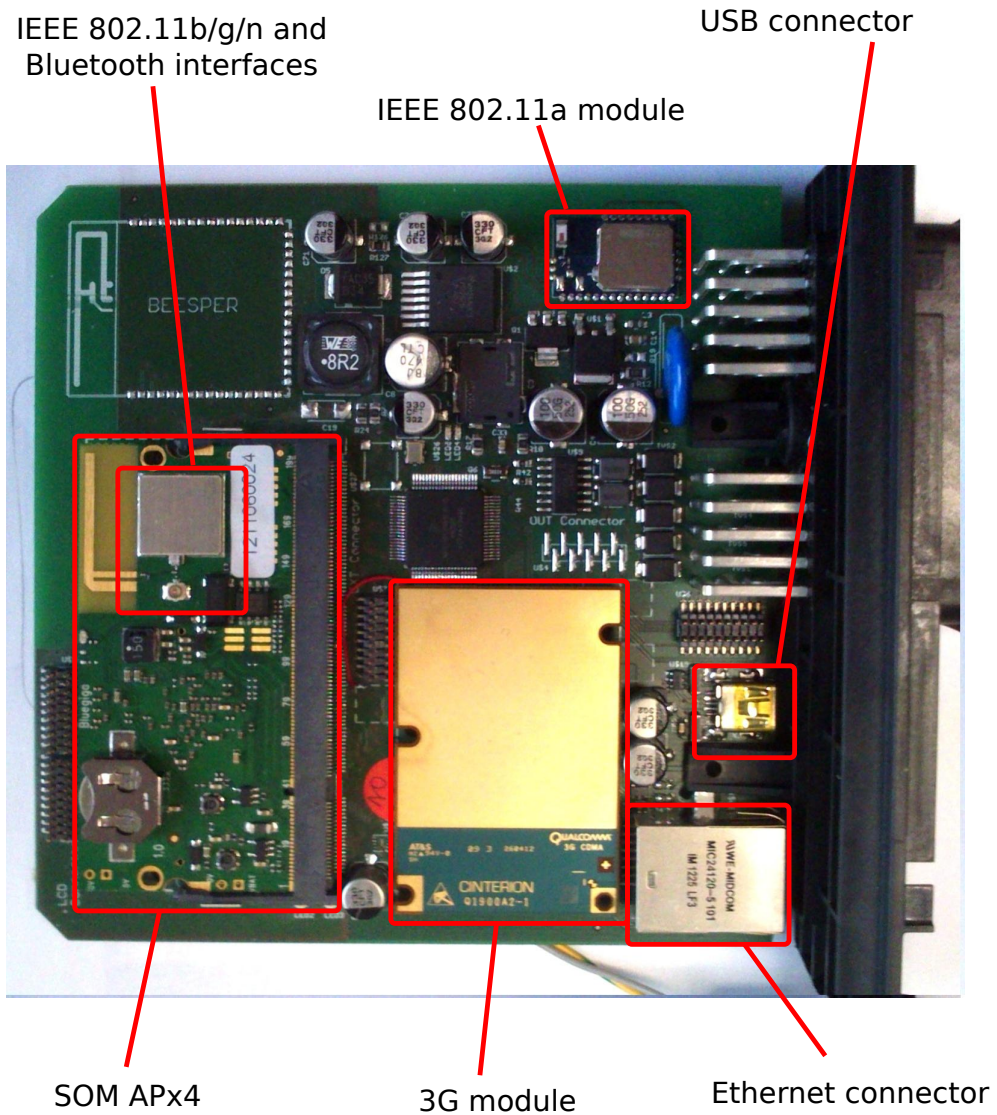


Figure 4.6: The prototype of the main module of the WiCar system.

- to forward the REQ messages to the Brokers,
- to answer to the DISCOVERY requests,
- to send periodically a PING message to all the registered entities,

The tasks of the Broker depend on the service the Broker wants to provide. Nevertheless every Broker must carry out the following tasks:

- register to the Engine,
- send publish request to the Engine,
- answer to the PING message,
- handle START and STOP requests,
- possibly handle REQ message and send RESP message,
- possibly send NOTIFY messages,
- possibly send data stream.

We decided to implement a Consumer entity for Android platform. The native language used by Android is a Java language with additional specific classes. The software application we developed can handle direct stream and notify services. The application connects to the Engine through the IEEE 802.11n interface by using the access point created by the main module (APx4). The tasks carried out by the Android application are the following:

- register to the Engine,
- send DISCOVERY request and display all available and supported services,
- answer the PING requests,
- send subscribe request for the service selected by the user,
- handle and display NOTIFY messages,
- handle and display STREAM service.

4.4 Conclusions

In this chapter we described a platform that enables the integration of heterogeneous networks in a vehicular environment. The work presented in this chapter didn't focus on the applications that such an integration enables in this specific environment, rather the protocol and the architecture of the platform was described in deep. However such a platform could be a test bed for many applications that exploit the integration of a vehicular ad-hoc network, a wireless local area network and a cellular network. In this platform the integration and interaction of networks are performed according to the applications' needs: if an application (or Broker, according to the WiCar terminology) needs to connect to the vehicular network, another application (the VANET Broker) will be asked to provide access to the ad-hoc network. The same way, if a Consumer wants to get Internet connectivity it will ask to subscribe the service provided by the Internet Broker. In this configuration there isn't a centralized management of the networks, rather there is a distributed management of heterogeneous networks.

Conclusions and future directions

In this thesis, we have presented some protocols, technologies, and applications that make the integration and the automatic management of multiple and heterogeneous networks feasible. The main concepts behind the interest on this subject are the Always Best Connected (ABC) concept—leveraging on the best capabilities of different network technologies—and cellular network offloading through WiFi networks. The existing technologies that make the integration of heterogeneous networks possible are the following: the IEEE 802.21 standard; the Mobile IP protocol; the Multipath TCP (not yet standardized) protocol; the Stream Control Transmission Protocol (SCTP); and the Session Initiation Protocol (SIP). In this thesis, we have described the IEEE 802.21 standard and Multipath TCP protocol, since these technologies could provide, together, a powerful stack for the integrated management of various connections. The IEEE 802.21 standard provides functions and information for the network discovery and network selection stages of the vertical handover (VHO) procedure, but it does not provide tools to create session continuity. While Mobile IP could manage the IP address transition and, therefore, guarantee session continuity, we believe that Multipath TCP could be a better solution for the management of multiple interfaces, since it can provide session continuity and can also combine multiple flows without the need to change existing applications and network protocols.

A specific scenario where the integration of heterogeneous networks could be very useful is given by the vehicular scenario. For this reason, we have also considered the IEEE 802.11p technology, an amendment to the IEEE 802.11 standard that allows the creation of vehicular ad-hoc networks with very fast association and very

low delay. In the same context of vehicular networking, leveraging on the results of the X-NETAD project, a smartphone-based system that integrates cellular network and ad-hoc networks for the delivery of road traffic information messages, we have focused on the design of a WiCar multi-interface system.

A large portion of this thesis has been dedicated to the VHO. This procedure is fundamental for the integration of different networks. It can rely on the technologies described previously, although our approach was to implement VHO on common operating systems (Windows, Android) without modifying the systems themselves. We then proposed two novel simple VHO algorithms based on two metrics: the received signal power and the network bandwidth. The performance of these algorithms was first analysed with the Opnet simulator, and then we developed applications for a Windows 7 and Android operating systems. By analysing the performance of these applications, we noticed that despite the simplicity of the proposed VHO algorithms, the applications can provide a constant connectivity. Exploiting the distribution of the Android application through Google Play, we were able to collect a large volume of statistical data about the utilization of WiFi and cellular networks.

Finally, the activities of the WiCar project, (a joint project between our department and three Italian companies) were aimed at integrating cellular, WiFi, and ad-hoc networks into a unique platform. We designed the architecture and the communication protocol of the WiCar system, implementing a software application on a properly chosen hardware platform.

Integrating different technologies, exploiting the best characteristic of every network, and making the selection of the best network automatic is still an open question. In particular, the problem of session continuity must be well investigated. For example, the widespread utilization of Multipath TCP could improve considerably the availability and the quality of the Internet connectivity.

Bibliography

- [1] T. Inzerilli and A. M. Vegni. A reactive vertical handover approach for WiFi-UMTS dual-mode terminals. In *Proc. of IEEE International Symposium on Consumer Electronics (ISCE)*, pages 1–4, Algarve, Portugal, April 2008.
- [2] E. Gustafsson and A. Jonsson. Always best connected. *IEEE Wireless Commun. Mag.*, 10(1):49–55, February 2003.
- [3] 3rd Generation Partnership Project. 3GPP. Website: <http://www.3gpp.org>.
- [4] Institute of Electrical and Electronics Engineers. IEEE Std 802.16TM-2009. Part 16: Air Interface for Broadband Wireless Access Systems, 2009.
- [5] European Telecommunications Standards Institute. HIPERLAN Type 2 Broadband Radio Access Networks (BRAN); System Overview, February 2000.
- [6] Institute of Electrical and Electronics Engineers. IEEE Std 802.11TM-2007. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 2007.
- [7] SIG Bluetooth. Bluetooth Specification. Version 4.0, 2010.
- [8] J. Manner, M. Kojo, T. Suihko, P. Eardley, and D. Wisely. IETF RFC 3753, Mobility related terminology, 2004.

-
- [9] J.G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M.C. Reed. Femtocells: Past, present, and future. *IEEE J. Select. Areas Commun.*, 30(3):497–508, April 2012.
- [10] V. Chandrasekhar, J. Andrews, and A. Gatherer. Femtocell networks: a survey. *IEEE Commun. Mag.*, 46(9):59–67, September 2008.
- [11] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T.A. Thomas, J.G. Andrews, P. Xia, H.S. Jo, H.S. Dhillon, and T.D. Novlan. Heterogeneous cellular networks: From theory to practice. *IEEE Commun. Mag.*, 50(6):54–64, June 2012.
- [12] Kyunghan Lee, Injong Rhee, Joohyun Lee, Yung Yi, and Song Chong. Mobile data offloading: how much can WiFi deliver? *SIGCOMM Comput. Commun. Rev.*, 40(4):425–426, October 2010.
- [13] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. Augmenting mobile 3G using WiFi. In *Proc. of the Int. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, pages 209–222, San Francisco, CA, USA, June 2010.
- [14] Damon Wischik, Mark Handley, and Marcelo Bagnulo Braun. The resource pooling principle. *SIGCOMM Comput. Commun. Rev.*, 38(5):47–52, September 2008. URL: <http://doi.acm.org/10.1145/1452335.1452342>, doi:10.1145/1452335.1452342.
- [15] J. Postel. IETF RFC 793, Transmission Control Protocol (TCP), September 1981.
- [16] C. Hopps. IETF RFC 2992, Analysis of an Equal-Cost Multi-Path Algorithm, November 2000.
- [17] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. IETF RFC 6824, TCP Extensions for Multipath Operation with Multiple Addresses, December 2013.

- [18] R. Stewart. IETF RFC 4960, Stream Control Transmission Protocol, September 2007.
- [19] M. Scharf and A. Ford. IETF RFC 6897, Multipath TCP (MPTCP) Application Interface Considerations, March 2013.
- [20] A. De La Oliva, A. Banchs, I. Soto, T. Melia, and A. Vidal. An overview of iee 802.21: media-independent handover services. *Wireless Communications, IEEE*, 15(4):96–103, 2008. doi:10.1109/MWC.2008.4599227.
- [21] D. Johnson, C. Perkins, and J. Arkko. IETF RFC 3775, Mobility Support in IPv6, 2004.
- [22] D. Jiang and L. Delgrossi. Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040, 2008. doi:10.1109/VETECS.2008.458.
- [23] Eureka project 6252 X-NETAD. Website: <http://www.eurekanetwork.org/project/-/id/6252>.
- [24] Cellint TrafficSense. Website: http://cellint.com/traffic_data/traffic_system.html.
- [25] S. Panichpapiboon and G. Ferrari. Irresponsible forwarding. In *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*, pages 311–316, Phuket, Thailand, 2008. doi:10.1109/ITST.2008.4740277.
- [26] G. Lampropoulos, N. Passas, L. Merakos, and A. Kaloxylos. Handover management architectures in integrated WLAN/cellular networks. *IEEE Communications Surveys & Tutorials*, 7(4):30–44, October 2005.
- [27] X. Yan, Y. A. Sekercioglu, and S. Narayanan. A survey of vertical handover decision algorithms in 4G heterogeneous wireless networks. *Elsevier Computer Networks*, 54(11):1848–1863, August 2010.

- [28] Insitute of Electrical and Electronics Engineers. IEEE Std 802.12TM-2008. Part 21: Media Independent Handover Services, 2008.
- [29] Internet Engineering Task Force. Website: <http://www.ietf.org/>.
- [30] S. Sharma, I. Baek, and T. Chiueh. OmniCon: A mobile IP-based vertical hand-off system for wireless LAN and GPRS links. *Software: Practice and Experience*, 37(7):779–798, 2007.
- [31] C. Perkins. IETF RFC 3344, IP Mobility Support for IPv4, 2002.
- [32] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. IETF RFC 3261, SIP: session initiation protocol, 2002.
- [33] S. Salsano, A. Polidoro, C. Mingardi, S. Niccolini, and L. Veltri. SIP-based mobility management in next generation networks. *IEEE Wireless Communications*, 15(2):92–99, April 2008.
- [34] J.-Y. Song, S.-W. Lee, and D.-H. Cho. Hybrid coupling scheme for UMTS and wireless LAN interworking. In *Proc. IEEE Vehicular Tech. Conf. (VTC-Fall)*, volume 4, pages 2247–2251, Orlando, FL, USA, October 2003. doi: 10.1109/VETECONF.2003.1285929.
- [35] M. Wasserman and P. Seite. IETF Draft (Work in progress) - Current Practices for Multiple Interface Hosts, July 2011. Available at: <http://tools.ietf.org/html/draft-ietf-mif-current-practices-12>.
- [36] H. K. Lee, V. Hall, K. H. Yum, K. I. Kim, and E. J. Kim. Bandwidth estimation in wireless LANs for multimedia streaming services. *Hindawi Journal on Advances in Multimedia*, 2007(1), January 2007.
- [37] M. Li, M. Claypool, and R. Kinicki. Wbest: a bandwidth estimation tool for IEEE 802.11 wireless networks. In *IEEE Conference on Local Computer Networks (LCN)*, pages 374–381, Montreal, Canada, October 2008.

-
- [38] Matlab. Website: <http://www.mathworks.com>.
- [39] N.D. Tripathi, J.H. Reed, and H.F. VanLandinoham. Handoff in cellular systems. *IEEE Personal Communication*, 5(6):26–37, December 1998.
- [40] Guglielmo S.r.l. Website: www.guglielmo.biz.
- [41] B. Anton, B. Bullock, and J. Short. Best current practices for wireless internet service provider (WISP) roaming. Technical report, Wi-Fi Alliance, February 2003.
- [42] Remote Access Service (RAS) Windows API. Available: [http://msdn.microsoft.com/en-us/library/bb545687\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb545687(VS.85).aspx).
- [43] R. Braden. IETF RFC 1122, Requirements for Internet Hosts - Communication Layers, 1989.
- [44] Opnet Website. Website: <http://www.opnet.com>.
- [45] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski and Hank Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.
- [46] BabelTen on Google Play. URL: <https://play.google.com/store/apps/details?id=com.guglielmo.babelten> [cited 2012].
- [47] Informa Telecoms & Media. Understanding today’s smartphone user: Demystifying data usage trends on cellular & wi-fi networks [online]. August 2012. URL: http://www.informatandm.com/wp-content/uploads/2012/02/Mobidia_final.pdf.
- [48] T. Jie, A.G. Klein, and D.R. Brown. Natural cooperation in wireless networks. *IEEE Signal Processing Mag.*, 26(5):98–106, September 2009.
- [49] M. Farsi, K. Ratcliff, and M. Barbosa. An overview of controller area network. *Computing Control Engineering Journal*, 10(3):113–120, 1999. doi: 10.1049/cce:19990304.

- [50] DQuid S.r.l.u. Website: www.dquid.com.
- [51] Henesis S.r.l. Website: www.henesis.eu.
- [52] Bluegiga - APx4. Website: hwww.bluegiga.com/en-US/products/wireless-systemonmodule/apx4-wireless-systemonmodule.
- [53] Freescale - i.MX28 Processors. Website: www.freescale.com/webapp/sps/site/taxonomy.jsp?code=IMX28_FAMILY.
- [54] Yocto Project. Website: www.yoctoproject.org.

Ringraziamenti

Un ringraziamento speciale va fatto innanzitutto al mio tutor Prof. Gianluigi Ferrari, soprattutto per avermi convinto, anche nei momenti più bui di questi anni, a continuare e a portare fino in fondo il dottorato. Grazie anche a Nicola e Giovanni Guerri per il sostegno tecnico. Un ringraziamento a tutti i colleghi della palazzina, in particolare Marco Picone e Simone, tuttologi sempre pronti ad aiutare e a dare consigli preziosi, grazie a Matteo, compagno dei pranzi. Grazie ai colleghi del nuovo WASN: Andrea, Stefania, Laura e Asim (quest'ultimo per i preziosi solleciti), a quelli del vecchio WASNLAB: Vito (VAG), Busa, Marco Martalò e Giubbo, da loro penso di aver imparato quasi tutto quello che un PhD deve sapere. Grazie agli altri colleghi che non ho mai capito se fanno parte del WASNLAB o no. Tipo Guy, Carlo, Goniz, Luca, compagni di pizzate. Grazie a Demi e a Picio per la pazienza nell'aspettarmi la mattina e per le risate. Grazie ai caffè dello zio Aurelio.

Ai miei genitori va un ringraziamento speciale, senza di loro non mi sarei mai potuto permettere questo lusso. Grazie anche ai miei fratelli, dal primo all'ultimo, loro forse non lo sanno, ma sono per me un costante punto di riferimento.

Grazie a Richard Raffaele che mi ha sempre aiutato a svegliarmi presto perché papa Arbeit.

E poi grazie a te Chri, che hai accettato e supportato questo mio capriccio.