Exploiting Network Diversity

German Castignani

N° d'ordre : 2012telb0250

Sous le sceau de l'Université européenne de Bretagne

Télécom Bretagne

En habilitation conjointe avec l'Université de Rennes 1

Ecole Doctorale – MATISSE

Exploitation de la Diversité des Réseaux

(Exploiting Network Diversity)

Thèse de Doctorat

Mention : « Informatique »

Présentée par **German Castignani**

Département : Réseaux, Sécurité et Multimédia (RSM)

Laboratoire : IRISA

Directeur de thèse : Xavier Lagrange

Soutenue le 7 novembre 2012

Jury :

M. Cesar Viho	- Professeur, Université de Rennes 1 (Président)
M. Giuseppe Bianchi	- Professeur, Université de Rome Tor Vergata (Rapporteur)
M. Christian Bonnet	- Professeur, Eurecom (Rapporteur)
M. Raphaël Frank	- Associé de Recherche, Université du Luxembourg (Examinateur)
M. Sébastien Houcke	- Maître de Conférences, Télécom Bretagne (Examinateur)
M. Nicolas Montavont	- Maître de Conférences, Télécom Bretagne (Examinateur)
M. Xavier Lagrange	- Professeur, Télécom Bretagne (Directeur de thèse)

To my parents, Mara and Antonio, for absolutely everything.
To Ana, who is always close to me.

ACKNOWLEDGMENTS

I am truly indebted and thankful to my supervisor, Dr. Nicolas Montavont, who has trusted me since the beginning and gave me the possibility to discover research.

I would like to thank also my thesis director, Professor Xavier Lagrange and the responsible of the RSM department, Professor Jean-Marie Bonnin, for their support during my thesis.

I wish to thank Professor Cesar Viho, Professor Giuseppe Bianchi, Professor Christian Bonnet, Dr. Raphael Frank, and Dr. Sébastien Houcke for accepting to be part of this Ph.D jury.

I would like also to address a special thank to the students and interns I have co-supervised during my thesis, Alejandro Lampropulos, Renzo Navas and Maria Ciminieri, who gave me the possibility to extend my research in many directions.

I wish to thank my colleagues at the RSM department and the Labo4G team at Télécom Bretagne, with whom I have worked on many projects.

I would like also to especially thank Dr. Andres Arcia-Moret and Dr. Alberto Blanc, with whom I have collaborated a lot during my thesis, giving me many helpful advices.

I wish also to thank all my colleagues and professors from the Faculty of Engineering of the University of Buenos Aires, in particular to Professor Alberto Dams, for showing me the way to start my experience in France.

I wish to thank a lot to my family: my parents, my brothers, my aunts and uncles, my grandmother, my girlfriend and my in-laws, who have always supported me.

ABSTRACT

Nowadays, mobile users integrate multiple wireless network interfaces in their devices, like IEEE 802.11, 2G/3G/4G cellular, WiMAX or Bluetooth since these heterogeneous technologies can provide Internet access in urban areas. In this context, there is a potential for mobile users to exploit the diversity of the wireless interfaces in order to be always best connected to the networks at any given time and place. However, taking advantage of this network diversity requires an efficient mobility and multihoming management. Regarding mobility, mobile users need to discover wireless networks and perform seamless handovers between different points of attachment. In order to support multihoming and allow using multiple wireless networks simultaneously, there is a need to define network selection mechanisms to assign the applications to the different wireless interfaces in the most optimal manner. In this thesis, we first provide a characterization of the network diversity by exploring and analyzing the performance of current wireless deployments in urban areas, especially considering cellular and IEEE 802.11 community networks. Then, we focus on IEEE 802.11 mobility, particularly on the access point scanning process, by providing two adaptive algorithms that aim to set the most suitable scanning parameters for each scenario condition. We evaluate these algorithms by experimentation and compare their performance against common fixed-parameters scanning strategies. Finally, we study the network selection in such a multi-homed scenario and provide a decision-making algorithm to find optimal flow-interface assignations, considering QoS and energy consumption criteria. This decision algorithm is modelled using a multi-objective optimization problem and genetic algorithms. We evaluate, by the means of simulations, the performance of our approach against preference-based decision-making algorithms.

RÉSUMÉ

Aujourd'hui, les utilisateurs mobiles intègrent plusieurs interfaces sans fil dans leurs dispositifs mobiles, tels que IEEE 802.11, des technologies cellulaires 2G/3G/4G, WiMAX ou Bluetooth, car ces technologies hétérogènes peuvent fournir un accès Internet dans les zones urbaines. Dans ce contexte, il existe un potentiel pour les utilisateurs mobiles d'exploiter la diversité des interfaces sans fil, afin d'être connectés aux réseaux de la meilleure manière possible, à tout moment et partout. Cependant, afin de profiter de cette diversité des réseaux il est nécessaire d'avoir une gestion efficace de la mobilité et de la multi-domiciliation. En ce qui concerne la mobilité, les utilisateurs mobiles ont besoin de découvrir les réseaux sans fil et basculer entre des points d'accès d'une façon transparente et sans coupures. Afin de supporter la multi-domiciliation et de permettre l'utilisation de plusieurs réseaux sans fil simultanément, il est nécessaire de définir des mécanismes de sélection des réseaux visant à attribuer les flux d'applications aux différentes interfaces sans fil d'une manière optimale. Dans cette thèse, nous avons d'abord caractérisé la diversité des réseaux en explorant et en analysant les performances des déploiements sans fil actuelles dans les zones urbaines, en particulier les réseaux cellulaires et les réseaux communautaires basés sur IEEE 802.11. Ensuite, nous avons étudié la mobilité dans les réseaux IEEE 802.11, particulièrement le processus de découverte des points d'accès, en fournissant deux algorithmes adaptatifs qui visent à utiliser les paramètres de découverte les plus appropriés dans chaque scénario. Nous évaluons ces algorithmes par l'expérimentation et nous comparons leurs performances par rapport aux stratégies utilisant des paramètres par défaut. Enfin, nous étudions la sélection des réseaux dans un scénario multi-domicilié et nous proposons un algorithme de prise de décision pour trouver l'attribution optimale des flux aux différentes interfaces, en prenant en compte des critères de qualité de service et de consommation d'énergie. Cet algorithme de décision est modélisé par un problème d'optimisation multi-objectif et est résolu avec des algorithmes génétiques. Nous évaluons, par le biais de simulations, les performances de notre approche contre des algorithmes de décision basés sur des préférences.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 MOTIVATION AND OBJECTIVES

In the current wireless environment users can find different types of access networks, like Wireless Personal Area Networks (WPAN), Wireless Local Area Networks (WLAN) and Cellular Networks. Originally, these networks have been designed to satisfy different needs in terms of capacity and coverage, oriented to different markets and user characteristics. For example, the Bluetooth technology (WPAN) has been designed for device-to-device communications and proximity data sharing, IEEE 802.11 (WLAN) to provide high data-rate in very small coverage areas and cellular network technologies to cover very large areas with a reliable and high data-rate wireless access.

These wireless technologies are based on different technical specifications. They operate in different frequency bands, use diverse modulation and coding schemes, manage the radio resources in different manners and consume different amounts of energy. Moreover, in the most general case, the network deployments belonging to different technologies are loosely-coupled, i.e., there is not a common control or management layer allowing collaboration among them. Then, the heterogeneity of networks involves two aspects. From the mobile user point of view, the heterogeneity implies having multiple accesses any time and any place. From a technological point of view, the heterogeneity implies different specifications and mechanisms for the physical, medium access control and radio resource management for each particular technology.

All these network technologies have been intensively deployed in the last years, especially in urban areas. At the same time, since none of these technologies could lead the market, mobile device manufacturers have integrated multiple wireless technologies in any single device model. In this scenario, mobile users can now take advantage from multi-homing, having the possibility of dynamically selecting different access networks, corresponding to different wireless interfaces in an alternative or simultaneous manner. Such an usage of the wireless networks is referred to as the Always Best Connected (ABC) paradigm. This concept has been first proposed by Gustafsson and Jonsson [5], where the authors characterized the ABC scenario, in which a user always selects the best available wireless interface to transmit its flows. The different functional components that need to be implemented in order to completely achieve the ABC paradigm are listed below:

- **Access Discovery**: to find the available networks and evaluate their expected performance.
- **Access Selection**: to select the best network at any time considering different criteria.
- **Authentication, Authorization and Accounting (AAA)**: to facilitate AAA by unifying procedures among access network operators
- **Mobility Management**: to guarantee session continuity, session transfer and user reachability
- **Profile Handling**: to manage user subscriptions, credentials and accounting information
- **Content Adaptation**: to be capable of detecting changing network conditions and adapt applications demands to the new context

Far from being a fact, there are some limitations that restrict users from intelligently exploiting this heterogeneous wireless scenario in an ABC manner. In this thesis, we propose contributions for the two first functional components of the ABC paradigm: Access Discovery and Access Selection. Regarding access discovery, a mobile user moving out from the boundaries of a point of attachment needs to discover new point of attachments of different technologies and connect to the best one while assuring a seamless transition (i.e., handover).

With regard to access selection, we currently observe, a lack of multihoming management in mobile devices. Nowadays, mobile users get connected to a single wireless interface at a time, i.e., mobile devices do not use several network interfaces at the same time. Recently, different solutions and protocols have been proposed to manage the usage of multiple interfaces. Shim6 [6] and HIP [7] are two examples of this kind of protocols, giving a single identifier for the applications (i.e., a default address) and a set of intermediate mechanisms that are able to transparently spread the application flows on the different interfaces. However, there is still a lack of mechanism to decide how the different flows have to be assigned to the different interfaces. This decision-making process, i.e., the network selection process, can consider multiple criteria, leading to a variable complexity to come out with optimal flow-interface assignations.

1.2 THESIS OVERVIEW AND CONTRIBUTIONS

In this thesis, we study the diversity of current wireless accesses in order to facilitate the exploitation of multi-homing in such a mobile heterogeneous environment. In particular we focus on the two aforementioned limitations related to handover support and decision-making for multihoming support.

First, since we have observed that WLAN, and particularly Community Networks, are as dense as cellular networks in urban areas, we study the handover process in IEEE 802.11, in order to reduce its duration and allow a continuity of WLAN connectivity while moving. Second, since different wireless technologies cohabit at any given place, we study the decision-making process performed by multi-homed users aiming to simultaneously use several interfaces at a time. In such a process, the user aims at assigning different application flows to multiple wireless interfaces in the most efficient manner, considering different criteria.

We can divide the contribution of this thesis into four aspects:

DESIGN AND IMPLEMENTATION OF A WIRELESS SENSING AND ANALYSIS PLATFORM (WI2ME) In order to characterize the diversity of wireless networks, there was the necessity of analyzing the real deployments, which required a mobile sensing platform. There are nowadays a number of sensing tools, but none of them provided a complete access to the collected traces and, moreover, they did not allow automatic connection to existing networks in order to evaluate their performance. We have designed and developed a new sensing tool for the Android system, Wi2Me, providing fine-grained statistics for WLAN and cellular networks. This tool is open-source and we plan to distribute it to the general public.

EVALUATION STUDY OF CURRENT HETEROGENEOUS WIRELESS DEPLOYMENTS Using the Wi2Me platform, we have performed a complete evaluation study of wireless deployments in the city of Rennes, France, particularly focusing on Community Networks (CN), a new WLAN-based communication paradigm that uses existing residential Access Point (AP) to provide Internet connectivity for urban users. We show that CN provide a ubiquitous wireless access, with a coverage equivalent to cellular technologies in urban areas. However, we point out some existing limitations for CN. Due to the high density of CN, a mobile user could seamlessly roam to a new AP, but in practice, all application flows are interrupted even if the mobile user can associate to a new CN AP. Additionally, using Wi2Me, we analyze the impact of handover on the performance of on-going communications, showing that they are greatly degraded.

ACCESS POINT DISCOVERY OPTIMIZATIONS IN IEEE 802.11 We aim at minimizing the impact of handovers on on-going communications. Particularly in IEEE 802.11, this impact is mostly related to the AP discovery process, which is the most time-consuming phase during a handover. In this thesis, we provide a set of mechanisms that reduce the duration of the discovery phase (i.e., the scanning latency) by efficiently configuring the scanning parameters. We define two parameter adaptation algorithms, the Adaptive Discovery Mechanism and the Cross-Layer Adaptive Scanning and evaluate

them using open-source IEEE 802.11 drivers. By the means of experimentation, we show that an adaptation of the scanning parameters to the current scenario can optimize the discovery process performance not only regarding its duration but the number of discovered AP as well.

A DECISION-MAKING FRAMEWORK TO ASSIGN APPLICATIONS TO WIRELESS INTERFACES IN MULTI-HOMED DEVICES In order to exploit network diversity, a decision-making mechanism to efficiently assign the different application flows to the available interfaces needs to be defined. Several network selection mechanisms have been proposed in the last years, considering a wide spectrum of parameters and criteria. These criteria include for instance, Quality of Service (QoS) requirements for each application, user preferences and monetary or energy costs of each interface. In all existing solutions, these criteria are combined using preference values in the form of weights. Using a preference-based strategy to solve a decision-making problem simplifies its complexity and allows the decision-maker to rapidly obtain a solution. However, the use of weights adds a high level of subjectivity to the decision making and prevents from finding the real trade-off between the different criteria. In order to overcome to this limitation, we design and evaluate a multi-objective approach to support network selection based on two relevant criteria: the energy consumption and the bandwidth. In order to obtain a complete view of the trade-off, we solve the optimization problem using evolutionary algorithms.

1.3 OUTLINE

The thesis contribution is organized in three chapters. First, in order to characterize the current wireless environment we propose in Chapter 2 an evaluation study of current wireless diversity including cellular networks and WLAN, giving special attention to wireless CN. We propose a set of metrics that not only show the characteristics of the current deployments but also highlight the lack of mobility and multi-homing support. Then in Chapter 3, we focus on the handover limitation in current IEEE 802.11 networks. We propose a set of algorithms that aims at reducing the impact of the IEEE 802.11 discovery process duration on the on-going communications. These algorithms use optimized parameters settings for the discovery process. Finally, in Chapter 4, we consider the decision-making support in multi-homed devices, that aims at assigning applications flows to the the available wireless interfaces in an optimal manner. For such a decision-making process, we propose a multi-objective optimization approach that considers the mobile device's energy consumption and the bandwidth demand of the different applications flows.

2

WIRELESS HETEROGENEOUS NETWORKS

2.1 NETWORK DIVERSITY

In the last twenty years, wireless communications have become a relevant part of everyday life. Nowadays, mobile users need to permanently access the Internet not only for professional purposes but also for getting communicated with their entourage using different mobile applications and services. In the current ecosystem of mobile communications, there is not a single wireless technology covering all user requirements. For that reason, different wireless technologies have been deployed and periodically enhanced, giving a fully heterogeneous environment. This ecosystem is illustrated in Figure 1, where it can be observed that different wireless technologies have been designed to provide different performance in terms of data-rate, coverage area or mobility pattern. However, when considering real deployments, the performance of the different networks is unpredictable, depending on several parameters (e.g., the radio condition, the network load, the Mobile Station (MS) characteristics). Moreover, there is not still a tight integration among the different network technologies, which prevents mobile users from fully exploiting the diversity and the ubiquity of the deployment, i.e., to seamlessly transit among different access networks or to have the possibility of simultaneously use more than one network without impacting on-going flows.

Before proposing any mechanism to exploit the diversity of the networks, we need to identify which are the particular characteristics of currently deployed networks, i.e., if there is the real possibility that a given mobile user could benefit of the presence of several networks at any given place, providing dissimilar performance. To this end, we propose in this chapter an inventory of the current wireless Internet accesses in urban areas. This inventory consists in a complete measurement study to evaluate the presence of the networks and their performance in a mobility scenario. We consider the two most popular technologies embedded in current mobile devices, cellular-based and IEEE 802.11 networks, particularly focusing on a new communication paradigm based on IEEE 802.11: the Community Networks, which aims at offering an ubiquitous IEEE 802.11 coverage using residential access points. To perform this measurement study, we have designed and developed a set of Android applications that allow gathering and analyzing traces from existing wireless networks: the Wi2Me platform. We found that CN deployments are as dense as cellular networks, providing acceptable performance. However, we have also observed that there are a number

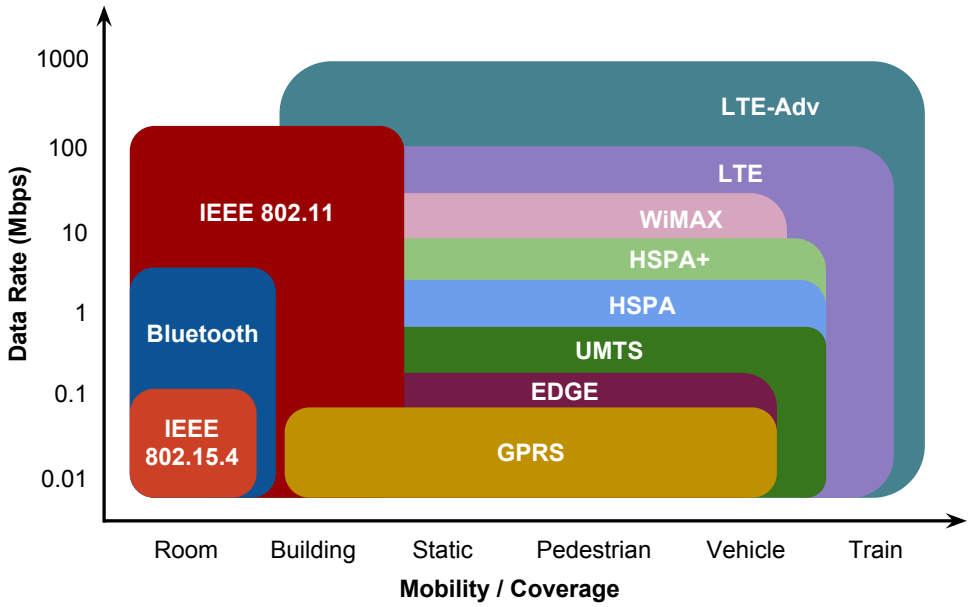


Figure 1: The current wireless ecosystem

of limitations related to a low received signal strength and a lack of mobility support which prevents mobile users from having seamless connectivity while moving.

The chapter is organized as follows. In the following sections, the technical details of each wireless technology in the ecosystem and their evolutions are proposed, including the main concepts in energy consumption of the most popular broadband wireless technologies IEEE 802.11 and 3G cellular networks. Then, in Section 2.2, we present the related work on wireless networks evaluation studies tending to characterize heterogeneous deployments. In Section 2.3, we present the *Wi2Me* platform, an Android tool to gather traces from cellular networks and WLAN, having the ability to automatically test the performance of CN and open networks. A detailed evaluation study of IEEE 802.11 CN and cellular networks is proposed in Section 2.4, which characterizes the different networks in terms of different metrics. Then, since we have observed a lack of mobility support in CN, we propose in Section 2.5 a comparative study between CN and a managed IEEE 802.11 deployment to evaluate the impact of mobility on the transport layers. Finally, in Section 2.6 we discuss the possible evolutions in CN and in Section 2.7, we conclude the chapter.

2.1.1 *IEEE 802.11*

IEEE 802.11 [8] is a standard for WLAN that has been released in 1997 to provide high data-rate wireless connectivity in the Industrial Scientific and Medical (ISM) frequency band. The medium access method is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). IEEE 802.11 has been

Standard	PHY Layer Technology	Data-Rate	Frequency Band	Channel Width
802.11-1997	DSSS/FHSS	$1-2\,\mathrm{Mbps}$	$2.4\,\mathrm{GHz}$	$22\,\mathrm{MHz}$
802.11b-1999	DSSS/CCK	$5.5-11\,\mathrm{Mbps}$	$2.4\,\mathrm{GHz}$	$22\,\mathrm{MHz}$
802.11a-1999	OFDM	$6-54\,\mathrm{Mbps}$	$5\,\mathrm{GHz}$	$20\,\mathrm{MHz}$
802.11g-2003	DSSS/OFDM	$1-54\,\mathrm{Mbps}$	$2.4\,\mathrm{GHz}$	$20\,\mathrm{MHz}$
802.11n-2009	OFDM	$6-600\,\mathrm{Mbps}$	$2.4\,/\,5\,\mathrm{GHz}$	$20\,/\,40\,\mathrm{MHz}$

Table 1: IEEE 802.11 physical layers [1]

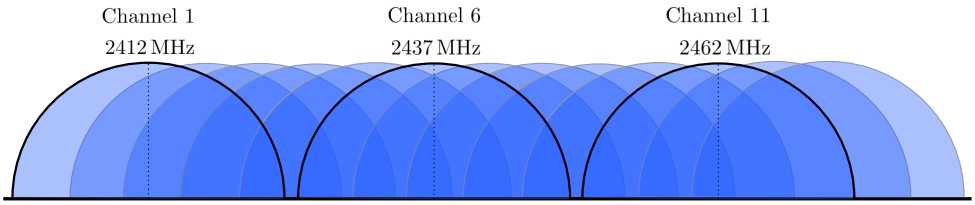
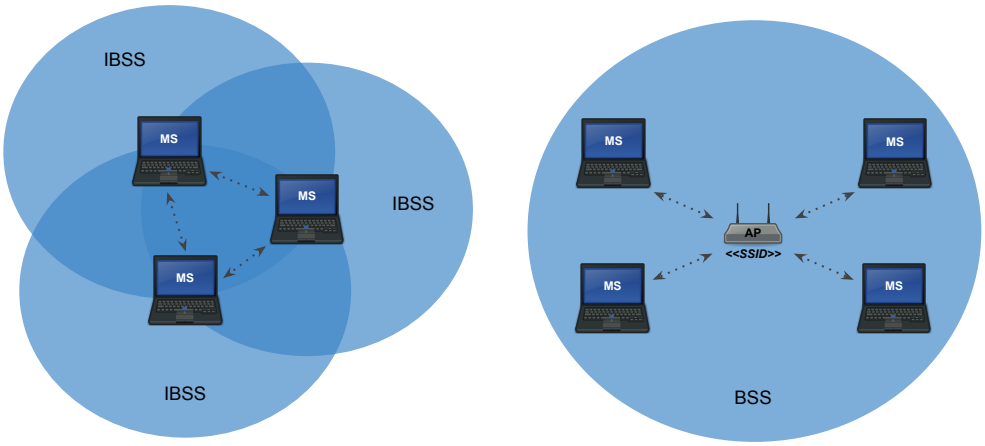


Figure 2: IEEE 802.11b channel layout

continuously evolving, as detailed in Table 1, to provide nowadays a maximum theoretical data-rate of $600\,\mathrm{Mbps}$ and an average coverage area of $100\,\mathrm{m}$. The first IEEE 802.11 standard was intended to provide a megabit-per-second wireless access using Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) in the $2.4\,\mathrm{GHz}$ and the infrared band. A data-rate improvement was achieved two years after with the introduction of IEEE 802.11b (based on DSSS but using Complementary Code Keying (CCK) as modulation scheme), and IEEE 802.11a, which uses the $5\,\mathrm{GHz}$ band and Orthogonal Frequency Division Multiplexing (OFDM) modulation. The latest standard, IEEE 802.11n, significantly increases the maximum data-rate from its predecessor, including a number of enhancements in the physical layer [9], such as Multiple Input Multiple Output (MIMO) antenna system, enhanced OFDM (providing more sub-carriers, a reduced guard-interval and optimized error correction codes) and an optional extended channel bandwidth, up to $40\,\mathrm{MHz}$. It also provides some medium access optimizations, like Frame Aggregation (allowing the MS to send multiple frames per medium access) and Block Acknowledgement (to acknowledge multiple IEEE 802.11 packets with a single frame).

Regarding channel allocation in IEEE 802.11, the typical channel layout for the European regulatory domain in the $2.4\,\mathrm{GHz}$ (IEEE 802.11b) is illustrated in Figure 2. We observe that only three fully non-overlapping channels are available (1,6 and 11). In the American regulatory domain, having only 11 channels, other possible channel combination of three of more channels suffers from partial frequency overlap.

The on-going work in the IEEE 802.11 Working Group includes the specification of IEEE 802.11ac, a gigabit-per-second wireless access in the $5\,\mathrm{GHz}$ band, IEEE 802.11y (approved in 2008) a high-power access (giving more



(a) IBSS Mode (b) Infrastructure Mode

Figure 3: IEEE 802.11 deployment modes

than 5 km range) in the 3.7 GHz band and IEEE 802.11ad, a very-high data-rate access in the 60 GHz band, mainly intended for a multimedia home wireless access.

2.1.1.1 *Deploying IEEE 802.11*

The main building block of an IEEE 802.11 network is defined by the Basic Service Set (BSS), consisting in a group of stations communicating together. The area where the communication takes place is identified as the Basic Service Area, which is conditioned by the wireless medium characteristics (i.e., path-loss, fading, interferences). In order to communicate within a BSS, two different approaches exist (see Figure 3): the Independent Basic Service Set (IBSS) (or adhoc) mode and the Infrastructure Mode.

INDEPENDENT BSS MODE Stations communicate directly with each other when they are within a common coverage range. The *adhoc* designation is related to the fact that this kind of networks are designed for specific purposes and usually for a sporadic or opportunistic usage. Commonly, when using an adhoc topology, one of the nodes has access to the Internet, so the other nodes in the network without an Internet connection may use the former node as a relay to reach the public network.

INFRASTRUCTURE MODE In this case, an AP bridges the MS connected through the wireless medium with other hosts connected to a wired Ethernet link, called the Distribution System (DS). This kind of architecture provides several advantages. First, it allows extending the coverage of a wired network to a large number of MS connected to different AP. Differently from IBSS, in the infrastructure mode the BSS is defined by the radio coverage of the AP, which is normally located in a fixed position, giving a stable coverage area, allowing scalability. Additionally, the AP may buffer frames targeted to

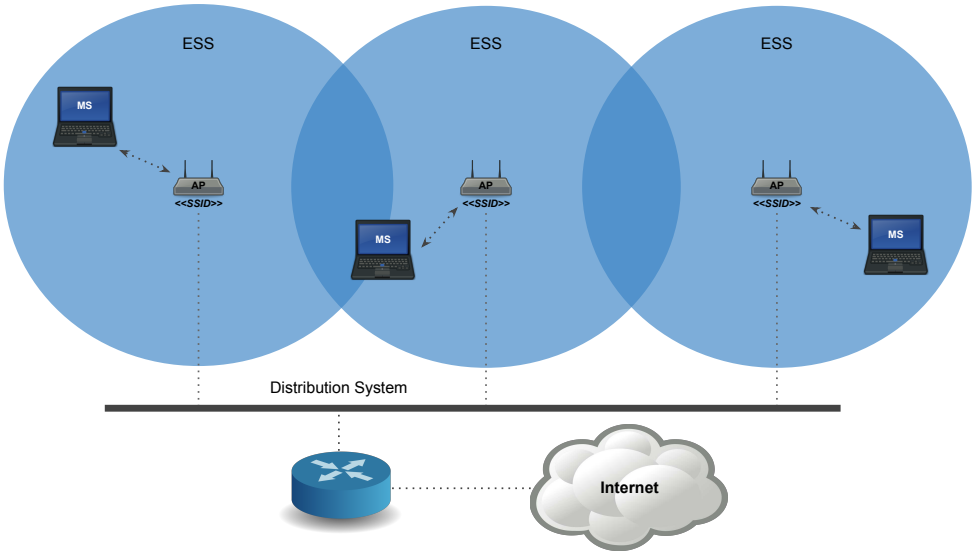


Figure 4: The Extended Service Set

an MS, allowing it to enter in Power Saving Mode (PSM) mode and save evergy by switching-off the MS radio circuits (energy considerations for IEEE 802.11 are given in Section 2.1.4). In order to achieve a larger coverage area, several BSSs may be chained together using a backbone network and conceiving an Extended Service Set (ESS). The main feature offered by an ESS is that an MS can send/receive frames to/from any other MS, even though they belong to different BSS. In an ESS all BSS are identified with a common Service Set Identifier (SSID). Figure 4 illustrates a typical ESS. In this case, the AP is responsible for locating the MS in the ESS and deliver frames to its final destination.

SECURITY IN IEEE 802.11 In both modes, the standard provides a set of security mechanisms for authentication and encryption. Originally, the Wired Equivalent Privacy (WEP) was based on using a common shared key for all MS in the network. Since WEP suffered from several weaknesses enabling users to easily crack the network key, the IEEE has rapidly proposed IEEE 802.11i, including the Wi-Fi Protected Access (WPA) and WPA2 protocols, providing stronger encryption techniques, evolved pre-shared key authentication and centralized authentication based on Remote Authentication Dial-In User Service (RADIUS) servers and the Extensible Authentication Protocol (EAP).

2.1.1.2 *Community Networks*

The concept of CN has been developing in the last years in order to provide ubiquitous WLAN coverage in urban areas. These networks differ from well-known WLAN hot-spot, metropolitan or municipal networks which only cover a number of public places and point-of-interests around different cities (e.g., train stations, transportation hubs, bars, shopping malls) requiring for

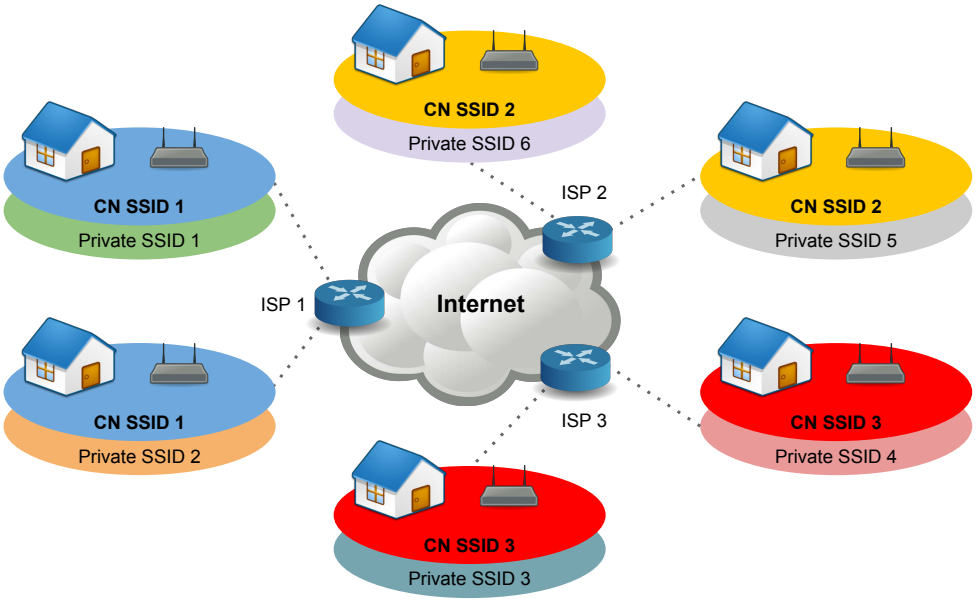


Figure 5: A common CN deployment

special subscriptions to access the Internet. The CN concept is based on extending the usage of existing IEEE 802.11 residential networks. In CN, the Internet Service Provider (ISP) gives to each customer a "box", which contains an Asymmetric Digital Subscriber Line (ADSL) modem, a router and an IEEE 802.11 AP. This box typically provides users with an Internet access through a Network Address Translation (NAT). These AP are capable of broadcasting multiple network identifiers, enabling users to share their ADSL access with other customers of the same ISP using a common open-system authentication network identifier, the CN SSID. In a common use case, when a user is within the range of his own AP, the secured SSID, implementing WEP or WPA, is used. Whenever a user is out of range of his own AP, he can associate to a shared CN AP that broadcasts the CN SSID. Then, in order to access the Internet, the user must provide his own identifiers through an Hypertext Transfer Protocol Secure (HTTPS) captive portal. A common CN deployment is depicted in Figure 5, where different ISP provide a CN access to their subscribers, who also manually configure a private secured SSID.

This concept of sharing private broadband ADSL accesses was originally proposed by FON[1], who has been deploying its platform using a special WLAN AP that users can plug to their routers at home. In France, wireless CN have been increasingly deployed by the subscribers of four of the most important ISP, counting around 13 millions AP: Free (claiming over 4 million AP), SFR (claiming over 4 million AP), Bouygues Telecom (claiming over 700 thousand AP) and, very recently, Orange (claiming over 4 million AP).

In those CN, there is a concern about the incentive scheme to encourage users to share their IEEE 802.11 access. Several authors have designed complex incentives and credit-based schemes, like Manshaei et al. [10], who

1 http://www.fon.com

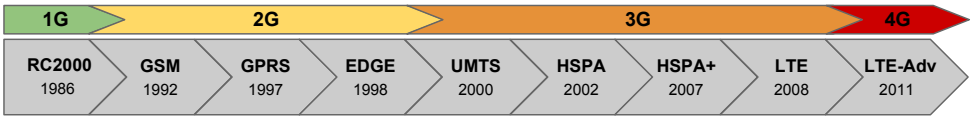


Figure 6: Cellular network evolution

derived an incentive scheme using a game theoretical approach and Ai et al. [11], who proposed a credit-based scheme, where users earn credits for sharing bandwidth. In practice, the incentive scheme of the FON CN is based on three user profiles that are chosen upon the first registration on the network. The *Alien* users are those who do not share their IEEE 802.11 access but still want to get connected to other AP without paying; the *Bill* users are those that sell their access using a time-based pricing and share the earnings with FON; finally the *Linus* users are those that fully share their access without asking for money. Regarding the CN operators in France, there is not complex pricing and incentive scheme, since users willing to access the CN can do so only if they share their own Internet access with other subscribers. In this case, the user can only access the CN belonging to his own ISP.

2.1.2 *Cellular Networks*

2.1.2.1 *Technological Evolution*

The other main broadband wireless access is provided by cellular networks, giving a larger coverage and much better mobility capacities than the aforementioned IEEE 802.11. The evolution of cellular networks has been divided in generations, as depicted in Figure 6. The first generation included a number of analog systems, such as Radiocom 2000 in France. The transition to digital systems arrived with the Second Generation Wireless Network (2G), with the popular Global System for Mobile Communications (GSM) specification which allowed digital voice services over cellular networks. In GSM, uplink and downlink communications are divided using Frequency-Division Duplexing (FDD) and an eight time-slot division using Time Division Multiple Access (TDMA) to guarantee the medium access to up to eight users for 200 KHz channel. The evolution to data-packets networks based on GSM has been first specified in General Packet Radio Service (GPRS), generally called 2.5G, providing a typical data-rate of 18 Kbps. It was based on allowing the MS to use more than one time slot in the TDMA frame. Data-packet access networks have then evolved to Enhanced Data Rates for GSM Evolution (EDGE), or 2.75G, being four times faster than GPRS, thanks to enhanced modulation and coding schemes.

With the increasing demand of data services against traditional voice services, the transition to the Third Generation Wireless Network (3G) was a strong technological shift, since the FDD/TDMA architecture implicitly posed

a performance limit. This technological shift, the Universal Mobile Telecommunications System (UMTS), is based on Wideband Code Division Multiple Access (WCDMA), an evolution of the Code Division Multiple Access (CDMA) technique, which required a completely new network deployment, i.e., new base stations and frequency allocations. In WCDMA, instead of using narrow frequency channels like in GPRS/EDGE, a larger channel of 5 MHz is used. Then, communications are based on spread spectrum, so each MS uses different codes to allow multiple access. This new technique allows a theoretical maximum data-rate of 384 Kbps, which enabled new online applications and a more fluid web-browsing. However, the increasing demand for bandwidth (mainly related to audio and video applications) pushed the evolution of UMTS to High Speed Packet Access (HSPA), called the 3.5G, including High Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA). HSDPA allows a maximum downlink data-rate of 14 Mbps by providing some enhancements from its predecessor, like a shared channel transmissions, higher rate modulations and shorter transmission time intervals. Regarding HSUPA, a new transport channel is defined, namely the Enhanced Dedicated Channel (E-DCH), giving a maximum uplink data-rate of 5.8 Mbps. A further evolution has been designed in the Evolved High Speed Packet Access (HSPA+), including MIMO and higher modulation rates, providing 84 Mbps and 10.8 Mbps in the downlink and uplink respectively.

Even if HSPA and HSPA+ were designed to provide very high data-rates, these technologies are limited in the distance in which data can be delivered, since the symbol duration decreases with the increasing data rate, becoming much smaller than the multipath distance, hindering the data retrieval. Additionally, another limitation is related to the architecture of the core network, which is not an all-IP architecture, forcing a co-existence with legacy voice circuits and gateways.

The 3rd Generation Partnership Project (3GPP) started to define the requirements for a new generation of mobile communications, the Fourth Generation Wireless Network (4G), establishing a peak data-rate of 100 Mbps for high speed mobility and up to 1 Gbps for low speed mobility or static users. In this direction, the Long Term Evolution (LTE) standard has been released in 2008. However, it does not still satisfy the requirements imposed for a 4G technology. For that reason, LTE is commonly referred to as 3.9G, even if it is commercially referred as to 4G. LTE can provide up to 300 Mbps in the downlink and 75 Mbps in the uplink, with a very low latency (5 ms in optimal conditions) and supporting high mobility speeds, up to 500 km/h. It uses the Orthogonal Frequency-Division Multiple Access (OFDMA) mechanism for multiple medium access, supporting both FDD and Time-Division Duplexing (TDD) with variable channel bandwidth, from 1.4 MHz to 20 MHz (recall that a fixed 5 MHz channel was used in 3G systems).

The first 4G standard has been finally defined in the LTE-Advanced specification, allowing gigabit-per-second data-rates by including carrier aggrega-

tion, enhanced MIMO and relay nodes, providing better performance at the cell edge.

2.1.2.2 *Current Deployments*

Mobile phones penetration rate has surpassed 85 % in 2011 [12], being more than 100 % in some regions like Europe and the Americas, giving almost 6 billion subscribers worldwide. Moreover, after the transition to HSPA the volume of mobile data has considerably increased (surpassing the amount of data generated for voice-based services). Nowadays, more than 900 million users have access to broadband wireless accesses through a 3G technology. In parallel, there has also been a great expansion of the mobile devices market, giving more than 3000 different mobile device models (supporting HSPA) that have been launched in the last years.

To respond to this demand, worldwide network operators have been deploying new networks. The current deployment of cellular networks is in practice a mixture of technologies belonging to different generations. In most countries, voice-based services are still carried out using the GSM network, covering nowadays the 90 % of the world population [13]. Regarding the deployment of worldwide data-packet networks, we find that HSPA is the most used technology, since all the 3G commercial deployments have launched the HSPA service. The mainstream on the deployment is marked by the implementation of HSPA+, giving a total of 234 commercial networks in 112 different countries (in July 2012 [13]). However, there is also a fast development of LTE in parallel with HSPA+. This is because networks operators have still to amortize their 3G infrastructure and maximize benefit before shifting to LTE, that will require important investments. There are currently 89 commercial LTE deployments in 45 different countries, expecting up to 150 deployments for the end of 2012.

2.1.3 *Other Wireless Technologies*

Even if the current wireless heterogeneous environment is dominated by a mixture of cellular technologies and IEEE 802.11, there are other wireless technologies that are also embedded in most mobile devices. First, the Bluetooth [14] technology, operating in the 2.4 GHz ISM band, has been originally designed to provide wireless connectivity for data sharing among devices (without any infrastructure) over short distances at relatively low data rates, forming a WPAN. Common usages of Bluetooth include file sharing and messaging between smartphones and computers, peripheral pairing (e.g., headsets, input devices, remote controls) and and Internet connection sharing between smartphones and laptops (also called Bluetooth modem). Bluetooth has been evolving over the last years and is able to provide, in its latest versions (v3.0 and v4.0), up to 24 Mbps, which is comparable to the common data-rates obtained in IEEE 802.11.

The IEEE 802.15.4 [15] standard has been conceived to provide low data rates (up to 250 Kbps), short range and low energy-consumption communications. Its most popular applications are the Wireless Sensor Networks (WSN) and the Internet of Things (IoT). Even if this technology is not strongly present in current mobile devices (e.g., laptops, smartphones, tablets), it has been recently reported[2] the launch of a 802.15.4-enabled Android smartphone, also providing an IEEE 802.11, a UMTS/HSPA and a Bluetooth interface.

Finally, the IEEE 802.16 family of protocols, known as Worldwide Interoperability for Microwave Access (WiMAX) has been designed to offer both fixed and mobile wireless communications at high data-rates, up to 40 Mbps with a very large coverage range. Even if it is currently part of the 4G specification, it did not meet the UMTS and HSPA success. However, there exists a large number of deployments worldwide, serving around 20 million users in 2011[3].

2.1.4 *Energy Consumption of Wireless Interfaces*

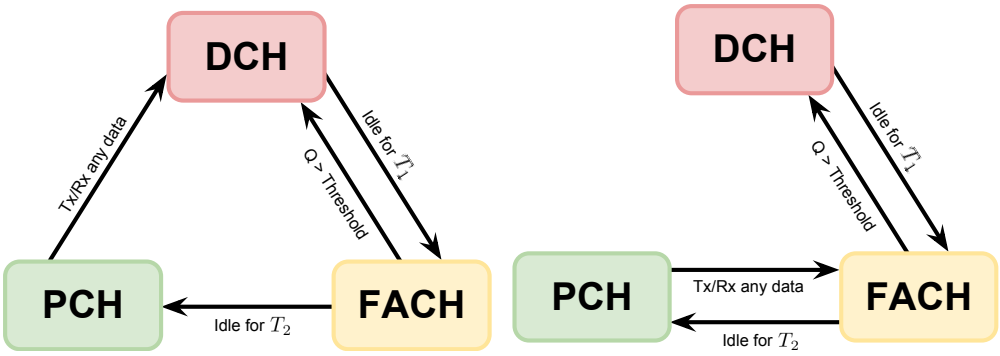
Energy efficiency of mobile devices has become a major issue in the design of modern mobile communication devices, since the improvement of battery technologies has not followed the exponential growth of mobile devices and systems performance [16]. As stated earlier in this chapter, mobile devices, and especially smartphones and tablets, are equipped with a great number of wireless interfaces and have the ability to run several applications at the same time, demanding a huge amount of energy. Moreover, the miniaturization of mobile devices still limits the size of batteries, which have not greatly improved their energy density (measured in Wh/l), commonly providing a total energy of 1500 Wh for a typical smartphone battery, giving less than one day of autonomy for a normal usage. If we consider a multi-interface smartphone (i.e., having at least IEEE 802.11, 2G/3G/4G and Bluetooth wireless radio), in a common use case, the radio interfaces contribute, in average, to the 58 % of the total energy consumption of the device [17]. Note that this contribution may increase if multiple interfaces are simultaneously used (multi-homing). We detail in this section the differences, in terms of energy consumption and operating modes of the two most popular broadband wireless technologies: 3G cellular and IEEE 802.11 networks.

2.1.4.1 *UMTS/HSPA*

As explained in [18], an MS communicating through a UMTS/HSPA interface transits between three different operating states, as listed below:

2 http://www.taztag.com/

3 http://www.wimaxforum.org/news/2866



(a) Carrier 1 (b) Carrier 2

Figure 7: RNC state machines

- Paging Channel (PCH): No radio resource is allocated to the MS since no Radio Resource Control (RRC) connection exists.
- Forward Access Channel (FACH): A RRC connection is established without dedicated channels, limiting the bandwidth up to a few kbps.
- Dedicated Channel (DCH): An RRC connection is established and a dedicated channel has been allocated, giving a high data-rate access to the MS.

The power consumption at each state depends on the particular MS hardware, but in all cases the PCH is the less consuming state and DCH the most consuming state (see Table 4). However, the logic to transit among these states, i.e., the state machine, is not managed by the MS but by the Radio Network Controller (RNC) in the network side, which is responsible for controlling a set of base stations (or NodeB in the 3GPP nomenclature). Then, different network operators may implement different configurations for the state machine, leading in a different energy consumption for the same usage for a given MS.

Two example of state machines are given in [18] for two of the most popular carriers in the USA. Figure 7 illustrates these state machines. In the case of *Carrier 1*, each time an MS in the PCH state has to establish an RRC connection to send or receive data, it switches to the DCH state to exchange packets. Then, it keeps staying in this state if the packet exchange continues or, in the case it becomes idle for an inactivity timer of T_1 seconds, it switches to the FACH state, which consumes less power but provides lower bandwidth than DCH as well. The MS then continously monitors the uplink and downlink packet queue (Q) and if its length becomes larger than a threshold, it switches again to the DCH state to exchange packets at a higher bandwidth. On the other hand, if the MS remains idle in FACH for an inactivity timer T_2, it comes back to the idle state (PCH). A more conservative state machine is implemented in the RNC of *Carrier 2*. In this case, an MS being in the PCH state first switches to the FACH state for any RRC connection request. If a

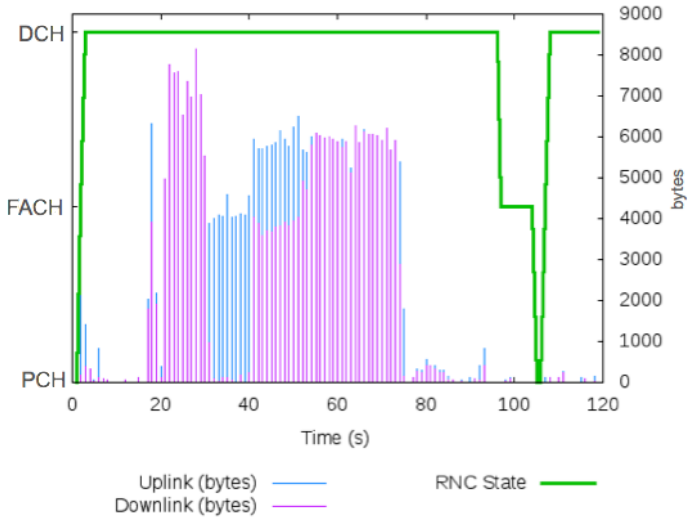


Figure 8: 3G state transitions for a Skype test-call

high throughput is demanded, it then switches to the DCH state. The inactivity timers T_1 and T_2 follows in this case the same logic than for Carrier 1.

We performed an experiment to analyze the transitions of a 3G state machine, as illustrated in Figure 8 [19]. We have obtained this trace by performing a Skype test call in an HTC Dream smartphone using the SFR 3G network and logging the activity of the MS using PowerTutor [20]. We can observe in this trace that once the call ends at time 75 s, the MS remains in the DCH state and switches to the FACH state at time 95 s. Then, 10 s after the transition to the FACH, the MS becomes idle (transition to PCH).

The state machine is set by the network because it is the network operator who has to efficiently manage its limited wireless resources, i.e., the dedicated channels. The network operator can set the logic of the state machine, values for T_1 and T_2 and the queue threshold. In the state machines presented in [18], these parameters are set as detailed in Table 8. Note that there is a trade-off while assigning dedicated channels. In order to negotiate a transition to a DCH, several signalling packets have to be exchanged between the MS and the network, which roughly takes 2 s. An MS willing to consume a low amount of energy may request a DCH transition every time there is some data to exchange and go immediately to the FACH or PCH state (i.e., release the DCH channel) without waiting for an inactivity timer. However, in this situation the user performance is strongly degraded since there is a signalling overhead caused by multiple negotiations of a DCH channel. On the other hand, if the MS wants to maximize the performance, it should always remains in the DCH state, but in this case the negative impact is twofold. First, a high energy consumption may be observed and so the MS battery autonomy is reduced. Second, the network operator may run out of wireless resources and so a great number of users may not be able to perform RRC connections.

Parameter	Carrier 1	Carrier 2
T_1 (s)	5	6
T_2 (s)	12	4
Queue Threshold (bytes)	475	119

Table 2: State machine parameters

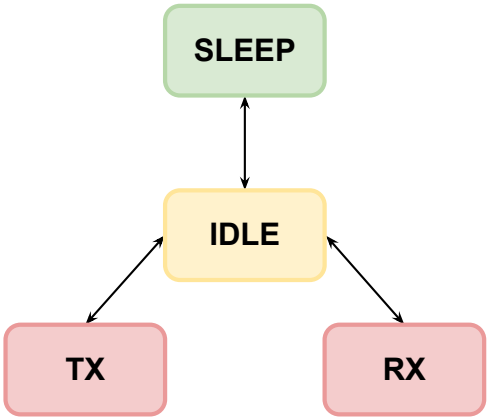


Figure 9: WLAN state machine

2.1.4.2 *IEEE 802.11*

Differently from UMTS/HSPA cellular interfaces, in IEEE 802.11 the operation mode is much simpler and the state transitions are exclusively managed by the MS, i.e., there is not a centralized entity (like the RNC in cellular networks) affecting the energy consumption of the device. Figure 9 illustrates the different states and transitions for a WLAN interface [21]. During an active communication, the interface enters in the transmit (TX) and receive (RX) mode intermittently. If no transmission is active, the MS remains IDLE. Additionally, if the interface supports the PSM it can enter in the SLEEP mode, that consumes very low energy (see Table 3). When being in such state, the MS switches its circuits off and requests the AP to buffer its incoming packets. Then, the MS will periodically wake up to receive AP beacons and will look for a Traffic Indication Map (TIM) that announces buffered packets for the MS. The transition to the SLEEP mode is usually performed after a timeout (an analysis of different PSM strategies can be found in [22]).

2.1.4.3 *Empirical studies and preliminary observations*

Several studies exist in the literature trying to characterize how the different wireless interfaces impact on the global energy consumption of an MS. Petander [23] proposes an overview of energy consumption in terms of the battery drain (measured in percentage) while using the IEEE 802.11 (WLAN) and the 3G (UMTS) interfaces in an HTC Dream smartphone. Some experiments are carried out by varying the traffic load and the signal strength between the MS and the AP or cellular base station. Results show that even if

WLAN State	Nokia N810[21]	HTC G1[21]	Nokia N95[21]	Nokia N90[26]
SLEEP	42	68	88	40
IDLE	884	650	1038	800
TRANSMIT	1258	1097	1687	2000
RECEIVE	1181	900	1585	900

Table 3: Power consumption (mW) of WLAN states

3G State	N/A[17]	N/A[26]	Nokia N95[24]	HTC TyTN II[18]
PCH	< 18.5	19	282	0
FACH	370-740	555	549	400-460
DCH	740-1480	1100	742	600-600

Table 4: Power consumption (mW) of 3G states

UMTS consumes more energy per unit of data (between 0.2 % and 2.8 % per MB), both interfaces slightly consumes the same amount of energy per unit of time (around 0.01 % per second). Xiao et al. [24] proposed a measurement analysis of YouTube video streaming using both UMTS and WLAN interfaces and different scenarios (e.g., progressive download and view, download first play next, local playback). The authors observe that in the case of UMTS, even if the video download is finished and the MS only plays the video, the power consumption does not immediately decrease, leading to a higher total energy consumption compared to WLAN. This is caused by the effect of cellular networks inactivity timers (see Section 2.1.4.1). We proposed in [19] an evaluation study of energy consumption of different mobile applications using 3G (UMTS/HSPA) and WLAN interfaces in an Android MS. In particular, these measurements have been performed using PowerTutor [20], a real-time power estimation tool that allows tracing the instantaneous energy consumption of the different components of the MS (e.g., procesor, screen, WLAN, 3G, Bluetooth, GPS). As in [24], we have also observed the effects of inactivity timers in 3G, especially when doing web-browsing, since during the idle time between two web page requests (i.e., the reading time) the MS is not capable of reducing the 3G power consumption. More particularly in [25], Haverinen et al. study the energy consumption of always-on applications (based on keep-alive message exchanges) using UMTS. They show the impact of inactivity timers and keep-alive frequency, proposing some configurations to both parameters to mitigate this negative impact on energy efficiency.

2.1.5 *Discussion*

This panoply of wireless technologies are commonly present in urban areas, giving multiple access possibilities to mobile users. In particular for Internet access, IEEE 802.11 and 3G networks appear as the most convenient wireless

accesses in urban areas. In order to analyze their coexistence and the possibility of simultaneously using both interfaces, there has been in the last years the necessity of exploring and evaluating the wireless deployments. These evaluation studies were usually taken as an input to design and evaluate new protocols aiming at optimizing the user experience while using those networks.

In the context of this thesis, a fine-grained knowledge of the wireless environment and the collaboration between different technologies help to better understand the drawbacks and limitations of current deployments and also to identify the opportunities and challenges for optimizing those networks. We particularly focus on optimizing IEEE 802.11 AP discovery in Chapter 3 and the decision-making processes for network selection in Chapter 4.

In the following, we present past evaluation studies of wireless diversity so as to motivate the development of the Wi2Me platform, a new wireless sensing tool that allow characterizing not only the presence and performance of wireless networks but also their limitations for mobile users.

2.2 RELATED WORK ON WIRELESS DIVERSITY EVALUATION

In this section, we present the related work on measurement and evaluation studies aiming to characterize wireless networks in urban environments, including the existing tools and applications to perform such studies. Regarding WLAN deployments, we focus not only on measurement studies for open AP deployments in urban areas but also for research dedicated networks, that have been specifically deployed to analyze the performance of IEEE 802.11 under different usage conditions. However, in the study that we present in Section 2.4, we focus on open urban WLAN deployments, particularly in CN and their coexistence with operator-based cellular networks.

2.2.1 *Evaluation Studies*

In [2], Bychkovsky et al. propose an evaluation study of IEEE 802.11 AP deployments in urban areas to provide network connectivity for moving vehicles. This study consisted in 290 hours of evaluation of existing urban WLAN deployments using Linux-based computers with an IEEE 802.11 card and a 5.5 dBi antenna and a Global Positioning System (GPS) device installed in several vehicles. These computers were responsible for discovering the available networks, associating to a (good) candidate AP, obtain an IP address, ping a remote server and finally attempt to upload data through TCP connections. During one year, nine vehicles have been moving around some urban areas of Boston and Seattle (USA), discovering more than 32.000 AP and associating to more than 5.000 AP. However, since their system was only capable to join open WLAN, i.e., without any authentication mechanism at the Medium Access Control (MAC) layer, they could only successfully ping their remote

Median time between AP association and IP address acquisition	3 s
Median time between AP association and first AP ping	8 s
Minimum / Average / Maximum Scan delay	120/750/7030 ms
Minimum / Average / Maximum Association delay	50/560/8970 ms
Average time between two successful associations	75 s
Average time between two successful end-to-end connections	260 s
Median connectivity duration to an AP	13 s
Median AP coverage	96 m
Average packet delivery rate	78 %
Median per-connection throughput	30 KB/s
Median per-connection uploaded data	216 KB

Table 5: General results [2]

server in only 20 % of the connections, i.e., the 3 % of the discovered APs. Using the traces collected during these connections, the authors evaluated the networks using different performance metrics as given in Table 5. For moving vehicles, they have observed very short connection durations (in median 13 s) allowing to upload 216 KB in median.

Since this study focused on wireless connectivity for vehicles, the authors provided a set of metrics to analyze the impact of speed on the connection performance. They find that the number of successful associations to an AP is uniform up to 60 km/h. Above this speed, they observe a very few number of connections. The connection duration linearly decreases with an increasing speed, up to 60 km/h. For greater speeds the connection duration behave somehow random. However, they observe no correlation between the vehicle's speed and the packet delivery ratio.

With the same aim, Balasubramanian et al. [27] study the potential for existing WLAN to provide connectivity to moving vehicles. Instead of inventorying existing wireless networks like in [2], they focus their attention on two particular deployments, VanLAN [28] and DieselNet [29], that have been specially conceived to analyze vehicular mobility under IEEE 802.11 deployments. In this case, they perform several measurement studies to collect beacons from different AP in both networks so as to analyze, by the means of trace-driven simulations, the performance of different handover algorithms seeking to minimize the disruption in IEEE 802.11 connectivity.

In a more recent work, Balasubramanian et al. [3] perform a measurement study to analyze the feasibility of offloading users' connections from cellular to WLAN networks. They conducted measurements in three different testbed in the urban areas of Amherst, Seattle and San Francisco (USA), sensing both 3G cellular and IEEE 802.11 networks. They installed a computer carrying an IEEE 802.11b and a 3G modem (HSDPA) in several vehicles. This computer has a software that is able to scan for networks and transfer fixed amounts

Average 3G availability over the time	87 %
Average WLAN availability over the time	11 %
Average Unavailability using 3G/WLAN simultaneously	5 %
Median 3G TCP throughput uplink/downlink	500/600 Kbps
Median WLAN TCP throughput uplink/downlink	280/200 Kbps
Median 3G UDP throughput uplink/downlink	850/1000 Kbps
Median WLAN UDP throughput uplink/downlink	400/500 Kbps
Average packet loss 3G/WLAN	7/22 %

Table 6: General results [3]

of data. In Amherst, these computers have been deployed on public buses, performing fixed routes, while in Seattle and San Francisco the mobility pattern was random. The results of this measurement study is presented in Table 6. They observed a much higher TCP throughput than in [2]. This could be due to more reduced speeds and because in some cases they connect not only to open AP but to other AP they have deployed along the path. The authors analyze, using trace-driven simulations, the offloading capacity of 3G data over WLAN. They estimate that in the 53 % of the locations, at least 20 % of the 3G data could be sent/received over WLAN. Moreover, in 9 % of the locations all 3G data could be sent over WLAN. The total amount of data that a user may offload to WLAN not only depends on the availability of AP but also on the tolerance of users to delay the transmission of some flows until a WLAN becomes available.

A most recent evaluation study analyzing the offloading capacity of mobile devices in urban heterogeneous networks is proposed by Lee et al. in [30]. In this study, the authors distributed 100 iPhones to different users moving around some metropolitan areas in Seoul (South Korea) during 20 days. These devices ran a especial application, *DTap*, that records the statistics of available WLAN every three minutes and perform connections to open AP to estimate throughput and Round Trip Time (RTT) using the ping command. The analysis of the statistics gives that the average temporal WLAN coverage for a user is around 70 % while spatially, the area covered is between 10 % and 20 %. Users get connected through a WLAN in average 120 minutes per day with a time between connections of around 41 minutes. They model the connection duration and the interconnection time using the Weibull distribution and evaluate the performance of different offloading strategies using trace-driven simulations. The authors estimate that up to 65 % of the traffic can be offloaded from 3G to WLAN, giving a maximum energy saving of 55 %.

A measurement study of WLAN deployments in the city of Chicago (USA) is presented in [4], aiming at collecting a large number of traces to evaluate best AP selection algorithms using trace-driven simulations. This study

	Downtown	Residential	Suburban
AP found	797	464	256
Open AP found	78	81	43
AP per scan	2.4	2.0	1.8
Usable AP	53.9 %	100 %	97.7 %
Estimated Bandwidth (KB/s)	60	80	200
Estimated RTT (ms)	22	29	20

Table 7: General results [4]

is performed in three different urban areas (i.e., downtown, residential, suburban), walking around a grid of 1.3 km^2. Traces have been collected using an iPaq handheld with an IEEE 802.11b card. In addition to discover AP, the device estimated the RTT and ran a set of scripts to determine which port numbers (e.g., HTTP, SMTP, Samba) were open. A summary of the results encountered in this measurement study is presented in Table 7. They observed different performance for different urban areas, achieving the highest throughput in suburban areas, where the lowest density of AP (i.e., the lowest interference) is found.

Another interesting case of study for urban WLAN deployments is the Google Muni WiFi Network[4]. This is a public accessible IEEE 802.11 mesh network in Mountain View, California (USA), covering a 31 km^2 urban area with more than 500 Tropos[5] AP, serving up to 2.500 (in 2008 [31]) and most recently 19.000 (in 2009 [32]) simultaneous users and daily transporting more than 600 GB of data at a maximum downlink data-rate of 3 Mbps. Contrary to CN, in which public indoor AP are shared to the public, the Google Muni WiFi Network enters in the category of metropolitan or municipal WiFi Networks, which commonly consists in a dedicated deployment using outdoor AP deployments which are managed by a governmental institution or a third-party operator. Several municipal deployments exist nowadays all over the world, specially in Europe and North America. In the case of the Google Muni WiFi Network, different authors have performed measurement studies to analyze its performance under urban mobility patterns. A very complete evaluation of the usage patterns of this network is provided by Afanasyev et al. [31]. Even if this evaluation study does not focus on the radio and wireless aspects, it provides a number of metrics to analyze what users can expect from the usage of this kind of networks. They distinguish three types of users. The *smartphone* users (those holding a handheld or similar device), the *hotspot* users (those users connecting with a laptop) and the *modem* users (those users that deploy a high-transmit power equipment at home that connects to the Google Muni WiFi Network and converts the wireless signal

4 http://wifi.google.com/
5 http://tropos.com

into a wired signal that users can access at home using an Ethernet interface). Regarding the session duration, in 65 % of the cases modem users get connected for less than one day, while for hotpot and smartphone users the median session duration is 30 and 10 minutes respectively. Smartphone users mainly generate HTTP and TCP traffic while modem and hotspot users generate additional traffic, like Peer-to-Peer, Virtual Private Network (VPN) and interactive traffic (e.g., streaming, VoIP). The authors however find that there is an order of magnitude more hotspot users than modem users but even in this scenario, modem users generate a similar amount of traffic than hotspot users. Regarding users' mobility, in a one hour period a smartphone connects to six different AP in median. The 10 % most moving smartphones get connected to 32 different AP. They also observe some oscillations in fixed modem users who change AP at least once. This could be due to the long range of the Tropos AP, up to 500 m, which impacts the radio signal and forces a handover on the modem side.

A second measurement study over the Google Muni WiFi Network is proposed by Arjona et al. [33]. In this case, the authors analyze the capacity of this network to support Voice over Internet Protocol (VoIP) for mobile users. An experimentation campaign was carried out in sub-urban, urban and corporative areas around the city using Skype-to-Skype and Skype-to-cellular phone calls. Results show a poor performance of VoIP calls using the Google Muni WiFi Network, measured with the Mean Opinion Score (MoS), especially for Skype-to-cellular phone calls. The authors estimate that, in order to achieve a similar MoS than for cellular network phone calls, the Google Muni WiFi network should increase its AP density from 30 to 81 AP/km^2, which they estimate as costly as a cellular network deployment.

All these studies show that, in urban environments, users have the possibility to intermittently connect to WLAN while moving. Note that in these studies, open AP networks have been generally considered, which at that time represented a considerable part of the deployment. These deployments represent now a very low number of AP (only 3.12 % of the AP, excluding CN AP, in our measurement study [34]). This motivated us to perform a completely new measurement study for 3G and WLAN using a new wireless sensing tool that allows connecting to multiple CN. In the following, we present the existing wireless sensing tools and we propose a new platform Wi2Me, for Android mobiles.

2.2.2 *Existing War-driving Applications*

Even if the measurement studies introduced in 2.2.1 used *ad hoc* software and hardware platforms to gather the traces from wireless networks, there are nowadays several publicly available tools to do so, running on different

mobile platforms. These tools are usually referred to as *war-driving* applications in the literature. In this section, we summarize some of them, mainly those operating under the Android system, like the Wi2Me platform.

OPENBMAP OpenBMap[6] is a free and open source wireless sensing tool for WLAN AP, cellular base stations (GSM, UMTS, HSPA and LTE) and Bluetooth devices that aims to build a free accessible data base. It is available for Android, Windows Phone and openmoko mobile platforms, providing in a website a limited view of the global traces, including coverage maps of cellular networks and WLAN AP and a full view for self collected traces (containing location information). In July 2012, the database contained cellular traces from 171 countries and more than 780.000 cells from 604 different operators. Regarding WLAN traces, more than 650.000 AP from 57 different countries have been inventoried.

SENSORLY The Sensorly[7] project is an Android participatory sensing platform that aims at creating a very precise wireless network cartography, showing the network coverage for 2G, 3G and LTE technologies for more than 120 network operators in different countries. The application allows mobile users to manually test the data-rate of different networks, which is then uploaded to the remote Sensorly database, in order to feed the network maps. Regarding CN in France, Sensorly allows obtaining an estimation of the position of individual AP for the four main ISP in France: Free, Orange, Bouygues Telecom and SFR, without providing any information about the overall performance of those networks.

WIGLE Very similar to OpenBMap, the Wigle platform[8] is a multi-platform participatory sensing tool for WLAN and cellular networks that has been actively gathering traces since 2001. It actually runs on Android mobiles and Linux, Mac and Windows computers, counting more than 125.000 users. In July 2012 it counted more than 68 million WLAN AP and 1.5 millions cellular base stations all around the world, mainly in Europe and North America. They also propose a partial view of the traces and some interesting statistics (e.g., evolution of the number of discovered networks and encryption protocols over the time).

OPENSIGNALMAPS OpenSignalMaps[9] is an Android application providing similar functionalities than the Sensorly platform but particularly for 3G networks. The traces are also represented in a dynamic heat-map while giving some metrics to compare two or more network operators in terms of average signal strength, uplink and downlink data-rate and round-trip latency.

6 http://openbmap.org/
7 http://www.sensorly.com/
8 http://wigle.net/
9 http://opensignalmaps.com/

It also compares, for some cities in USA, United Kingdom, Italy, Germany and Spain, the network performance compared to the average country or worldwide performance. The OpenSignalMaps database contains coverage and performance information from a large number of countries. The application also discovers WLAN AP, but these traces are not available online.

2.3 THE WI2ME PLATFORM

2.3.1 *Introduction*

As it has been previously presented in Section 2.2, existing evaluation studies of wireless heterogeneous networks provide a characterization of current deployments by using *ad hoc* platforms, i.e., specific software installed in some specific hardware to discover the networks and evaluate their performance. On the other hand, public available war-driving applications are available for different platforms, but, in all cases, even if they can successfully discover wireless networks and locate them in a map, they lack of automatic mechanisms to trigger connections and evaluate the performance of the networks by downloading and uploading data packets in the background, without requiring the intervention of the user. Moreover, the access to the traces is very limited for the users, since only some predefined metrics are available. Unfortunately, the user has not the possibility to manipulate raw traces (e.g., databases, files) and calculate their own metrics. Moreover, as we have shown in a first measurement study [35] in Rennes (France), the high density of AP and especially of CN, required for a new sensing tool, capable to analyze CN in an automated manner.

To reach our goal of analyzing the wireless diversity, we have designed and implemented a new war-driving platform, called Wi2Me. The main goal of this platform is to allow continuous AP and base station scanning and automatic connection to cellular networks and CN, while gathering all the collected traces into an internal database. In practice, this platform is composed of a common Android core, containing the main functionality and two front-ends, conforming two application versions: a user version, *Wi2Me-User* and a research version, *Wi2Me-Research*. Both applications are detailed in the next section.

2.3.2 *Design and Implementation*

2.3.2.1 *Wi2Me-User*

The Wi2Me-User version acts as a network manager for common smartphone users, aiming to automatically connect and authenticate to CN while moving. With Wi2Me-User, a mobile user first sets his/her CN accounts (username and password) that the application will use to attempt automatic con-

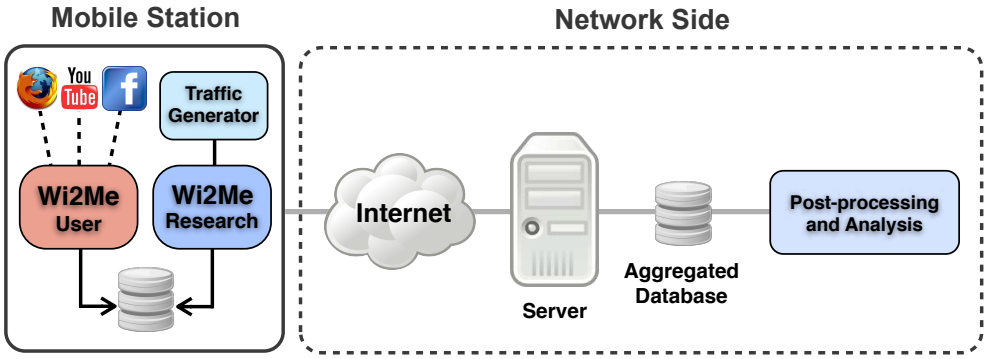


Figure 10: Wi2Me Architecture

nection and authentication. Then, he/she simply pushes the "Start" button to run the network manager, that will perform scanning, connection and HTTPS authentication. After tracing the scanning, authentication and association process, once the MS gets connected, it will start tracing the activity of the mobile in a SQLite database. As illustrated in Figure 10, Wi2Me-User logs information about the usage of the running applications. For instance, it logs the number of bytes and packets transmitted and received (per application and for all applications together) over the WLAN interface. It can also log the evolution of the TCP connections by registering the state transitions for each single connection.

2.3.2.2 *Wi2Me-Research*

FUNCTIONALITIES Wi2Me-Research [36] is intended for participatory sensing, i.e., to set up measurement campaigns to discover and deeply analyze the performance of existing cellular networks and WLAN deployments by taking in consideration not only CN but any captive portal based network. Differently from Wi2Me-User, Wi2Me-Research allows a very fine-grained trace collection, at a higher sampling rate, giving to the application the full control of the wireless interfaces (i.e., it prevents other applications from making use of the interfaces) by using a firewall.

This application performs geo-location, network scanning, automatic connection and data traffic generation using the IEEE 802.11 and the cellular interfaces. It differs from other war-driving applications, since for the best of our knowledge, it is the first war-driving application that performs automatic connection and authentication to existent CN, supporting different CN providers and captive portal based networks. Wi2Me-Research also allows easily adding to the application captive-portal authentication algorithms (in JavaScript) for any existing network. The application runs in background mode and logs different events and information in a local SQLite database.

Three main modules define the application's operation, as shown in Figure 11. A first module obtains the MS position by using the GPS or network location services (based on GSM and WLAN positioning). A second module is

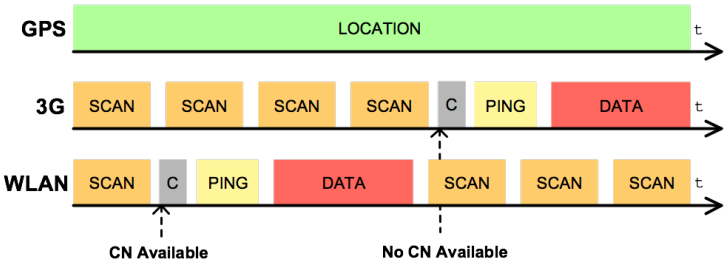


Figure 11: Wi2Me-Research modules

responsible for managing scanning, connection and traffic generation over 3G, while logging base station information. The third module is responsible for managing scanning, connection and traffic generation for WLAN. After a WLAN scanning, the application logs the list of discovered AP, including for each AP the Basic Service Set Identifier (BSSID), the SSID, the signal strength, the channel number and the supported security capabilities. If the user configures at least one CN account (i.e., username and password), Wi2Me-Research can select and connect to an AP and generate data traffic in order to evaluate the application level performance of the CN. Each time the application finds a known CN SSID having a signal strength greater than a threshold (by default, -85 dBm), it tries to associate with the AP and automatically performs HTTPS authentication using JavaScripts plugins we have developed to this end. Then the application may download and/or upload files of different sizes (50, 100, 250 and 500 KB) from a web server located at Telecom Bretagne. The WLAN connection lasts until the MS finishes transferring all the files or there is a disconnection due to poor radio link quality. If no CN SSID is available, the application attempts a cellular connection and performs the same file uploads and downloads. During each connection, the application logs the current signal strength, the data rate and the number of transmitted or received bytes.

As one of the goals in developing the Wi2Me application was to have a complete picture as possible of the network state and its performance, we have also monitored the network traffic on the server side. We use a Common Gateway Interface (CGI) script on the web-server that triggers the TCP connection tracing using the `ss` command (a socket statistics tool) to measure the TCP Congestion Window (CWND), the RTT and the throughput. Additionally, for each connection, all the packets are captured on the server side using `tcpdump`. After a measurement campaign the user can upload the traces (in SQLite) to a remote server via the application. Then, traces from different users and measurement campaigns are merged in a single SQL database. Finally we use Perl scripts to analyze the traces and compute relevant performance metrics. We have also used `tcptrace` to analyze individual TCP connections.

We have observed that for Wi2Me-Research, the intensive usage of the wireless interfaces and the GPS has a strong impact on the battery autonomy,

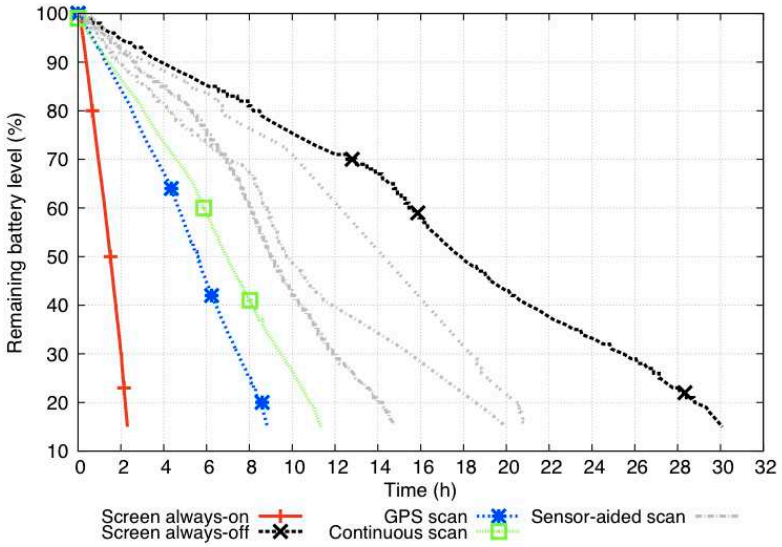


Figure 12: Battery drain different scanning strategies

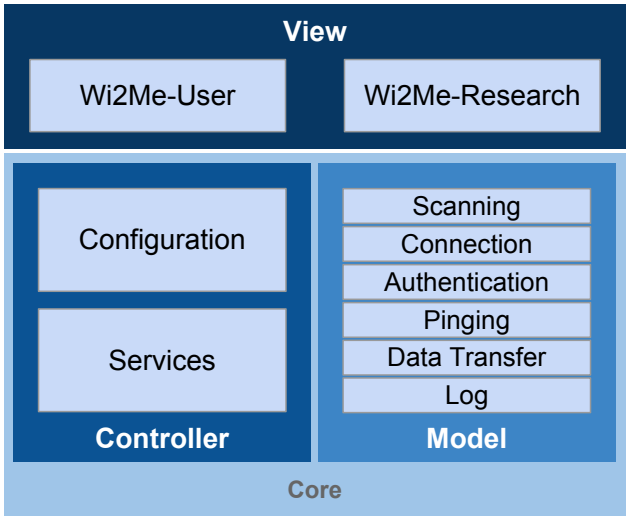


Figure 13: Wi2Me MVC model

especially due to WLAN scanning. For that reason, in order to prolong the battery lifetime, the application is paused if the MS stops moving during a certain time (by default, 60 s). In order to detect movement, we use the internal accelerometer, since it consumes much less energy. We illustrate the energy consumption of different scanning strategies in Figure 12, where the grey curves (sensor-aided scan) represent the battery drain (in %) for different runs of the application using the accelerometer-based movement detection. We observe that the usage of the accelerometer allows improving the battery duration between 3 h and 10 h (depending on user's mobility) compared to continuous scanning (green and blue curves).

CORE IMPLEMENTATION Wi2Me-Research was designed to be used by any contributor having an Android smartphone with GPS, WLAN and 3G inter-

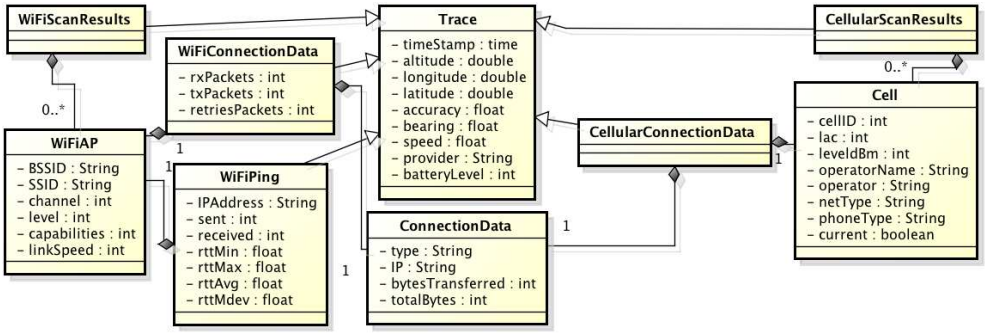


Figure 14: Traces database schema

faces. To guarantee scalability, we have used a Model View Controller (MVC) architectural pattern, as shown in Figure 13, that isolates the application logic (the core) and the user interfaces. The *View* can be a complete client or a simple experimental view. The application's *Controller* and *Model*, belonging to the Core module, contain the classes that collect and store the traces and run in background mode, allowing the user to continue using the smartphone as long as the other applications do not interact with the network interfaces. The *Controller* contains the configuration classes that allow selecting different parameters and options for the application. It also contains the *Services* package which includes all the classes running the main flows and providing interfaces to the device hardware, e.g., network interfaces, GPS, battery, sensors. The *Model* contains all the classes that are responsible for applying the control logic and policies, deciding when and how to perform a certain action, e.g., trigger a scanning or a connection, transfer data, log events. The model classes interact with the *Controller* services to transform policies in specific actions. This allows extending the application functionality by just implementing a new Model class or inheriting from the existing one without modifying the *Controller* services.

The application is completely parametrizable. Some of the possible configuration variables are: the time between successive WLAN scans (scanning interval), the signal strength threshold for attempting a connection, the delay to attempt a cellular connection after finding a new cell or the amount of data to transfer during a connection.

TRACES ORGANIZATION When running Wi2Me-Research, the device starts collecting information and organizes it in a database (see Figure 14). Every information collected represents a `Trace`, e.g., a scanning result, an association or connection event, an authentication event, a ping result or a connection information. A `Trace` is composed of a timestamp and the MS location information (altitude, latitude, longitude and speed). Location information can be provided by either the GPS module or by network interfaces, which give a low-precision location that is used when the GPS is unavailable. For each connection information (`WiFiConnectionData`), scan result (`WiFiScanResult`)

or ping command (`WiFiPing`), the application stores AP information in a `WiFiAP` trace, which contains the BSSID (layer-2 address of the AP), SSID, channel number, signal level (in dBm), the supported security protocols and link data-rate (in Mbps). For cellular-related information, the `Cell` trace stores the Cell Identifier (CID), the Location Area Code (LAC), the operator name and code, the signal level (in dBm) and the network type (GSM, GPRS, EDGE, UMTS or HSPA). Regarding the connection information, the application logs the connection type (download or upload), the IP address and the amount of transferred data for both WLAN and 3G connections. During a connection, a `ConnectionData` trace is logged every 50 ms. Additionally for WLAN connections, the `WiFiConnectionData` trace logs the number of received and transmitted layer-2 packets and the number of layer-2 retransmissions. All these traces are gathered and locally stored in a SQLite database on the smartphone. Unlike other war-driving applications that store the information in plain text files, we decided to organize traces in a database for scalability reasons and to simplify the post-processing phase. Once the application is stopped, it is possible to send the collected traces to our remote server via an FTP client integrated in the application. This feature allows aggregating traces from multiple devices in a central server for off-line processing.

2.4 CHARACTERIZING WIRELESS NETWORKS WITH WI2ME

2.4.1 *Experimental Setup*

The results presented in this section are based on the data obtained during two measurement campaigns using *Wi2Me-Research* on Samsung Galaxy S (GT-I9000) smartphones running Android 2.2.1. The first campaign [34], illustrated in Figure 15, consisted in a single user walking in the city center of Rennes, France[10] carrying two smartphones. One of the smartphones performed WLAN scanning and connected to CN to exchange data while the second smartphone was scanning every 2 seconds for IEEE 802.11 networks and receiving cellular beacons, while also performing cellular connections. The aggregated path length was 34 km and the total experimentation time was 10h19. We have gathered traces from 6761 unique APs and 61 unique cellular base stations. The *Wi2Me-Research* application was configured to alternate between downloading and uploading files of increasing size, between 50 KB and 500 KB.

We have also performed a second measurement campaign where we focused on the performance of TCP connections observed by a mobile user. In this case we performed multiple runs in urban areas by connecting to two CN (FreeWiFi and SFR), as well as several connections to a campus WLAN network at Telecom Bretagne, called SALSA, an open-system HTTPS captive-portal based indoor network covering the campus. We used Wi2Me-Research

10 A map with the estimated AP locations: `http://labo4g.enstb.fr/wi2me`

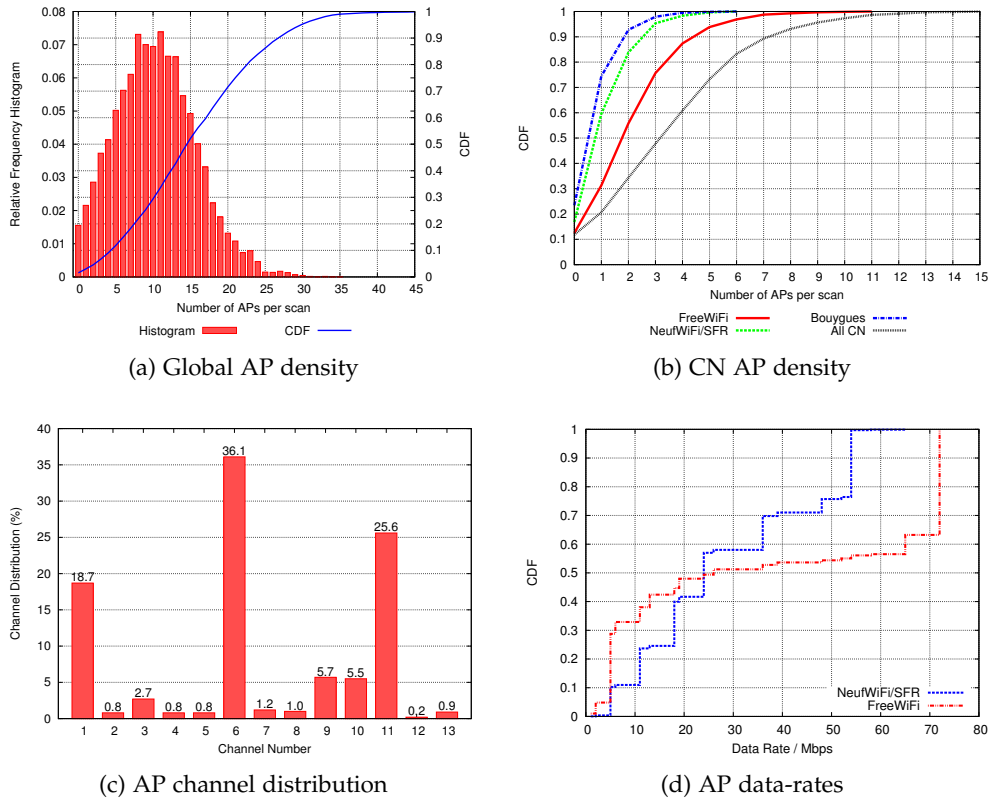
Figure 15: First measurement campaign

over the SALSA network in order to analyze the impact of handovers on TCP connections since, in CN, layer 3 mobility is not currently supported and so flows are interrupted after a handover. We performed 291 connections and we obtained traces from the MS and the server side. In both campaigns, an external 7 dBi antenna has been added to the smartphone to maximize AP discovery and CN connection success rate.

2.4.2 *Experimentation Results*

2.4.2.1 *Topology Discovery*

DENSITY During the first measurement campaign, we have measured the CN deployment density in terms of the number of discovered AP per scan. A scan is triggered every second while the MS is moving. In Figure 16a, all discovered WLAN AP are considered, resulting in a median density of 15 AP per scan, which indicates a very dense environment. The CN AP use different IEEE 802.11 physical layers. We observe in Figure 16d that in the case of FreeWiFi, the MS can reach data rates higher than 54 Mbps, indicating an IEEE 802.11n network. In the case of SFR, we can infer from the traces that the MS is connected to the AP in IEEE 802.11b/g mode. Figure 16b shows that FreeWiFi has a denser deployment compared to SFR and Bouygues. A FreeWiFi user may find more than one AP in 70 % of the scans. In the case of SFR, this value falls to 40 % and only 25 % for Bouygues. The number of AP discovered per scan by a user having access to all the CN has a median of 3.3, denoting fairly dense community networks.



(a) Global AP density

(b) CN AP density

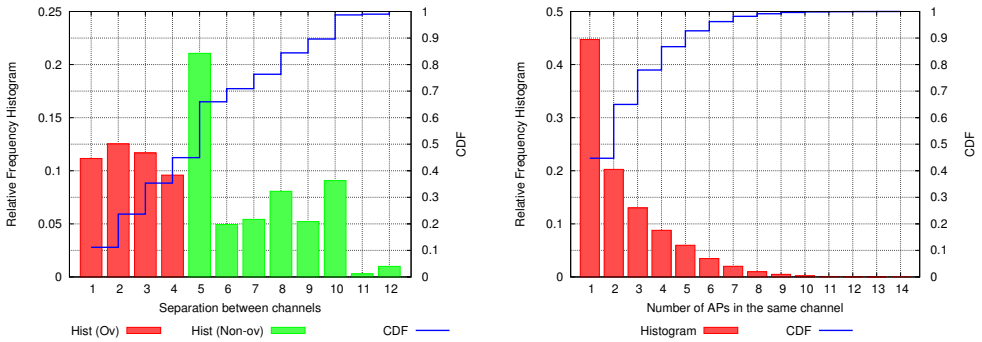
(c) AP channel distribution

(d) AP data-rates

Figure 16: Topology discovery metrics

CHANNEL OVERLAP We study the AP distribution on the different channels. This is a critical issue, since in IEEE 802.11 adjacent channels partially overlap, causing high levels of interference and frame loss [37]. In current WLAN deployments, there is a coexistence of different IEEE 802.11 physical layers that use various modulation schemes and channel bandwidth (as previously shown in Section 2.1.1). In all the cases, the channel separation is 5 MHz, but in IEEE 802.11b channels are 22 MHz wide while in IEEE 802.11g/n, channels typically are 20 MHz wide.

Figure 16c shows the distribution of the AP on different channels. Around 80 % of the total number of AP are deployed in the IEEE 802.11b non-overlapping channels (1, 6 and 11). This result confirms the channel distribution previously found in [38] and in our previous evaluation study [35]. To evaluate the level of overlapping, two metrics are proposed. Figure 17a shows the Cumulative Distribution Function (CDF) of the distance between the channels of all the AP observed in a single scan, called the *inter-channel overlap*. We found that 45 % of the times the APs overlap with other neighboring APs in the (up to four) adjacent channels, which may produce link degradation and packet loss due to partial overlap (the corresponding cases are shown in red). A four channel separation is needed, since we consider that there are IEEE 802.11b users, that require a 22 MHz wide channel. We observe in Figure 17a that the most likely separation is five channels, which is the case of



(a) Inter-channel overlap

(b) Intra-channel overlap

Figure 17: Channel overlap metrics

the classical non-overlapping deployment 1-6-11. Then, we define the *intra-channel overlap* as the number of APs operating in the same channel. Figure 17b shows that in the 55.3 % of the cases a given channel is shared by two or more APs. Even if most recent AP models implement an intelligent channel selection mechanism to find the most convenient channel, the high AP density, the uncontrolled deployment and the limited spectrum in the 2.4 GHz band generate an inevitable intra-channel overlapping. These interferences notwithstanding, CN are capable of supporting user communications at reasonable data rates, as illustrated in the following sections.

2.4.3 *End-User Experience in Community Networks*

2.4.3.1 *General Results*

Table 8 summarizes the results obtained in the first campaign, during which we have performed 661 connections to two CN (FreeWiFi and SFR) and 17 connections to cellular base stations belonging to a single operator (SFR). Such a difference between the number of connections to CN and cellular base stations is due to the *Wi2Me-Research* connectivity policy, which attempts a connection to a CN every time a CN AP having a signal strength greater than a threshold (by default, -85 dBm) is available. Given the high density of CN AP and the fact that the MS cannot simultaneously connect to both WLAN and cellular networks, the MS was connected to a CN AP most of the time. We downloaded 158 MB (in 1136 files) and uploaded 125 MB (in 900 files) using CN. We observe different connection success rates among CN operators. In the case of FreeWiFi, a connection attempt succeeds in almost 60 % of the cases (510 CN connections over 858 attempts) while a SFR connection succeeds only in 30 % of the attempts (151 CN connections over 504 attempts). However, SFR shows a slightly higher average throughput than FreeWiFi. As shown in Table 8, we have observed higher average and peak throughput in cellular than in CN connections. This may be due the low signal strength

Network	APs / Cells	Connection Attempts	L2/L3/CN Connections	Number of Files (Down/Up)	Tx/Rx MB (Down/Up)	Average Throughput (Down/Up) (KB/s)	Peak Throughput (Down/Up) (KB/s)
FreeWiFi	720	858	802/626/510	846 / 675	114.5 / 90.2	51.2 / 61.2	162.9 / 172.4
SFR	908	504	626/410 /151	290 / 225	43.5 / 35.1	65.5 / 41.2	367.9 / 126.5
Cellular	61	17	-	57 / 54	11.9 / 11.6	92.7 / 73.7	381.7 / 225.3

Table 8: Performance results for CN and cellular network connections

received from CN AP, as detailed in Section 2.4.3.3, and CN rate limitations imposed by ISP as discussed in Section 2.4.3.5.

2.4.3.2 *Complementarity between CN and 3G*

The MS activity during the first experiment campaign is shown in Figure 18. A cellular base station was available 99.2 % of the time, 45.5 % of the time the service offered was HSDPA (supporting downlink speeds between 3.6 and 14.4 Mbps) and a UMTS access (up to 384 Kbps on the downlink) for the rest of the time. Regarding WLAN, a CN AP with a signal level greater than −85 dBm was available 98.9 % of the time. This dense deployment allowed the MS to have a link layer connection to a CN for 93.9 % of the time.

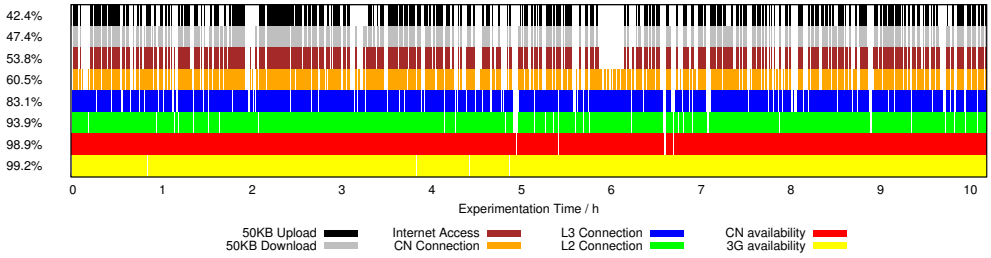


Figure 18: Smartphone activity during the first campaign

On the other hand, the MS has been successfully connected to the Internet through a CN (i.e., at least one packet was successfully received) only for 53.8 % of the time. This is mainly due to packet loss during Dynamic Host Configuration Protocol (DHCP) exchanges or to captive portal authentication problems due to poor signal strength.

2.4.3.3 *Signal Strength*

Figure 20 shows the signal strength distribution for the different CN operators. FreeWiFi and SFR have an equivalent signal strength distribution (in median, −80 dBm) while Bouygues has a weaker signal strength probably due to a smaller gain of the embedded AP antenna or lower transmit power. The low signal strength in CNs is related to the fact that residential APs are commonly deployed indoor, as their main goal is to cover an apartment or a house, reducing the power of the signals received outdoor. This issue is also raised in [39], where the authors analyze the impact of verticality of indoor AP in an outdoor usage. However, considering that a 7 dBi gain antenna has

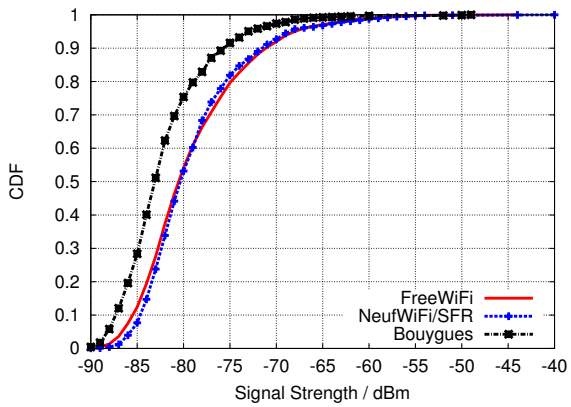
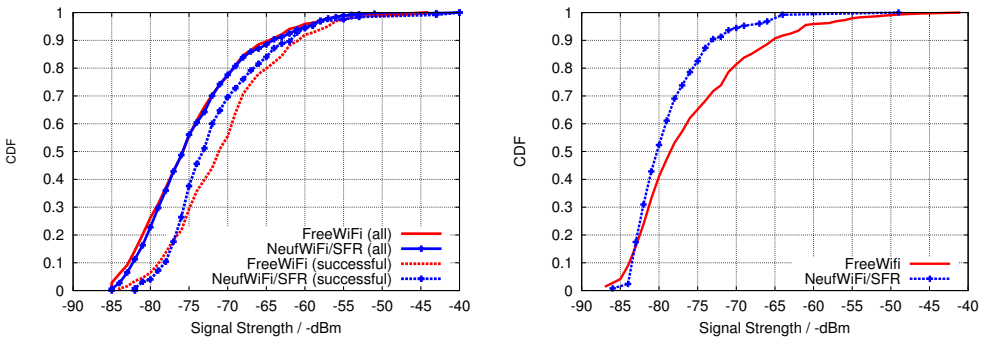


Figure 19: Signal strength distribution

been attached to the MS, a larger gain might be needed in both the MS and the AP in order to guarantee a good user experience when connecting to CN in a urban mobility use case. We have also investigated the distribution of the signal strength during connection and disconnection events. In Figure 20a, we can observe that the median signal strength during a connection attempt is around $-76\,\mathrm{dBm}$ for both FreeWiFi and SFR. However, successful connections (i.e., those with a successful CN authentication) require a higher signal strength, giving a median value of $-71\,\mathrm{dBm}$ for FreeWiFi and $-73\,\mathrm{dBm}$ for SFR. At disconnection, we obtained a median signal strength of $-78\,\mathrm{dBm}$ for FreeWiFi and $-80\,\mathrm{dBm}$ for SFR (see Figure 20b).



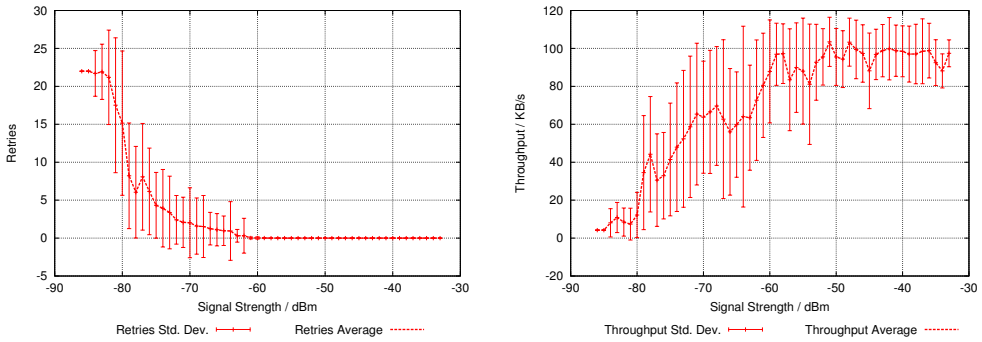
(a) Signal strength at connection

(b) Signal strength at disconnection

Figure 20: Signal strength distributions at connection and disconnection

In order to evaluate the signal strength impact on the download performance, we have calculated the average throughput and the average number of link layer retransmissions for different signal strength levels observed in all connections to FreeWiFi during the second measurement campaign. Figure 21a shows the average number of retransmissions per second for different signal strength levels. We observe that there is no retransmission for signal strength levels greater than $-60\,\mathrm{dBm}$; for lower levels, the link quality degrades, forcing the MS to retransmit packets. Figure 21b shows the relation

between the signal strength and the average throughput. We observe that for stronger signal strengths, the MS reaches the maximum average throughput provided by the CN (around 100 KB/s). We can also observe that for low signal strength levels, between −65 dBm and −80 dBm the throughput is extremely variable, resulting in unstable performance. Observe in Figure 21b that the maximum achievable throughput is reached for signal strength values greater than −60 dBm. However, such a level of signal strength has been observed during less than 2 % of the time, leading to fairly low average data rates in our experimentation.



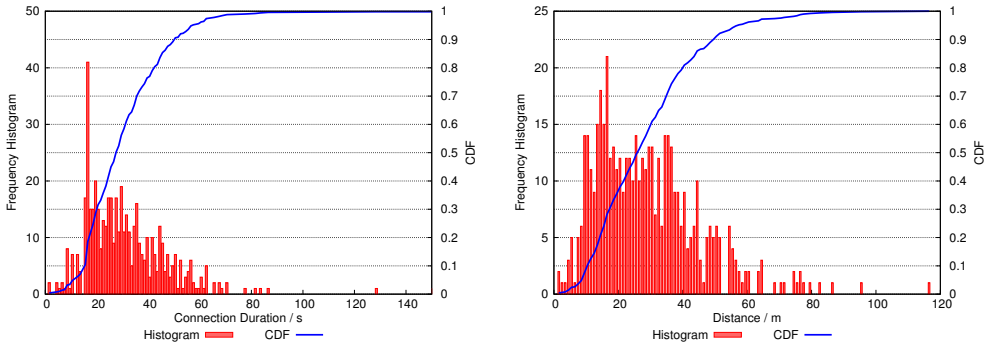
(a) Retries vs Signal Strength

(b) Throughput vs Signal Strength

Figure 21: Signal strength correlations

2.4.3.4 *Connection and Disconnection Periods*

We analyzed the connection duration and the distance covered by the user while being connected to a CN AP. In our campaigns, the MS moves at a walking speed (roughly 1 m/s). In Figure 23 we plot the relation between the distance covered and the connection duration. Most of the points cluster around the dashed reference line which corresponds to 1 m/s. Figures 22a and 22b show the distribution of the covered distance and the duration of a connection respectively. We observe a median connection duration of 27.5 s corresponding to a displacement of 26 m. This result doubles the connection time observed in [2]. We observe that the IP disconnection time, i.e., the time between a disconnection and a new connection to a CN, has a median of 5 s. More precisely, the time needed to establish a layer-2 connection has a median of 2 s and the time to have a functioning Internet connection after a successful layer-2 connection has a median of 3 s. Even if these disconnection times dramatically perturb real-time and interactive applications, they may not greatly affect applications like email, micro-blogging or browsing, that may tolerate a higher latency. Compared to results presented in [2], thanks to the high density of CN, permanent connectivity could be feasible because most of the time a CN AP is available. So instead of having the 23 s between two connections calculated in [2], we observe a disconnection time of 5 s that is only due to protocol latencies and not to the network deployment.



(a) Connection duration

(b) Connection distance

Figure 22: Connection and disconnection metrics

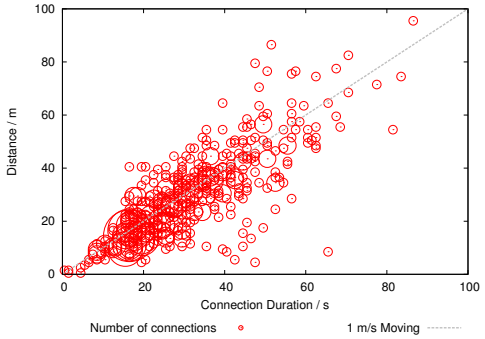


Figure 23: Duration-distance correlation

Note that these 5 s could be significantly reduced by carefully tuning the parameters used in the protocol implementations (e.g., better triggering the link going down event).

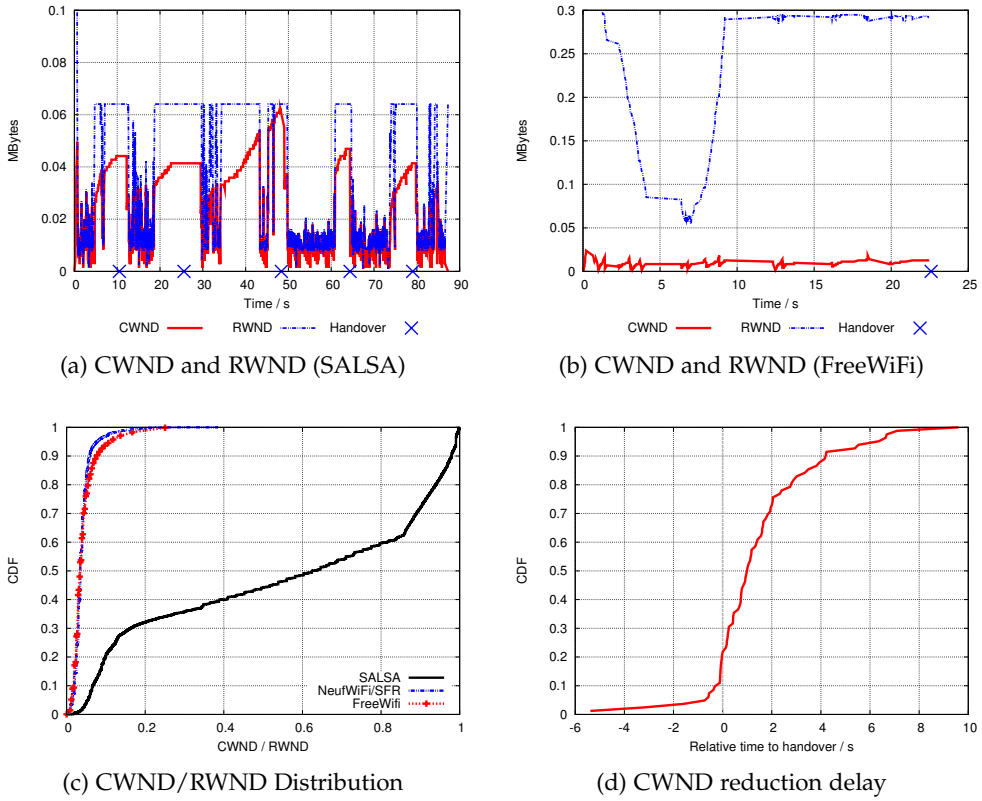
Regarding session continuity, we observed in all CN that the connections are interrupted when the MS handovers to a new AP. In the case of FreeWiFi, once associated to an AP, the MS obtains a public IP address which is leased for 130 s, but even if the same IP address is used in the new AP, no communication is possible. This could be caused by a missing routing update causing IP packets to be routed to the wrong AP (recall that in CN, AP are routers as well). In the SFR CN, the MS is behind a NAT and acquires a private IP address using DHCP. Then, after a handover, the IP address is no longer valid in the new AP and a complete reconfiguration is needed. In the case of the university campus network, called SALSA, all the AP belong to the same subnet, thus an MS is able to pursue its communication after a transition between two AP. In this case, we observe an uninterrupted download, even if the bandwidth is highly degraded when a handover occurs.

2.4.3.5 *Transport Layer Aspects*

Thanks to the traces collected on the server side, which used TCP Reno, we can reconstruct individual connections with a good level of detail. We use the

`tcptrace` utility which can compute, among other parameters, an estimate of the CWND, the value of the TCP Receiver Window (RWND), an estimate of the RTT and of the TCP throughput. Figures 24a and 24b illustrate the evolution of CWND and RWND for one SALSA and one FreeWiFi connection respectively, where the MS was downloading the file and it was, therefore, the TCP receiver. Note that in the case of the FreeWiFi connection, RWND is never a limiting factor as it is always much larger than CWND. Instead, in the case of the SALSA network (Figure 24a), at times the two windows are identical, meaning that the application layer throughput is being limited by the receiver. As we have observed this in multiple connections, we computed the distribution of the ratio CWND/RWND for FreeWiFi, SFR and SALSA (Figure 24c). As CWND is always lower than RWND the ratio falls in the interval $(0, 1]$. We observe that in CN, in almost all the cases, the ratio between the two windows is lower than 0.2, meaning that RWND is at least five times CWND. For the SALSA connections, instead, RWND is at most $1.25 \times$ CWND in the 40 % of the cases. This confirms the fact that Figures 24a and 24b are representative of our sample.

As far as handovers are concerned, we have observed that no TCP connection was able to continue after a handover in CN. Even in the case of FreeWiFi, which (at least in some cases) uses the same public IP address before and after a handover, there is no traffic on the server side after the handover. As one can observe in Figure 25c the same is true on the MS side, even though the received signal strength (Figure 25a) is high. One possible explanation is that the IP packets are still routed to the previous AP, preventing the TCP connection from working after the handover. In the SALSA network, instead, TCP connections do keep working after a handover but, in most cases, the TCP sender (i.e., the server in our case) experiences a timeout after a handover, leading to an important reduction in the congestion window. Figure 24a shows a typical evolution of the congestion window for a TCP connection of a mobile user in the SALSA network. To quantify this behavior, we have calculated the relative delay between the handover and the reduction of the CWND to at least half its maximum value before the handover (see Figure 24d). In some cases (less than 20%) the dynamics of the TCP connection were such that the congestion window was cut shortly *before* or about the same time as the handover but in 70% of the cases the window was reduced during the 4 s following a handover. Given the small RTT of the TCP connections in the SALSA data set, these timeouts and window reductions do not affect significantly the TCP throughput but such a behavior could decrease the performance in the case of a wide area network where RTT, and competing traffic can be significantly larger. Finally, it is also interesting to observe that, as shown in Figure 24a, the aforementioned problems with the RWND limiting the CWND at times, happen right after a handover. One possible explanation is that the algorithm used by the MS to compute the RWND is confused by the timeout and the corresponding in-



(a) CWND and RWND (SALSA)

(b) CWND and RWND (FreeWiFi)

(c) CWND/RWND Distribution

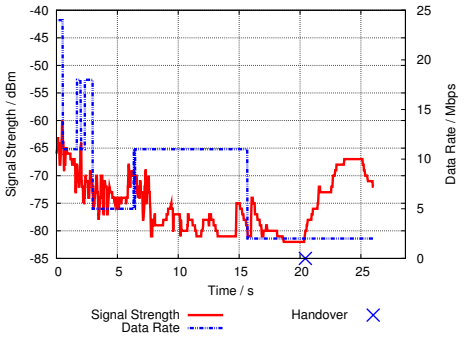
(d) CWND reduction delay

Figure 24: CWND and RWND metrics

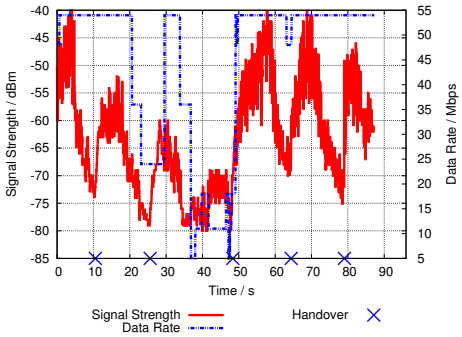
crease in the RTT, leading to a much smaller RWND. All this shows how, even if the application-level performance is acceptable in the case of the SALSA network, this could not be the case for a wide area network.

2.5 MOBILITY ISSUES IN COMMUNITY NETWORKS

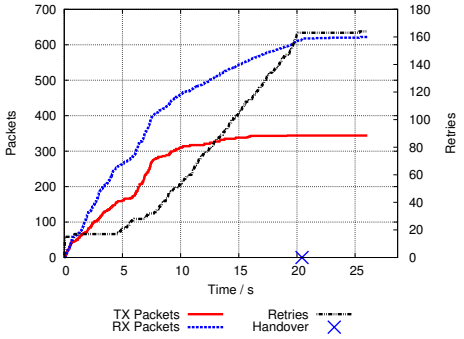
During our measurement campaigns, we found a high density of WLAN AP along our itinerary. One could deduce that there is a potential for a low cost and high data rate Internet access in urban areas. However, we have observed a low received signal strength most of the time (even if an external antenna has been used); 90 % of the measured signal strength samples are lower than -70 dBm, and 50 % are lower than -80 dBm. This results in short connection times, between 10 and 40 seconds, when a user is moving at 1 m/s (see Figure 23). This high density also makes the disconnection time (with a median of 5 s) to be only dependent on the protocol latencies since when a handover occurs there is always a candidate CN AP. Thus, while CN are currently used mainly by static users, they have the potential to offer a reliable Internet connectivity to mobile users, provided optimized mobility protocols are deployed to support handover between AP. In this way, once the current connection with an AP is lost, the MS can switch to another AP while maintaining its communication. In the following sections, we study



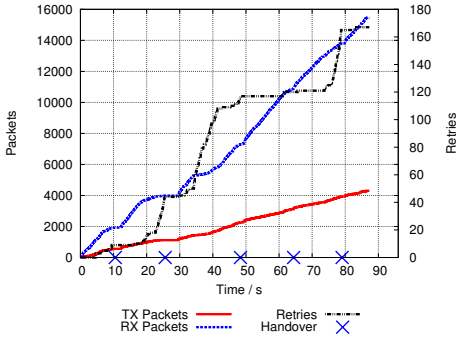
(a) FreeWifi: Power and data-rate



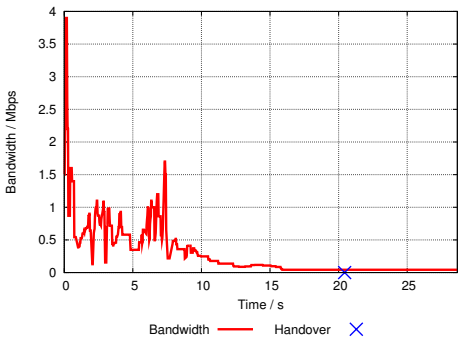
(b) SALSA: Power and data-rate



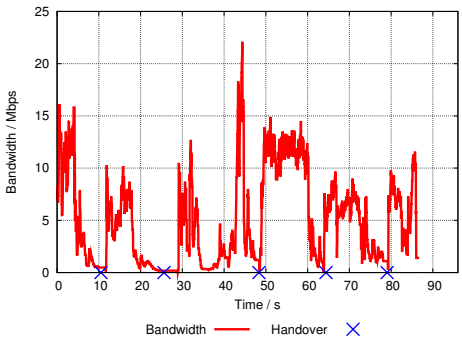
(c) FreeWifi: Packets and retries



(d) SALSA: Packets and retries



(e) FreeWifi: Throughput



(f) SALSA: Throughput

Figure 25: Performance metrics for CN and SALSA

the impact of handover on the applications and explain what could be done to provide mobility support in CN.

2.5.1 *Handover impact*

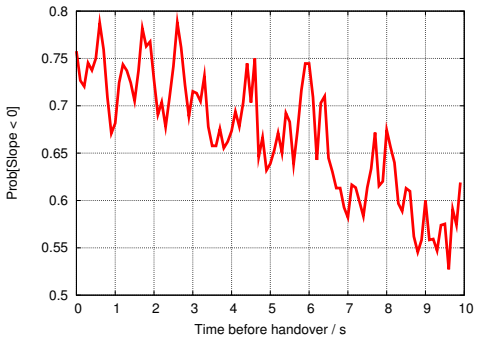
A handover, which is deeply studied in Chapter 3, is the process performed by an MS to change AP. It may be triggered by a low signal strength from the current AP or bad link layer performance. A handover consists in discovering potential APs operating in the MS radio range and then authenticating and associating with one of them. Then, IP layer operations may be needed if the new AP requires that the MS acquires a new IP address.

Authentication after a handover is one of the critical aspects in CN. In some cases, when an MS disconnects from a CN AP and re-associates to a new AP of the same CN, HTTPS authentication through a captive portal has to be carried out again. It means that a user typically enters his login and password at each connection with an AP. In our application, no interaction is needed with the user, scripts perform the authentication on the user behalf. However, it still takes between 1 and 2 seconds, which is unacceptable for several applications.

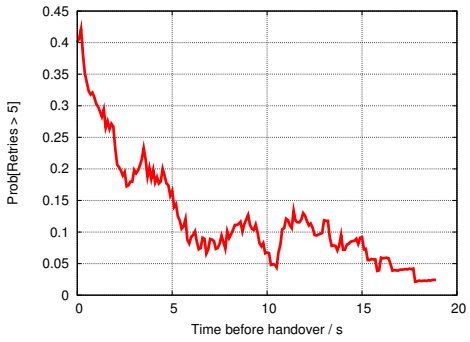
In order to further analyze the handover, we propose to study the impact of handover on a data transfer by making a comparative analysis between CN and the SALSA network, where only a layer 2 handover is needed (all AP are on the same subnet). Figure 25 shows the signal strength, the total number of packets transmitted (TX) and received (RX) as well as the link layer retransmissions (retries) and the bandwidth (on the server side) for two individual connections to FreeWiFi and SALSA. We have observed that, in the case of CN, when the MS reaches the limit of its AP coverage area, the MS automatically switches to a new AP. However, in all the cases, we have observed that the download is interrupted as shown in Figure 25c. After a handover (marked with a blue cross), even if a good signal strength is received from the new AP (see Figure 25a), the RX packets curve from Figure 25c stops increasing. On the other hand, we observe exactly the contrary in SALSA, where the MS can always continue the download after a handover. Figure 25b shows the data rate and power after handovers (indicated by a blue cross) while Figure 25d shows that the MS continuously receives and sends traffic. Figures 25d and 25f also show the number of retries and throughput. We can see that before each handover, the number of retries increases and the throughput decreases. We also observed that, in some cases, the throughput is low for some time after the handover, before the MS can fully exploit the capacity of its new connection. This is the time needed by the transport layer to adjust its transmission rate to the new capacity of the network.

2.5.2 *Predicting a handover*

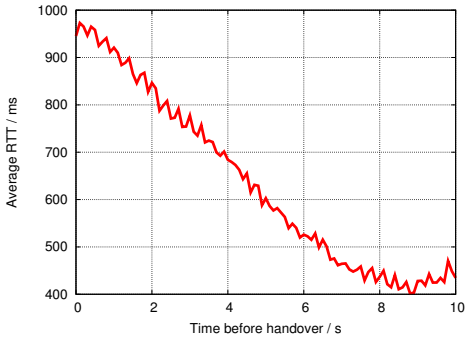
Two optimizations can be set up to reduce the handover impact on applications. First, the handover process can be enhanced, by proposing mechanisms at different layers to either better manage the handover or limit the impact on transport protocols. Second, it might be possible, at least in some cases, to use a (short-term) prediction mechanism to forecast impending disconnections in order to execute a handover before the link conditions are significantly degraded. We have observed that, in most cases, shortly before an MS is disconnected from the current AP, the signal strength and the throughput are both decreasing, while the number of link-layer retransmissions in-



(a) Power Slope Probability



(b) Retries Probability



(c) Average RTT

Figure 26: Signal strength, retries and RTT before a handover

creases significantly as shown in both FreeWiFi and SALSA connections in Figure 25.

In order to better characterize and therefore better detect an impending handover, we have computed three parameters. First, in Figure 26a, we show the percentage of cases where the short-term trend of the signal strength (slope) is negative as a function of the time before the handover. When the handover approaches, around 80 % of the times there is a degradation of the signal strength. Figure 26b the percentage of cases where there are more than 5 retries per second. We observe that 5 s before the handover occurs, a high level of retries becomes more probable. Finally, in Figure 26c we consider the average RTT measured on the server before the handover. We observe a linear increase of the RTT when a handover approaches, reaching almost 1 s.

In order to anticipate handover decisions, we could monitor the signal strength and the number of retransmissions. While link-layer retransmissions can be measured only when the MS is sending data, this is exactly the case where handover anticipation is most important.

2.5.3 *Mobility Support in upper layers*

If CN operators want to offer seamless connectivity, they need to deploy mechanisms to quickly and efficiently handle frequent handovers at layers

2, 3 and 4. First, in order to improve the radio link quality and so reduce the handover frequency, CN AP may replace common internal antennas, that have become popular due the miniaturization of devices, with high-gain external antennas. This could reduce the handover frequency and give more time to the MS to anticipate and prepare the handover. Second, layer 3 mechanisms are needed to allow users to keep continuous IP-layer connectivity while moving. Operators may configure CN AP in a single IP subnet or in a set of geographically distributed subnets (e.g., one for each neighborhood). However, this solution may not scale to the large number of current AP (i.e., several millions). Another approach is to introduce Mobile IP [40] in the network architecture. In this case, each operator may deploy a Home Agent, which allows the MS to continue receiving flows while moving from one network to another. Moreover, a single Home Agent for several CN operators having a roaming agreement could allow mobile users to roam between APs belonging to different communities. Nevertheless, a negative handover impact at the transport layer (i.e., layer 4) is observed for both CN and SALSA. In the case of CN, the CWND can never recover due to a network disconnection. In the SALSA network, we observed that the throughput is drastically reduced just after and before the handover due to the combination of poor signal strength and the reduction and later recovery of the CWND. To overcome the impact of handover at layer 4, some solutions already exists, like FreezeTCP [41], which can prevent the CWND from dropping when a handover occurs, provided the appropriate control message can be sent at least one RTT before the handover takes place.

2.6 POSSIBLE EVOLUTIONS IN COMMUNITY NETWORKS

2.6.1 *Managing and Controlling Deployments*

The fact that ADSL subscribers can easily share their WLAN access allows, on the one hand, proposing a simple model to set up a dense network, but, on the other hand, the AP coverage and capacity planning are uncontrolled. Users can place AP and manually set the operating frequency (channel) and the modulation scheme (IEEE 802.11b/ g/n and different data-rates). Users can also turn-off and on the AP whenever they want, resulting in a completely dynamic topology. This uncontrolled deployment leads to high level of interference, limited per AP coverage and relative low network performance as also studied in [39].

A potential solution would be for CN operators to take control of the radio configuration on behalf of their subscribers. First, they could automatically select the AP channel depending on the environment (i.e., neighboring APs and other devices operating in the ISM bands). This is currently implemented in several AP, but since it is not mandatory, users may manually set the channel without having any knowledge of surrounding sources of interference.

Second, operators could dynamically modify the CN AP deployment by remotely regulating the AP output power (or directly turning them off). This is similar to cell-breathing techniques in cellular networks. Some cell breathing algorithms for WLAN have been proposed in [42]. Using such a mechanism, operators may adapt the deployment to the current users needs, possibly leading to a more energy efficient network as a positive side effect.

2.6.2 *Using multiple network interfaces*

We have observed in Figure 18 that in urban areas, the density of wireless community deployments is such that at least one CN AP having a signal strength greater than -85 dBm is available 98.9 % of the time, which is equivalent to the availability of cellular base stations (99.2 %). This should allow a user to use both interfaces at the same time instead of alternating between them. Existing mobile devices use a pre-established connectivity policy (embedded in the OS) dictating the use of a single interface at a time for all flows. On the contrary, if using multiple interfaces simultaneously were enabled, users could implement smart decision-making schemes. In Chapter 4, we survey the decision-making in network selection and propose a multi-objective decision-making problem that we solve using genetic algorithms.

Recent studies try to optimize this deterministic interface selection in order to offload cellular data to WLAN networks [30]. Offloading strategies focus on reducing the cellular overload by delaying data transmissions until the user enters in a WLAN covered zone. As we have shown in the proposed measurement study, in a urban environment, WLAN deployments appear to be highly ubiquitous, therefore a user may have the opportunity to offload a significant amount of traffic using CN or even to enhance his experience by simultaneously using both access technologies.

2.7 CONCLUDING REMARKS

In this chapter, we have characterized CN using the *Wi2Me-Research* application. This application was designed to periodically scan the radio environment, connect to IEEE 802.11 AP and cellular base stations and evaluate the application layer performance. CN are created by millions of users sharing their residential AP with other subscribers. Particularly in urban areas, CN have a high AP density, albeit coupled with an uncontrolled deployment. During a CN connection, that we have measured to last between 10 and 40 seconds, we observed different values of throughput for the same signal strength and vice versa. So that signal strength alone can hardly determine the application performance. We have also shown that handovers are not supported in CN, because after changing AP, the application flows are not redirected to the new AP. In a controlled IEEE 802.11 deployment, such as a campus network, a user may change AP without interrupting its commu-

nication, if handovers are supported at layer 2 and 3. But even in this case, the handover is not transparent for TCP as, in most cases, the congestion window is significantly reduced for up to 6 seconds after a handover. Furthermore we observed that sometimes after a handover, the TCP receiver is still limiting the sender window size by announcing a low RWND.

We have shown that mobility support is required in CN because, due to the limited connection time, handovers are frequent. We also observed that the used protocols to manage handovers are not optimized: the time needed to reconnect a new CN AP has a median of 5 s. To reduce the contribution of the authentication process on the handover delay, some operators have started to deploy Extensible Authentication Protocol Method for GSM Subscriber Identity Module (EAP-SIM) [43], aiming at automatically authenticating WLAN-enabled smartphones using the GSM Subscriber Identity Modules (SIM) instead of using captive portal based authentication. However, in order to still reduce the handover delay, the MS may reduce the AP discovery latency by optimizing the scanning algorithm. We address in Chapter 3 the handover management, particularly the AP discovery process in IEEE 802.11.

3

HANDOVER OPTIMIZATIONS AND ACCESS POINT DISCOVERY IN IEEE 802.11

3.1 INTRODUCTION

The usage of wireless networks has incredibly evolved during the last years. Current mobile users have the possibility to connect to different network technologies using multi-interface devices. In a typical use case, mobile users access the Internet through a 2G/3G/4G cellular network or through an IEEE 802.11 AP. Particularly for IEEE 802.11, there has been a proliferation of hot-spots and community networks, providing a high throughput Internet access in urban environments. However, a single AP covers very short ranges, limiting the users' mobility.

In Chapter 2, we have shown that community networks appear to be highly dense in urban areas, generally providing several access points (15 in median) per location. Under these conditions, a mobile user may be able to connect to community networks and compensate the low AP coverage area by transiting across AP. Such AP transition is called a *handover*. However, as stated in the previous chapter, two main issues currently limit users from moving across urban IEEE 802.11 and performing seamless handovers. First, in current hot-spots and CN deployments there is a lack of mobility support at the IP layer that guarantees session continuity of users' applications. To overcome to this drawback, the network operators may deploy existing IP mobility support protocols, like MobileIP [40]. However, even if a handover is managed at the IP layer, the MS has still to manage the disconnection from the current AP and the transition to a new one at the MAC layer, i.e., the layer-2 handover. Nowadays, existing mechanisms to support layer-2 handovers lead to long handover delays, which strongly impacts the mobile users experience. In current MS, when a handover occurs, a degradation of the on-going flows is observed, corresponding to a dramatic reduction of the TCP CWND and the throughput.

In this chapter, the handover process at the layer-2 and, in particular, the AP discovery are investigated. We analyze the AP discovery process and identify the main trade-offs while designing a scanning algorithm. We highlight the importance of adapting the scanning parameters in order to provide low latency scanning phases while discovering the maximum number of AP. In this context, we propose two adaptive scanning algorithms, Adaptive Discovery Algorithm (ADA) [44] and Cross-Layer Adaptive Scanning [45], which aim at dynamically setting the scanning parameters for each particular scenario in order to better manage the performance trade-off.

The chapter is organized as follows. In Section 3.2, we describe the handover process in IEEE 802.11 and provide, in Section 3.3, a set of preliminary results that illustrate the impact of scanning on on-going communications. Then, in Section 3.4, a description of the related work on handover optimization is presented. In Section 3.5, we expose the limitations of current scanning optimizations and present the motivations for an adaptive approach while performing scanning. The contribution of this chapter is divided in two parts. First, in Section 3.6, ADA is proposed with the goal of optimizing the AP scanning performance in terms of its latency and its success rate. Then, a second adaptation algorithm for IEEE 802.11 scanning is proposed in Section 3.7, based on cross-layer communication between the physical and the MAC layer. Both adaptation algorithms are implemented in open-source drivers and evaluated under experimentation. Finally in Section 3.8 we conclude the chapter.

3.2 THE IEEE 802.11 DISCOVERY PROCESS

The IEEE 802.11 standard [8] has been designed to provide a high bandwidth wireless access for short coverage areas, limited to some tenths of meters. In order to provide mobility over IEEE 802.11, the standard has defined a set of mechanisms to manage handovers, i.e., the MS transition across different AP in the topology.

When defining handovers, a distinction has to be done depending on the impact of handover on the communication layers. A layer-2 handover implies the modification of the point of attachment in the network, while in a layer-3 handover, the mobile user also get connected to a new access network (i.e., packets needs to be transmitted over a different access router after the handover). That is, for AP deployed in the same IP network, the MS performs a layer-2 handover. For AP deployed in different networks, such as the case of different CN AP in a urban environment, the MS has to perform first a layer-2 handover and then a layer-3 handover in order to redirect the flows through the new network.

Particularly in layer-2 handovers, the process can be divided in different phases:

1. **Triggering:** The MS monitors the current connection and decides when to declare the link going down and start the handover.

2. **Scanning**: In this phase the MS discovers candidate AP in its wireless range.

3. **Best AP Selection**: A single AP is chosen to attempt connection

4. **Authentication**: The MS identity is verified by using one authentication method: Open System authentication, Shared Key authentication,

Fast Transition (FT) authentication or Simultaneous Authentication of Equals (SAE)

5. **Association**: The association procedure grants the MS with a full access to the DS through the AP the MS selected in order to communicate with other devices in the wireless and wired network.

Regarding the scanning process, the standard proposes two different scanning algorithms namely *passive* and *active* scanning. In passive scanning, the MS simply tunes its radio on each channel and listens for periodic beacons sent by different AP. In active scanning, the MS proactively sends requests on each channel as illustrated in Figure 27. In a given channel, the MS sends a Probe Request management frame and waits for Probe Responses from the AP. Since Probe Request frames are not acknowledged at layer-2, the MS needs to wait for a pre-established amount of time to get responses from AP. If at least one Probe Response is received before the expiration of a first timer, called MinChannelTime (MinCT), the MS waits until the expiration of a longer timer, called MaxChannelTime (MaxCT), in order to gather Probe Responses from additional AP on the same channel. If no Probe Response has been received when MinCT expires, the MS declares the channel empty, switches to another channel and starts over the process. Once candidate APs have been found, the MS selects the best AP and attempts the authentication and association processes.

To completely scan all channels using passive scanning, the MS may switch to every channel and remain listening for beacons for at least one beacon period (by default, 100 ms), giving a scanning latency of around one second. Using active scanning, the MS can look for AP in a proactive manner, reducing the time spent on each channel to just the time required to receive Probe Responses. For active scanning, the standard does not provide any values for MinCT and MaxCT but it only imposes the condition that MinCT $\leqslant$ MaxCT. Existing IEEE 802.11 drivers implement scanning algorithms which usually last between 100 ms and 800 ms.

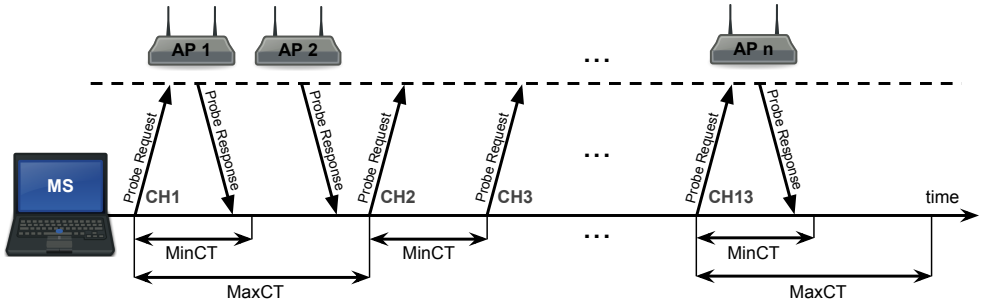


Figure 27: IEEE 802.11 active scanning

In the following, we present a set of experimentations aiming to identify the impact of the layer-2 handover process on the on-going data communications.

Device	OS Version	Chipset	Avg. Thr Static (MB/s)	Avg. Thr Mobile (MB/s)	Num. of hdv.	Avg. RTT (ms)	σRTT (ms)
Asus N10J	Win XP SP2	AR5006	5.19	0.878	4	103	43
Asus N10J	Ubuntu 10.04	AR5006	4.88	0.601	2	161	360
Nexus S	Android 4.0.3	BCM4329	3.80	0.568	5	129	114
MacBook	MAC OS 10.7.4	BCM4322	8.44	0.613	3	167	276

Table 9: Handover performance of different OS

3.3 HANDOVER IMPACT ON DATA COMMUNICATIONS

During a layer-2 handover, the MS is not able to send or receive application flows. This is because, usually, when an MS triggers a handover, the link quality does not allow exchanging frames anymore, and because the MS is often switching channel to discover APs. In this section, the handover and, in particular, the scanning impact on applications flows is evaluated.

OPERATING SYSTEMS BENCHMARK To illustrate how the handover impacts data flows, a set of experiments to evaluate the degradation of TCP performance for different devices and Operative Systems (OS) has been performed. To this end, we have generated TCP connections between a server and the different devices and we have gathered all the packets on the server side using `tcpdump`. Then, the performance of the TCP connections has been analyzed using `tcptrace`. Table 9 shows the number of handovers and the average TCP throughput that has been observed for the same path and same speed using different devices and OS. As a baseline, the maximum achieved throughput is showed for each device remaining static and connected to a single AP. The best result is observed using Windows, since the MS performs up to four handovers, reaching an average throughput of 0.878 MB/s. Additionally for Windows, the time in which no data is downloaded (i.e., the disconnected time) is relatively short compared to the other OS. The Asus netbook running Ubuntu reacts slowly to variations on the channel conditions. In this case the MS remains disconnected for more than 20 s and executes only two handovers. This indicates that the MS waits until the quality of the radio link is significantly degraded to perform handover. Figure28 shows the evolution of the downloaded data for each case. Additionally, it has been observed that for the Windows device, the average RTT is the lowest one (103 ms) having also a low standard deviation. This differs from the other devices which reach higher and more unstableRTT.

SCANNING IMPACT Several studies in the literature [46] [47] has shown that, during a handover, scanning is the most time-consuming phase, which may represent up to 90 % of the handover duration, depending on the authentication method. During a scanning, the MS switches to different channels and so is not able to send or receive packets. An MS may use the PSM defined in IEEE 802.11 to request its current AP to buffer incoming packets

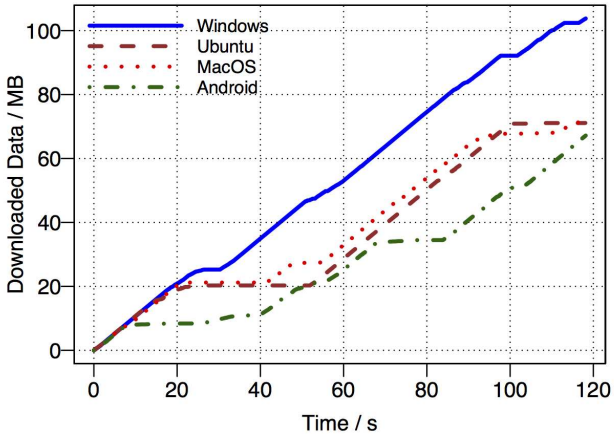


Figure 28: Downloaded data for different OS

during the scanning. This way, instead of loosing packets during the scanning phase, an MS can receive the packets after the scanning phase, albeit with an extra delay

An illustration of the scanning impact can be observed in Figure 29, which shows the evolution of the amount of downloaded data for a TCP flow that we have monitored using `tcpdump`. The MS is an Android smartphone (Samsung GT I9023) and the TCP flow was generated using iperf. We have configured the MS to periodically perform active scanning on the 13 available channels with $MinCT = 50\,ms$ (and $MaxCT = MinCT$). We observe that at time 0.6 s the throughput slows down since the MS requests to enter in PSM. Then, no packets are received between time 0.8 s and 1.5 s. Once the scanning is finished, the MS comes back to its current AP, and starts receiving TCP packets again. Finally, the complete interruption of the flow (i.e., just before the MS request to enter in PSM and the recover of the reception of new TCP packets) lasts for around 0.9 s. Note that in this case, the station comes back to the original channel, so no additional authentication or association delays have to be considered. This level of latency greatly affects the user experience, specially in applications like VoIP, online gaming or live streaming.

3.4 HANDOVER OPTIMIZATIONS AND RELATED WORK

Most of the related work done for the IEEE 802.11 handover process concerns the optimization of the layer-2 handover, when an MS roams from one AP to another. In this section, we present the main strategies to reduce the scanning latency. The objective of any handover optimization focusing on scanning is to reduce the latency, i.e., the time spent to discover new candidate AP, while maximizing the number of discovered AP. In the following, we list the most relevant scanning optimization techniques, that mainly focus on providing fast handovers.

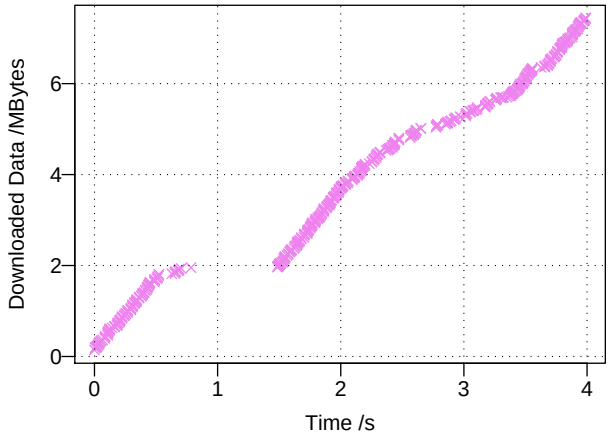


Figure 29: Communication disruption in active scanning

3.4.1 *Selective Scanning*

The most common manner to perform scanning in IEEE 802.11, as depicted in Figure 27, involves probing all the available channels. We refer to this as a full scanning. Evidently, the scanning latency is directly proportional to the number of channels to scan. As stated in Chapter 2, in IEEE 802.11, the number of available channels depends on the particular physical layer and the regulatory domain. In the most common IEEE 802.11b and IEEE 802.11g deployments, 11 (in USA), 13 (in Europe) and 14 (in Japan) channels are available in the 2.4 GHz ISM band.

One simple way to reduce the full scanning latency is to only scan a subset of channels from the complete list of available channels. Shin et al. [48] suggest the utilization of a channel binary mask to select which channels to scan. This mask is updated after each handover. During the first handover, the mask is initialized with `1` for all channels, meaning that all channels are scanned. Then, for the next handover, the MS builds a new mask containing a value of `1` for the non overlapping channels (1, 6 and 11) and for those channels where a Probe Response has been received in the previous scanning. The mask contains `0` for channels where no Probe Response has been received. The channel in which the MS's AP was previously operating is turned to `0` in the mask, since authors consider that a neighboring AP operating on the same channel is not probable. This consideration contradicts the statement presented in [49], where a neighboring AP in the same channel is considered highly probable. Moreover, we have shown in the evaluation study in Chapter 2 that in 55.3% of the cases a single channel is shared by two or more AP. When a new scanning has to be performed, channels marked as `1` in the binary mask are scanned, if no Probe Responses is received on those channels, the mask values are logically inverted and the MS continues probing these new channels. If the scanning process is still unsuccessful, a standard full scanning process is executed all over again (i.e., to scan the complete list of channels). In addition, the authors propose to use a Caching

method where neighbor AP information is stored in a table during the MS operation. This table will allow the MS to directly probe (using a Probe Request) a neighbor AP when the MS returns to an AP that has already been visited. Selective Scanning reduces the full scanning latency in average 43 %. Applying the caching mechanism, the layer-2 handover latency is reduced to reauthentication and reassociation delays, which are negligible compared to the scanning latency in open system authentication networks. Regardless of these results, even if selective scanning may be easily implemented in an MS without introducing modifications in the AP side, the neighbor AP table has to be carefully maintained. Erroneous information in the caching table, such as unavailable AP that have been previously discovered, would leads to handover failure (i.e., the impossibility to reassociate to the candidate AP). On the other hand, as both the cache and the binary mask are incrementally built (i.e., as the MS moves across AP), the first handovers will use the standard technique, i.e., scanning the whole list of available channels, resulting in higher latencies.

3.4.2 *Reduced Scanning Timers*

A simple method to reduce the scanning latency is to reduce the value of the scanning timers. As previously stated in this Chapter, the IEEE 802.11 standard does not propose any specific values for the scanning timers, i.e., MinCT and MaxCT. Then, every firmware or driver implementation sets specific values, giving different scanning performance.

MinCT is, by definition, the minimum time to wait for Probe Responses in a channel. At the same time, it is the maximum time an AP has to successfully answer a Probe Request. On the other hand, MaxCT is the maximum time to wait for Probe Responses on a channel that allows collecting the largest number of Probe Responses from different AP. Using a low value for MinCT may prevent receiving the first Probe Response on the channel and so erroneously declaring the channel empty. Also, setting a low value for MaxCT may prevent discovering all the APs in the channel but only a subset.

Velayos and Karlsson [50] try to infer the best values for both timers presenting theoretical considerations and simulation results. For MinCT, the authors base on the maximum time an AP needs to answer a Probe Request, considering that both the AP and the channel being probed are idle. If propagation delay and Probe Response generation time are neglected, then the IEEE 802.11 MAC Distributed Coordination Function (DCF) establishes that the maximum response time has the form of Equation 1. MinCT should allow the station to wait for DCF Interframe Space (DIFS) and the backoff (considering the maximum value for the contention window during the first transmission attempt, aCW_{min}), reaching 670 µs. The authors decide then to set MinCT to 1 Time Unit (TU), which is equal to 1024 µs.

$$\begin{aligned} MinCT &= DIFS + (aCWmin \cdot aSlotTime) \\ &= 50\mu s + (31slot \cdot 20\frac{\mu s}{slot}) \\ &= 670\mu s \\ &\cong 1\,TU \end{aligned} \tag{1}$$

For MaxCT, the authors analyze the Probe Response delay depending on traffic load and the number of stations on each channel. They conclude that MaxCT is not bounded as long as the number of stations increases. They recommend to set MaxCT to 10 TU to avoid responses from overloaded AP. This is based on the hypothesis that ten MS associated with the same AP is an adequate number in order to achieve good throughput in a channel.

However, as it will be detailed in the following sections, using fixed timers may not allow maximizing the number of discovered AP while giving a reduced latency in all possible scenarios. Authors introduced several considerations regarding the number of MS operating on each channel and data traffic conditions. The experimental results presented in this chapter will prove that the proposed fixed value for MinCT, 1 TU, is not long enough to receive the first Probe Response in a channel in the most typical AP deployments. In this case, if no AP is found due to a short MinCT, a scanning failure occurs, giving a link layer disconnection if no AP is found after scanning all the channels.

3.4.3 *Handover Anticipation*

Other handover optimizations in the literature aim to reduce the impact of handover on data communication by intelligently deciding the most suitable moment to trigger the scanning, authentication and association. These optimizations differ from previously presented works since in those cases the authors aim at providing optimal parameters and heuristics for the scanning process itself. In this section, we first present a set of solutions that consider different mechanisms to trigger the handover process. Then, we present the periodic scanning and the synchronized passive scanning mechanisms.

HANDOVER TRIGGERS The simplest mechanism to trigger a handover is to monitor the Received Signal Strength Indication (RSSI) as an estimation of the link quality and start the handover process if the current RSSI is lower than a pre-established threshold. This is commonly referred to as a *Link Going Down* indicator in the context of IEEE 802.21 Media Independent Handover [51]. When declaring a Link Going Down, there is a trade-off between the number of times a handover is triggered and the link quality. If the MS declares the link going down when the MS has still and acceptable link layer

connection with an AP, handovers will be more frequently executed, generating additional delays for the user. On the other hand, if the MS waits until a very weak link situation, the data communications start degrading before the handover occurs. The main goal is to anticipate the handover and trigger the transition to a new AP under optimal conditions. Mhatre and Papagiannaki [49] propose a set of handover algorithms based on continuously monitoring the wireless link, i.e., listening to beacons from the current and neighboring channels. The authors propose a taxonomy of triggering algorithms to anticipate handover based on different criteria. In particular, they propose five different algorithms. First, the *Beacon* approach triggers a handover based on the number of consecutive lost Beacon frames from the current AP without analyzing the condition of neighboring AP. The *Threshold* algorithm uses the current RSSI to trigger a handover, i.e., it waits until a very low RSSI value and does not consider neighbor AP information neither. Then, three algorithms are defined by considering and comparing neighboring AP information in the current and the neighboring channels. These algorithms aim to avoid triggering a handover if no better AP is deployed in those channels. The *Hysteresis(Δ)* algorithm considers RSSI values from the current and neighboring AP deployed in the overlapping channels. it triggers a handover if the RSSI of one of the AP in the overlapping channels exceeds the current AP RSSI by a value of Δ. Then, the *Trend* algorithm uses the RSSI measurement of neighboring AP and calculate, for each one of them, the trend of the smoothed RSSI during a time window. The handover is triggered if the new candidate AP has a positive trend Δ while the current AP has a negative trend having the same absolute value, $-\Delta$. Finally, the *LSE* uses a Least Square Estimator to predict the RSSI value in the next time interval. Then, a handover is triggered only if the least square estimator of the candidate AP RSSI plus the associated error of the estimation is greater than the least square estimator of the current AP RSSI plus the error. The authors evaluate these approaches in a real testbed. The *Beacon* and *Threshold* algorithms give average handover latencies between 530 and 860 ms since they always rely on scanning all the channels after deciding to trigger the handover. On the other hand *Hysteresis*, *Trend* and *LSE* succeed in avoiding to scan all the channels by using information from AP in the current and overlapping channels and so average handover latencies are between 140 and 450 ms. However, since these approaches need to listen to beacons from neighboring channels, it is necessary to modify the firmware of the wireless card, which may not always be possible.

Similarly to the approaches presented in [49], Yoo et al. [52] propose a number of handover triggering mechanisms based on predicting RSSI samples at a given future time using Least Mean Square (LMS) filters. In this algorithm, the device continuously monitors the RSSI and computes the LMS prediction if the RSSI is below a certain threshold (P_{Pred}). Then, if the predicted RSSI value is lower than a second threshold (P_{Min}) the MS starts a handover.

The authors propose simulation results showing that the difference between the simulated RSSI samples and the LMS 500 ms ahead predictions is lower than 0.35 dB for a MS speed lower than 4 m/s. However, the simulation results only consider a single value for P_{Min}, without providing an analysis of the trade-off between the handover frequency and the link quality.

PERIODIC SCANNING AND SMOOTH HANDOVER The periodic scanning concept is based on decoupling AP scanning (i.e., the most time-consuming handover phase) from the actual AP transition (i.e., AP selection, authentication and association). Wu et al. [53] propose *Proactive Scanning*, which consists in two phases. First, when the MS is connected to its current AP, it alternates short scanning phases with data communication. Each short scanning phase is approximately 10 ms long, so the effect on data traffic is minimized. To achieve this, the scanning interval (i.e., the time between two scanning phases) and the channel sequence (i.e., the list of channels to scan in each short phase) are dynamically adapted. The scanning interval is adapted depending on the current signal level and varies between 100 and 300 ms. The channel sequence is selected based on a priority list that is built based on historical information. To build the priority list, the MS stores the channel where AP have been discovered during long scanning phases (i.e., where all possible channels are probed). The authors also propose an analysis of the scanning timers to allow receiving Probe Responses from AP. After performing some experiments introducing some traffic load on the AP, they conclude that using $MinCT = MaxCT = 5$ ms allows the MS to successfully receive Probe Responses even in highly loaded situations. Experimental results are proposed showing that service interruption is reduced using proactive scan compared to standard handoff procedures of existing wireless cards. Moreover, the overhead of proactive scan, measured as the degradation of the TCP throughput, strongly depends on the scanning interval and the link rate, varying between 1 % and 36 %.

The *Smooth Handover* [54] and the *Periodic Scanning* [55] methods are also based on splitting the discovery phase into multiple sub-phases to allow an MS to alternate between data packet exchange and the scanning process. Liao et al. [54] propose to scan a group of channels in each sub-phase, while in [55] only one channel is scanned during MinCT. Each sub-phase is triggered depending on the current RSSI. The Smooth Handover [54] performance was evaluated in a real testbed and showed that the data packet loss is strongly reduced (up to 93 %) compared to a full scanning based handover. In Periodic Scanning [55], network simulations on six different scenarios are proposed. The handover delay and the packet loss rate result lower than in full scanning and selective scanning [48] techniques. However, they observe that for Periodic Scanning, the MS generates a larger number of packets per handover, contributing to a high energy consumption.

The main limitation of periodic scanning techniques is that, after collecting Probe Responses or Beacons from different AP before the actual handover is triggered, those AP may no longer be available or may have a lower signal strength due to the MS mobility. In this case, the MS has no option but to perform a full scanning that may greatly impact the on-going communication flows.

3.4.4 *Synchronized Passive Scanning*

Unlike common handover optimizations focusing on active scanning, the SyncScan [56] method is based on the standard passive scanning, where an MS simply waits for periodic beacons on a given channel. The passive scanning latency is related to the number of channels and the *Beacon Period* timer, commonly set to 100 ms. Therefore, passive scanning latencies usually exceed one second when listening for beacons in all the channels. SyncScan synchronizes short listening periods at the MS with periodic Beacons reception from the AP. Then, the MS switches to a given channel when a beacon is about to arrive. Using SyncScan the MS has up-to-date information about AP without probing the channels. Once the MS decides to execute a handover, it only needs to authenticate and associate to the selected AP.

Like in [54] and [55], the scanning latency is spread during data communication, but some limitations should be analyzed. The fact that the MS must switch to a channel when a beacon is about to arrive adds a complex time synchronization management between the MS and all deployed APs. Clock accuracy becomes critical in this approach because even a minor deviation in time synchronization becomes non-negligible, preventing an MS from discovering neighbor AP. Authors propose the usage of Network Time Protocol (NTP) that maintains time within 10 ms accuracy over the Internet, achieving precisions of 200 µs or better in local area networks, under ideal conditions. Under these considerations, SyncScan implementation appears as a solution limited to very specific deployments (e.g. enterprise or campus deployments), where a central administrator can manage the channel allocation and synchronization between beacons from all AP. Synchronizing AP in a fully heterogeneous environment like the one presented in Chapter 2 (e.g. hotspots or community deployments from multiple operators around a city) for the implementation of SyncScan seems impractical. In all the cases, the SyncScan procedure is performed regularly, resulting in several unavailable periods for data packets transmissions, so packet loss may be observed while exploring other channels, just as in periodic scanning.

3.5 MOTIVATION

3.5.1 *Preliminary Considerations*

The previously presented handover optimizations aim at reducing the handover delay by proposing low-latency scanning mechanisms. For every fast handover approach, an MS still needs to scan channels one after the other to discover AP and this requires an appropriate configuration of the scanning process. In smooth handover [54] or in periodic scanning [55], the discovery phase is split into several independent sub-phases that are separated by a certain time period (during which the MS may still exchange data packets). During each of these sub-phases, the MS scans the channels one by one, just as in a continuous scanning phase. In the selective scanning [48], the order in which channels are scanned is determined by a binary mask built from previous scanning phases. For each channel, the different AP also need to be probed and thus the time to spend on each channel (i.e., the value of the timers) needs to be defined. In the synchronized passive scanning [56], channels are not actively probed but still the MS needs to define a time to wait on each channel. Moreover, the optimization proposed in [56] cannot be applied in an heterogeneous scenario, since synchronisation between possible neighbours is not achievable. We observe that in those mechanisms there is still a lack of work in the determination of the most adequate values defining the specific parameters to use while doing scanning. In Section 3.5.1.1, we describe the configuration of the scanning parameters that can be currently found in the literature.

3.5.1.1 *Configuring the scanning parameters*

In every scanning mechanism, the MS needs to define appropriate parameters in order to assure a low scanning latency while discovering the maximum number of AP in the environment. We differentiate between two different parameters for a scanning mechanism: the *timer values* and the *channel sequence.*

The timer values define the amount of time that an MS waits for Probe Responses on a channel after sending a Probe Request. Since these values are not defined in the standard, several studies in the literature [50] [47] [53] suggest using fixed values for $MinCT$ and $MaxCT$. Velayos et al. [50] derives $MinCT = 1\,ms$ and $MacCT = 10\,ms$ using some theoretical considerations. Wu et al. [53] proposes using $MinCT = MaxCT = 5\,ms$. On the other hand, Mishra et al. [47] provide an experimental study to deduce the values of $MinCT$ and $MaxCT$ for three different IEEE 802.11b wireless cards (Cisco, Lucent and ZoomAir). They show that for Cisco cards $MinCT$ equals $13\,ms$ and $MaxCT$ equals $38\,ms$. In the case of Lucent and ZoomAir cards, several Probe Requests are send on different channels and the time to wait for responses (the authors do not find a correlation that indicates the usage of two

timers MinCT/MaxCT) varies between 10 and 75 ms. We have investigated the values used by two popular open-source drivers: ath5k[1] and MadWiFi[2]. Their timer values vary between 20 ms and 200 ms, resulting in a scanning latency that can be greater than one second.

Regarding the channel sequence, the MS has to decide which channels to scan and their ordering. The channel sequence and its order are relevant since a scanning strategy can decide to stop the scanning once an AP has been discovered (or if another condition is satisfied). Common implementations in open-source drivers switch channels sequentially. Mishra et al. [47] showed that Cisco cards probe all the channels sequentially, while Lucent and ZoomAir cards only probe channels 1, 6 and 11 and take profit from the channel overlap to gather Probe Responses from neighboring channels with a lower probability (like in [49]). We have observed that in Android mobile devices, for example, the MS implements a channel switching technique similar than the mechanism proposed in [53], interleaving short scanning phases to data communication.

3.5.2 *The Probe Response Delay*

In all the previously cited strategies, different values for the timers and the channel sequence are proposed. These values are fixed on each particular firmware or driver implementation and used each time an MS needs to scan.

However, in the case of the timers, using fixed values cannot assure that Probe Responses will be received before their expiration in all possible scenarios. Probe Responses may be delayed because of channel congestion, causing packet collisions and retransmissions, or due to a high traffic load on the AP, adding extra delays to treat a Probe Request and generate Probe Responses. To evaluate this, we have performed different measurements of the Probe Response delay under different deployment conditions.

In 2009 [44], we have set an MS that performed scanning using the MadWiFi driver over a DLink DWL-AG660 card. The network deployment was composed of Linksys and DLink AP that have been set in a three non-overlapping channel deployment (i.e., one AP in channel 1, 6 and 11) and logged the First Probe Response Delay (FRD), i.e., the time elapsed between sending the Probe Request and receiving the first Probe Response on each channel. Figure 30 shows the distribution of the FRD. We observe that the introduction of traffic in the network (using D-ITG [57]) delays Probe Responses significantly. For instance, waiting 6 ms after sending the Probe Request allows receiving the 87 % of the Probe Responses without traffic while it allows receiving only the 43 % of the responses with traffic.

Later in 2010 [45], we have evaluated the FRD by performing scanning with a Netgear WG511T card using the ath5h open source driver. We have

1 http://linuxwireless.org/
2 http://madwifi-project.org/

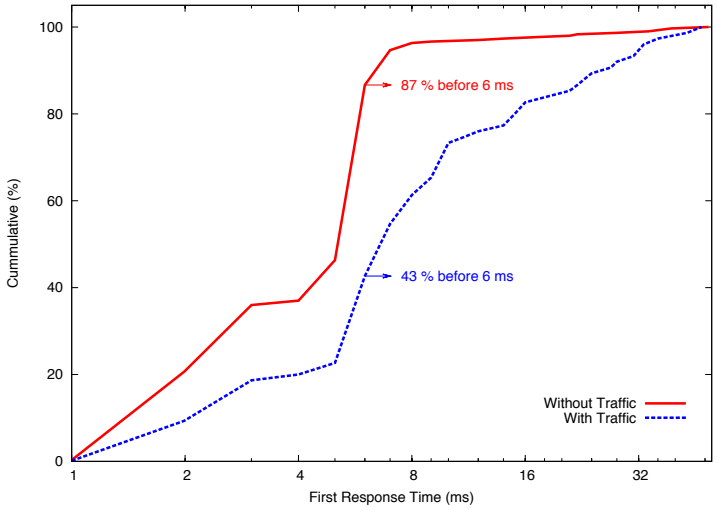


Figure 30: Experiment 2009 - First Probe Response Delay distribution

performed this test in a different laboratory deployment than in the previous experiment, using Linksys WRT54GL AP. In this case, we have introduced different levels of traffic using iperf[3]. The distribution of the first FRD is illustrated in Figure 31 and the variation of the FRD with different levels of traffic is shown in Table 10. In Figure 31, the FRD distribution has been calculated in an environment with a single AP deployed (i.e., no interferences from neighboring AP). In the case of the FRD with traffic, it corresponds to the highest traffic level, i.e., 12.5 Mbps. In this experiment, we have observed shorter FRD than in the previous experiment (2009), possibly due to the different wireless cards and AP that have been used.

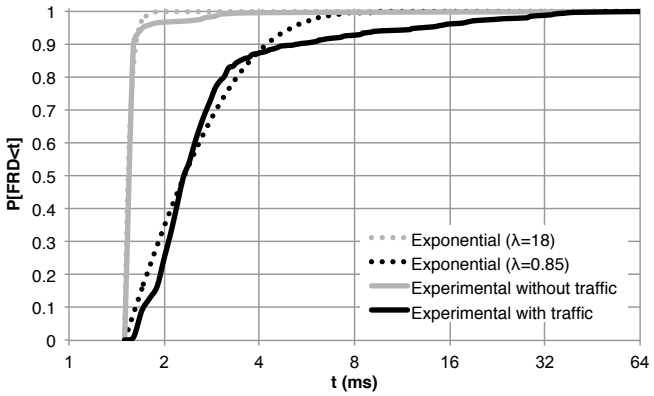


Figure 31: Experiment 2010 - First Probe Response Delay distribution

Finally, we aimed at gathering a large number of Probe Responses in an heterogeneous deployment [35]. In 2011, we have performed a measurement study using an the internal wireless card of an Asus N10J laptop carrying an external 7 dBi antenna. This MS has continuously performed scanning along an 8 km path in the city center of Rennes, gathering more than 51,000 first

3 iperf available at: http://sourceforge.net/projects/iperf/

Table 10: FRD experiments

Flow (Mbps)	Load (%)	$\bar{E}[FRD]$ (ms)	$\bar{\sigma}(FRD)$ (ms)
Bkg	1.52	1.83	2.12
1	5.62	1.79	1.19
2	9.68	1.84	1.03
4	20.05	1.81	0.58
10	51.97	2.07	0.62
11.5	73.11	3.9	5.7
12.5	74.49	3.58	4.87

Probe Responses from different AP. The distribution in Figure 32 shows a median FRD of 3 ms.

Along these experiments, we can observe that the delay of Probe Responses varies depending on multiple parameters, e.g., the AP deployment, the specific hardware and drivers. Then, the specification of the timer values is not straightforward, since unique fixed values for MinCT and MaxCT may be appropriate only for some scenarios. For example, if we consider the MinCT specified by Velayos et al. [50] (MinCT = 1 ms), it does not allow an MS to successfully discover the first Probe Response in a given channel with a high probability. Particularly in our experiments carried out in 2009 and 2010, we have observed that no Probe Response arrived before MinCT = 1 ms while in the 2011 experiment only the 3.7 % of the first Probe Responses arrived before 1 ms.

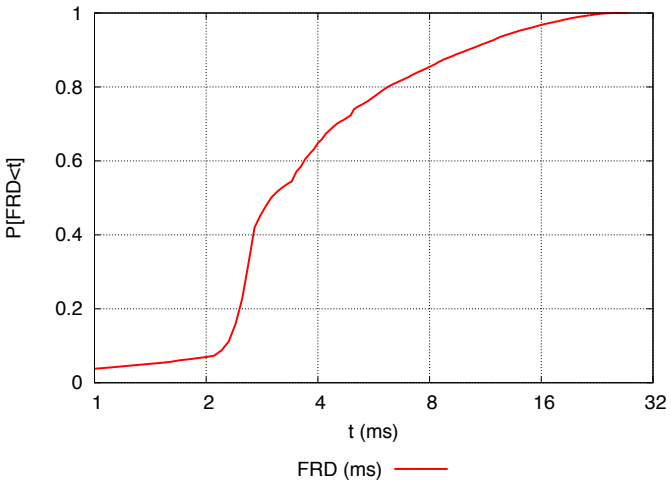


Figure 32: Experiment 2011 - First Probe Response Delay distribution

Then, we can assume that in order to reduce the latency and maximize the number of discovered AP, the values of the timers have to be dynamically adapted to the different scenario conditions rather than using fixed values. In the following, we define the trade-off between the different performance metrics when doing scanning. Then, in Sections 3.6 and 3.7, we define, implement and evaluate two adaptive strategies for IEEE 802.11 scanning.

3.5.3 *Scanning Performance Metrics: A Trade-off*

In every active scanning process, the MS has to configure a set of parameters (i.e., the timer values and the channel sequence) to discover for AP on the different channels. The main objective is to minimize the disruption of ongoing communications while discovering a large number of AP. In order to measure the performance of the scanning process, a set of metrics is defined:

1. **Scanning Latency**: The time spent for the discovery process, i.e., to scan all the channels in the channel sequence. The scanning latency is measured as the difference between the time of the Probe Request in the first channel in the sequence after the expiration of the last timer in the last channel in the sequence (i.e., $MinCT$ if no Probe Response is received or $MaxCT$ if at least one Probe Response is received).

2. **Scanning Failure**: A Scanning Failure corresponds to the case where no AP is discovered after scanning all channels in the channel sequence. A scanning failure occurs if no AP is deployed in the scenario or if the scanning parameters do not allow to discover at least one of the existing AP.

3. **Discovery Rate**: The number of discovered AP over the total existing number of AP. This metric provides a measure of the proportion of the deployment that has been discovered.

4. **First Discovery Time**: The time elapsed between the beginning of the scan (i.e., the sending of the first Probe Response) and the reception of the first Probe Response. This metric is useful to analyze the effect of different channel sequences and timers in a scanning algorithm aiming at discovering a single AP as soon as possible.

The purpose of adapting the timers and the channel sequence is not to determine the best values that fit a particular scenario, since we assume unknown and unpredictable deployments and so the topology on every discovery process is rather unknown. Thus, an adaptive scanning algorithm aims at finding a trade-off between a minimum scanning latency, a minimum scanning failure, a maximum discovery rate and a minimum first discovery time. Recall that when decreasing the latency we increase the failure and decrease the discovery rate.

In the following, two adaptive algorithms for IEEE 802.11 scanning are defined and evaluated by the means of experimental testbeds. The first adaptive algorithm, ADA, is based on dynamically varying $MinCT$ and $MaxCT$ while scanning, depending on the previously discovered AP (i.e., in the previous channels) and using a random channel sequence giving priority to the non-overlapping channels. The second adaptive approach uses physical layer information, i.e., the channel load and the power, to dynamically set the timers values and the channel sequence.

3.6 ADA: ADAPTIVE DISCOVERY ALGORITHM

The ADA consist in dynamically changing the values for MinCT and MaxCT during the scanning process based on the previously discovered AP. This approach allows an MS to spend less time on some channels once candidate APs have been already found whereas a fixed timers scanning would spend the same time on each channel. The main goal is to reduce the timers channel by channel while APs are discovered. This is because the MS can assume the risk of missing further APs (by using lower timers) if other APs have been discovered before. On the contrary, timers may be increased if no AP has been found, in order to increase the chances of finding an AP on the next channel(s). In this section, we analyze the behavior of such an adaptive scanning algorithm against a fixed timers scanning approach. A particular adaptation function is presented, but other adaptation function could also be implemented.

The selection of the channel sequence becomes important if we consider timers adaptation, because timers are adapted according to the activity on each channel. The sooner an AP is found, the faster the timers will be decreased, and thus, it is important to scan first channels in which APs may be operating. In IEEE 802.11 networks, only three non-overlapping channels exist. As stated in [58] and [59], a proper deployment typically uses only these channels. Results found in Eriksson et al. [60], Giordano et al. [38], Castignani et al. [35] and our own measurement study in Chapter 2, confirms that the non-overlapping channels are the most likely used in real deployments. In [60], the authors propose an optimal scanning strategy that gives more priority to channels 1, 6 and 11, since they found that 83 % of AP are assigned to those channels. On the other hand, experimenting over a different deployment, the authors of [38] states that 77.98 % of AP are deployed in the non-overlapping channels. Our own experiments on a city-wide WiFi deployment [35] results in 78.21 % of occupancy in non-overlapped AP. Then, it could be assumed that prioritizing those channels, as stated in [48] and [60], the MS has more chances to find an AP in the first scanned channels and so reduce the timer values for the following channels. In ADA, the channel sequence is built using two different random sub-sequences. In the first subsequence, the MS randomly switches between the non-overlapping channels (1, 6 and 11). Then, the rest of the channels are also randomly considered. If an AP with relative good signal level is discovered in channels 1, 6 and/or 11, the adaptive system will set lower timers for the next channels to scan. In all cases, timers are adapted between pre-established bounds (defined by experimentation in Section 3.6.2.2). A different strategy to compute the channel sequence is proposed in [60], where the ordering of the channels is proportional to the channel occupancy, e.g., if 20 % of the AP have been observed to be deployed on channel 6, then channel 6 is scanned with a probability of 0.2. ADA considers to scan all the channels in the sequence, but giving more

priority to the non-overlapping channels, i.e., increasing the probability that a high timer will be used on those channels. However, a different adaptation function may consider to use shorter channel sequences, and so stop scanning at any given time.

3.6.1 *Design and Implementation*

In this Section, the logic of the proposed ADA is detailed. Given the high variability of the Probe Response delay, we propose to adapt $MinCT$ and $MaxCT$ between pre-established bounds (defined in Section 3.6.2.2). The main goal is to allow the MS to use low timers once APs have been previously discovered. As illustrated in Figure 33, the MS starts scanning using half the maximum bounds for both timers. This strategy aims at balancing the trade-off between the scanning latency, the scanning failure and the discovery rate. In this first adaptation algorithm we do not consider the first discovery time performance metric, since we consider scanning all the channels in ADA. Observe that, if an MS started scanning using the maximum bounds of the timers, it would end up with a higher scanning latency. Otherwise, if an MS started scanning with the lower bounds, it could fall in a high scanning failure. For a channel in which at least one AP has been discovered, the MS calculates the greatest quality of all discovered AP on that channel (Q) and the number of APs that have replied on that channel (N). N is obtained by simply counting the number of Probe Responses received from different APs. Regarding Q, the RSSI of each Probe Response is considered. Both Q and N are combined in order to establish the criteria that will be used to rank discovered AP, and decide whether the timers for the next channel (T_{n+1}) can be reduced or increased. In Figure 33, T_{n+1} represents the tuple $(MinCT, MaxCT)$, since both timers are simultaneously adapted (i.e., $MinCT$ and $MaxCT$ are equally increased or decreased). Then, T_{n+1} is calculated considering a *decision making* parameter (R, in Equation 2) calculated for each channel by using Q and N.

$$R = \frac{Q}{N} \tag{2}$$

This simple relation allows adapting timers differently, depending on the quality of the APs that have been discovered on the previous channels. Two different APs having the same RSSI and operating in different channels are considered in different ways, since populated channels (having a high N) will not be prioritized as well as those with a lower N. This choice comes from the observation that in a wireless environment, *weak signal* and *collisions* are considered as the factors limiting link performance. A packet transmitted through an IEEE 802.11 link may be lost because of a weak signal or a collision, but discerning the real cause is quite difficult. We consider the re-

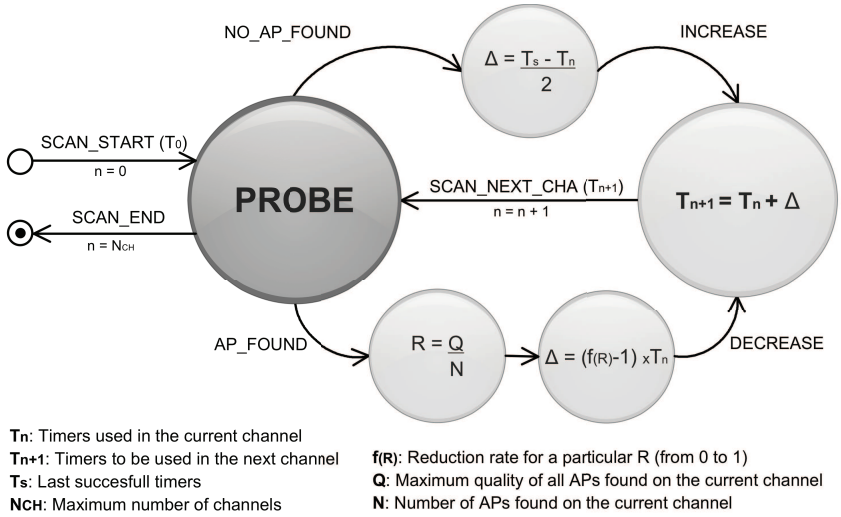


Figure 33: ADA implementation

sults obtained in [61], in which a set of testbeds were implemented so as to independently analyze the effect of a weak signal (due to a low RSSI between the MS and the AP) and collision (due to several MSs and APs operating on the same channel). The authors have empirically showed that for 98 % of the packets with errors, they observed a Bit Error Rate (BER) lower than 12 % in a weak signal scenario (low RSSI, without collisions) against a BER of 50 % in a collision scenario (without weak signal effects). We can infer that the fact of having a high N, i.e., multiple AP sharing the channel, which produce collisions, is less desirable than a low RSSI scenario since it causes a much higher BER. Then, using N as the divisor in Equation 2 prevents from drastically reducing the timer if a high congested channel has been previously found, giving a higher chance to discover a better AP in the next channels.

Then the MS takes into account the value of R calculated on the channel and adapts both MinCT and MaxCT using the same proportional factor ($f(R)$). $f(R)$ is implemented in a way that for higher values of R, timers are more strongly reduced. In contrast, as shown in Figure 33, if no AP is discovered, no R is calculated on the correspondent channel and then timers are increased to half the difference between the last successful timers (those from whom at least one response has been received from an AP) and the timers used on the previous channel (those from whom no response was received from any AP). Increasing timers using this approach avoids overshooting, since timers are smoothly augmented.

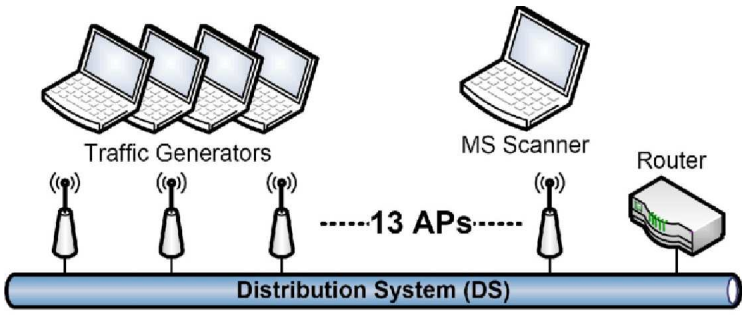


Figure 34: Testbed configuration

3.6.2 *Experimentation and Results*

3.6.2.1 *Testbed setup*

A real testbed was implemented using up to thirteen AP from different manufacturers and seven MS for traffic generation on the different channels (see Figure 34). All the devices in the testbed implemented 802.11b as physical layer. The MS performing scanning uses an Atheros based D-LINK DWL-AG660. The ADA algorithm and a set of fixed-timers strategies have been implemented in the MadWiFi driver (Version 0.9.4). Up to 64 different network scenarios were evaluated using eight different channel allocations, two different traffic conditions and four configurations for MinCT and MaxCT timers. Traffic was generated using *D-ITG* (Distributed Internet Traffic Generator) [57], injecting, in all configurations, a UDP traffic load of $8\,\mathrm{Mbit/s}$ between one sender and one receiver, which leads to overloaded cells in IEEE 802.11b. With regard to the channel allocation, the following configurations were evaluated.

- **Configuration 1**: Thirteen APs allocated on channels 1 to 13 (one AP per channel).
- **Configuration 2**: Thirteen APs all allocated on channel 11 (13 AP on the same channel).
- **Configuration 3**: Three APs allocated on channels 1-6 -11 (one AP per channel).
- **Configuration 4**: Twelve APs allocated on channels 1-6-11 (4 AP per channel).

Note that the performance of ADA depends on each particular scenario. However, these configurations have been chosen from all the possible cases since they are a valid representation of real deployments (as shown in [38] and [60]) and include favourable cases (for the discovery process) as well as challenging environments. In each of our experiments, the scanning process is measured a hundred times in every configuration. For each scanning, all

channels are probed by the MS one by one, i.e., the MS do not prematurely stop the scanning process.

3.6.2.2 *Bounds Determination for* MinCT *and* MaxCT

In order to set up ADA, the bounds for MinCT and MaxCT need to be defined, i.e., the intervals in which MinCT and MaxCT will vary. We define *MinLower* and *MinUpper* as the lower and upper bounds for MinCT and *MaxLower* and *MaxUpper* as the lower and upper bounds for MaxCT. For this purpose, we measured the delay of the first and further received Probe Responses on each channel for each particular AP deployment configuration, with and without traffic. We define Further Probe Responses as those that arrive after the first Probe Response. We configured the MS with $(50\,ms, 200\,ms)$ for $(MinCT, MaxCT)$ in order to allocate enough time for the discovery of all operating AP (we were not interested in the scanning latency, but in collecting the delay of each Probe Response). Note that, as illustrated in Figure 30, a MinCT equal to $50\,ms$ allows the MS to gather all the first Probe Responses in a channel.

Table 11 gives the main conclusion for bounds determination. Optimistic and pessimistic scenarios have been considered to define the upper and lower bounds. We also considered different percentages of received Probe Responses for each bound (e.g., we considered a high percentage of received Probe Responses for *MinUpper* because this bound will be used when no AP has been found). As presented before, Figure 30 shows the FRD in a three non-overlapping channel configuration with and without traffic (configuration 3). This scenario is considered as an ideal AP configuration where interferences are minimized and thus helps to determine the minimum limits for MinCT. If we observe the FRD received over all the trials without traffic, it can be observed that 87 % of the first Probe Responses were received before $6\,ms$. Thus *MinLower* is set to $6\,ms$ as stated in Table 11. We allowed such a low percentage ($8ms$ could have been taken where 96 % of the Probe Responses were received) because it can be afforded to risk few unsuccessful discoveries in our adaptive strategy when this minimum value is used. Note that in the considered adaptive strategy, this minimum value is only used when AP(s) have been previously discovered (See Section 3.6.1).

With the same aim and considering configuration 4, without traffic, *MaxLower* is set at $8ms$ where 50 % of further Probe Responses from other AP were already received. Then, MaxCT can be adapted up to a low limit that covers less cases than MinCT (only 50 %), since the situation of not discovering more AP is not as risky as not discovering the first AP, in which the channel will be declared empty.

On the other hand, the upper bounds *MinUpper* and *MaxUpper* are determined using the results obtained in the other scenarios (which are highly affected by interference and congestion), like configuration 1 and 2, including traffic. *MinUpper* has been set to $34\,ms$ (96 % of further Probe Responses

Table 11: Bounds for MinCT and MaxCT

Bound	Value	% of Probe Resp. received	Conf.	Traf.
MinLower	6 ms	87%	3	No
MinUpper	34 ms	96%	1	Yes
MaxLower	8 ms	50%	4	No
MaxUpper	48 ms	87%	2	Yes

received in configuration 1) and *MaxUpper* has been set to 48 ms (87 % of further Probe Responses received in configuration 2). These upper bounds are meant to be used when no AP has been previously discovered, in order to have a higher probability of getting Probe Responses.

3.6.2.3 *General Results*

ADA was tested using bounds defined in Table 11 and the fixed timers strategy was evaluated considering three different values for the timers, (10 ms, 20 ms), (25 ms, 50 ms) and (50 ms, 200 ms) for (MinCT, MaxCT) respectively. Table 12 shows the results organized by configuration scenario, where the *full-discovery rate* indicates in how many scanning processes all available AP were discovered. The *failed scanning* values describe the scanning failure as a percentage from the total cases. Finally the average scanning latency, including the standard deviation (σ) for the adaptive strategy, shows a low dispersion of the obtained latencies.

The main observation of these experiments is that the discovery process performance highly depends on the deployment scenarios and a high scanning failure may be observed for the fixed timers strategy in some common network scenarios. Considering all the scenarios, ADA only shows 2 % of scanning failure in a single scenario (configuration 2 and loaded cells) and keeps a low scanning latency. A detailed analysis of results of Table 12 is presented in the following paragraphs.

IMPACT OF TRAFFIC LOAD Figure 30 illustrates configuration 3, where Probe Responses are notably delayed when traffic is injected. While before 6 ms the 87 % of the Probe Responses are received in non loaded scenarios, only 43 % is received when there is traffic. As shown in Table 12, in the case of configuration 3 with traffic, for several scanning attempts a Probe Response is not received before 25ms, causing 20 % of scanning failure. Even when using a MinCT equal to 50 ms the scanning failure reaches 13 %. The effect of traffic also produces a decrease in the percentage of discovered AP in all evaluated scenarios. ADA helps to reduce the effects of traffic, since no scanning process fails except in configuration 2 (with traffic), where we observe only 2 % of scanning failure.

Table 12: Comparative results

Scenario				**Discovery Rate (%)**				**Scanning Failure (%)**				**Scanning Latency (ms)**				
AP conf.	Channels	Number of APs	Traffic	Fixed Timers			ADA	Fixed Timers			ADA	Fixed Timers			ADA	
				10-20	25-50	50-200		10-20	25-50	50-200		10-20	25-50	50-200		σ
1	1 to 13	13	No	65%	87%	93%	49%	0	0	0	0	275	708	2567	256	11%
lightgray 1	1 to 13	13	Yes	24%	69%	82%	40%	2%	0	0	0	317	636	2378	248	13%
2	11	13	No	75%	92%	94%	96%	2%	2%	0	0	152	360	807	423	3%
lightgray 2	11	13	Yes	54%	88%	98%	83%	29%	3%	0	2%	159	363	814	434	5%
3	1-6-11	3	No	92%	94%	99%	94%	0	0	0	0	117	414	1119	190	11%
lightgray 3	1-6-11	3	Yes	38%	51%	61%	81%	52%	20%	13%	0	227	403	1025	210	18%
4	1-6-11	12	No	98%	98%	100%	95%	0	0	0	0	179	419	1121	390	3%
lightgray 4	1-6-11	12	Yes	39%	60%	87%	84%	13%	1%	0	0	239	450	1110	378	13%

THEORY VS. EXPERIMENTATION Our experimental results do not match theoretical considerations and simulations presented in [50]. In this work, a value of 1 ms for MinCT is considered enough to wait for the first Probe Response before switching to the next channel in the sequence. On the other hand, our experience shows that MinCT needs to be greater than 10 ms to receive the 97 % of first Probe Responses in an ideal three non-overlapping channel scenario without traffic. Moreover, as shown in Figure 30, the earliest first Probe Responses only appear after 2ms in the same configuration. Regarding MaxCT, authors of [50] state that it is unbounded and claims for a MaxCT equal to 10 ms to be enough. It has been shown during the bound determination that this value could not be sufficient for some scenarios. This gap between results proposed in [50] and our experimentation (that are close to those presented in Mishra et al.[47]) may be explained by additional delays neglected in the literature, such as channel switching, congestion condition and processing time for management frames.

IMPACT OF THE NUMBER OF APS In configurations 3 and 4 where only non-overlapping channels were used, a reduced scanning failure is observed when there are four AP operating on the same channel (configuration 4). When there is a single AP per channel (configuration 3), a higher scanning failure is attained for the fixed timers strategy. This may be due to the back-off timer of the MAC protocol, since there are more chances to pick a small random number when there are more active AP.

SCANNING LATENCY The scanning latency depends on the values of MinCT and MaxCT during the discovery process. In ADA, MinCT is initially set to half the value of *MinUpper* and it gradually decreases until *MinLower* if APs are discovered. Figure 35 shows scanning latency values for all configurations with traffic including the scanning failure (in percentage) for each case. Even if fixed timers strategies may give good results in some scenarios, the adaptive strategy provides lower or equivalent scanning latency from 190 ms to 434 ms. The fixed timers strategy configured with (10 ms, 20 ms) gives better latencies in AP configuration 2, around 150 ms, against 420 ms for ADA. But in this case the scanning failure reaches 29 %, while the adaptive strategy only gives 2 % of failure. Moreover, in configuration 3 with traffic, the adaptive strategy gives the best scanning latency without any failure, while all other evaluated strategies reach high levels of failure, up to 52 %.

3.6.3 *Discussion*

ADA has been proposed to better manage the trade-off between the different scanning performance metrics, i.e., to achieve a low-latency discovery while discovering the maximum number of AP and reduce failed scanning

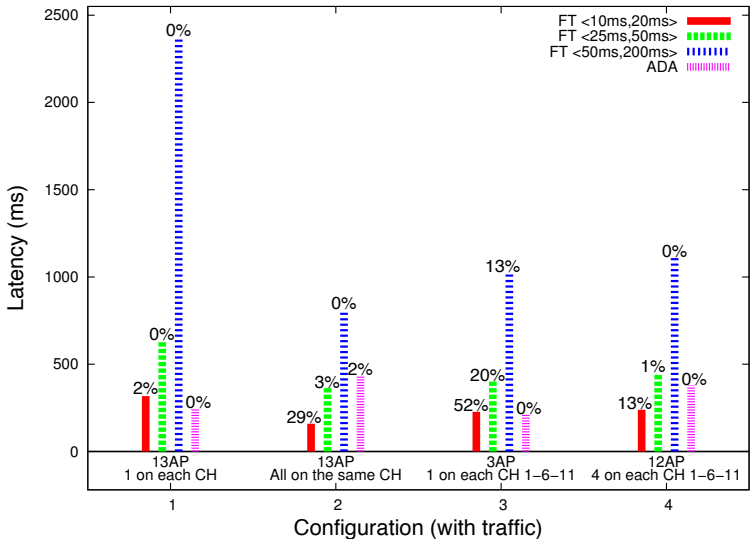


Figure 35: Testbed results

attempts. However, other adaptation strategies can be studied in order to avoid using fixed timers for the discovery process. In Section 3.7, we define a different adaptive strategy that uses physical layer information in order to use the most adequate timers for each specific channel condition.

3.7 CROSS-LAYER SCANNING: A PHY-MAC APPROACH

As proposed in Section 3.6, ADA is based on dynamically adapting the scanning timers, i.e., the value of the timers to use in the current channel is based on the discovered resource on the previous channels. This allows reducing the scanning latency while having a controlled scanning failure. However, a different adaptation strategy could find the most suitable timers for each channel based on reliable information characterizing the particular condition of each channel. In order to have a knowledge of the channel activity before performing scanning at the MAC layer, an MS has different options. First, an MS may rely on IEEE 802.11k [62], a recent amendment for Radio Resource Management in IEEE 802.11. Using IEEE 802.11k, an MS is able to request for channel measurement information to its current AP and to other MS in the cell. Some of the information that can be requested are the BSSID and operating channel of neighboring AP, the radio condition of the different channels or the load and the error rate of different AP. Athanasiou et al. [63] propose a cooperative handover mechanism based on IEEE 802.11k. However, since IEEE 802.11k only defines the protocol for message exchange to perform the radio measurements but not any particular algorithm to calculate those parameters (i.e., channel load estimation, access delay), there is still a lack of real IEEE 802.11k implementations. In [63], the cooperative handover approach is only evaluated by the means of simulations.

Another approach to assist the scanning mechanism is based on the cooperation between the MAC layer and the physical layer in every single MS. In such a cross-layer approach, the MS may request the physical layer for some link-related parameters that characterize the current situation on the different channels. In the following sections, we propose an adaptive cross-layer scanning algorithm that uses physical layer information such as the channel load and the power measured on each channel in order to improve the scanning performance. This information provides the MAC layer with a preliminary knowledge of wireless deployment, so as to accurately select the channels to scan and the timer values. An implementation of two adaptive approaches in the *ath5k* IEEE 802.11 open-source driver is evaluated in different scenarios. The evaluation of the proposed cross-layer adaptive scanning considers the performance metrics described in Section 3.5.3. Note that in the case of the first discovery time metric, that has not been considered for the ADA approach, it becomes now important since, using the cross-layer information, the MS may stop the scanning process as soon as an AP is found. The trade-off between these performance metrics is identified and evaluated.

3.7.1 *Preliminary Considerations*

In Section 3.5.2, we have presented a set of experiments to analyze the delay of the first probe response (FRD). In particular, for the design and evaluation of the cross-layer adaptive scanning, we consider the experimental FRD distribution of Figure 31 and the results of Table 10, showing the mean and standard deviation of FRD for different levels of traffic that has been injected to the network. It can be appreciated that both the empirical mean ($\bar{E}[FRD]$) and standard deviation ($\bar{\sigma}(FRD)$) tends to increase as the traffic load increases. One relevant observation is illustrated in Figure 31, which shows the empirical cumulative distribution function of FRD ($P[FRD < t]$). The effect of loading the cell produces a great dispersion of the FRD. In the case of background traffic (i.e., only management frames circulate on the channel), almost no FRD dispersion is observed. However, when a high traffic load is injected (12.5 Mbps in Figure 31), the FRD tends to follow a displaced exponential distribution.

3.7.2 *Physical Layer Measurements*

In the proposed cross-layer approach, two physical parameters are considered: the *channel load* and the measured *power* on each channel. An MS can measure the power on the channel and the load in order to determine the presence of AP on the channels and to estimate the most suitable time to wait for Probe Responses while doing scanning. Regarding the power on each channel, it is simply measured from the captured signal during a given time window. On the other hand, the channel load is estimated by calculat-

ing the ratio between the signal-plus-noise and the noise-only samples. The estimation mechanism is presented below. This physical-layer information could be estimated by a wireless network card just before scanning. However, in the particular case of this experimentation, as it will be explained in Section 3.7.4, this physical-layer information is obtained using a dedicated device, a Universal Software Radio Peripheral (USRP2)[4], that allows sensing and processing IEEE 802.11-based physical signals.

3.7.2.1 *Estimation Mechanism*

For the channel load, we use the estimation mechanism that has been defined by Oularbi et al. [64]. The medium access control in IEEE 802.11 defines different Interframe-Spacing (IFS) intervals between two consecutive frames that guarantee different priorities to access the channel. At the receiver side, the observed signal is a succession of signal plus noise samples corresponding to data frames or noise samples corresponding to the IFS intervals or to idle periods.

We assume that there is only one data frame in the observation window. Let $\mathbf{y} = [y(1), \ldots, y(N_s)]$ be a set of N_s observations on a given channel, such that

$$\begin{cases} y(m) = w(m) & 1 \leqslant m \leqslant m_1 - 1 \\ y_i(m) = \sum_{l=0}^{L-1} h(l)x(m - m_1 - l) + w(m) & m_1 \leqslant m \leqslant m_2 \\ y(m) = w(m) & m_2 + 1 \leqslant m \leqslant N_s \end{cases} \tag{3}$$

where the x for $j = 1, \ldots, M$ is the data transmitted signal, $h(l)$ is the channel response from source signal to the receiver's antenna, L is the order of the channel h. $w(m)$ is a complex additive white Gaussian noise with zero mean and variance σ_w^2. The variance σ_w^2 is assumed to be known or at least estimated. In practice, the noise power is captured by the USRP2 device. A channel with no traffic and no active AP is first observed. In this case, no data signal is present and the only signal observed is due to thermal noise and background noise. Thus the noise power is equal to the variance of the observed samples.

The vector $\mathbf{y}$ can be divided into three parts : noise , signal plus noise and noise. Starting from the set of observation $\mathbf{y}$, the goal is to find which samples correspond to noise and which ones correspond to signal plus noise. The used approach relies on the following : since the samples are supposed to be independent in the noise areas and correlated in the signal plus noise area due to the channel effect and their OFDM structure, a likelihood function is used to provide information about the independence of the processed sample.

Let $\mathbf{Y}(u)$ in Equation 4 denote the following set of observations :

$$\mathbf{Y}(u) = [y(u), \ldots, y(N_s)] \qquad 1 \leqslant u < N_s \tag{4}$$

4 http://www.ettus.com/

And let us define f_Y (see Equation 5) the joint probability density function of $\mathbf{Y}(u)$. If $\mathbf{Y}(u)$ is composed of only noise samples

$$f_Y(\mathbf{Y}(u)) = \prod_{m=u}^{N_s} f_w(y(m)), \tag{5}$$

where f_w (see Equation 6) is the probability density function of a complex Normal law centered and variance σ_w^2, given by

$$f_w(x) = \frac{1}{\pi\sigma_w^2} e^{-|x|^2/\sigma_w^2}, \tag{6}$$

The log-likelihood that the vector $\mathbf{Y}(u)$ is formed of $(N_s - u)$ noise independent samples is expressed as in Equation 7.

$$\begin{aligned} \mathcal{J}(u) &= \log\left[\prod_{m=u}^{N_s} f_w(y(m))\right] \\ &= -(N_s - u)\log(\pi\sigma_w^2) - \frac{1}{\sigma_w^2}\sum_{m=u}^{N_s} |y(m)|^2 \end{aligned} \tag{7}$$

As u varies in the interval $[1, m_1)$, the number of noise samples composing $\mathbf{Y}(u)$ decreases and so does $\mathcal{J}(u)$ until it reaches a minimum bound at m_1 (see Fig 36).

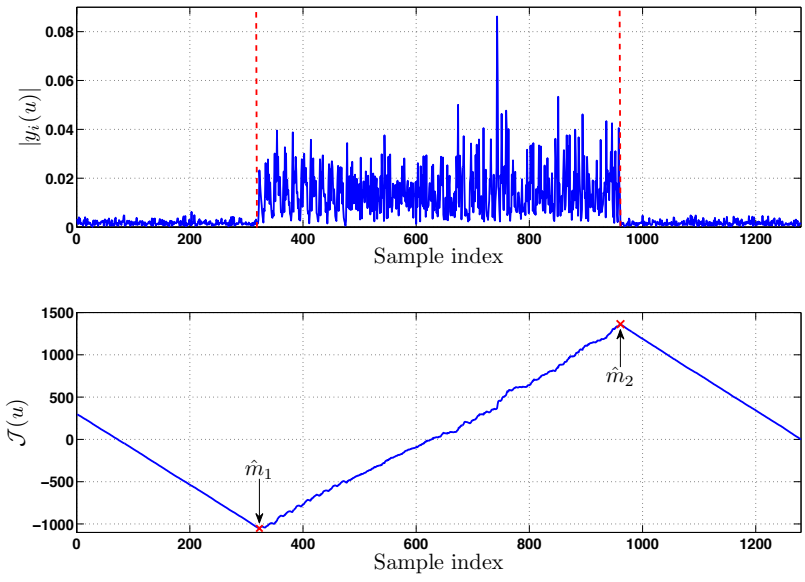


Figure 36: Example with one frame and corresponding criterion behaviour.

However, for u varying from m_1 to m_2 the number of signal plus noise samples decreases, therefore the ratio between noise samples and signal plus noise samples increases and by the way $\mathcal{J}(u)$ increases. It reaches its maximum value if and only if $\mathbf{Y}(u)$ contains only noise samples, i.e when $u = m_2$.

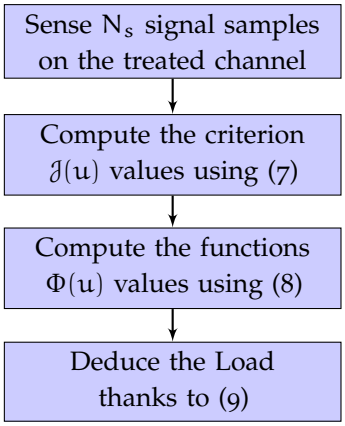


Figure 37: Estimation algorithm phases

Finally for $m_2 < u < N_s$, $\mathcal{J}(u)$ decreases again for the same reason than the one explained for $1 < u < m_1$.

Based on the behavior of $\mathcal{J}(u)$, it can be clearly seen in Fig 36 that the slope of $\mathcal{J}(u)$ is positive when u corresponds to the index of a signal plus noise sample and negative when u corresponds to the index of a noise sample. Therefore, the gradient of $\mathcal{J}(u)$ can be used to distinguish the nature of the observed samples. The function $\Phi(u)$ is introduced in Equation 8.

$$\Phi(u) = \frac{1}{2}\left[\text{sign}\{\nabla(\mathcal{J}(u))\} + 1\right] \tag{8}$$

Here ∇ denotes the gradient of $\mathcal{J}(u)$ and sign{.} denotes the sign operator. According to this, $\Phi(u)$ equals 1 when signal plus noise samples are present and zero when it is only noise, and the channel load (or occupancy rate) is estimated by $\widehat{C_{or}}$, as shown in Equation 9.

$$\widehat{C_{or}} = \frac{1}{N_s}\sum_{u=1}^{N_s}\Phi(u) \tag{9}$$

The whole algorithm is summarized in Figure 37.

3.7.3 *Algorithm Design and Implementation*

3.7.3.1 *Theoretical Analysis*

The proposed cross-layer scanning algorithm aims to obtain an expression to generate the waiting time on each channel. In Figure 31 the empirical FRD distribution is approximated using a random variable that follows a displaced exponential distribution. For the cross-layer adaptive scanning algorithm, a single-timer approach is considered (i.e., only t_{min} is used), that differs from the standard two-timers approach, (i.e., MinCT and MaxCT). A single-timer approach is also used in common wireless devices, like Android smartphones.

Let T be the FRD, then T is modelled as a displaced exponential distribution, i.e., $T \sim a + exp(\lambda)$. Remark that for each traffic load, an exponential law with a different parameter λ is obtained. The value of a is defined as the minimal observed time for a Probe Response to arrive. The goal is to find an expression that represents the amount of waiting time on each channel (t_{min}) that allows receiving a Probe Response with a given probability ($P[T \leqslant t_{min}] > p$). Then, the probability density function (see Equation 10) of the displaced exponential variable T is used to calculate probabilities. Then $P[T \leqslant t_{min}]$ can be expressed as shown in Equation 11.

$$f(t,\lambda) = \begin{cases} \lambda e^{-\lambda(t-a)} & t \geqslant a \\ 0 & t < a \end{cases} \tag{10}$$

Focusing on the side $t \geqslant a$, then we aim to find $T = t_{min}$ that satisfies $P[T \geqslant t_{min}] > p$. This yields to Equation 11.

$$\begin{aligned} P[T \leqslant t_{min}] &= \int_a^{t_{min}} \lambda e^{-\lambda(t-a)} dt \\ &= \lambda \int_a^{t_{min}} e^{-\lambda(t-a)} dt \\ &= -\lambda \frac{e^{\lambda(a-t)}}{\lambda} \Big]_a^{t_{min}} \\ &= 1 - e^{\lambda(a-t_{min})} \end{aligned} \tag{11}$$

Then, t_{min} can be expressed in terms of λ and p.

$$\begin{aligned} P[T \geqslant t_{min}] &> p \\ 1 - e^{\lambda(a-t_{min})} &> p \\ e^{\lambda(a-t_{min})} &< 1-p \\ \lambda(a-t_{min}) &< \ln(1-p) \\ t_{min} &> a - \frac{\ln(1-p)}{\lambda} \end{aligned} \tag{12}$$

Note that Equation 12 is an expression for t_{min} that depends on the parameter of the distribution (λ, which varies with the traffic load), the minimum observed FRD and the probability p, which represents the confidence interval (a grade of precision for the calculated t_{min}). Then, giving that the variance of an exponential random variable is well known ($\sigma^2 = 1/\lambda^2$), the parameter of the distribution (λ) can be estimated by using the empirical standard deviation ($\bar{\sigma}$) obtained in the preliminary experimentation (see Table 10). Finally, in Equation 13, an estimator for t_{min} is expressed considering $\widehat{\lambda} = 1/\bar{\sigma}$.

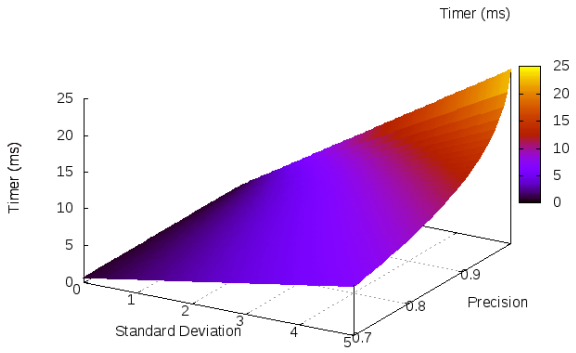


Figure 38: Timer values for different σ and p

$$\widehat{t}_{min} > a - \bar{\sigma}\ln(1-p) \tag{13}$$

This two-variable function gives values for t_{min} that are used on each channel depending on the standard deviation (σ) and the precision (p). Figure 38 illustrates the behavior of this function, the x-axis represents the standard deviation, the y-axis represents the precision and the z-axis gives the timer value (t_{min}). If the value of σ increases (i.e., the traffic load increases) the value of t_{min} linearly increases for a same value of p. Then, when increasing the confidence interval p, for a fixed value of σ, the value of t_{min} increases exponentially.

In summary, we propose a simple mechanism that gives the waiting time on a channel using two variables, p and σ. The value of σ is varied depending on the channel load. Then the value of p is set by considering other parameters, like the channel power or the priority of the channel.

3.7.3.2 *Implementation*

Based on Equation 13, the timer setting strategy is implemented in the ath5k open-source driver as follows:

```
Timer = FRD_min + Deviation * Precision
```

Where the `FRD_min` component is the absolute minimum observed FRD, `Deviation` is the empirical standard deviation of the FRD (Table 10) and `Precision` is the calculation of the term $-\ln(1-p)$ for different values of p. Using different `Precision` values, the timer for each channel can be increased or decreased. `Precision` values are indicated in Table 13 and have been set after several experimental trials, in order to find the best ones in terms of the scanning performance. In general, higher values of p are used, i.e., higher `Precision` for the first channels to scan, since, as it has been observed in Figure 39, the power level generally indicates the presence of AP in the channel. For that reason, a higher precision value (i.e., a longer timer)

is given to channels with relative high power in order to avoid missing a response due to a scarce timer. Channels with relative low power will use shorter timers (due to a smaller `Precision` value), since an AP is less likely to be found. Two algorithms have been envisaged, the main difference between them is related to how they consider the measured channel power to calculate the channel sequence.

SIMPLE PRECISION ALGORITHM The Simple Precision Algorithm (SPA) was implemented using the timer calculation mechanism of Equation 13 with a `FRD_min` equal to 1.69 ms, since it has been the minimum FRD we have observed in the preliminary experiments. Then, the `Deviation` term was implemented considering an extrapolation of the empirical relation between the channel load and the standard deviation of the FRD (Table 10). In this case, the channel load values are used as an input, separated in steps of 5 %. The channel sequence was calculated by simply ordering the channels by power (decreasingly), i.e., different values of p (i.e., different `Precision` values) are used for different position in the channel sequence (the first channels have greater p).

LOCAL MAXIMUM PRECISION ALGORITHM The Local Maximum Precision Algorithm (LMPA), takes into account the problem of channel overlap. As shown in Figure 39, even if only channels 1, 6 and 11 have an AP deployed, the power measured in the neighboring channels is still high due to channel overlap. As it has been previously explained in Section 2.1.1, this is due to the channel allocation in IEEE 802.11, where channels are 20 MHz or 22 MHz wide, with a separation of 5 MHz. In order to prevent from considering high power channels that do not have any AP deployed, instead of simply ordering the channels by decreasing power, we propose to use a high `Precision` value for the local maximums in terms of measured power. In the example of Figure 39, the sequence will first consider channels 1, 6 and 11, which are the local maximums. Note also that the power levels of channels 1, 6 and 11 are quite different among them. This can be explained by the fact that different AP brands and models have been used, which may yield in different radio modules, antennas, etc.

Table 13: Precision values

p	`Precision` $(-\ln(1-p))$	No of Channels to scan
0.95	2.996	3 (or all local maximum)
0.85	1.897	3
0.80	1.609	3
0.75	1.386	4

Figure 40 illustrates the cross-layer adaptive algorithms logic. The MS first computes the channel sequence depending on the approach (SPA or LMPA)

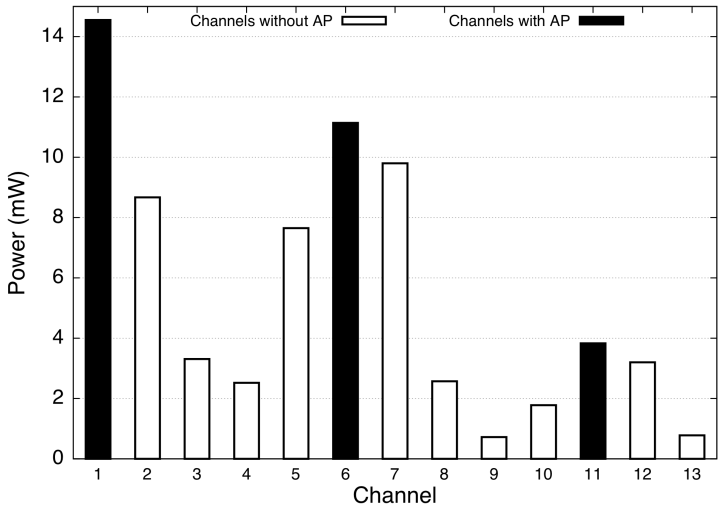


Figure 39: Measured power in scenario 2

by considering the power measured on the channels (`Ch_Power_List`). For each channel to scan, the MS calculates a timer (`T[i]`) based on the channel load (`Ch_Load[i]`) and the precision (obtained after computing the channel sequence using `Ch_Power_List`). The scanning process finishes when the number of scanned channels reaches `MAX_CH` (whose value depends on the number of channels in the sequence).

3.7.4 *Performance Evaluation*

3.7.4.1 *Testbed*

To evaluate the performance of SPA and LMPA, a testbed using real IEEE 802.11b/g devices has been deployed. An MS implementing a Netgear WG511T card based on an Atheros chipset is used for scanning. We used a modified version of the ath5k driver that allows controlling all the parameters of the active scanning function. Up to five Linksys WRT54GL AP were deployed in different channels (see Section 3.7.4.2) in an isolated environment without interferences from any other wireless device. Traffic load was generated using iperf on a set of DELL Latitude laptops and ASUS netbooks. Since the Atheros card using ath5k driver does not allow physical-layer measurements in the MAC layer, a dedicated USRP2 device has been used. This device allows sensing and processing IEEE 802.11-based signals. The performance of the scanning algorithms are evaluated in two steps. First, after deploying the scenario (AP and MS for traffic generation) the channel load and the power are estimated as explained in Section 3.7.2 over signal samples gathered by the USRP2 device. Second, both parameters are statically set on the ath5k driver before scanning a particular scenario using SPA, LMPA and a set of fixed-timer scanning algorithms. The scenario's conditions remain constant during the measurement and the scanning phase. Note that these measurements could

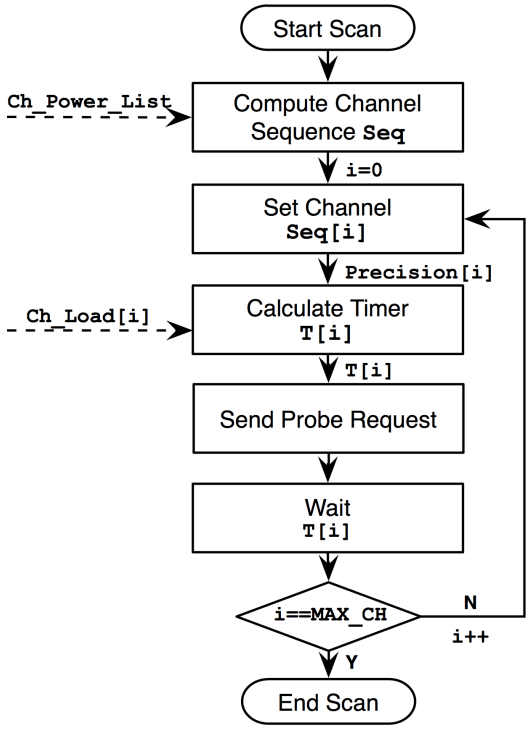


Figure 40: Cross-layer adaptive algorithm

also be performed before triggering the handover, in periodic sub-phases. Moreover, physical-layer information could be directly requested by the MAC layer using IEEE 802.11k to an AP or an MS by simply using Channel Load Request/Report and Link Measurement Request/Report messages.

We consider the metrics introduced in Section 3.5.3 to compare the performance of the different evaluated scanning algorithms. As stated before, these metrics define a trade-off, since using a fixed timer for scanning all channels cannot simultaneously keep a reduced latency, provide a high discovery rate, reduce the failure rate and spend a low time to discover the first AP.

3.7.4.2 *Scenarios*

Like for the ADA evaluation, we consider a set of scenarios aiming to represent real deployments. Five different scenarios have been considered, using different channel allocations and traffic load. The scenarios are as follows:

- **Scenario 1:** Three AP deployed in channels $1-6-11$ (one on each channel). This could be an example of an enterprise or campus non-overlapping deployment. Uplink and downlink traffic from 5 to 15 Mbps is injected in all channels.
- **Scenario 2:** Five AP deployed in channels $1-6-11$. One AP in channel 1, and 2 AP in channels 6 and 11. This is another non-overlapping scenario. Uplink and downlink traffic from 5 to 15 Mbps is injected in all channels.

- **Scenario 3:** Five AP deployed in channels $3-4-8-9-13$. This could be an example of an heterogeneous city-wide deployment. Uplink and downlink traffic from 1 to 15 $Mbps$ is injected in all channels.
- **Scenario 4:** Two AP deployed in channels $1-11$. This is another non-overlapping scenario (separation of 9 channels). Uplink and downlink traffic from 5 to 10 $Mbps$ is injected in both channels.
- **Scenario 5:** One AP deployed in channel 6. This is a common non populated scenario. Downlink traffic of 20 $Mbps$ is injected in the channel.

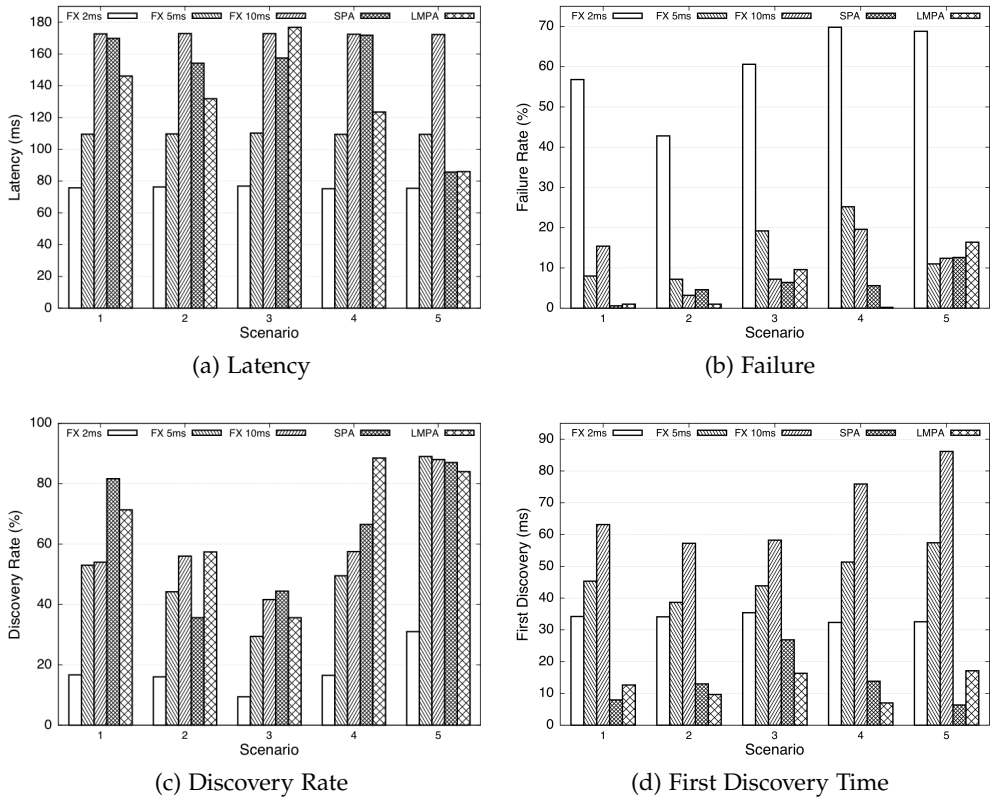
3.7.4.3 *Results*

GENERAL RESULTS Five scanning approaches have been considered, three fixed strategies using a single fixed timer (FX 2 ms, FX 5 ms and FX 10 ms) and two cross-layer adaptive approaches (SPA and LMPA), in the five different scenarios described above.

For the fixed timers approaches (FX), we consider that there is not a-priori information to build the channel sequence. Then, the channel sequence is always randomly calculated on each scanning trial. This is to avoid that a pre-established channel sequence penalizes or benefits a scenario. Regarding the timers, note that they are lower than the timers used in the ADA evaluation, which used different devices for the testbed.

For the two adaptive approaches, the t_{min} timer, calculated using Equation 13, varies between 2 ms to 18 ms, depending on the traffic load estimation and the measured power on each channel. Also in this case, the standard deviation for all the performance metrics is always comparable to fixed timers strategies. Focusing on the scanning latency, Figure 41a shows that the cross-layer adaptive approaches give a latency lower than or equal to the FX 10 ms strategy, being always between 85 ms and 176 ms. Moreover, LMPA can reduce the latency in non-overlapping scenarios 1, 2 and 4, since it prevents from setting a high timer in the neighboring channels due to a high power. Regarding the failure (see Figure 41b), the adaptive strategies always give reduced rates, between 0.2 % and 16 %. Only in scenario 5 LMPA failure is slightly higher than the FX 10 ms approach. Figure 41c shows that the adaptive approaches keep a high discovery rate (up to 84 %) even for scenarios where the latency is lower than the fixed timers strategies. Finally, focusing on the first discovery time, Figure 41d shows that both SPA and LMPA always discover the first AP sooner than all of the fixed timer approach. The first discovery time for the adaptive approaches varies between 6.35 ms and 26.87 ms, but for fixed timer approaches it can reach up to 86.14 ms in average.

COMPARATIVE RESULTS In order to provide a comparative view for the metric results, we define a simple score function (see Equation 14), where D



(a) Latency (b) Failure

(c) Discovery Rate (d) First Discovery Time

Figure 41: Experimentation results

is the discovery rate, L is the latency, F is the failure and T defines the first discovery time. All metrics are equally considered.

$$S_i = 1 - \frac{D_i}{max(D_i)} + \frac{L_i}{max(L_i)} + \frac{F_i}{max(F_i)} + \frac{T_i}{max(T_i)} \tag{14}$$

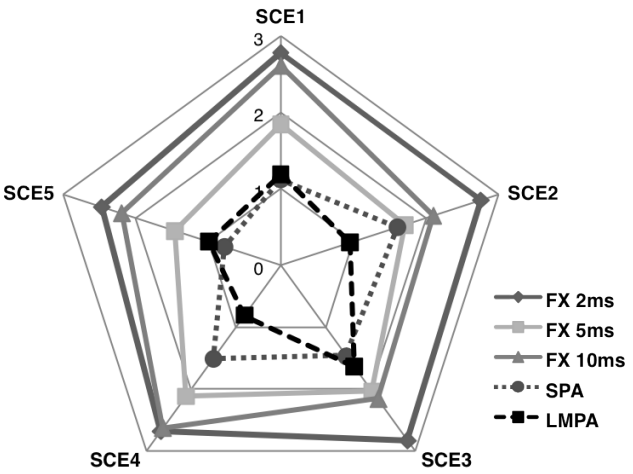


Figure 42: Score function

For each approach (i) a score is assigned. The approach managing better the trade-off between the performance metrics is the one that tends to minimize the score function (S_i). Figure 42, illustrates the score functions for each scenario. This figure gives a global view of each approach and also illustrates how a particular scenario influences the scanning performance. The adaptive cross-layer approaches minimize the score in every scenario, since both SPA and LMPA curves are closer to the origin. Moreover, LMPA gives better scores than SPA in scenarios 2 and 4, since it allows setting longer timers only on channels where an AP is deployed. Regarding fixed timers approaches, FX 5ms behaves better than the rest of the fixed strategies. The adaptive approaches are capable to behave differently for each particular scenario, by taking into account its specific constraints in terms of interference and traffic load. They help to keep a reduced latency and failure rate while also importantly reducing the time to discover the first AP, and giving high discovery rates. Remind that only one timer (t_{min}) is used in the five evaluated approaches. Although this is not the case of the standard specification, we focus on the most critical timer and we expect that a second timer ($MaxCT$) may only improve the discovery rate in a fixed or adaptive approach, by increasing the latency in both cases.

3.7.5 *Discussion*

We proposed and evaluated a cross-layer mechanism to improve the IEEE 802.11 scanning process in layer-2 handovers. We have shown that scanning timers can be adapted using samples of signal power and congestion levels. Like in ADA, we have evaluated the scanning process in terms of a trade-off between different performance metrics. Since an optimal timer value for scanning (i.e., the one to wait for Probe Responses) depends on each particular AP deployment, we proposed and evaluated two different adaptive algorithms, SPA and LMPA, that consider the same physical layer information. These cross-layer adaptive approaches ensure that each timer matches the characteristics of each channel.

The main limitation of a cross-layer scanning approach is how the physical layer information is shared with the management functions at the MAC layer. In our evaluation, we gather physical layer information with a different device than the one that is performing scanning. In our experiments, the load and the power are estimated in a separate phase by gathering signal samples for each individual channel during 1 to 10 ms using the USRP2 device. In order to reduce this phase in the case of a real implementation of the estimation mechanism in a wireless card, we are actually studying a new mechanism that takes advantage from the channel overlap to infer the load and the power of neighboring channels by only considering physical layer measurement of a limited number of channels that partially overlap (e.g., channels 3, 5, 7 and 9). We have used the USRP2 device since in existing IEEE

802.11 wireless cards signal samples are not available at the driver level and so the power and load estimation cannot be assured. This limitation could be overcome by using the IEEE 802.11k amendment, providing radio resource management for WLAN or by implementing the estimation mechanism at the firmware level. Such a protocol allows requesting physical layer measurements to other nodes in the network.

3.8 CONCLUDING REMARKS

In this chapter, the handover process in IEEE 802.11 has been presented together with existing optimizations focusing on reducing the impact of handover on the user experience. Handovers in IEEE 802.11 have become an essential issue, since, as shown in Chapter 2, the number of IEEE 802.11 networks have dramatically increased, giving high dense deployments in urban deployments. The challenge for IEEE 802.11 networks is to evolve from providing static wireless connections to provide seamless mobility, and in this context, a fast and reliable handover support arises as one of the most important issue.

A special attention has been given to the AP discovery process, the most time consuming process while performing a handover. As a result, a performance trade-off for scanning has been defined and analyzed: while scanning, the MS aims to discover the largest number of candidate AP by spending a low delay to discover them and avoiding the case in which no candidate is found due to misconfiguration of the scanning parameters. We have proposed two different approaches to manage the performance trade-off while scanning. First, ADA is a simple adaptation function for scanning timers that is based only on local information, i.e., it decides to use longer or shorter timers for scanning the different channels depending on previously found AP. The second approach, is based on a tight collaboration between the physical and the MAC layers in IEEE 802.11. It has been demonstrated that the average time to receive the first Probe Response in a channel while scanning and its dispersion increase with the AP load. The proposed cross-layer approach is capable to adaptively select the channels to scan and the time to spend on a channel in order to better manage the performance trade-off. Both approaches have been evaluated under experimental testbeds, using different type of devices and open-source wireless drivers.

4

AN ENERGY-EFFICIENT APPROACH FOR NETWORK SELECTION

4.1 MOBILITY AND MULTI-HOMING IN A WIRELESS CONTEXT

Nowadays, users run different kind of applications over the Internet: instant messaging, mailing, voice-over-IP, video-on-demand , web-browsing or social networking. These applications generate an important number of flows and gather a high variety and amount of information. Moreover, users want to be always best connected [5] when running their applications, receiving the best possible performance at any time while moving. Different types of portable devices have been introduced in the market in the last years, and very frequently, they embed different wireless access technologies, like IEEE 802.11 (WiFi), 2G/3G/4G cellular, WiMAX or Bluetooth interfaces. In general, these access technologies have been designed to support different use-cases, e.g., cellular networks for large coverage and high speed mobility communications, IEEE 802.11 networks for local area communications or Bluetooth for proximity services. As previously shown in Chapter 2, due to the increasing number of deployments of these networks and because no wireless technology could cover all market needs, every single MS is able to access one or several networks in any given place. Then, it is usual that users associate with more than one network, increasing the number of IP addresses allocated by user and opening the way to multi-homing, which gives users the possibility to exploit the diversity of their networks by gaining in reliability and performance. In particular, the IPv6 protocol allows flow distribution for a very high number of connected nodes implementing different link-layer technologies. However, in a mobile and multi-homed environment (i.e., in which multiple wireless interfaces are available on a single mobile device), the definition of the policies establishing how the different flows are mapped to the different available interfaces/paths is still an open research topic. We generally refer to this issue as the *network selection* process. In the literature, solutions for network selection follows two different directions. First, there are solutions aiming to manage vertical handovers, i.e., the decision-making process to establish when to switch all the on-going flows to a different interface. A typical example for vertical handover is an MS connected to a 3G network aiming to switch all the flows when entering in a WLAN covered zone. A second group of solutions for network selection consider the decision-making problem aiming to assign the different flows to the available interfaces, enabling load spreading and a simultaneous usage of the wireless interfaces.

As it will be exposed in this chapter, we present solutions for vertical handover and load spreading but we particularly focus on the decision-making process for the flow-interface assignation. Note that the flow-interface assignation strategy in the network selection process can highly impact the user experience. The lack of an intelligent decision-making process can lead, for example, to situations where some wireless interfaces become overloaded while other wireless interfaces having an acceptable available capacity are not even used. Also, it may produce that high bandwidth demanding application flows are assigned to high energy consuming interfaces, which drastically drain the MS battery.

In this chapter, a decision-making framework for network selection is defined and evaluated. This framework allows searching an optimal flow-interface assignation in a multi-homed MS. The optimality of this assignation is measured in terms of the bandwidth satisfaction of a flow using a particular interface and its energy efficiency. The flow-interface assignation is modelled as a multi-objective optimization problem, and we propose using evolutionary algorithms to search for optimal solutions.

The chapter is organized as follows. The related work on decision-making mechanisms for network selection is introduced in Section 4.2, including Multi-Attribute Decision Making, Neural Networks, Combinatorial and Multi-Objective optimization. Then in Section 4.3, an energy-efficient network selection mechanism is defined and modelled as a multi-objective optimization problem. We propose a set of simulation results in Section 4.4. The proposed simulator generates random scenarios and solves the multi-objective problems using the PISA [65] framework for evolutionary computation. We compare the results obtained for such a multi-objective approach against existing preference-based techniques. Finally, the chapter is concluded in Section 4.5.

4.2 DECISION MAKING FOR NETWORK SELECTION

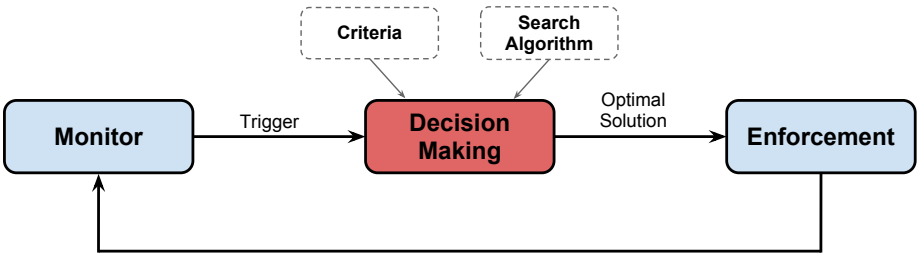


Figure 43: Network selection process

The network selection in wireless multi-homed devices implies different processes, as illustrated in Figure. 43. As stated before, the final goal while performing network selection in a load spreading approach is to find the assignation of the on-going application flows to the different available interfaces that optimizes a given criteria. First, in order to identify when the MS

needs to look for a new assignation one may suppose that the MS needs to *monitor* the current status of the flows and the interfaces, while also doing network discovery to find the available points of attachment. Then, under certain conditions, the MS may decide that the current assignation is no more optimal and so a *decision-making* process has to be triggered in order to find a new optimal assignation. The conditions to trigger the decision-making process may be related, for instance, to some interfaces or networks becoming active or inactive, to the initialization or termination of a new application flow or some QoS variations on the available networks or the application flow requirements. The decision-making process itself has to consider a set of *criteria* to evaluate the optimality of the assignation. Finally, an *enforcement* process is carried out to set up the new assignations on the device. During this enforcement process, the MS connects to new interfaces or turns-off other interfaces that will no longer be used in the new flow-interface assignation. It also considers the seamless switching of some flows to different interfaces based on some existing multi-homing support protocols like shim6 [6] or Host Identity Protocol (HIP) [7].

The decision-making process itself is the core mechanism that will provide the most optimal assignation as an output. To achieve this, at least the following issues have to be considered:

1. Which criteria has to be considered to measure the optimality of a certain flow to be transmitted over a certain interface ?
2. How to combine the different criteria in a way to represent a realistic decision-making ?
3. How to manage the combination of contradictory criteria ?
4. Once the criteria has been combined, how to search for optimal solutions ?
5. How to select a single optimal solution if different optimal solutions exist ?

Different works in the literature propose a set of frameworks, models and mechanisms to give response to one or more of the aforementioned questions. In the following sections, a number of mechanisms are described.

4.2.1 *Multi-Attribute Decision Making*

Multi-Attribute Decision Making (MADM) [66] is a technique used for decision-making based on preference decisions over an available set of possible alternatives. These alternatives are usually characterized by multiple discordant attributes, that are measured in different units or, in some cases, some attributes may have incommensurable units. As an example, in the context of

network selection, one may consider that an alternative is a particular flow-interface assignation (i.e., in the case of load spreading) or the interface to use by default for all the flows in the MS (i.e., in the case of vertical handover). Each assignation may have attributes, e.g., QoS satisfaction, interface and flow characteristics, security issues, associated cost.

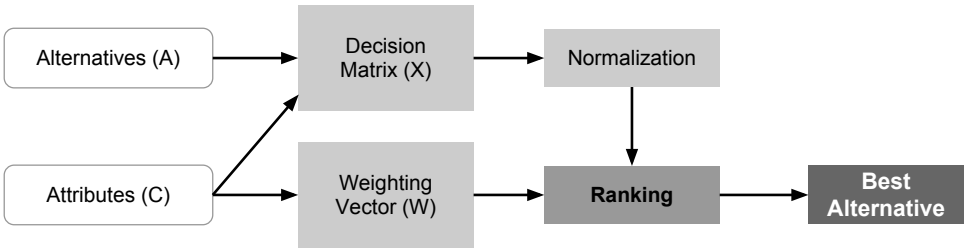


Figure 44: MADM flow chart

A generic flow chart for MADM is illustrated in Figure 44, describing the different phases to find the best alternative. More formally, an MADM problem may be modelled as a set of m alternatives, A, as shown in Equation 15 [66]:

$$A = \{a_1, a_2, a_3, \cdots, a_{m-1}, a_m\} \tag{15}$$

And a set of n attributes C in Equation 16

$$C = \{c_1, c_2, c_3, \cdots, c_{n-1}, c_n\} \tag{16}$$

In every MADM mechanism, a weight vector W is defined (see Equation 17) in order to express the relative importance of each attribute in the problem, satisfying the following condition: $\sum_{i=1}^{n} w_i = 1$.

$$W = w_1, w_2, w_3, \cdots, w_{n-1}, w_n \tag{17}$$

Additionally, a decision matrix X, in Equation 18, contains the elements x_{ij} representing the performance of each alternative a_i for each attribute c_j where $i \in [1, m]$ and $j \in [1, n]$.

$$X_{m,n} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix} \tag{18}$$

Then, after defining the weight vector W and the decision matrix X, the goal is to combine them to come out with a single optimal alternative. This

process is composed of two different phases. First, since the different attributes are certainly measured in different units, a *normalization* phase is required. Then, an *aggregation* process is performed in order to combine normalized attributes with the weight vector W, and so rank the different alternatives to come out with a decision.

Regarding normalization, different methods may be applied [67], mainly based on linear-scale transformation. Let r_{ij} be the normalized performance x_{ij} forming the normalized matrix R. In the following, we describe the different methods to obtain r_{ij}.

The *Vector Normalization* given in Equation 19, also referred to as Euclidean Normalization, is based on dividing each performance value x_{ij} by its norm (i.e., the square root of the sum of squared performances for the n alternatives).

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \tag{19}$$

In the *Max Method*, each performance value is divided by the maximum performance value among all possible alternatives. Equation 20 shows the normalization formula for benefit attributes (i.e., attributes that should be maximized), while Equation 21 shows the formula for cost attributes (i.e., attributes that should be minimized).

$$r_{ij} = \frac{x_{ij}}{max(x_j)} \quad \text{(for benefit attributes)} \tag{20}$$

$$r_{ij} = 1 - \frac{x_{ij}}{max(x_j)} \quad \text{(for cost attributes)} \tag{21}$$

The *Max-Min Method* (see Equations 22 and 23), has the advantage of producing normalized performance values in the range $[0, 1]$, which facilitates comparisons. It is obtained by dividing the difference between the performance x_{ij} and the minimum for all the alternatives (for the benefit attributes) or the difference between the maximum performance and x_{ij} (for cost attributes) by the difference between the maximum and minimum performance for all the alternatives.

$$r_{ij} = \frac{x_{ij} - min(x_j)}{max(x_j) - min(x_j)} \quad \text{(for benefit attributes)} \tag{22}$$

$$r_{ij} = \frac{max(x_j) - x_{ij}}{max(x_j) - min(x_j)} \quad \text{(for cost attributes)} \tag{23}$$

Finally, the *Sum Method* (see Equation 24) divides the performance value by the sum of the performance values among all the alternatives.

$$r_{ij} = \frac{x_{ij}}{\sum_{j=1}^{n} x_{ij}} \tag{24}$$

Once the MADM problem is defined, i.e., the decision matrix (X) has been defined and normalized, different *aggregation* techniques exist to combine the normalized decision matrix (R) and the weight vector (*W*) to outcome with the most feasible alternative. In the context of network selection, the final goal is to select one of the interfaces (in a vertical handover context) or flow-interface assignations (in a load spreading context) among all the available alternatives. Different works in the literature have modelled the network selection problem using MADM considering different attributes (C), weight vectors (*W*) and aggregation methods. In the following sections, the different aggregation methods are described. Then, existing network selection methods based on MADM are discussed.

4.2.1.1 *Simple Additive Weighting*

In the Simple Additive Weighting (SAW) aggregation method, each alternative is rated by calculating the weighted sum of its attributes. More formally, let a_i^{SAW} be the rating of the alternative a_i using SAW. Then, a_i^{SAW} is calculated as shown in Equation 25.

$$a_i^{SAW} = \sum_{j=1}^{n} r_{ij} \cdot w_j \tag{25}$$

Then, the optimal alternative s^{SAW}, is the one that maximizes the weighted sum: $s^{SAW} = \max_{\forall i}(a_i^{SAW})$.

4.2.1.2 *Multiplicative Exponential Weighting*

In the Multiplicative Exponential Weighting (MEW), the rating of each alternative a_i^{MEW} is calculated as the exponentially weighted product of the attributes (as in Equation 26). In this case, as weights are exponentially considered, they are positive for benefit values ($x_{ij}^{w_j}$) and negatives for cost attributes ($x_{ij}^{-w_j}$). Note that, in the case of MEW, normalization is not required since attributes are combined in a product rather than a sum.

$$a_i^{MEW} = \prod_{j=1}^{n} x_{ij}^{w_j} \tag{26}$$

In contrast to SAW, a_i^{MEW} values have not an upper bound when using exponential weights [66]. Then, to come out with the optimal alternative (s^{MEW}), it is necessary to compare each alternative with an ideal performance vector $a^* = (x_1^*, x_2^*, \cdots, x_m^*)$ representing an ideal alternative,

that is defined as an imaginary alternative composed of the best values for each individual attribute, i.e., $a_j^* = \max_{\forall i} x_{ij}$ for benefit attributes and $a_j^* = \min_{\forall i} x_{ij}$ for cost attributes. Finally, the optimal alternative s^{MEW} is calculated in Equation 27, with ρ_i defined in Equation 28 as the ratio between the MEW rating of the evaluated solution a_i and the ideal solution a^*.

$$s^{MEW} = \max_{\forall i} \rho_i \tag{27}$$

$$\rho_i = \frac{\prod_{j=1}^{n} x_{ij}^{w_j}}{\prod_{j=1}^{n} (x_{ij}^*)^{w_j}} \tag{28}$$

4.2.1.3 *Technique for Order Preference by Similarity to Ideal Solution*

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is an MADM method proposed by Yoon and Hwang [66]. Optimal solutions calculated by TOPSIS are those having the shortest distance to the positive ideal solution and, at the same time, the farthest from the negative ideal solution.

In order to solve an MADM problem using TOPSIS, once having defined the decision matrix X, Euclidean normalization is applied to obtain the normalized R matrix. Then, a new matrix V is calculated by computing the product $V = W^T \times R$ (see 29).

$$V = W^T \times R = \begin{pmatrix} w_1 r_{1,1} & w_2 r_{1,2} & \cdots & w_m r_{1,n} \\ w_1 r_{2,1} & w_2 r_{2,2} & \cdots & w_m r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_1 r_{m,1} & w_2 r_{m,2} & \cdots & w_m r_{m,n} \end{pmatrix} \tag{29}$$

Each element v_{ij} of matrix V represents the weighted normalized performance of the alternative a_i with regard to the attribute c_j. Then, in order to determine the ideal positive V^+ and negative V^- solutions, consider J the set of benefit attributes and J' the set of cost attributes. Then, the positive ideal alternative V^+ is defined in Equation 30 and the negative ideal alternative V^- is defined in Equation 31.

$$V^+ = \{v_1^+, \cdots, v_m^+\} \text{ where } v_j^+ = \{\max_{\forall i}(v_{ij}) \text{ if } j \in J; \min_{\forall i}(v_{ij}) \text{ if } j \in J'\} \tag{30}$$

$$V^- = \{v_1^-, \cdots, v_m^-\} \text{ where } v_j^- = \{\min_{\forall i}(v_{ij}) \text{ if } j \in J; \max_{\forall i}(v_{ij}) \text{ if } j \in J'\} \tag{31}$$

Then, for each alternative, the distance from the positive ideal alternative (S_i^+, in Equation 32) and the negative ideal alternative (S_i^-, in Equation 33)

are calculated to finally compute the relative closeness to the ideal solution γ_i (Equation 34). TOPSIS finally selects the optimal alternative as the one with γ_i closest to 1.

$$S_i^+ = \sqrt{\sum_{j=1}^{m} (v_{ij} - v_j^+)^2} \tag{32}$$

$$S_i^- = \sqrt{\sum_{j=1}^{m} (v_{ij} - v_j^-)^2} \tag{33}$$

$$\gamma_i = \frac{S_i^-}{S_i^+ + S_i^-} \tag{34}$$

4.2.1.4 *Analytic Hierarchy Process*

The Analytic Hierarchy Process (AHP) [68], proposed by Thomas Saaty, is an MADM method to solve complex decision problems, having a large number of alternatives and attributes. The versatility of AHP allowed its application in different fields, like in economics, management and education, among others. In general, AHP allows prioritizing the different attributes (i.e., finding the weights vector *W*) in a pairwise comparison manner. AHP is based on decomposing a complex problem in a hierarchy of simpler sub-problems (i.e., the decision factors). This hierarchy is modelled as illustrated in Figure 45.

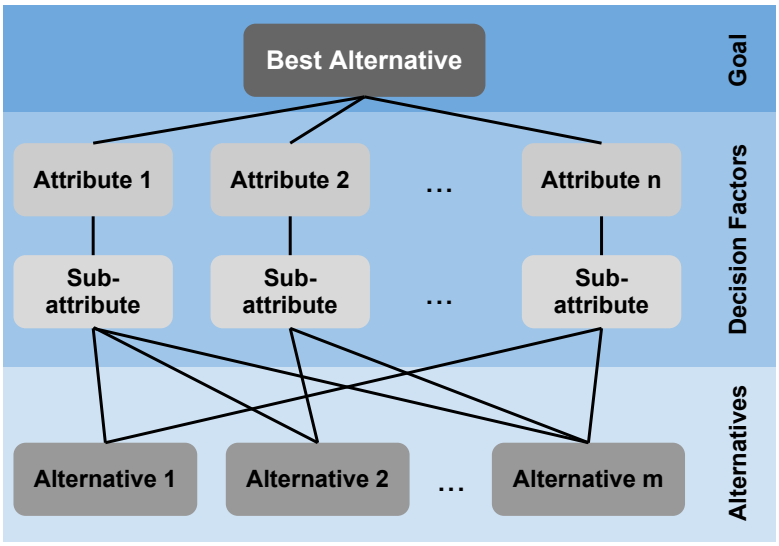


Figure 45: The Analytic Hierarchy Process

The final goal, i.e., choosing the best alternative, is placed at the top of the hierarchy. Then, in the middle of the hierarchy, the decision factors are placed and finally, at the bottom, the different alternatives are placed. After

decomposing the problem, the different decision factors within the same parents are being compared among them using pairwise comparison. That is, for two given factors, it has to be decided which one of them is more important than the other and by how much (using a 1 to 9 numeric scale). After completing the pairwise comparison, a reciprocal symmetric matrix M $(n \times n)$ is obtained (see Equation 35). Each element of M has been calculated using pairwise comparison, representing a ratio between weights of different factors with respect to their parents.

$$M = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & w_1/w_2 & \cdots & w_1/w_n \\ w_2/w_1 & 1 & \cdots & w_2/w_n \\ \vdots & \vdots & \ddots & \vdots \\ w_n/w_1 & w_n/w_2 & \cdots & 1 \end{pmatrix} \tag{35}$$

As demonstrated in [69], for a reciprocal matrix, Equation 36 is satisfied. Where the *eigenvector* V is a non-zero vector and λ is the *eigenvalue*.

$$MV = \lambda V \tag{36}$$

In the case of the matrix M, the weights of the decision factors w_i $i \in [1, n]$, that are the elements of vector W, can be calculated by considering $V = W$ and $\lambda = n$ [69]. The obtained weight vector can suffer from inconsistencies originated by pairwise comparisons misjudgements. To address this issue, a consistency ratio is defined so that the weighting vector is consistent if this ratio does not exceed 10 % (details on the consistency ratio calculation can be found in [68]).

4.2.1.5 *Grey Relational Analysis*

The Grey Relational Analysis (GRA) [70], similarly to the TOPSIS algorithm, analyzes the level of similarity and variability of the different alternatives. The problem is modelled as any MADM approach, having a set of n alternatives A and a set of m attributes C. To perform the best alternative selection, three phases are required in GRA: normalization, definition of the ideal alternative and calculation of the Grey Relational Coefficient (GRC). The normalization procedure follows the rules of the Max-Min method (see the normalization methods in Section 4.2.1), giving normalized performance values r_{ij}. Then, the ideal alternative $S^* = \{r_0^*, \cdots, r_m^*\}$ is calculated as the alternative containing the best values for each attribute. Finally, the GRC index for each alternative, GRC_i is calculated as shown in Equation 37.

$$GRC_i = \frac{1}{m} \sum_{1}^{m} \frac{\min_{\forall i}(\delta_i) + \max_{\forall i}(\delta_i)}{\delta_i + \max_{\forall i}(\delta_i)} \text{ where } \delta_i = |r_j^* - r_{ij}| \tag{37}$$

The best alternative is simply chosen by maximizing the coefficient, i.e., $\max_{\forall i}(GRC_i)$.

4.2.1.6 *Existing MADM Mechanisms for Network Selection*

After introducing the most popular MADM techniques, we aim now to present the application of MADM particularly for the network selection process. Most part of existing MADM-based mechanisms in the literature have been proposed to support vertical handover in multi-homed mobile devices.

TOPSIS-BASED NETWORK SELECTION In [71] and [72] two different network selection mechanisms are defined based on TOPSIS. Savitha and Chandrasekar [71] propose a network selection scheme to support vertical handovers, i.e., the transition between two access networks belonging to different technologies. In such a mechanism, they consider as the alternatives the different available access networks at a given time. They consider jitter, cost, bandwidth and delay as attributes for the alternatives and a set of weights, that unfortunately are not clearly specified. The best network is then obtained using TOPSIS. On the other hand, Bari et al. [72] proposed to use a variation of TOPSIS, namely Iterative-TOPSIS, to tackle the ranking abnormalities observed when using traditional TOPSIS. A ranking abnormality [73] occurs when the best selected alternative changes if an alternative (different to the best one) is replaced by a worse alternative without changing the weights. Ranking abnormality in the context of a network selection problem is also identified in [74], where the authors show that SAW and MEW provide more stable solutions when removing an alternative different from the best one. Moreover, Triantaphyllou and Shu [75] have shown that in TOPSIS, these abnormalities increase with the number of alternatives and attributes. To avoid this problem, the authors propose an MADM problem similar than in [71] but using the monetary cost per byte, the total bandwidth of the network, the allowed bandwidth for the MS using a network, the network load, the packet delay and jitter as the set of attributes to characterize each alternative. Regarding the definition of weights, they consider the user QoS subscription. For example, a "Bronze" subscription indicates that more priority (i.e., higher weights) is given to cost rather than QoS attributes. Using Iterative-TOPSIS, once the best alternative has been selected (i.e., a problem iteration), the worst alternatives are removed from the set. Simulation results show that even if Iterative-TOPSIS can manage ranking inconsistencies, for the first iterations, there are still some ranking abnormalities that cannot be eliminated.

COMBINED MADM NETWORK SELECTION A different MADM approach is proposed by Puttonen et al. [76], as a part of the VERHO vertical handover framework. In this case, the network selection is also modelled as an MADM problem considering a set of alternatives corresponding to the available interfaces, a set of attributes and a dynamic weighting scheme. With regard to attributes, the signal strength, the bit-rate, the power consumption of the interface, the cost of using the network, the coverage range and the

security level are considered. However, differently from the previous mechanisms, attributes are considered in a *fuzzy* manner. In such a Fuzzy Logic approach [77], attributes are no longer considered as absolute values but as linguistic variables (e.g., low, medium, high) that can be easily transformed to a 0 to 1 range through a membership function obtaining a fuzzy set. Then, instead of dealing with values, units and normalization, decisions can be directly taken combining the fuzzy sets with logic operators. In [76], power consumption, cost , coverage range and security are fuzzified in a scale of length 5 (very low, low, medium, high, very high). The authors propose a comparative study of SAW, MEW and TOPSIS aggregation mechanisms against some fixed policies for network selection, such as best bit-rate, best signal strength and a fixed priority list (i.e., in this order: Ethernet, WLAN, Bluetooth, Cellular). They also consider three weighting vectors (referred to as "profiles") that are dynamically chosen depending on the user's and applications' requirements; one of them prioritizing signal strength and bit-rate, the second one prioritizing mobility-related attributes and the third one being neutral (i.e., equal distribution of weights). A set of simulations is presented and authors conclude that due to the large difference observed in the selected network using different MADM methods, a combined mechanism may be used. This combined algorithm performs the decision-making using SAW, MEW and TOPSIS and selects the network using the MADM mechanism that minimizes the distance to a potential ideal solution. A limited implementation of the VERHO framework is proposed in [78] for the Maemo platform (formerly supporting some Nokia Internet Tables) based on Mobile IPv6 [79]. In this implementation, the authors only consider SAW for network selection, and no evaluation for the other MADM methods is given.

AHP-GRA NETWORK SELECTION As introduced in Section 4.2.1.4, AHP provides a mechanism to define attribute preferences (i.e., the weights) without asking the decision-maker to directly define the W vector but to perform pairwise comparisons and obtain the W vector after doing some algebra. The network selection mechanism proposed in [80] takes advantage from the AHP weight definition and uses it as input in a GRA problem to rank the alternatives and find the best network. In such an approach, an AHP hierarchy is defined considering user-based parameters, including QoS parameters (e.g., availability, throughput, timeliness, reliability, security and cost) at the top of the hierarchy. Then, the availability, timeliness and reliability elements are decomposed in sub-attributes. Regarding availability attributes, signal strength and coverage area are considered. Then, delay, response time and jitter are considered for the timeliness attribute and BER, packet loss, burst error and retransmissions for the reliability attribute. Two different alternatives are considered at the bottom of the hierarchy, UMTS and WLAN. The GRA problem is modelled using network parameters as attributes for each interface. Simulation results are proposed for a single scenario with only

four alternatives, one UMTS base station and three available WLAN, and no comparative study with other solutions is proposed. The simulation results consider a single set of AHP weights, but the authors also show the variability of the selected networks for variations in the weights.

A similar approach is presented by Zhang et al. in [81], using AHP for weighting and a modified GRA for ranking. The problem modelling corresponds exactly to the model presented in [80], using the same set of attributes and alternatives. However, the authors propose a slight modification in the raking mechanism using GRA. This modification implies not only comparing the alternatives against the (positive) ideal solution but also to the worst solution, i.e., the negative ideal solution. Finally, the proposed modified GRA is very similar than the TOPSIS algorithm, which reduces the contribution of such a modified GRA.

Balasubramaniam et al. [82], proposed a different AHP-based approach. In this work, a network selection framework for vertical handover is presented, considering context information of user devices, location, network performance and application's requested QoS. The authors consider a system in which there is a preliminary knowledge of the available networks at each geographical location and their provided QoS. In this environment, an MS can request for the most suitable network to handover (called Impending Network Profile, INP) based on its location. The authors define two types of network selection, the locality-based and the QoS-based network selection. Every time the selection process has to be triggered, the MS performs a locality-based network selection, in which the MS asks for the set of networks that it could connect to. Then, it selects a subset of them in which there are users or devices in the proximity of the MS. A QoS-based selection is performed among this last networks sub-set. This QoS-based selection is modelled as an AHP problem, in which the final goal is to maximize the user's preferences and the application bandwidth and minimize jitter, delay, loss and bandwidth fluctuations. The framework is evaluated with simulations by focusing on the evolution of the MS received QoS over the time. However, since this framework relies on a centralized entity that is capable to provide very detailed information about all the available networks at every single location, it appears to be unrealistic for the current wireless environment, which is mainly characterized by a diversity of technologies and network operators.

4.2.1.7 *MADM Limitations: Attributes and Weighting Subjectivity*

As it has been previously described in Section 4.2.1.6, different MADM-based mechanisms have been proposed for network selection. However, two main problems can be highlighted. First, since MADM algorithms have been designed to support a very large number of attributes, the proposed network selection mechanisms consider a large number of attributes, including very detailed network performance parameters, user preferences or constraints

and application QoS requirements. In all cases, the authors do not consider how to gather those parameters in a real environment, i.e., how to actually feed the MADM algorithm with those attributes and if it is useful to gather such an amount of attributes for the decision-making. Additionally, the monetary cost is frequently considered as an attribute, but, since wireless accesses have been evolving to flat-rate pricing schemes, there may be no longer the interest to consider the cost as an attribute. The second problem is related to the logic to define priorities among attributes. In SAW, MEW, TOPSIS and GRA, it is the decision-maker who manually set the weights. AHP introduces a more complex logic to define weights, defining the relative importance of an attribute over the others. Even if the latter avoids manually setting the weights, which leads to subjectivity by nature, AHP may still introduce subjectivity and, even worst, inconsistency.

Several studies propose comparative analysis for different MADM algorithms (e.g., SAW, MEW, TOPSIS, GRA) mainly pointing out the limitations related to weights definitions due to subjectivity. Stevens-Navarro et al. [83] modelled the network selection with MADM using the following attributes: the available bandwidth, the end-to-end delay, the jitter and the BER. As in the previous models, the alternative set contains the different available interfaces. The authors propose a comparative study that focuses in two different aspects. First, the observed performance (in terms of average bandwidth and average delay) is analyzed for different MADM methods. Second, a sensibility study is proposed to show how the decision-making is affected by weight variations for different MADM methods. In those studies, four different traffic classes are considered (i.e., conversational, streaming, interactive, background), assigning different weights to the attributes depending on the traffic class. Regarding weights, they are defined as described in Section 4.2.1.4, using AHP pairwise comparisons. Simulation results show that, depending on the traffic class, MADM methods have different performance in terms of average bandwidth and delay. To analyze the effect of weight variation, a simulation is carried out by modifying the weights (from 0 to 1) of the jitter attribute (for conversational and streaming traffic classes) and the BER attribute (for background and interactive traffic). Two main results have been observed. First, for the same weight, there are up to 30 % of the cases in which the different MADM mechanisms select different interfaces. Second, the authors have observed that the decision-making is highly variable for slight weight variations, which highlights the importance of weights definition in MADM.

More deeply in [84] weight subjectivity is analyzed for different MADM algorithms. Differently from previous studies and frameworks, in [84], two types of network selection are being considered: a single-homed (SH) network selection, in which all application flows are assigned to the best selected network (i.e., vertical handover), and multi-homed (MH) network selection, in which different applications may be assigned to different inter-

faces (i.e., load spreading), enabling the simultaneous usage of multiple wireless interfaces. By the means of simulations, the authors conclude that, even if MADM algorithms provide dissimilar selection results for the same problem, these results are generally reasonable. However, regarding the weighting method, it is inconvenient to manually evaluate weights based on AHP pairwise comparison matrices. To partially solve this issue, the authors propose to classify weights in two different types: objective weights (such as network attributes or provided QoS) and subjective weights (such as MS properties, user preferences or application QoS requirements). As the authors state, it is more important to evaluate the relationship between changes in subjective attributes and changes in subjective weights than to define concrete values for subjective weights each time using AHP. Then, a trigger-based scheme is proposed, in which a *mapping pot* is used to store the effects of changes of the subjective attribute on the subjective weights. In a second phase, subjective and objective weights are combined to perform the best alternative selection.

In any case, trying to avoid subjectivity in the weight definition adds a relatively high level of complexity to the decision-making which prevents the decision maker to easily set up and solve network selection problems.

4.2.2 *Artificial Neural Networks*

Artificial Neural Networks (ANN) [85] have been proposed to model systems that behave as a biological nervous system, i.e., they have a number of interconnected elements (neurons) that work in parallel to solve a specific problem. An ANN has the capacity to learn using examples in what it is called a *training phase*. These systems can solve different types of problems, including pattern recognition, event predictions or complex optimization problems. When modelling an optimization problem with ANN (such as a network selection problem), the ANN is defined and iteratively operates to converge to an optimal solution.

Some applications of ANN to network selection can be found in [86] and [87]. Espi et al. [86] model the network selection using a Hopfield ANN [88], which is commonly used to solve optimization problems [89]. A cost function is defined and the ANN will iteratively converge to a solution that minimizes the cost of assigning a particular application flow to one of the available interfaces. The cost function is defined in a way that guarantees that the same application flow is not spread among different networks, that only one network is selected for each particular interface, that the MS does not demand more than the maximum available capacity of each interface and, finally, that the total available bandwidth utilization is maximized. The authors propose a simulation study, comparing the Hopfield ANN approach against a round-robin network selection (i.e., resources are cyclically assigned to applications) and an optimum bit-rate approach (i.e., it assigns the application to the inter-

face whose available bandwidth is the closest to the application demands). These approaches are compared in terms of blocking probability, buffering time and latency for a 1 MB file download. The Hopfield ANN approach offers a zero bloking probability and lower buffering time and latency than the other selection mechanisms.

A combined Fuzzy-Logic ANN approach, called Adaptive Multi-Criteria Vertical Handover (AMVHO) is proposed by Guo et al. [87] to support vertical handover decisions in a UMTS/WLAN environment. The decision-making for vertical handovers are modelled in a Fuzzy Inference System (FIS) [77]. The required bandwidth, the MS velocity and a prediction of the number of users attached to each available access network are used as the input for the FIS. The authors state that the number of users connected to the network is an important input because it has a strong impact on UMTS (CDMA-based) and WLAN (ALOHA-based) networks, since their capacity depends on the number of users. However, since the number of attached users is not commonly provided by the networks, its value has to be predicted. In this case, the authors use a Modified Elman Neural Network, an extension of the basic Elman ANN [90] consisting in four layers (i.e., input, hidden, context and output layers). Then, the three inputs of the FIS are fuzzified using three level membership functions (low, medium and high) and the vertical handover decision is taken by looking to the fuzzy inference rules table, that combines the required bandwidth, MS velocity and predicted number of users fuzzy data to decide if a UMTS to WLAN or WLAN to UMTS handover is necessary. The authors propose a simulation evaluation of AMVHO (without deeply detailing the simulation environment and parameters) against a common vertical handover algorithm, which only considers signal strength as a trigger. They show that AMVHO can reduce the bit-error rate, the delay and the number of retransmissions, while also improving the received signal strength at the MS.

4.2.3 *Combinatorial Optimization*

Combinatorial Optimization [91] mechanisms allow looking for an optimal alternative from a large set of alternatives that are typically represented in a graph. The most common example for a combinatorial optimisation problem is the Travelling Salesman problem [92], aiming to find, for a known number of cities and distances between each pair of cities, the shortest route traversing all cities only once.

Ben Rayana [93] models the decision-making for network selection using a combinatorial optimization approach. In this case, the author considers network selection in a load spreading manner, aiming to assign flows to the different interfaces. The problem is modelled by considering m flows having up to k different QoS levels (i.e., each flow could be transported in different modes, consuming a different amount of bandwidth t_i and providing a profit p_i) and n different networks. For each assignation of a flow mode j

and a network i, a *score* S_{ij} is calculated mainly considering QoS constraints, power consumption and security. The problem is then modelled with a directed graph, in which each node corresponds to an assignation of a flow mode to a network (a_{ij}). Each edge has an assigned weight corresponding to the cost (in terms of energy for example) of activating a particular interface. In order to find the optimal assignation for all flows, authors propose to use the Ant Colony Optimization (ACO) [94] heuristic to assign each flow (in a particular mode) to an available network while maximizing an utility function, calculated by the difference of the total score (i.e., the sum of the score for each node) and the total cost (i.e., the sum of the cost of each edge).

As in previously presented MADM-based frameworks for network selection, in this case the calculation of scores still requires manual weights definition for the different criteria. Moreover, the cost of activating a particular interface while assigning flows are manually fixed. In this case, activating the UMTS interface is three times more expensive than the WLAN interface. However, as it will be further discussed in the following sections, such a situation is unrealistic, since the energy cost of assigning a set of flows to a given interface is not constant.

4.2.4 *Multi-Objective Optimization*

Multi-Objective Optimization (MOO) [95] allows modelling problems where a defined number of conflicting objectives have to be simultaneously minimized/maximized subject to a number of constraints. The main difference between MOO and MADM is that in MADM, the set of alternatives (or solutions) is discrete and finite, while in MOO the solutions set can be continuous and infinite. Moreover, in MOO, the objectives are explicitly defined as mathematical functions, while in MADM, attributes are aggregated in a pre-established manner (e.g., SAW, MEW, TOPSIS). More formally, an MOO problem has the form given in Equation 38.

$$\begin{aligned} \text{Minimize/Maximize} \quad & f_m(X), \quad m \in [1, M] \\ \text{subject to} \quad & g_j(X) \geqslant 0, \quad j \in [1, J] \\ & h_k(X) = 0, \quad k \in [1, K] \\ & x_i^L \geqslant x_i \geqslant x_i^u, \quad i \in [1, \nu] \end{aligned} \tag{38}$$

The decision variables $X = \{x_1, x_2, \cdots, x_n\}$ are bounded within lower (x_i^L) and upper (x_i^U) limits, delimiting what is called a *decision space* Λ. The problem may also be constrained by equality ($h_k(X)$) or inequality ($g_j(X)$) constraints. Constraints are useful to limit the set of feasible solutions to particular regions in the space. The *objective space* Θ is defined by M objective functions $f(X) = \{f_1(X), f_2(X), \cdots, f_M(X)\}$, that can be maximized or minimized. However, due the duality principle [96], the whole problem can be converted to a minimization problem by multiplying by -1 the objective

functions to be maximized. Then, for each point X (i.e., a solution) in Λ, there is a corresponding point in Z, denoted by $f(X) = Z = \{z_1, z_2, \cdots, z_M\}$. Figure 46 illustrates an MOO problem example with $n = M = 2$.

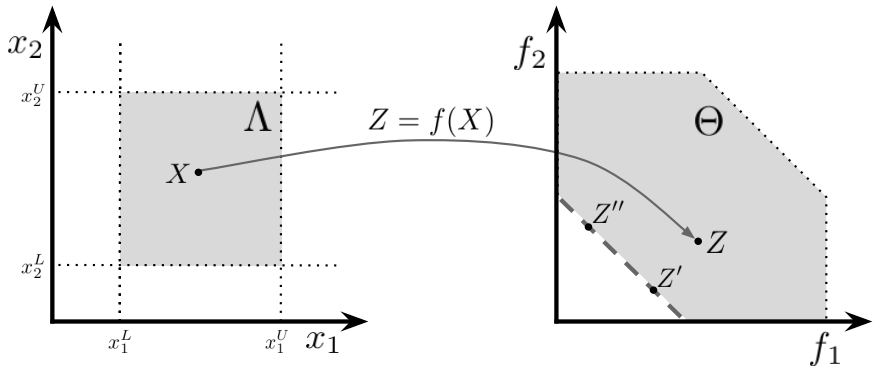


Figure 46: Decision and objective space in MOO

4.2.4.1 *Pareto Optimality*

In the minimization example of Figure 46, depending on the constraints, it is possible that not every X in Λ corresponds to a feasible decision vector. Then, a point in the feasible region of Λ can be mapped to a feasible solution in the feasible region of Θ. Two individual feasible solutions in Θ can be compared to decide which of them is more optimal than the other. In the proposed example, consider the solutions Z, Z' and Z''. The solutions Z' and Z'' lie on the grey dashed curve, representing the closest boundary to the origin of the feasible objective region. The feasible solutions in this curve minimize $f(X)$ and are called the *Pareto-optimal front* (P). Observe in our example that the solution Z is worse than Z' or Z'' in at least one of the objectives and that Z' is only better than Z'' in one of the objectives ($f_1(X)$). These comparison between solutions can be defined in terms of the domination of one solution over the other. A solution Z' is said to *dominate* Z ($Z' \triangleleft Z$) if Z' is *not worse* than Z in *all of the objectives* (in our case, $f_1(Z') \leqslant f_1(Z)$ and $f_2(Z') \leqslant f_2(Z)$) and if Z' is *strictly better* than Z in *at least one objective* (in our example, $f_1(Z') < f_1(Z)$ or $f_2(Z') < f_2(Z)$). Note that, $Z' \triangleleft Z \equiv Z \triangleright Z'$. For P to be a valid Pareto-optimal front, the following conditions must be satisfied:

- There is not a domination relationship between any pair of solutions belonging to P (in our example, $Z' \not\triangleleft\!\!\triangleright Z''$).
- Any solution that does not belong to P is dominated by at least one solution of P.

4.2.4.2 *Solving Algorithms*

In a multi-objective optimization problem, all solutions belonging to P are equally important. Then, any algorithm searching for optimal solutions may

find as many solutions in P as possible. While looking for these solutions, one may consider the following goals:

- **Closeness to** P: to find solutions as close as possible to the real Pareto-optimal front.
- **Diversity of solutions**: to find solutions barely spaced in the Pareto-optimal front, giving the most complete set of solutions.

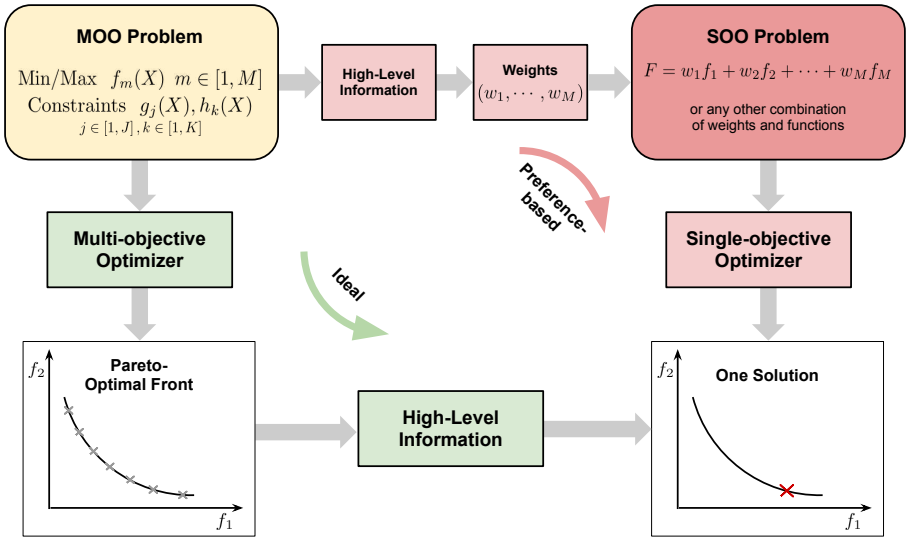


Figure 47: Ideal versus preference-based algorithms

Several searching algorithms exist, giving optimal solution sets with a different closeness to P and level of diversity. For these algorithms, a classification may be considered as illustrated in Figure 47, differencing between ideal and preference-based algorithms. In an ideal algorithm, multiple trade-off solutions are found to converge to the Pareto-optimal front. Then, from this front, one individual solution may be chosen using high level information provided by the decision maker. In a preference-based approach, objectives are first combined using high level information in the form of weights (as in MADM problems), reducing the complexity of the problem to a single objective optimization problem (i.e., a weighted sum in the example of Fig 47). Observe that even if in both cases high level information is used, in an ideal approach, this information is not used to generate and evaluate new solutions but to simple pick a single solution from an existing optimal trade-off. The most popular preference-based algorithm is the weighted sum, which is defined as an MADM SAW. We observe that for a preference-based algorithm, there is the same limitation observed in MADM approaches, i.e., the subjectivity of defining weights (see Section 4.2.1.7). The main concern in a preference-based approach is that the decision maker has not information about the existing trade-off among objectives, i.e., which is the relation between objectives in a given problem. In the case the user has a knowledge of

this trade-off, then it is possible to use a preference-based approach. However, as it will be shown in the simulation results (see Section 4.4), one cannot assume a previous knowledge of the trade-off in the network selection problem. An explanation and classification of some of the existing multi-objective algorithms following the ideal optimization process, including Evolutionary Algorithms, is presented in Section 4.3.

4.2.4.3 *Existing MOO-based Frameworks for Network Selection*

Suciu [97] proposed a multi-objective based framework for network/interface selection. In particular, two maximization objectives are defined, S_i the *score* of the interface i and U_i the utility of assigning a set of flows to the interface i.

Regarding S_i, the objective function of Equation 39 is considered. In this case, for interface i, B_i is the mean bit-rate, E_i the monitored packet error, D_i the average delay and C_i the cost of utilization. The parameter γ ($\gamma \in [0, 1]$) indicates the relative preference (i.e., the weight) of QoS. Then, $1-\gamma$ is the preference given to the cost attribute. Reference values indicated in the equation are fixed by the author. The natural logarithm function is used as a sort of normalization, even if a well-defined normalization could be used (see Section 4.2.1).

$$S_i = \begin{cases} \gamma \left(\ln \frac{B_i}{B_{ref}} + \ln \frac{E_{ref}}{E_i} + \ln \frac{D_{ref}}{D_i} \right) + (1-\gamma) \ln \frac{C_{ref}}{C_i} & \text{if} \quad B_i > 0 \\ 0 & \text{if} \quad B_i \leqslant 0 \end{cases} \tag{39}$$

For the second objective, the utility of a flow j being assigned to interface i, U_{ij} is first calculated as shown in Equation 40. Here, the difference between the interface offer (B_i, E_i, D_i and T_i) and the application demand (b_j, e_j, d_j and t_j) for each QoS parameter is considered. In this case, T_i is a measure of the security offered by interface i and t_j the required level of security. The parameter α indicates (if $\alpha = 1$) that the interface satisfies all the QoS parameters (i.e., the offer of QoS and security is greater than the demand). Finally, to calculate the objective U_i, a weighted sum of assigning J flows to the interface i is calculated as $U_i = \sum_{j=1}^{J} w_j U_{ij}$, where w_j is the importance of flow j in the set (i.e., the weight).

$$U_{ij} = \begin{cases} \ln \frac{B_i - b_j}{B_{ref}} + \ln \frac{e_j - E_i}{E_{ref}} + \ln \frac{d_j - D_i}{D_{ref}} + \ln \frac{T_i - t_j}{T_{ref}} & \text{if} \quad \alpha = 1 \\ 0 & \text{if} \quad \alpha = 0 \end{cases} \tag{40}$$

Two preference-based approaches are proposed in this work. A weighting sum using equal weights for S_i and U_i is first proposed. Then, a second approach is based on a metric that minimizes the distance to an ideal solution (as in TOPSIS). This network selection approach suffers from the same limitations observed in the previously presented approaches, mostly related to the

important amount of weights that have to be defined. In this particular case, not only the preference values of the weighted sum approach are needed (to ponder S_i and U_i) but the parameter γ and the importance of each application flow on the set as well. Additionally, the author does not present a sensibility study for the weight definition and the difference between the two solving algorithms for the multi-objective model.

4.2.5 *Existing Energy-Aware Network Selection Mechanisms*

The network selection mechanism that will be proposed in Section 4.3 considers the energy consumption as a criteria for the decision-making. In this section, we summarize the related work on energy-aware vertical handover and network selection mechanisms.

4.2.5.1 *Energy-Efficient Vertical Handovers*

A number of energy-aware network selection mechanisms exist in the literature. Petander [23] proposes a decision-making rule to support vertical handover (i.e., switching from UMTS to a WLAN when it becomes available) based on minimizing the energy cost. This decision-rule is formalized in Equation 41, and establishes that a vertical handover from UMTS to WLAN is energetically advantageous by comparing the amount of energy spent to exchange N bytes of data through the UMTS (NE_U) with two different cases while using WLAN. In the first case, if the vertical handover succeeds (this occurs with probability P_{VHO}), the energy consumed when using WLAN is equal to the energy needed for a vertical handover (E_{VHO}) and the energy consumed to transmit the data over WLAN (NE_W). The second case considers an unsuccessful vertical handover, in which case a WLAN scanning has been performed and finally data is transmitted using UMTS.

$$NE_U = P_{VHO}\,(E_{VHO} + NE_W) + (1 - P_{VHO})\,(E_{scan} + NE_U) \tag{41}$$

Finally, the MS attempts a vertical handover to WLAN if the amount of data to exchange is greater than the threshold N_T (see Equation 42), obtained by solving the Equation 41 for N.

$$N_T = \frac{P_{VHO} E_{VHO} + (1 - P_{VHO})\,E_{scan}}{P_{VHO}\,(E_U - E_W)} \tag{42}$$

The authors provide empirical values for the different parameters in the previous equation, obtained in a measurement study using a Nokia N95. They observed that the energy needed for a vertical handover (E_{VHO}) was in average between 0.127% and 0.212% of the battery capacity and the energy of a scan (E_{scan}), 0.122%. Then, depending on the conditions (e.g., load and signal strength) of the UMTS and the WLAN connections, N_T may oscillate between 0.1 and 0.9 MB of data.

A similar energy-aware network-selection mechanism, called WISE, is proposed by Nam et al. [98]. In this case a tightly coupled 3G-WLAN system is considered, in which the same network operator provides 3G and WLAN networks in a centralized manner. The authors propose a new entity in the 3G core network, the Virtual Domain Controller (VDC), that manages the vertical handover decisions in such a tightly-coupled system, i.e., it manages vertical handover requests from MS and executes them by transferring the context of the MS to guarantee session continuity of the on-going applications. In WISE, the MS is able to dynamically request to switch the network interface in order to consume less energy while being aware of possible throughput degradation caused by the selection of a new interface. The MS permanently monitors its uplink and downlink traffic and decides which interface consumes less energy under the current traffic situation. Particularly for the decision-making, the authors propose a fixed rule based on the fact that 3G interfaces consume more energy than WLAN while transmitting and less energy while receiving or being idle. This assumption is based on two measurements using a CDMA modem and an external WLAN card. When an MS decides to perform a handover, it transmits a vertical handover request to the VDC, which evaluates the impact of this decision on the overall network performance. The VDC can then reject the request if the overall network performance is degraded.

4.2.5.2 *Delay-Tolerance and Energy Efficiency*

The network selection mechanisms proposed by Balasubramanian et al. [99] and Ra et al. [100] are based on delaying the use of some network interfaces depending on the reduction of energy consumption that an MS may observe if it delays the transmission of some application flows for a certain period of time, that varies according to a degree of tolerance for each application. An example of a delay-tolerant mechanism for network selection [100] is illustrated in Figure 48, where an MS has access to three different wireless networks (EDGE, 3G and WLAN), having variable bandwidth and availability (as shown in the highest part of Figure 48) and consuming different levels of power. In this example, the user triggers two application flows (Flow 1 and Flow 2) and the network selection mechanism reacts differently depending on the delay tolerance (e.g., Minimum Delay, WLAN Only, Optimal Energy). If the user desires a minimum delay, the flows are immediately sent over the best available wireless interface (in Figure 48, it first uses EDGE then 3G and finally WLAN for Flow 1). Then, both flows are successfully transmitted after 246 s consuming 246 J. However, if the user can support longer delays to transmit these flows by waiting for a WLAN to become available for example, it will spend a longer time to successfully transmit the flows but also much less energy (up to 50 J) will be consumed, since the usage of high energy-consuming interfaces is minimized.

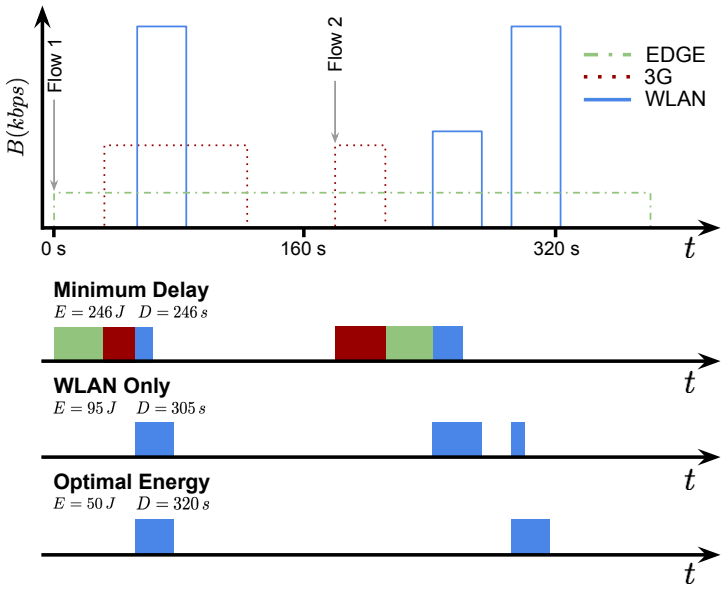


Figure 48: Impact of delay tolerance on energy consumption

Ra et al. [100] particularly focus on the analysis of the energy-delay trade-off to design a network selection algorithm that decides whether to use one of the available interfaces or to defer the transmission of a set of flows since a more energy-efficient network will become available later. Such an algorithm is modelled using a single objective optimization problem with one constraint. This is to minimize the total energy consumption subject to keeping the flow queue length finite (i.e., to avoid deferring the flow indefinitely).

In the same perspective, the *TailEnder* mechanism [99] takes advantage from the operating logic of 2G/3G cellular networks, in which the MS remains in a high consuming energy state after completing a flow transmission (as explained in Section 2.1.4.1). Thanks to a measurement study, the authors have determined that nearly 60 % of the total energy corresponds to tail energy, i.e., caused by the MS remaining in the high energy consuming state during an inactivity timer (as explained in Section 2.1.4). Then, if application flows are delayed, the MS could amortize this tail energy by scheduling several flows one after the other, separated by a time interval lower than the inactivity timer. For example, when writing e-mails, the users normally send each one of them just after completing the edition. However, the user can tolerate a short delay and cumulate several e-mails to perform a single access and send them altogether. *TailEnder* is also modelled as a single objective optimization problem, that aims at minimizing the total time spend in high power states by satisfying that each flow is transmitted before a predefine deadline (i.e., the delay tolerance). Regarding the WLAN interface, the authors state that it is still more energy efficient than the 3G interface and, additionally, the energy consumed is independent from the time between two transmissions, so there is no necessity to use *TailEnder* over WLAN. Then, WLAN can be used each time it becomes available.

4.2.6 *Discussion*

In the precious sections, a number of decision-making models and techniques together with existing frameworks for network selection have been presented. Most part of the existing frameworks are based on MADM or, most commonly, on a combination of methods. As stated in Section 4.2.1.7, the main limitation of these approaches is the high level of subjectivity that influences the decision-making. This subjectivity usually takes the form of weights, i.e., a measure of the relative preference between different attributes or criteria. When dealing with a high number of attributes and alternatives, weighting techniques are very convenient, since they reduce the complexity of the optimization problem to a single function to be optimized. However, a reasonable weighting is very hard to obtain. Several authors propose to use AHP to facilitate the weight definition to decision makers, but as shown in 4.2.1.7, AHP may introduce inconsistencies due to pair-wise comparisons.

Another approach is to use MOO, which is a more general method than MADM since instead of merging attributes and alternatives in a decision matrix, in MOO, the goal is to optimize a set of discrete or continuous mathematical functions that may model more evolved metrics than a single attribute (like a QoS parameter). In this case, depending on how the decision maker models the optimization problem (i.e., which decision variables and which objectives), solving the problem may give a trade-off of optimal solutions instead of a single optimal solution (like in MADM, a single-objective optimization or a preference-based MOO). However, when using a preference-based algorithm to solve the problem (like the weighted sum or SAW approach), a single optimal solution is obtained for each weight combination. Then, in order to approximate the Pareto-optimal front (i.e., the trade-off of solutions) several runs using different weight combinations must be performed. However, using a weighted sum algorithm and different weight combinations over a non-convex problem may not give a good approximation of the Pareto-optimal front [95].

These issues may be addressed using an optimization algorithm that allows obtaining the complete Pareto-optimal front without using preference values or weights. In the following sections, a multi-objective optimization model for network selection is proposed and solved using evolutionary algorithms. These algorithms can provide a very good approximation of the Pareto-optimal front without using a priori high layer information (e.g., weights, priorities or preferences). Moreover, we have observed that the previous proposed frameworks for network selection commonly try to quantify user demands in terms of several QoS parameters, monetary cost or security metrics, which imposes very complex monitor and estimation processes to gather all this information. In the proposed approach, the number of attributes is reduced in order to provide a more realistic network selection framework, mostly focusing on energy consumption, since it has become

one of the most critical issues on today's mobile devices. As differently as the energy-aware network selection mechanisms presented before, we consider the energy consumption in fine-grained manner. Considering the different operating modes of IEEE 802.11 and 3G cellular networks and their power consumptions, as introduced in Section 2.1.4, in our proposed network selection mechanism in Section 4.3, we use traffic models to estimate the energy consumption of each interface by determining the amount of time the interfaces remain on each operating state. As in the solutions introduced in Section 4.2.5, we highlight the importance of inactivity timers in 3G networks and PSM in WLAN networks, and we consider these issues to model our network selection mechanism.

4.3 ENERGY-EFFICIENT NETWORK SELECTION USING GENETIC ALGORITHMS

4.3.1 *Introduction*

As illustrated in the previous sections, the most part of current network selection mechanisms suffer from the following shortcomings:

- They are commonly based on MADM decision-making, which implies the definition of preference values (in the form of weights) for the different criteria. This imposes a high user involvement, adding an important level of subjectivity to the decision-making process.
- They are particularly designed to support vertical handovers, i.e., the transfer of all on-going flows to a new selected interface. In those cases, the advantages of wireless multi-homing is not fully exploited, since a user is not able to simultaneously use different wireless interfaces.
- They consider energy consumption of wireless interfaces as a constant criteria, in which the interfaces are ranked using a constant energy cost. As it has be presented in Section 4.2.5, depending on how the application flows are arranged, the interfaces can be more or less energy-efficient.
- There are several solutions considering different QoS parameters and energy as criteria, giving an important number of criteria to be simultaneously optimized, which increases the complexity of the problem.

To overcome these drawbacks, we propose a network selection mechanism that does not consider subjective criteria but only objective criteria in a multi-objective optimization approach: the total energy consumption and the bandwidth satisfaction of flows assigned to the different interfaces. Such a mechanism does not require the definition of weights or preference values to solve the optimization problem. We consider network selection as a load

spreading problem in a per-flow granularity, i.e., each flow is individually assigned to a single interface.

Moreover, in order to consider energy consumption, we use energy models (as those detailed in Section 2.1.4) to estimate the average power consumed by each group of flows over a particular interface. In such a model, we consider the power consumption of the different operating states of the two most common wireless technologies, UMTS/HSPA and WLAN. This differs from previous network selection mechanisms, where the effects of inactivity timers (in 3G) and PSM (in WLAN) are not taken into consideration while assigning flows. As it has been introduced in Section 4.2.5.2, the flow-interface assignation strategy greatly affects the energy consumption. This is mainly due to the tail energy observed in UMTS/HSPA cellular networks, in which the MS is not able to come back to low energy consuming states just after the end of a transmission, since it has to wait for inactivity timers to expire. In the case of IEEE 802.11 interfaces, even if the MS comes back to the IDLE state just after a transmission terminates, it has the possibility to enter in the SLEEP state (PSM) after a timeout in order to consume much less energy.

Additionally, instead of modelling the decision-making problem using MADM, we propose to model the flow-interface assignation problem using Multi-Objective Optimization. It is solved following an ideal approach (see Section 4.2.4.2), which differs from a preference-based solving algorithm (like the weighted sum algorithm), since it provides a set of optimal solutions instead of a single solution. Using an ideal approach, the decision-maker can have a complete view of the energy-bandwidth trade-off to select the most suitable flow-interface assignation for each particular scenario. As it will be presented in Section 4.4, we search for solutions using a genetic algorithm, which gives a good approximation to the energy-bandwidth trade-off and can reduce the solving time for problems with a large number of applications and interfaces.

4.3.2 *Problem Statement*

We model the network selection as a flow-interface assignation problem in an MOO approach. In such a problem, there are n application flows $A = \{a_1, a_2, \cdots, a_n\}$ and m available interfaces $I = \{i_1, i_2, \cdots, i_m\}$. We aim at finding the optimal decision vectors in the discrete decision-space Λ, $X = \{x_1, x_2, \cdots, x_n\}$, in which each element x_k with $k \in [1, n]$ corresponds to the interface to use for the application flow a_k, i.e., $x_k \in I$. In the objective space Θ, the objective function is defined in Equation 43, where $E(X)$ is the interface energy consumption and $B(X)$ is the overall bandwidth dissatisfaction related to the decision vector X.

$$\text{Minimize} \quad F(X) = \{E(X), B(X)\} \tag{43}$$

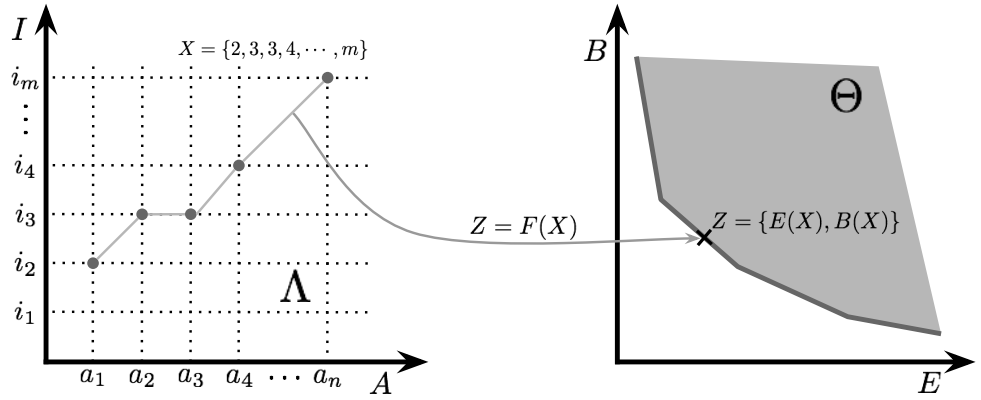


Figure 49: Decision and objective space for the MOO network selection

An illustration of the optimization problem, including the decision space Λ and the objective space Θ is presented in Figure 49. In this example, a decision vector $X = \{2, 3, 3, 4, \cdots, m\}$ is evaluated in terms of $E(X)$ and $B(X)$ giving a solution Z that belongs to the Pareto-optimal front in Θ.

4.3.3 *Solution Searching using Genetic Algorithms*

4.3.3.1 *Introduction*

In order to search for optimal solutions in a single-objective or a multi-objective optimization problem, a search algorithm is used to explore the decision space. Some of the classical search algorithms [96], like the Simplex method, the Newton method or the Gradient descent method use a set of heuristics that iteratively approach optimal solutions. On each iteration, the classical algorithms suggest a search direction and calculate a new decision vector that is closer to the optimal solution. The process is repeated for a pre-established number of times or until a convergence criterion is satisfied. However, as stated in [95], in general these algorithms present a number of limitations:

1. The convergence to an optimal solution strongly depends on the initial (random) solution provided to the search algorithm.
2. These algorithms could get stuck to suboptimal solutions, which prevents finding the real optimal solution, or, in a multi-objective problem, a good approximation of the Pareto-optimal front.
3. If the problem definition changes (e.g., objective and constraints definition, decision-space modelling) a classic search algorithm may no longer be suitable to search for solutions in the new problem.
4. Classical algorithms are not efficient in solving discrete decision space problems

The latter two drawbacks especially limit the use of classical algorithms in the proposed network selection problem. Each time the MS triggers a decision-making problem to solve the flow-interface assignation, it will set-up a new problem, that may have different number of interfaces (m) and applications (n). Moreover, as it has been defined in Section 4.3.2, the bi-dimensional decision-space is made of a combination of two discrete variables (I and A).

For that reason, we have decided to use genetic algorithms to search for optimal solutions. Genetic algorithms have been first proposed by Jhon Holland [101] and have the ability to explore a decision space inspired on the principles of genetics and natural selection. To achieve this, the decision-space is modelled in a binary representation, that allows applying genetic operations between different decision vectors.

4.3.3.2 *Binary-Coding*

In a binary-coded problem, the decision vectors X are represented using bit-strings (also called chromosomes or individuals). In our problem, we first assign a bit-string b_l ($l \in [1, m]$) to each available interface by converting the integer interface index to binary. The length of b_l is exactly $\lceil \log_2 m \rceil$, which guarantees that we use the necessary number of bits for the total number of interfaces in the problem. Then, the binary representation of X consists in joining, for each application flow, the binary value b_l for the interface to which it has been assigned. For example, for a $n = 4$ and $m = 6$ problem (six applications to be assigned to four different interfaces), the interfaces are coded as follows: $b_1 = \texttt{00}$, $b_2 = \texttt{01}$, $b_3 = \texttt{10}$ and $b_4 = \texttt{11}$. Consider as an example the decision vector $X = \{1, 1, 2, 1, 1, 4\}$, then, its binary representation is:

$$X = \texttt{00 00 01 00 00 11}$$

4.3.3.3 *Genetic Algorithm Working Principles*

Figure 50 illustrates a generic genetic algorithm, which can be divided in four main processes: generation of an initial population, fitness evaluation, selection and variation (mutation and crossover).

After defining the problem and representing the decision space in a binary-coded manner, the genetic algorithm first creates an *initial random population* of solutions (we refer to solution or decision vector indistinctly). In our case, this is simply to assign each application flow in A to a random interface in I and represent them in a bit-string. Then, for each solution, the algorithm evaluates its *fitness*. The fitness is a way of combining the objective values and the constraints violations. In our problem, since it is unconstrained, the fitness of a solution X is simple calculated as $F(X) = \{E(X), B(X)\}$. At this point, all initial random solutions have an assigned fitness. Then, the *selection* process identifies good solutions in the population, makes several copies

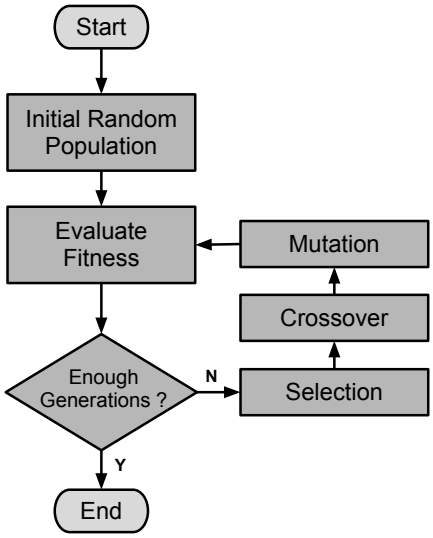


Figure 50: Genetic algorithm operations

$F(X_1) = \{0.8, 2.8\}$ $X_1 = 0010101001 \longrightarrow X_1' = 0010100001$ $F(X_1') = \{1.2, 1.9\}$

Figure 51: A one-bit mutation example

of them (referred as to reproduction in the literature) and eliminates bad solutions. In order to differentiate between good and bad solutions, the fitness metric is considered in different manners, depending on the particular genetic algorithm. After the selection process is performed, the different solutions (and copies of solutions) remaining in the population form what it is called a *mating-pool*. This mating-pool is taken as the input of the *variation* process, which generally performs two operations over the existing solutions: Mutation and Crossover. These operators are responsible for combining solution chromosomes to create new (and hopefully better) solutions and keep diversity on the population, which is one of the goals of the searching process.

Regarding mutation, there are several mutation operators. The *one-bit mutation randomly* selects a single bit in the solution and inverses it (from `0` to `1` or vice-versa), as illustrated in Figure 51. Another common mutation operation is the *independent-bit mutation*, which inverts each bit of the solution with probability p.

Unlike mutation, the crossover operator requires two solutions from the matting pool as an input. As illustrated in Figure 52, in a *one-point crossover*, a random position inside the bit-string is calculated. Then, two new solutions are created by taking the leftmost part of each original string in the new solution and switching the rightmost part of the strings. A different crossover operator is the *uniform crossover*, in which two new solutions are created by switching the bits of the original solutions with a probability p.

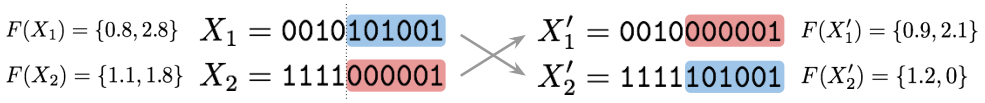


Figure 52: A one-point crossover example

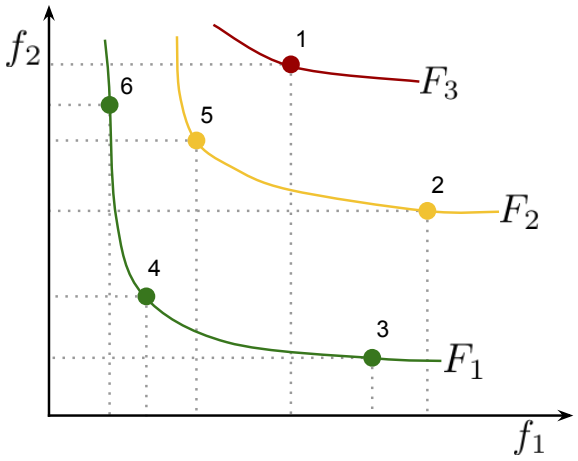


Figure 53: Non-dominated sets example

Finally, the process is repeated for a pre-established number of *generations*. The population remaining after the last iteration is the approximation to the Pareto-optimal front.

4.3.3.4 *The NSGA2 Genetic Algorithm*

The genetic algorithm description in Section 4.3.3.3 represents the operation of a basic genetic algorithm. However, there are a number of genetic algorithms introducing optimizations to reduce their computational complexity. These optimizations are mainly introduced in the selection procedure, i.e., the process that compare individuals and decide which are the individuals that remain for the next generation. In a single-objective approach, the comparisons between individuals are based on their fitness. However, as stated in Section 4.2.4, in MOO these comparisons are based on the dominance criteria since there is not a single value to compare individuals. Then, the selection procedure needs to rank individuals depending on its dominance level. This ranking procedure is called non-dominated sorting. To explain the non-dominated sorting, consider the example in Figure 53, an MOO minimization problem with a population P of size six. At a given time, the six individuals in the population are $P = \{1, 2, 3, 4, 5, 6\}$ and their fitness in terms of the objectives f_1 and f_2 are as shown in Figure 53. Then, the different individuals can be assigned to three different non-dominated sets: $F_1 = \{6, 4, 3\}$, $F_2 = \{5, 2\}$ and $F_3 = \{1\}$. Observe that a non-dominated set is composed of individuals that do not dominate each other and, moreover, each individual in F_n $(n > 1)$ is dominated by at least one individual in F_{n-1}.

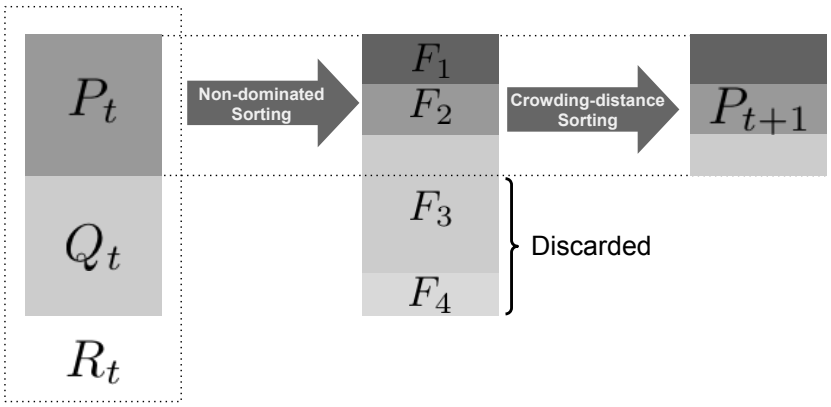


Figure 54: NGSA2 algorithm

The non-dominated sorting is used in several MOO selection algorithms, including Non-dominated Sorting Genetic Algorithm 2 (NSGA2) [102] which is used in our network selection mechanism to search for solutions in the decision space. NSGA2 is an elitist and low complexity selection algorithm that provides close convergence to the Pareto Optimal front. It has an $O(MN^2)$ computational complexity, where M is the number of objectives and N the population size, improving the $O(MN^3)$ complexity of its predecessor Non-dominated Sorting Genetic Algorithm (NSGA). Regarding elitism [95], it is a technique that allows using previously found good solutions in the next generations to avoid degrading the overall fitness of the population. Elitism can be introduced for instance by copying to the next generation a fixed number of best individuals from the previous generation. Particularly in NSGA2, elitism is achieved by combining the previous population P and the *offspring* Q (i.e., the population after crossover and mutation) in a new set R. This is illustrated in Figure 54 , where $R_t = P_t \cup Q_t$ is ranked using non-dominated sorting and finally the new population P_{t+1} is composed of the first and the second non-dominated sets F_1 and F_2 and by some individuals of F_3, obtained by crowding distance sorting [95] (which prefers solutions in a lesser crowded region so as to improve diversity), in order to have exactly N elements in the new population. Then, Q_{t+1} is obtained after crossover and mutation.

4.3.4 *Modelling Flows and Interfaces*

Up to this point, we have defined an MOO problem, including the decision space, the objective space and the genetic algorithm basics that allow exploring the decision space for optimal solutions. In this section, details are given on the variables modelling the decision space (i.e., the application flows A and the interfaces I) in order to evaluate the proposed network selection approach.

4.3.4.1 *Interface Model*

A multi-homed MS can get connected to the network through different interfaces, either alternatively or simultaneously. We consider the case of being simultaneously connected through multiple interfaces as a common scenario, since the density of wireless networks of different technologies has importantly increased (as detailed in our measurement study in Chapter 2). Note that when referring to a multi-homed MS, one may consider not only common user devices (e.g., smartphones, tablets, laptops) but also mobile routers, which may be connected to different wireless access networks and provide a local wireless access to a multiplicity of users moving together.

For the network selection mechanism, we consider that an interface i_j, $j \in [1, m]$ is characterized by three parameters. First, we assume that an interface has, at any time, an associated available bandwidth bw_j. Second, as stated in Section 2.1.4, each interface consumes different levels of power in each operating state. Thus, we define an energy vector e_j (see Equation 44) depending on the interface technology (i.e., WLAN or UMTS/HSPA, as introduced in Section 2.1.4).

$$e_j = \begin{cases} \{p_j^{DCH}, p_j^{FACH}, p_j^{PCH}\} & \text{if } i_j \text{ is a UMTS/HSPA interface} \\ \{p_j^{TxRx}, p_j^{IDLE}, p_j^{SLEEP}\} & \text{if } i_j \text{ is a WLAN interface} \end{cases} \tag{44}$$

For simplicity, we consider the WLAN interface having the same level of power in the transmission than in the reception mode. Finally, we consider that each interface has a vector of timeouts, t_j. In the case of UMTS/HSPA, we consider the inactivity timers T_1 and T_2 and in the case of WLAN, the PSM timeout T_T.

$$t_j = \begin{cases} \{T_1^j, T_2^j\} & \text{if } i_j \text{ is a UMTS/HSPA interface} \\ T_T^j & \text{if } i_j \text{ is a WLAN interface} \end{cases} \tag{45}$$

Each time the a flow-interface assignation optimization problem has to be solved, bw_j, e_j and t_j for $j \in [1, m]$ are provided as an input to the system.

4.3.4.2 *Traffic Model*

EXISTING TRAFFIC AND ENERGY MODELS A special effort has been given to model the energy consumption of wireless interfaces depending on the amount and type of application flows that are being transmitted. Basically, combining a traffic model with the state transition of the different wireless interfaces allows the MS to estimate how much time an interface spends on each different operating state (t_i, where $i \in [0, n]$ is an operating state) while transmitting/receiving an application flow. Then, assuming that each interface has an associated energy vector e_j (that gives the power consumption

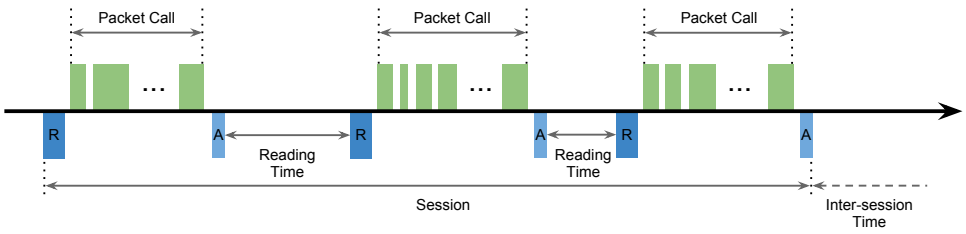


Figure 55: Traffic model for non real-time traffic

of each operating state), the energy consumption (measured in Joules) is calculated as the sum, for each state, of the product between the time spent on the state and its associated power. The power consumption may be easily estimated for each particular MS and can be considered as constant for each operating state. In Section 2.1.4, we listed the different power values found in the literature, showing very dissimilar values for different MS. Then, the traffic model has to be capable of giving an estimation of the time spent on each state, depending on the particular application flow.

Different models exist in the literature trying to estimate the energy consumed for different types of flows. Yeh et al. [103] proposed an analytical traffic and energy model for UMTS/HSPA considering two different types of application flows, non real-time and real-time. The non-real-time traffic is modelled as the web-browsing traffic model suggested by the 3GPP [104], as illustrated in Figure 55. It considers several browsing sessions, each session includes one or more packet calls (i.e., a sequence of packets after a web-request). Between two consecutive packet calls there is an idle time (i.e., reading time). These model variables follow different random distributions. The time between two sessions is modelled as a geometrically distributed random variable with a mean of 600 s; the number of packet calls per session, the number of packets per packet call and the reading time are also modelled as geometric random variables of mean 5 calls, 25 packets per call and 412 s respectively. On the other hand, real-time traffic is modelled as video streaming flows in an ON/OFF approach (see Figure 56), in which the duration of the different requests for videos (ON periods) follows an exponential random variable of mean 30 s and the time between requests (the OFF periods) are also exponential of mean 120 s. To estimate the energy consumed by each flow, they model the operation of the UMTS/HSPA interface using a Discrete Markov chain, and the average time spent on each operational state (DCH, FACH, PCH) is estimated using the steady-state probability of the chain.

A different traffic and energy model is also proposed in [21] for bursty traffic. The authors model bursty traffic, where S_B is the burst size, T_I is the time between two burst (both depends on the application flow) and T_B is the burst duration (that depends on the hardware data-rate). Then, the rate (r) of the bursty traffic is calculated as shown in Equation 46.

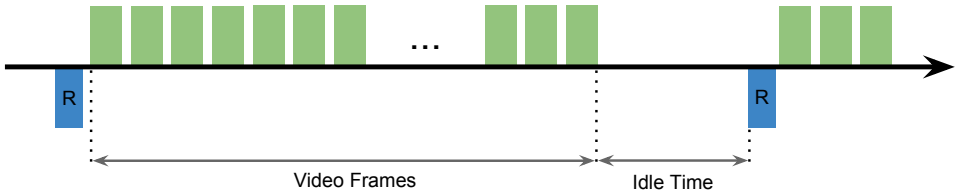


Figure 56: Traffic model for real-time traffic

Flow Type	Model Variable	Distribution
Non Real-Time	Number of requests	Geometric
	Number of packets per request	Geometric
	Packet size	Pareto
	Time between requests	Geometric
Real-Time	Number of sessions	Geometric
	On-Time	Geometric
	Off-Time	Geometric
	Media type	Uniform

Table 14: Traffic model

$$r = \frac{S_B}{T} = \frac{S_B}{T_B + T_I} \tag{46}$$

To model the energy consumption two scenarios are considered. In the first scenario, the MS does not enter into PSM (because it is not enabled or because the PSM timeout, T_T, in greater than T_I). In the second scenario, the MS has PSM enabled and $T_T < T_I$. In a download use case, the Energy (E, in Joules) is calculated as in Equation 47, where P_R, P_I and P_S are the power consumption in the RX, IDLE and SLEEP state respectively.

$$E = \begin{cases} P_R T_B + P_I T_I & \text{for Scenario 1} \\ P_R T_B + P_I T_T + P_S (T_I - T_T) & \text{for Scenario 2} \end{cases} \tag{47}$$

APPLICATION FLOWS MODELLING Like in [103], we consider two different types of application flows: real-time and non-real-time. For real-time flows, we consider up to four different types of flows, demanding different levels of bandwidth. Table 14 summarizes the random distribution chosen for each parameter modelling each flow type. The specific parameter values are then given in Section 4.4, since they will be used for the evaluation of the network selection algorithm.

We assume that a real-time or non real-time application flow a_k with $k \in [1, n]$ is a sequence of data of a given size s, separated by a given time interval t, as exemplified in Figure 57. Then, each application can be formal-

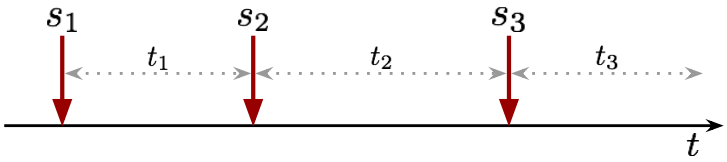


Figure 57: Application flow example

ized as follows, $a_k = \{(s_1, t_1), (s_2, t_2), \cdots, (s_r, t_r)\}$ where r is the number of sequences of data (i.e., packet calls) to transmit/receive.

4.3.5 *Computing Objectives*

Having defined application flows and interfaces for the decision-space, the calculation of the objectives for a particular flow-interface assignation (i.e., energy consumption and bandwidth dissatisfaction) is detailed in this section.

4.3.5.1 *Bandwidth Dissatisfaction*

We define the bandwidth dissatisfaction on each interface, B_j with $j \in [1, m]$, as in Equation 48.

$$B_j = \begin{cases} D_j - bw_j & \text{if} \quad D_j > bw_j \\ 0 & \text{if} \quad D_j \leqslant bw_j \end{cases} \tag{48}$$

Then, the overall bandwidth dissatisfaction is $B = \sum_{j=1}^{m} B_j$. As seen in Equation 48, the bandwidth dissatisfaction on each interface is calculated as the difference between the applications' bandwidth demand and the available bandwidth of the interface. The bandwidth demand on each interface (D_j) is equal to the sum of the bandwidth of each application assigned to that particular interface. Then, to calculate the bandwidth demand of each application, we average the bandwidth of the most recent (s, t) pairs of each application. To this end, consider the example of Figure 58, where three applications, a_1, a_2 and a_3 are assigned to the same interface i_j at time T_n (just when a new flow starts). If the last two transmitted/received (s, t) pairs are considered, then the bandwidth demand of the different applications are as in Equation 49. Note that for the new application (a_3 in the example), we consider the pair (s_1, t_1), since no past information is available. Finally, D_j is calculated as the bandwidth of all the applications assigned to interface j, which exclusively depends on the decision vector. For this example, $D_j = b_{a_1} + b_{a_2} + b_{a_3}$.

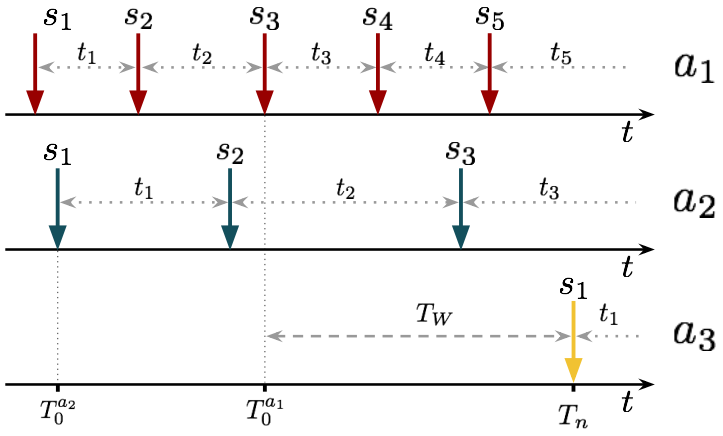


Figure 58: Bandwidth demand calculation example

$$
\begin{aligned}
b_{a_1} &= \frac{s_3^{a_1} + s_4^{a_1}}{t_3^{a_1} + t_4^{a_1}} \\
b_{a_2} &= \frac{s_1^{a_2} + s_2^{a_2}}{t_1^{a_2} + t_2^{a_2}} \\
b_{a_3} &= \frac{s_1^{a_3}}{t_1^{a_3}}
\end{aligned}
\tag{49}
$$

4.3.5.2 *Energy Consumption*

The energy objective, E, is calculated as the sum of the average power consumption (in Watts) of all the interfaces, i.e., $E = \sum_{j=1}^{m} p_j$. This is to avoid using the absolute energy values, in Joules, since this hinders the comparison if the flows last different amounts of times. To calculate p_j, we estimate the time the interface will remain in each operating state (as described in the energy and traffic models in Section 4.3.4.2). To achieve this, we first need to establish a time window (T_W^j) during which the average power consumption will be calculated. To calculate T_W^j we consider, as for the bandwidth, the two last sent/received (s, t) pairs. Then, we calculate the difference between the current time (T_n, in Figure 58) and the first considered (s, t) pair of each application (T_0 in Figure 58). Finally, we chose the minimum difference as the value for T_W^j, in our example $T_W = T_n - T_0^{a_1}$. Since we aim at calculating during how long the interface is being used/busy (T_B^j) or being idle (T_I^j), we consider the previous value for T_W^j since it is the most restrictive interval to send/receive all flows.

In order to calculate T_B^j and T_I^j we first consider the ratio r_T in Equation 50 between the total application demand and the available bandwidth of the interface

$$
r_T = \frac{\sum b_{a_i}}{bw_j} \quad \forall i \text{ assigned to } j \tag{50}
$$

Then, if $r_T \geqslant 1$, the applications' demand is equal to or exceeds the available bandwidth and so the interface will be used all the time, which means $T_B^j = T_W^j$ and $T_I^j = 0$. On the other hand, if $r_T < 1$, there will be a time interval in which the interface will be idle, then $T_B^j = r_T T_W^j$ and $T_I^j = (1 - r_T) T_W^j$.

Finally, using T_B^j, T_I^j, the power vector e_j and the timeouts t_j, we calculate the average power consumption like in the model proposed by Xiao et al. [21], introduced in Section 4.3.4.2. Algorithm 1 illustrates the calculation of E for UMTS/HSPA and WLAN interfaces. Observe that for UMTS/HSPA we consider a RRC state machine like in Figure 7a, in which the MS access to a DCH channel each time it has any data to transmit or receive.

Algorithm 1 Algorithm to calculate E

```
1: for j = 1 → m do
2:     if j ∈ UMTS/HSPA then
3:         if T_I^j ⩽ T_1^j then
4:             aux = (T_B^j + T_I^j) p_j^DCH
5:         else if T_I^j ⩽ (T_1^j + T_2^j) then
6:             aux = (T_B^j + T_1^j) p_j^DCH + (T_I^j − T_1^j) p_j^FACH
7:         else
8:             aux = (T_B^j + T_1^j) p_j^DCH + T_2^j p_j^FACH + (T_I^j − T_1^j − T_2^j) p_j^PCH
9:         end if
10:    else
11:        if T_I^j ⩽ T_T^j then
12:            aux = T_B^j p_j^TxRx + T_I^j p_j^IDLE
13:        else
14:            aux = T_B^j p_j^TxRx + T_T^j p_j^IDLE + (T_I^j − T_T^j) p_j^SLEEP
15:        end if
16:    end if
17:    p_j = aux/T_W^j
18:    E+=p_j
19: end for
```

4.4 SIMULATION RESULTS

In this section we aim at evaluating the proposed network selection mechanism. We have developed a simulator implementing the PISA Framework [65] for MOO. We first introduce the simulator environment and then we propose the simulator results.

4.4.1 *Simulation Environment*

4.4.1.1 *The PISA Framework*

The PISA framework [65] is an open source interface that allows modelling optimization problems, including MOO. It has the ability to split the optimization problem in two independent modules, the optimization problem definition itself (in our case, the network selection) and the algorithms to

search for solutions. In the context of genetic algorithms, the PISA framework allows defining an MOO problem, called the *Variator*, (i.e., decision and objective space definition, fitness evaluation, variation of solutions) independently from the *Selector* logic (i.e., the genetic algorithm). Then, different Selectors can be used to solve a particular Variator and analyze their performance when searching for optimal solutions. The interface between the Variator and the Selector in PISA is handled using plain text files. In a general case, the Variator, containing the problem, initializes a random population, calculates the fitness of each individual of the population and performs reproduction. Then, the Variator sets a value in a text file indicating the Selector the end of the reproduction process and the necessity to perform selection over the current population. This process loops for a fixed number of generations and the optimal solutions are given in an output file. PISA provides open source implementations for a number of Selectors (e.g., NSGA2 [102], Strength Pareto Evolutionary Algorithm 2 (SPEA2) [105], Fair Evolutionary Multiobjective Optimizer (FEMO) [106]) and some Variator examples from typical optimization problems (e.g., Leading Ones Trailing Zeros (LOTZ), Bi-objective Binary Value (BBV), Knapsack Problem).

For the purpose of our work, we developed a new Variator and used NSGA2 (Section 4.3.3.4) as a Selector which has shown better performance than other implemented Selectors. A comparison study of the two most popular elitism-based Selectors, NSGA2 and SPEA2 is proposed by Bui et al. [107]. In this study, the authors conclude that even if SPEA2 gives better results than NSGA2 in the early generations, NSGA2 outperforms SPEA2 in the last generations independently from the problem considered.

When solving a problem with PISA, some input parameters are required, as described in Table 15. Some of these parameters are common for the Variator and the Selector. On each run of the MOO problem, μ parent individuals are selected from the population of size α. After variation (i.e, crossover and mutation), λ individuals are generated in the offspring. Specifically for the Variator, the number of generations g indicates how many iterations have to be performed after outputting the solution trade-off. On each iteration, crossovers of type c_t and mutations of type m_t are performed using the probabilities indicated in Table 15.

4.4.1.2 *Network Selection Simulator*

We have developed a network selection simulator using the Practical Extraction and Report Language (PERL), as illustrated in Figure 59, that is responsible for the generation of random scenarios and the computation of network selection using different algorithms. In our simulator, we consider that a network selection process is triggered each time a new application has to be assigned (i.e., a simulation event). However, other triggers could be used, like the interface bandwidth variations or when application flows terminates.

Common Parameters	
α	Size of the initial population
μ	Number of parents
λ	Number of individuals after variation (offspring)
d	Problem dimension (number of objectives)
Variator Parameters	
g	Number of generations
c_t	Crossover Type (uniform or one-point)
p_c	Probability that a given pair of individuals undergoes crossover
m_t	Type of mutation (one-bit, independent-bit)
p_m	Probability that a given pair of individuals undergoes mutation
p_b	Probability that a bit is turned

Table 15: PISA parameters

RANDOM SCENARIOS In order to evaluate network selection by simulation, a scenario has to be defined in a first phase. In our context, as shown in Figure 59, a scenario includes an application flow arrival process (i.e., the process defining at what time each application needs to receive/send data over the network) and an interface availability process (i.e., indicating the time and bandwidth availability of each interface). We consider the application flow arrival process as a Poisson process of parameter λ_a. On each arrival an application flow is a non real-time flow with probability p_{nrt} and a real-time flow with probability p_{rt} (note that $p_{nrt} + p_{rt} = 1$). Each application flow is modelled as described in Section 4.3.4.2. In particular, for real-time flows, we consider four different types, consuming different levels of bandwidth (e.g., 0.5, 0.9, 2 and 3 Mbps). These values correspond to the most typical YouTube quality profiles [108]. Regarding the interfaces, their availability is calculated by interleaving exponentially distributed connected and disconnected periods of parameters λ_c^{3G} and λ_d^{3G} for UMTS/HSPA interfaces and λ_c^{WLAN} and λ_d^{WLAN} for WLAN respectively. In both interface types, the available bandwidth is uniformly distributed.

DECISION-MAKING ALGORITHMS In a second phase, for each application arrival (i.e., an event in the simulation), a decision-making process is triggered. Each event considers the subset of applications (A) corresponding to the active applications (i.e., those transmitting/receiving data) and the current available interfaces (I). For each event, we run an instance of our implementation of the PISA Variator and the NSGA2 Selector. A trade-off of solutions is obtained after g generations. Then, for comparison purposes, we calculate the set of solutions obtained using two preference-based algorithms explained in Section 4.2.1, SAW and MEW, with different weight combinations for each parameter. Since for preference-based algorithms only one solution is obtained on each run, we use different weight combinations to obtain a set of optimal solutions. In our particular case we consider 41 different weight

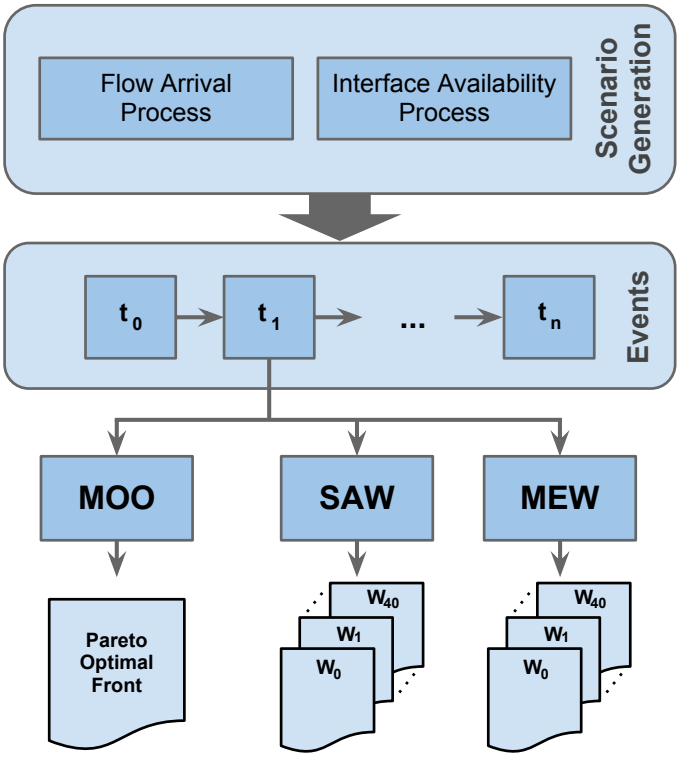


Figure 59: Network selection simulator

vectors $W = [w_E, w_B]$ for E and B from $W_0 = [0, 1]$ to $W_{40} = [1, 0]$ with steps of ± 0.025 (i.e., $[0, 1]$, $[0.025, 0.975]$, $[0.05, 0.95]$, ..., $[0.975, 0.025]$, $[1, 0]$).

4.4.2 *Simulation Results*

In this section, we present a set of simulations results aiming to evaluate the network selection mechanism. First, we provide a detail on the scenarios generation and the parameters chosen for the random variables modelling application flows an interfaces. Then, simulations results for a particular scenario are presented. Finally, a comparative study of the performance of the proposed multi-objective approach using genetic algorithms versus two of the most used preference-based algorithms (SAW and MEW) is presented.

4.4.2.1 *Scenarios and Simulation Parameters*

For each simulation run (that corresponds to a single set of parameters of Table 15), we consider one hundred different random scenarios. For each scenario, an arrival process of 20 application flows is considered. Additionally, a random interface availability process is considered for each scenario. Regarding application flows, they follow the parameters in Table 16. The interface availability process is also calculated using the parameters in Table 17.

Following a Poisson process, the application inter-arrival time is exponential ($\lambda = 1/30$). We have decided that each application is non real-time with

	Variable	Distribution	Parameter
General	Arrival Process	Poisson Process	$\lambda = 1/30$
	Type of flow	Bernoulli	$p = 0.6$ (Non Real-Time)
Non Real-Time	Number of requests	Geometric	$p = 1/5$ [103]
	Number of packets per request	Geometric	$p = 1/25$ [103]
	Packet size (B)	Pareto	scale $= 81.5$, shape $= 1.1$ [103]
	Time between requests (s)	Geometric	$p = 1/12$ [109]
Real-Time	Number of sessions	Geometric	$p = 1/10$
	On-Time (s)	Geometric	$p = 1/30$ [103]
	Off-Time (s)	Geometric	$p = 1/120$ [103]
	Media type	Discrete Uniform	$a = 1$, $b = 4$

Table 16: Simulation parameters for application flows

a probability of 0.6 and real-time with a probability of 0.4. For non real-time applications, the number of sessions is geometric of mean 10 sessions and the bit-rate is uniform between four different bit-rates, depending on the quality. The ON-Time and the OFF-Time follows the distribution and parameters proposed by Yeh et al. [103]. Regarding non-real time flows, they are also modelled as proposed in [103] but considering a more realistic time between request (i.e., the reading time) as proposed in [109] which is geometric of average 12 seconds.

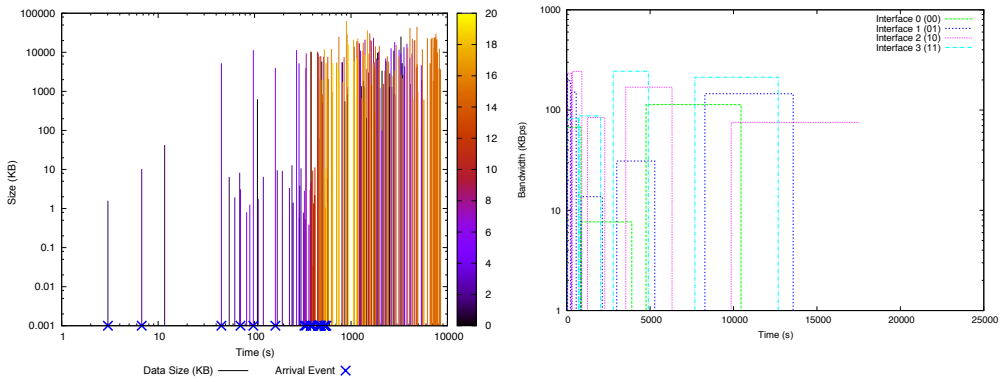
For the interface availability process, we define different parameters for the connected and disconnected time exponential distributions for WLAN and 3G. We consider that an MS remains connected longer to a 3G network and connects intermittently to a WLAN, having a longer disconnected time. We set then the power consumption of each state and the inactivity timers following uniform distributions between common ranges we have found in previous energy measurements (see Tables 3 and 4)

4.4.2.2 *A simulation case*

We consider in this section a set of simulation results for a representative scenario in order to illustrate the performance of the genetic algorithm approach for different number of generations and problem size (i.e., the chromosome size). The proposed scenario corresponds to the flow arrival process illustrated in Figure 60a and to the interface availability process depicted in Figure 60b. In Figure 60a, we illustrate for each application $a_k = \{(s_1, t_1), (s_2, t_2), \cdots, (s_r, t_r)\}$ ($k \in [1, 20]$)the size (s_i, in KB, in the y-axis) and the arrival time (t_i, in seconds, in the x-axis) of each sequence of data (s_i, t_i). Each application a_k corresponds to a different color in the scale. In this case the MS launches up to twenty application flows, the first at time 3 s and the last at time 576 s. On each event, we consider a flow a_k as active (and so to be assigned to an interface) if the event time (t_e) is in between t_1 and t_r, i.e., $t_1 \leqslant t_e \leqslant t_r$. The arrival of new flows, i.e., the time of the first pair (s_1, t_1)

	Variable	Distribution	Parameter
3G	Connected time (λ_c^{3G})	Exponential	$\lambda = 1/1800$
	Disconnected time (λ_d^{3G})	Exponential	$\lambda = 1/2$
	Available bandwidth (KB/s)	Uniform	$a = 1$, $b = 125$
	DCH Power p^{DCH} (W)	Uniform	$a = 0.6$, $b = 1.2$
	FACH Power p^{FACH} (W)	Uniform	$a = 0.2$, $b = 0.5$
	PCH Power p^{PCH} (W)	Uniform	$a = 0.01$, $b = 0.08$
	Inactivity Timer T_1 (s)	Uniform	$a = 5$, $b = 6$
	Inactivity Timer T_2 (s)	Uniform	$a = 4$, $b = 12$
WLAN	Connected time (λ_c^{WLAN})	Exponential	$\lambda = 1/300$
	Disconnected time (λ_d^{WLAN})	Exponential	$\lambda = 1/10$
	Available bandwidth (KB/s)	Uniform	$a = 10$, $b = 250$
	TX/RX Power p^{TxRx} (W)	Uniform	$a = 0.6$, $b = 1.2$
	IDLE Power p^{IDLE} (W)	Uniform	$a = 0.2$, $b = 0.5$
	SLEEP Power p^{SLEEP} (W)	Uniform	$a = 0.01$, $b = 0.08$
	PSM Timeout T_T (s)	Uniform	$a = 5$, $b = 6$

Table 17: Simulation parameters for interfaces



(a) Flow arrival process

(b) Interface availability process

Figure 60: Simulation scenario

for each application a_k, are indicated with blue crosses, which correspond also to the time of the events for the decision-making. Regarding the interfaces in Figure 60b, we consider up to four interfaces (from 0 to 3) having different connection and disconnection times (in the x-axis) and bandwidth availabilities (in the y-axis).

We present six different events of a single scenario. As said before, each event is treated as a unique optimization problem, which corresponds to a given chromosome size (related to the number of applications and interfaces) and has been solved using the NSGA2 genetic algorithm (called GA in Figure 61) using different generation values. We also provide the set of solutions found using SAW and MEW with different weight combinations. We consider, for NSGA, a population size (α) equal to 100 individuals, a number of parents (μ) equal to 50 and an offspring size (λ) equal to 50 individuals.

The Event 1 in Figure 61a is triggered at time 45 s and corresponds to an assignation of 4 applications to 4 interfaces, which gives a chromosome size of 8. The red crosses represent the value of the objectives (E and B) for all the possible flow-interface assignations (exactly 2^8 in this case). Then GA-10, GA-25 and GA-500 represent the solutions for NSGA2 solutions for 10, 25 and 500 generations. We observe that all the possible assignations have $B = 0$, which means that the bandwidth demand is always lower than the offer. The GA-25 approach gives a good approximation to the least energy consuming solution.

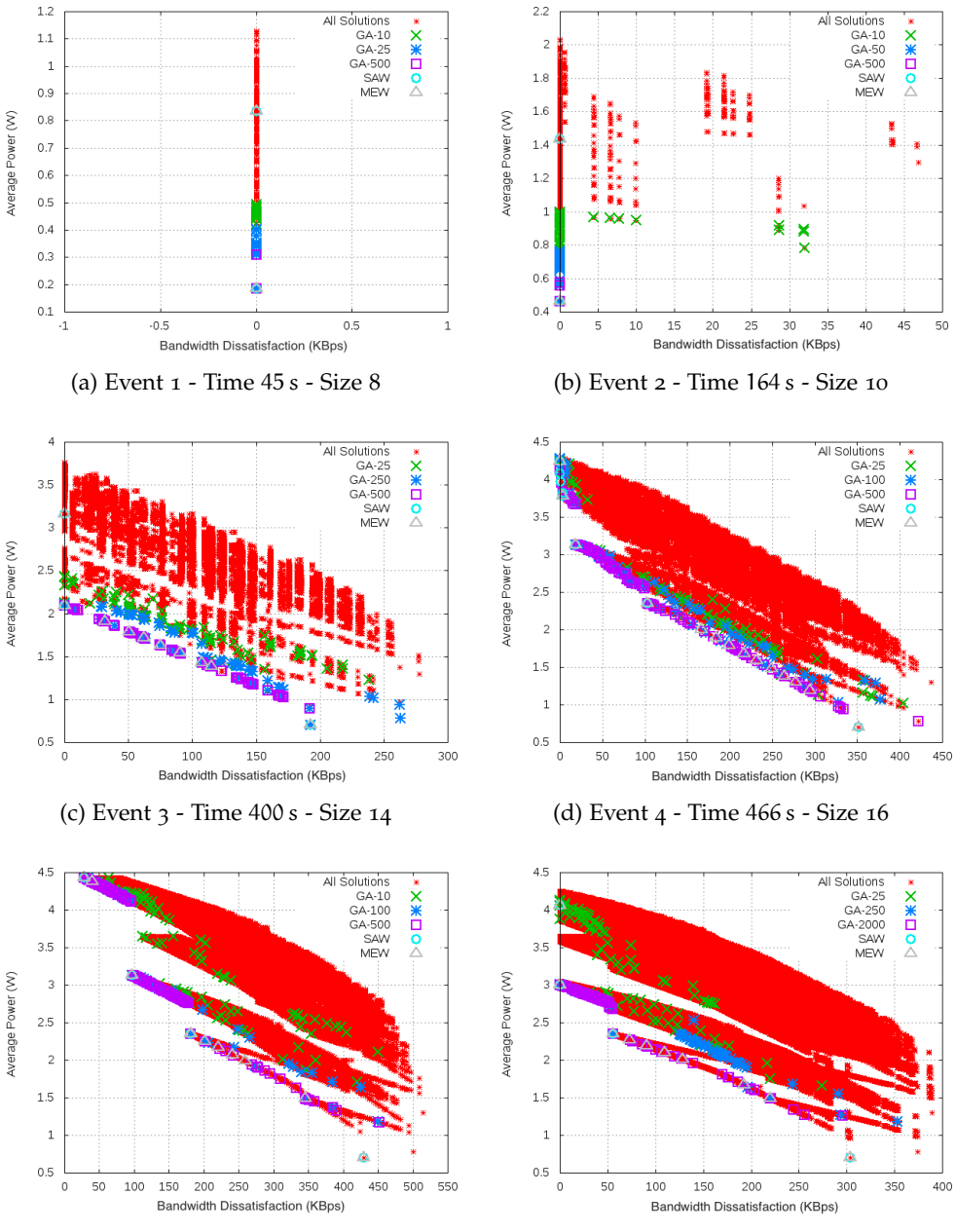
For a later event (Event 2), at time 164 (see Figure 61b), there are a number of solutions that have $B > 0$, meaning that the bandwidth demand is greater than the offer for some assignations. Using a low number of generations (e.g., GA-10) the NSGA2 approach is not capable to find the most optimal solutions, since it provides some dominated solutions on the last generation (e.g., the green crosses with B between 5 and 35). However, for GA-50 and GA-500 the algorithm only provide solutions with $B = 0$.

The problem becomes more complex at time 400 s, where 7 flows have to be assigned to 4 interfaces, giving a chromosome size of 14 and up to 2^{14} possible solutions. We observe that GA-500 gives a good approximation of the Pareto-optimal front. For lower number of generations, NSGA2 finds non-dominated sets that are relatively far from the real front. Particularly for the non-dominated set found by GA-25, it does not provide any solution belonging to the front. The same behavior is observed for later events, having longer chromosome size. Note that for greater chromosome size, in order to have a better approximation of the Pareto-optimal front, the Genetic Algorithm needs to iterate for a longer number of generations. This is the case of Event 6 at time 537 s, which requires between 1000 and 2000 generations to provide a good approximation of the trade-off.

Regarding the solutions found using preference-based algorithms SAW and MEW, in both cases the solutions belong to the Pareto-optimal front, since a full search (i.e., all solutions are evaluated) is performed on the decision space. This gives perfect closeness to the Pareto-optimal front. However, we observe that there is a larger separation between solutions than for the NSGA2 obtained solutions, affecting the diversity of the set, which is one of the main concerns while approximating the optimal set. In the following section, we focus on the comparison of the genetic approach against the preference-based algorithms.

4.4.2.3 *Comparative Analysis: Genetic vs. Preference-based Algorithms*

In this section we aim at studying the differences between solving the optimization problem using a genetic approach and a preference-based algorithm, focusing on different performance metrics. To this end, we have performed a set of simulations using different configurations for the genetic algorithm (NSGA2), i.e., the number of individuals in the initial population



(a) Event 1 - Time 45 s - Size 8

(b) Event 2 - Time 164 s - Size 10

(c) Event 3 - Time 400 s - Size 14

(d) Event 4 - Time 466 s - Size 16

(e) Event 5 - Time 477 s - Size 20

(f) Event 6 - Time 537 s - Size 22

Figure 61: Simulation results

Parameter	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
α	20	60	100	20	60	100
μ	10	30	50	10	30	50
λ	10	30	50	10	30	50
c_t	One-point	One-point	One-point	Uniform	Uniform	Uniform
p_c	1	1	1	1	1	1
m_t	Indep-bit	Indep-bit	Indep-bit	One-bit	One-bit	One-bit
p_m	1	1	1	1	1	1
p_b	0.5	0.5	0.5	0.9	0.9	0.9

Table 18: Simulation cases

(α), the number of parents (μ), the offspring size (λ) and the different parameters for mutation and cross-over. The different parameters and their values are listed in Table 18. On each simulation case, we have simulated 100 scenarios of 20 event each, in order to obtain good averaged values. As observed in Table 18, for NSGA2, we consider three population sizes (20, 60 and 100) and two sets of parameters for mutation and crossover. One of the configurations considers One-Point cross-over and Independent-Bit mutation. In this case, for each pair of solutions selected from the matting pool crossover is always performed ($p_c = 1$). Regarding mutation, it is performed to all the solutions ($p_m = 1$) and in this case all the bits are turned with a probability $p_b = 0.5$. The second set of parameters for NSGA2 considers a less aggressive mutation (only one bit is mutated with a probability $p_b = 0.9$) and uniform crossover. For uniform crossover, the bits of two solutions from the matting pool are switched with a probability $p = 0.5$.

We perform a set of simulations in order to compare both the genetic and the preference-based algorithms using four metrics:

- **Diversity of solutions**: the number of different solutions found on each approach
- **Optimality of the solutions**: a measure to the closeness to the Pareto-optimal front
- **Calculation time**: the delay to come out with a set of optimal solutions
- **Number of reallocations**: the number of flows that have to be reallocated to a different interface than its previous one after a decision-making

DIVERSITY OF SOLUTIONS We measure the diversity of the solutions of the different approaches as the number of different solutions in the obtained non-dominated set. In Figure 62, we show the number of different solutions obtained using SAW, MEW and the genetic algorithms for different chromosome size. For the genetic algorithms, we consider eight different number of generations, i.e., 10, 25, 50, 100, 250, 500, 1000 and 2000.

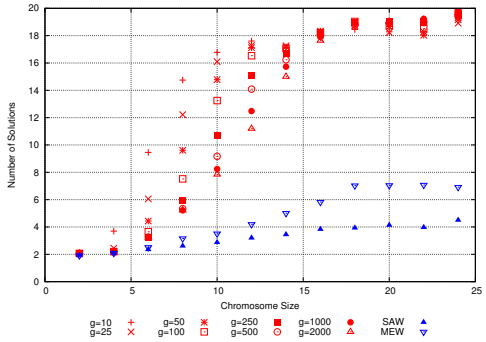
In the case of preference-based SAW/MEW solution sets, providing a single solution per weight vector, we observe in all simulation cases in Figure 62 that even if 41 different weight vectors are considered for each event, they cannot provide a large number of different solutions. For SAW and MEW, each weight combination gives a single solution. However, different weight vectors can give exactly the same solution, reducing the total number of solutions found and so the diversity. This limitation of preference-based algorithms is also raised by Deb in [95]. The author suggests that a decision-maker may pick up a single solution (using weights or other high level information) from the Pareto-optimal front (calculated after solving the problem using Genetic Algorithms) instead of using preference-based algorithms to obtain the front. In the latter case, it is not always possible to obtain a good approximation of the Pareto-optimal front using different weight combinations. In our simulations (see Figure 61) even if all the solutions provided in SAW/MEW belongs to the Pareto-optimal front, we observe a low number of solutions, giving a reduced diversity. Additionally, we observe that in all simulation cases, MEW is able to provide a greater number of solutions than SAW.

Regarding the genetic algorithm, we observe that the diversity increases with the chromosome size. This is because an increasing chromosome size indicates a large number of applications, which gives more possible solutions and then a larger range for E and B. Observe also that using the second set of parameters for NSGA2 (i.e., simulation cases 4, 5 and 6) we obtain almost the same number of solutions for every generation value (except for $g = 10$). However, since for the second set of parameters we use one-bit mutation (which leads to a less aggressive variation), the number of solutions as a function of the chromosome size seems to increase more slowly than for the first set of parameters.

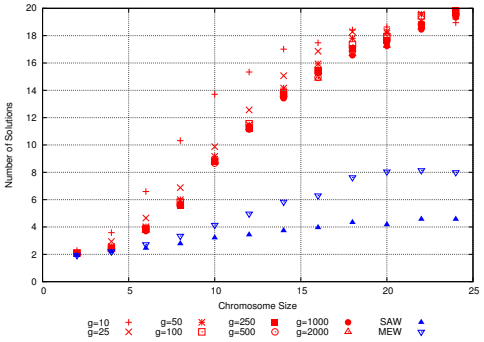
OPTIMALITY OF SOLUTIONS In order to measure the optimality of the solutions provided by the different optimization algorithms, we use a reference-point based quality indicator as proposed in [110]. We consider the set of solutions $Z^* = \{z_1^*, z_2^*, \cdots, z_l^*\}$ ($l = 2^c$, where c is the chromosome size and $z_i^* = \{E_i, B_i\}$) representing all the possible flow-interface assignations in the scenario. We choose two reference-points: the ideal solution Z^*_{min} (see Equation 51), that is composed of the minimum values for E and B in the problem and the worst solution Z^*_{max} (see Equation 52) that considers the maximum values for both E and B.

$$Z^*_{min} = \{min_{\forall i} E_i, min_{\forall i} B_i\} \; i \in [1, l] \tag{51}$$

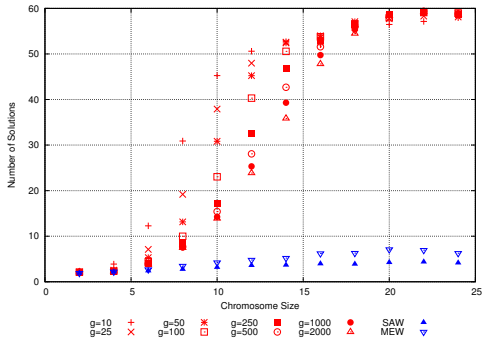
$$Z^*_{max} = \{max_{\forall i} E_i, max_{\forall i} B_i\} \; i \in [1, l] \tag{52}$$



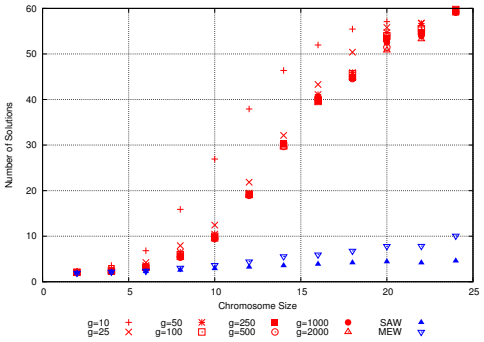
(a) Simulation Case 1



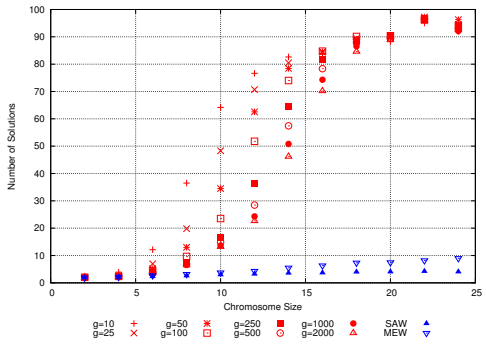
(b) Simulation Case 4



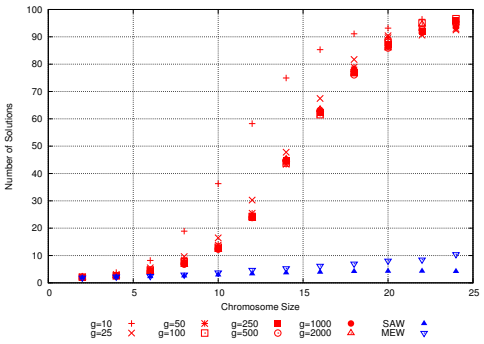
(c) Simulation Case 2



(d) Simulation Case 5



(e) Simulation Case 3



(f) Simulation Case 6

Figure 62: Diversity: number of different solutions

Then, consider the non-dominated sets obtained after solving the problem using the genetic algorithm (Z_{GA}), SAW (Z_{SAW}) and MEW (Z_{MEW}). We can apply Max-Min normalization (see Section 4.2.1) using Z^*_{min} and Z^*_{max} over these sets, giving the normalized sets Z'_{GA}, Z'_{SAW} and Z'_{MEW}. These normalized sets contain normalized values for the objectives in the range $[0, 1]$, which facilitates comparisons.

Since the goal is to compare the optimality of the solutions provided by the different solving algorithms, we propose to use the minimum distance to the ideal solution Z^*_{min}. The distance to the ideal solution is computed as the Euclidean distance (operator d) between the solutions in the normalized sets and Z^*_{min}. For each solving algorithm (GA, SAW and MEW) we find the solution having the closest distance to the ideal solution as shown in Equation 53.

$$\begin{aligned} d_{SAW} &= \min_{\forall i} d(z'_{SAW}, Z^*_{min}) \\ d_{MEW} &= \min_{\forall i} d(z'_{MEW}, Z^*_{min}) \\ d_{GA} &= \min_{\forall i} d(z'_{GA}, Z^*_{min}) \end{aligned} \tag{53}$$

Finally, we calculate the differences $D(GA, SAW)$ and $D(GA, MEW)$ between d_{GA} and d_{SAW} and d_{MEW} respectively (see Equation 54). Using this difference, we can see which approach provides solutions closer to the ideal solution. Then, if $DA(GA, SAW)$ is positive, it means that SAW found solutions closer to the ideal than the genetic algorithm. If $DA(GA, SAW)$ or $DA(GA, MEW)$ equal zero, it means that the genetic algorithms and the preference-based approaches provide solutions with the same distance to the ideal solution.

$$\begin{aligned} D(GA, SAW) &= d_{GA} - d_{SAW} \\ D(GA, MEW) &= d_{GA} - d_{MEW} \end{aligned} \tag{54}$$

In Figures 63 and 64 we illustrate $D(GA, SAW)$ and $D(GA, MEW)$ respectively, for different chromosome sizes. We observe that for an increasing chromosome size, the genetic approach gives increasing positive values for $D(GA, SAW)$ and $D(GA, MEW)$, meaning that the the closest distance to the ideal solution in genetic algorithms is greater than for SAW and MEW. However, for increasing α, μ and λ, the genetic algorithms can provide closest distance to the ideal solution. This is an indication of a better approximation to the Pareto-optimal front for the genetic algorithms. Regarding the number of generations, the closest distance to the ideal solution provided by genetic algorithms decreases for an increasing number of generations. Note that when using the second set of parameters for the genetic algorithm we obtain much lower values for $D(GA, SAW)$ and $D(GA, MEW)$, showing that, for our problem, uniform mutation and one-bit mutation can better explore the decision space.

We have observed that in addition to providing a greater number of different solutions than SAW, MEW is also capable to give closer solutions to the ideal solution. This is observed in Figure 64, where it is most common to find negative values for $D(GA, SAW)$ than for $D(GA, MEW)$ for the the two set of parameters

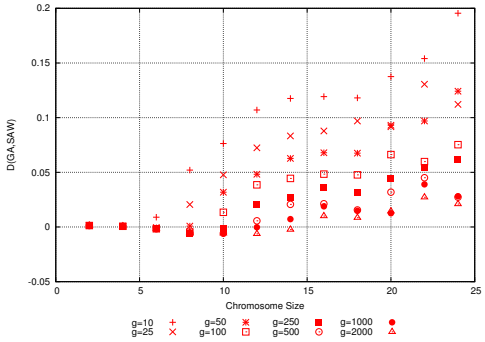
CALCULATION TIME The calculation time, i.e., the time to obtain an approximation to the Pareto-optimal front, is a main concern in network selection. In this context, mobile devices are usually on the move and need to take decisions as fast as possible in order to prevent impacting the on-going applications. We compute in our simulations the calculation time for the genetic algorithms (with different numbers of generations) and preference-based approaches (SAW and MEW).

Regarding genetic algorithms we compute the time spent between the initialization of the random population and the last iteration (i.e., the last generation of solutions). However, as we have stated before in Section 4.4.1.1, the PISA implementation uses two independent processes, the Variator (i.e., the optimization problem) and the Selector (i.e., the solving algorithm or optimizer, in our case NSGA2) that communicate through text files to indicate the start and the end of the variation and selection processes. Each time the Variator does mutation and crossover over the current population, it waits until the Selector performs non-dominated sorting and selection. This adds a time overhead that may not exists if the Variator and the Selector were implemented as a single process. For our simulations, we have reduced to 1 ms the frequency used by the Variator to check for the termination of the Selector, and vice-versa. Then, for the genetic algorithms, we can consider the calculation time shown in Figure 65 as an upper bound.

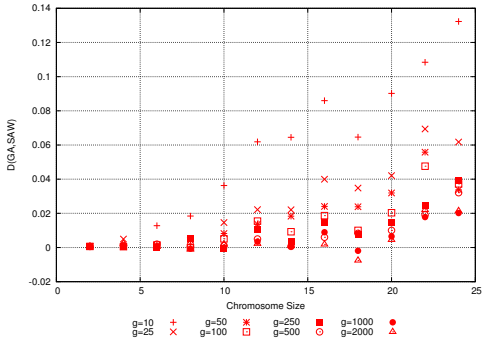
We observe in Figure 65 that, for the genetic algorithms, the calculation time does not depend on the chromosome size of the problem but on the number of generations, meaning how many times the algorithm will iterate before providing an approximation to the Pareto-optimal front.

In the case of SAW/MEW, there is an exponential relation between the calculation time and the chromosome size (note that the y-axis has log scale). Recall that for SAW and MEW, a full search is performed on the decision space. In this case, the algorithm first iterates to perform normalization (in the case of SAW) and determines the minimum and maximum bounds for E and B, since they are required for the computation of the normalization and the multiplicative weighting (see Section 4.2.1.2).

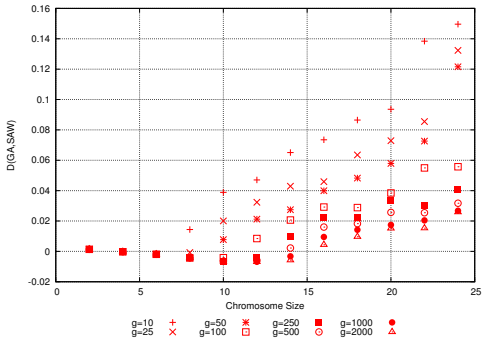
We observe that the genetic algorithms calculation time slightly increases, in all the considered generations, for greater α, μ and λ (i.e., the different simulation cases). Computing solutions using preference-based algorithms performing a full search on the decision space becomes as time consuming as genetic algorithms for chromosome sizes greater than 8 or 10 (i.e., 4 or 5 applications among 3 or 4 interfaces). For a chromosome size greater than 18



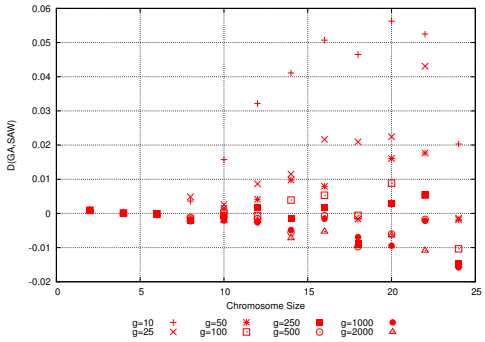
(a) Simulation Case 1



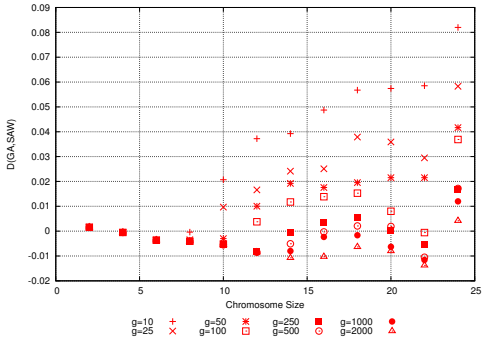
(b) Simulation Case 4



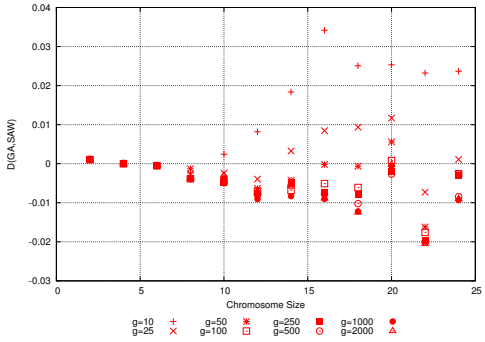
(c) Simulation Case 2



(d) Simulation Case 5

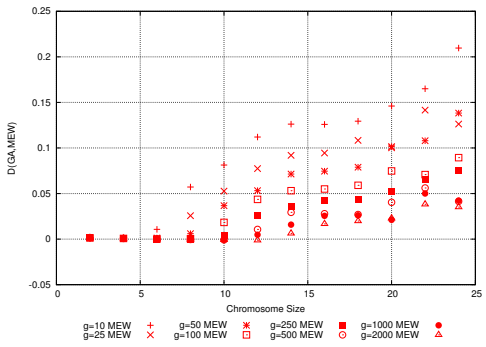


(e) Simulation Case 3

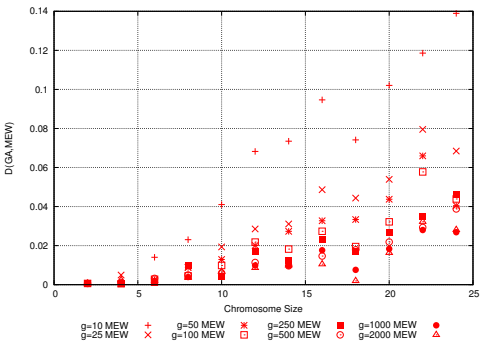


(f) Simulation Case 6

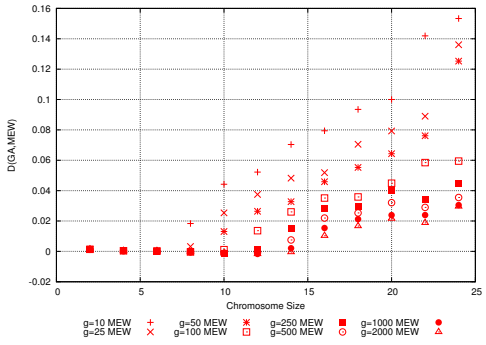
Figure 63: Optimality comparison against SAW



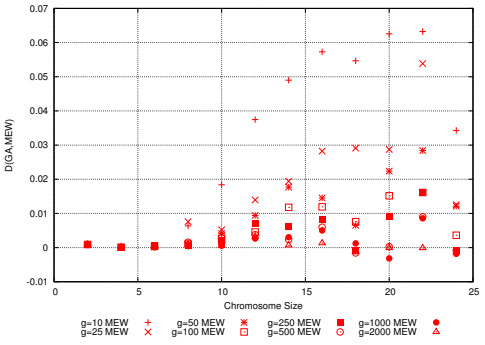
(a) Simulation Case 1



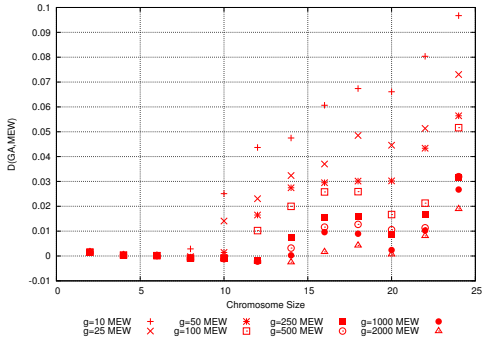
(b) Simulation Case 4



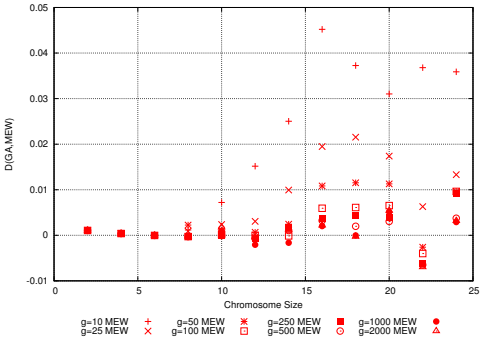
(c) Simulation Case 2



(d) Simulation Case 5



(e) Simulation Case 3



(f) Simulation Case 6

Figure 64: Optimality comparison against MEW

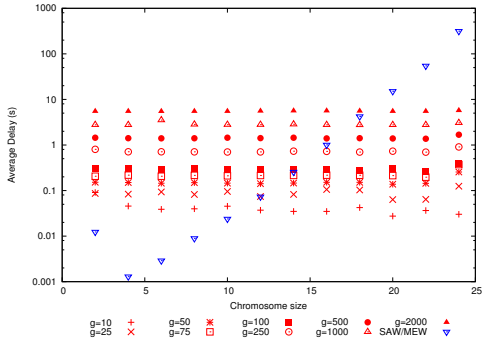
or 20 (depending on the simulation case), the preference-based algorithms becomes high time consuming, spending more up to 200 s for a 24 chromosome size.

NUMBER OF REALLOCATIONS When performing flow-interface assignation, it is not desirable for a given flow to be assigned to different interfaces each time a decision-making is performed, since it may affect its performance. The set of solutions obtained after solving the decision-making problem should provide optimal solutions that minimize the number of reallocations, i.e., the number of flows that have to be assigned to a new interface different than the previous one.

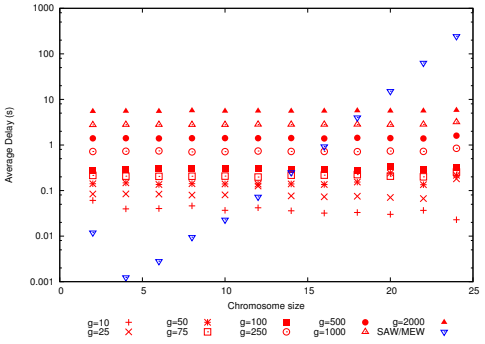
To measure the number of reallocations for each decision-making algorithm we first obtain, for each event in the scenario, the solution from the set giving the lowest number of reallocations from the previous event. We consider that, for the previous event, the closest to the ideal solution is chosen for each decision algorithm (NSGA2, SAW and MEW). Finally, for each simulation case, we calculate the average number of reallocations observed on each event. This is illustrated in Figure 66. We observe that the genetic algorithms using a reduced population size ($\alpha = 20$, $\mu = 10$ and $\lambda = 10$) are not able to provide solutions with lower number of reallocations than SAW and MEW. For higher α, μ and λ, the genetic algorithms can provide a lower number of reallocations. Comparing the two sets of parameters for the genetic algorithms, we observe a lower number of reallocations using the second set, regardless to the number of generations. This is observed in Figures 66b, 66d and 66f.

4.4.2.4 *Discussion*

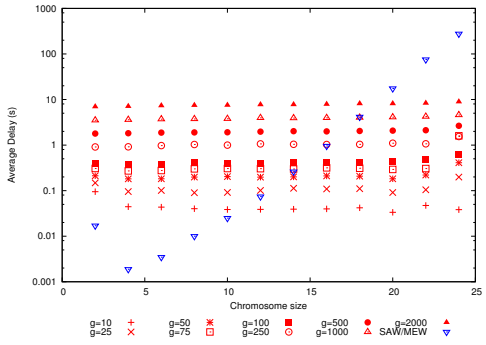
The simulation results show that using genetic algorithms to solve the network selection problem allows obtaining a set of optimal flow-interface assignations, representing the best trade-off between the two objectives: the energy consumption and the bandwidth dissatisfaction. For problems having a low number of applications to be assigned to a low number of interfaces, we observe that the calculation time for the genetic algorithms is higher than doing a full search using preference-based algorithms, since for the genetic algorithms, it only depends on the number of generations. Note that, in the results presented for the genetic algorithms, there are additional calculation delays caused by the inter-process communication between the Variator and Selector in PISA, which is based on state variables that are written in plain text files. In a real implementation of the genetic algorithm in a device, the Variator and Selector could be implemented in a single process, reducing the time overhead observed in PISA. However, for problems concerning more than 4 or 5 applications (e.g., a chromosome size of 8 or 10) the NSGA2 with low number of generations have equivalent calculation delays. We have also observed that the genetic algorithms, compared to the preference-based



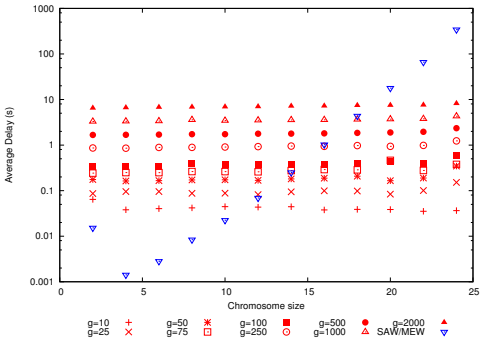
(a) Simulation Case 1



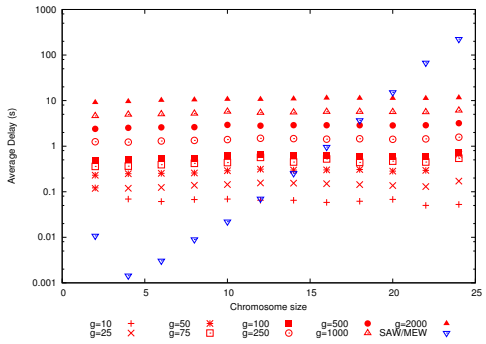
(b) Simulation Case 4



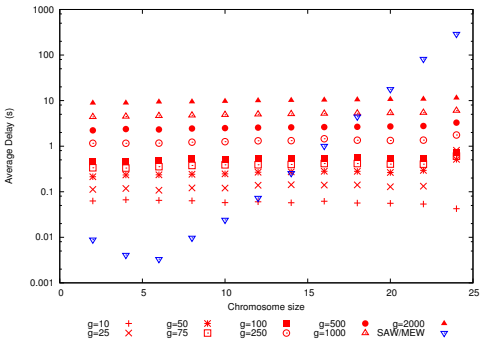
(c) Simulation Case 2



(d) Simulation Case 5

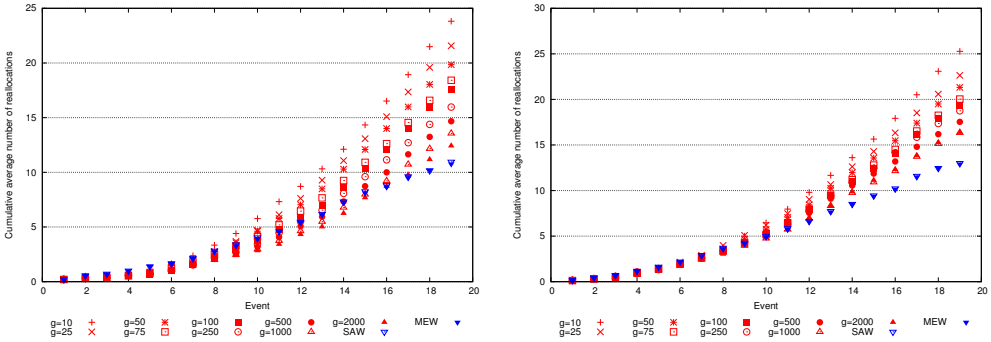


(e) Simulation Case 3



(f) Simulation Case 6

Figure 65: Calculation time



(a) Simulation Case 1

(b) Simulation Case 4

(c) Simulation Case 2

(d) Simulation Case 5

(e) Simulation Case 3

(f) Simulation Case 6

Figure 66: Number of reallocations

approaches always provide a greater diversity of solutions, giving a good approximation to the real Pareto-optimal front.

After performing decision-making and obtaining a set of optimal solutions, an MS must select one particular solution from the optimal set and perform flow-interface assignation. At this point, the decision-maker (i.e., the mobile user) can use high level information to take the final decision. Some possible alternatives to select the most suitable solution from the optimal set are as follows:

- **A posteriori preference-based**: The decision-maker may set up a single objective optimization using weights to combine E and B (like in SAW or MEW) but only considering the solutions obtained in the previously calculated optimal set using the genetic algorithm. This problem is easy and fast to solve since it only concerns a very limited number of decision vectors (depending on the population size used for the genetic algorithm). In this case, the mobile user can dynamically set a weight vector (giving more or less preference to E and B) depending on the current condition, e.g., if the mobile is running out of battery, one may give more preference to E.
- **Proximity to the ideal solution**: The decision-maker can select, as the most optimal solution, the solution with the closest distance to the ideal solution composed of the best values of E and B in the previously obtained set.
- **Similarity to the current assignation**: In order to minimize the impact of switching the application flows between the different interfaces, the mobile user can select the most similar optimal solution to the current assignation (i.e., the solution of the previous event). Then, it can consider the solution that minimizes the number of reallocations. Note that in this case the decision does not consider the objectives values.

4.5 CONCLUDING REMARKS

In this chapter, we have introduced the network selection problem in multihomed mobile devices. First, we have surveyed the existing mechanisms for network selection, that mainly focused on vertical handover between WLAN and 3G, where the problem modelling is based on deciding when to switch all the flows to a different interface. These mechanisms consider different criteria, including the QoS required by the applications and provided by the interfaces, the security level, the monetary and energy cost. Regarding energy, in existing frameworks, the energy consumption is usually measured as a constant cost or penalty produced by the event of turning an interface on.

In order to model the network selection problem, MADM, MOO, Neural Networks and combinatorial optimization frameworks have been proposed

so far. However, in the existing frameworks, we identified some limitations. First, they are based on multiple criteria to be optimized, but in most of the cases, the authors do not consider the corresponding monitoring mechanisms that provide all these criteria to the decision-making engine (i.e., the input of the network selection problem). Then, regarding the decision-making engine itself, it usually considers a combination of the different criteria using some weighting schemes, in order to simplify the problem. This is referred to as a preference-based decision-making. As it has been noted in [95], solving the decision-making problem using a preference-based approach may prevent the decision-maker from finding the complete trade-off of optimal solutions among different criteria.

Based on this, we propose to model the network selection based problem using MOO. We consider the model as a flow-interface assignation, which unlike traditional vertical handover decision-making, is capable to individually assign flows to different interfaces, enabling load spreading in multi-homing devices. In this mechanism, we consider two criteria that do not require complex monitoring. First, we consider the overall bandwidth dissatisfaction of the mobile device by computing the difference between the overall bandwidth demand (of the applications) and the available bandwidth of each interface. Second, we consider the overall energy consumption of the mobile device as an estimation of the average power consumed by each interface. In the latter case, for a given flow-interface assignation, we estimate the amount of time each interface spends on each operating state (e.g., idle, transmit, sleep). Then we use the power consumption for each state to estimate the average power.

To solve this optimization problem, we propose and evaluate the usage of genetic algorithms. For the evaluation, we provide a set of simulation results that consider different scenarios (i.e., different application flows and interface availability) and solve the optimization problem with traditional preference-based algorithm and different configurations for a particular genetic algorithm, NSGA2. The simulation results show that genetic algorithms provide a good approximation to the Pareto-optimal front and more reduced calculation delays than preference-based algorithms for problems aiming to assign more than 4 or 5 applications to 3 or 4 different interfaces. Observe that even if our proposed algorithm can enhance the decision-making in a single MS, in the case of a mobile network (i.e., composed of several nodes connected to a mobile router), the problem size scales and so our multi-objective approach can provide optimal assignations in a relatively short delay. For less applications, preference based algorithms seem to provide some optimal solutions faster than genetic algorithm, even if in the latter case some time overhead may be added by the simulator framework (i.e., PISA) architecture.

5

CONCLUSION AND PERSPECTIVES

5.1 CONCLUDING REMARKS

The current wireless environment is characterized by a diversity of technologies providing broadband Internet accesses, including WPAN, WLAN, and cellular networks (2G, 3G and 4G). Particularly in the last years, there has been a proliferation of community networks based on IEEE 802.11 that have been deployed by residential ADSL subscribers, providing ubiquitous WLAN coverage in urban environments. To access those networks, mobile users are now multi-homed, i.e., they can have access to different networks at any place and any time, using different wireless interfaces. In this context, mobile users may exploit the network diversity in an Always Best Connected manner, i.e., to obtain the best connection performance while being simultaneously connected to multiple wireless networks. Such an exploitation of the network diversity poses a number of questions that we tried to address in this thesis, e.g., how to optimize the handover process between different points of attachment or how to perform an optimal network selection while using multiple wireless networks.

In order to analyze the potential for an Always Best Connected scenario in current wireless networks, we provided in Chapter 2 an evaluation study of IEEE 802.11 and cellular networks, particularly focusing on the performance of CN deployments. This evaluation study has been performed using a new participatory sensing platform we have designed and implemented for the Android mobile system, called Wi2Me. We have gathered traces from more than 6000 AP and 60 cellular base stations and generated almost 700 connections to CN and cellular networks. We proposed a set of metrics to characterize the different networks. We have observed that CN provide the same level of coverage than cellular networks in urban areas, albeit with a low average received power from the IEEE 802.11 APs, limiting the connection performance and duration, which was in median 27.5 s at pedestrian speed. Regarding the deployment condition in IEEE 802.11, the uncontrolled AP location and channel settings may produce a high level of interference that is generated by several APs allocated on the same or in an overlapping channel. In this case, we have observed that around 80 % of the APs are deployed in the non-overlapping channels (1, 6 and 11), giving that in around 50 % of the cases a single channel is shared by two or more APs. We have also analyzed the impact of handovers on on-going communications. In managed deployments (like a corporate or campus deployment), even if handovers are supported and the on-going communications do not break after an AP

transition, we have shown that the TCP connections are highly degraded due to a reduction of the CWND. Particularly in existing CN deployments, we have found that even if the high density of CN AP could provide the MS a continuous connection through CN, there is still a lack of handover support providing seamless transitions between APs and a continuity of the on-going applications flows.

Regarding the optimization of the handover process, we have focused on the IEEE 802.11 AP discovery process in Chapter 3, since it is the most time-consuming process during a handover. The discovery process during an IEEE 802.11 handover consists in performing active scanning in the available channels. On each channel, the MS sends Probe Request frames and waits for Probe Responses from different AP during a certain waiting time. This waiting time is managed by two timers: *MinChannelTime* and *MaxChannelTime*, whose values are not defined in the IEEE 802.11 standard. The values of the timers on each channel strongly depend on the delay of the Probe Responses, which appears to have a high variability, depending on the channel condition (e.g., congestion, interferences). Then, there is a a trade-off when setting the timers, since fixing low values for the timers to reduce the scanning latency may also reduce the number of discovered APs or, even worst, avoid discovering any AP in the channels. Unlike most of the current state of the art, that mainly focused on reducing the latency of the discovery process to mitigate the impact of handovers on on-going communications, we considered the trade-off between the duration of the process and the amount and quality of the AP discovered during a scanning. To manage this trade-off, we have proposed a set of adaptive scanning algorithms aiming at setting the most suitable scanning parameters for each particular channel. In this context, we first defined the Adaptive Discovery Algorithm (ADA), which dynamically sets the waiting time on each channel depending on the previously discovered APs. Then, we defined and evaluated the Cross-Layer Adaptive Scanning Algorithm, which uses physical layer information (i.e., the channel load and the power measured on each channel) to set the most suitable channel sequence and the waiting time for each channel. We implemented both adaptive functions in open-source IEEE 802.11 drivers and performed an experimental evaluation in two different testbeds, in order to compare the adaptive algorithms against typical fixed-timers approaches (using different values for the timers). We have found that using an adaptive strategy while scanning allows managing the trade-off between the scanning latency and the topology discovery, i.e., for a given latency, an adaptive algorithm is able to discover a larger number of access points. In the case of ADA we have observed that at most in 2 % of the cases the MS is not able to discover any AP after scanning all channels. However, this percentage increased up to 52 % when using low fixed timers. Using Cross-Layer adaptive scanning, since the MS uses longer timers in the channels where APs are more likely to be deployed, we have found that the algorithm provides a low scanning

latency and a high number of discovered APs, while also reducing the time to discover the first AP. The latter metric becomes important in the case that the MS needs to find a single AP as soon as possible.

In Chapter 4, we have focused on the decision-making process for network selection in multi-homing devices. Two different approaches exist when performing decision-making for network selection. The first approach focuses on the network selection during a vertical handover, i.e., when to decide to switch all the on-going application flows to a new available wireless interface. The second network selection approach consists in finding the most optimal flow-interface assignation in the case the MS wants to simultaneously use several interfaces. In both cases, existing works on network selection have modelled the problem using different optimization tools, considering various criteria (e.g., QoS, cost, security, energy). In our approach, we have modelled the network selection as a decision-making problem for flow-interface assignation using multi-objective optimization and genetic algorithms to search for feasible solutions. We have considered two minimization objectives for the multi-objective approach: the energy consumption and the bandwidth dissatisfaction. Regarding the energy consumption, differently from existing network selection mechanisms, where the energy is considered as a fixed cost associated to each interface, we have taken into account the energy consumption in a fine-grained way. To this end, we use traffic models to estimate the amount of energy consumed by a given flow-interface assignation. For the second objective, we consider the bandwidth dissatisfaction as the difference between the bandwidth demand of the application flows and the available bandwidth provided by the interfaces. We have evaluated our mechanism by simulation, considering different scenarios corresponding to a mobile user running different types and number of applications over different interfaces. We have found that solving the decision-making problem using genetic algorithms can provide a complete view of the trade-off between the objectives (energy consumption and bandwidth dissatisfaction). Moreover, when the number of applications to assign scales, the genetic algorithms provide a lower calculation delay to obtain the Pareto-optimal front than performing full search with preference-based algorithms. Finally, depending on the user's policies or other high level information, one single solution may be chosen from the optimal set.

5.2 FUTURE WORK

In order to extend the contributions of this thesis, the following issues may be considered for the future work. Regarding the Wi2Me platform, we have recently published and distributed the source code of the Wi2Me-User and Wi2Me-Research applications and applied to several project calls in order to extend the possibilities for further collaboration with research groups, network operators and other industrial partners. In current community net-

works, the network operators have no return on their deployment characteristics and performance (e.g., channel allocation, density of APs, average bandwidth and QoS). In addition, the usage pattern of community networks by the mobile users remains unknown. Both limitations could be solved by performing very large evaluation studies using Wi2Me-User and Wi2Me-Research. Using Wi2Me-Research, we are able to analyze the deployment of wireless networks (IEEE 802.11, 2G, 3G or LTE). It is thus possible to characterize the environment, the coverage area and the complementarity of different access networks in different locations. At this time, there is no other tool to perform these studies with a high level detail. Concerning the use of community networks, the Wi2Me-User application is able to trace the user's activity. Thus it would be possible for the network operators to determine which applications are being used over their networks and derive the traffic model (i.e., the type of flows, the amount of data to transmit/receive, the arrival process of the flows, the place and time of the connections). A network operator providing access to different technologies could benefit from such a study by better knowing the possible connectivity options for each user at any place. For any geographic point, an operator could have a preliminary knowledge of the available technologies and their performance. Thus, these data may allow better sizing and planning the networks or deploying offloading techniques (e.g., transferring some 3G users to an IEEE 802.11 community network under certain conditions). In a second step, the Wi2Me-User application could be massively distributed through the Google Play platform, providing a community network connection manager that is able to trace the usage of the networks and the deployment characteristics. The distribution of this application would create a large scale automated tracing system that could be very beneficial not only to operators but to the research community in order to better understand the network diversity. In addition, it will allow the mobile users to take advantage of automatic connection tools for community networks that improve connectivity across heterogeneous networks.

Regarding the contributions presented in Chapter 3, the future work may focus not only on the definition of the most feasible parameters for the scanning algorithm but also on the handover triggering. We have found that current mobile devices lack efficient mechanisms to trigger the handover process. In most cases, they base their handover decision on the current link condition, i.e., they wait until the instantaneous value of the received signal strength from the AP reaches a fairly low level to trigger a handover. Then, new triggering mechanisms need to be designed in order to allow the MS to rapidly react to link degradations and so start the handover process as soon as better APs become available. The main goal of better triggering the handover is to enhance the mobile user experience by keeping a high signal strength, a low number of packet retransmissions and a high connection throughput while being on the move and traversing several APs. In such

a mechanism, the MS continuously monitors the received signal strength from the current AP and instead of considering its absolute value, it may use some filtering/smoothing technique (such as Kalman filters) to predict the short term evolution of the signal strength and so decide when to trigger the adaptive scanning. This adaptive scanning, as proposed in this thesis, may integrate physical layer information to set the most suitable parameters. This information could be based on some message exchange using the IEEE 802.11k amendment, like channel load reports or noise histograms. Moreover, the scanning configuration may also consider the effect of channel overlapping when receiving Beacons or Probe Responses. It has been previously observed [49] that when probing a channel, the MS may receive responses from APs in the neighboring channels, albeit with a lower RSSI. The adaptive scanning algorithm may consider this issue by intelligently selecting a channel sequence that takes profit from the overlapping nature of the channels. For instance, a channel may not be probed if a response from an AP deployed on this channel has been received when probing a neighboring channel. This allows the MS to reduce the latency of the scanning process. Another possible adaptation strategy for the scanning process may use triggers from the higher layers (like the transport or the application layers) and set the scanning parameters depending on some specific QoS profiles. Then, an MS may use low timers and a reduced channel sequence to achieve a low latency scanning if the QoS profile of current flows/applications require a new AP as soon as possible (e.g., if the user runs VoIP or streaming flows). On the other hand, the MS may set longer timers and larger channel sequences to discover a high number of APs if the QoS profile is more elastic (e.g., web-browsing, messaging or microblogging applications) and can tolerate a longer connectivity disruption during a handover.

With regards to the decision-making framework for network selection that we have proposed in Chapter 4, we have evaluated its performance by the means of simulation. For the future work, an implementation and evaluation of this mechanism may be carried out using real multi-interface mobile devices moving in a heterogeneous wireless deployment. Moreover, as stated before, we should consider an hybrid approach for network selection, that uses preference-based algorithms (or MADM) doing a full search on the decision space for small problems (i.e., a few number of applications to be assigned to the available interfaces) and uses a genetic algorithm to search for solutions when the problem size scales (i.e., a high number of applications and interfaces). In addition, other existing solution searching algorithms/heuristics may be evaluated and compared against the genetic approach, since in this thesis we have only compared the performance of our proposed mechanism against a full-search in the decision space. Additionally, the decision-making process needs to be integrated to the mobility and multi-homing support architecture. In this case, as stated before, multi-homing protocols like shim6 or HIP may be implemented in the mobile de-

vices to allow enforcing the output generated by the decision-making mechanism. Regarding the input of the decision-making mechanism itself, in order to select an optimal solution from the computed trade-off, high-level information or policies still may be required. These policies may be provided not only by the mobile user but by an external entity as well (e.g., the network operator). Then, there may be a need for a new component in the network to store and maintain those policies and a set of messages to request or modify them.

5.3 PERSPECTIVES

From a more general perspective, we consider that tomorrow's networks will not be composed of a single wireless access covering all the user needs but of several technologies, networks operators and deployments competing in the market. Then, in order to achieve the Always Best Connected paradigm in such an scenario, a high level of cooperation among the different actors will be required. This will allow the mobile users to fully exploit the network diversity by connecting to different access networks belonging to different technologies and operators, performing handovers and dynamically adapting the usage of the networks depending on the particular scenario. Such a cooperation will take place not only between network operators but between the users as well. Network operators will need to establish roaming agreements, allowing the subscribers from different operators to use their deployments belonging to different technologies. They will also need to deploy common mobility and multi-homing support protocols, which may guarantee the mobile user the continuity of the on-going communications even when a handover to a network managed by a different operator occurs. Regarding the collaboration among users, they may be able to exchange information about the best available networks at any given place, which can be used as an input for handover and flow-interface assignation decisions.

Second, from the mobile device point of view, even if network operators could provide a full mobility and multihoming support in future networks and if the mobile devices can assure a low latency discovery process and an optimal flow-interface assignation, the mobile devices currently lack of reactivity to rapidly detect changes in the topology (e.g., new networks, a link going down or a degrading QoS) and take smart decisions. However, being reactive requires monitoring different performance parameters and potentially exchange information among users, which may contribute to more overhead and, especially, an increasing energy consumption. Then, new algorithms and mechanisms should be designed to provide smart triggers to detect changes in the environment. Particularly for WLAN, these smart triggers may use physical and link layer information provided by a radio resource management layer based on IEEE 802.11k, that needs still to be deployed in current networks.

Finally, we have observed that CN has been increasingly deployed (at least in France) by the most important network operators. In Chapter 2, we showed that there is a very high density of APs in urban areas, giving in median more than 3 CN APs in any given place. Moreover, if all the deployed APs are considered, we observed in median 15 APs at one single point of measure. This yields a high level of interference since users can manually configure their APs in a particular channel and may also add external antennas to extend the coverage area of a single AP. Moreover, this high AP density clearly shows the potential for energy saving by turning off most of these access points when there are only few mobile users to serve. We may consider drastically reducing the number of APs given the current deployment density. The Internet boxes provided by ISP to ADSL subscribers which embed IEEE 802.11 APs and TV decoders, globally consumed around 1.6 TWh per year in 2008, and since then, the number of deployed boxes has increased (and will increase in the future, as the penetration rate is 70 % of the population today), as well as their energy consumption. Today, one box consumes around 35 W, between 10 W and 18 W for the Internet access and around 20 W for the TV decoder. With advanced standby and efficient wake-up techniques, we could reduce by half the total boxes energy consumption. In a sense, this could be viewed as a generalization of the concept of community networks to private network. A given ISP could allocate a unique SSID to all its boxes. This would lead users to always connect to the unique SSID (whether they are at home or not) and the network operator could simply choose which AP needs to be turned on, and which ones could be turn off, depending on the deployment conditions and the users' demand for bandwidth. The network operator can then mitigate the interferences and reduce the energy consumption, enhancing the overall user experience.

BIBLIOGRAPHY

[1] E. Perahia and R. Stacey, *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*. Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n, Cambridge University Press, 2008.

[2] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ wi-fi networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, MobiCom '06, (New York, NY, USA), pp. 50–61, ACM, 2006.

[3] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3g using wifi," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, (New York, NY, USA), pp. 209–222, ACM, 2010.

[4] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved access point selection," in *Proceedings of the 4th international conference on Mobile systems, applications and services*, MobiSys '06, (New York, NY, USA), pp. 233–245, ACM, 2006.

[5] E. Gustafsson and A. Jonsson, "Always best connected," *Wireless Communications, IEEE*, vol. 10, pp. 49 – 55, feb. 2003.

[6] S. Barré, A. Dhraief, N. Montavont, and O. Bonaventure, "MipShim6: une approche combinée pour la mobilité et la multi-domiciliation," in *Actes du 14ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 09), Strasbourg*, pp. 113–124, October 2009.

[7] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture," RFC 4423, IETF, may 2006.

[8] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1 –2793, 29 2012.

[9] H. Chaouchi and G. Pujolle, *Réseaux sans fil émergents: standards IEEE*. IC2: Série Réseaux et télécoms, Hermes science, 2008.

[10] M. Manshaei, J. Freudiger, M. Felegyhazi, P. Marbach, and J.-P. Hubaux, "On wireless social community networks," in *INFOCOM*

2008. The 27th Conference on Computer Communications. IEEE, pp. 1552 –1560, april 2008.

[11] X. Ai, V. Srinivasan, and C.-K. Tham, "Wi-sh: A simple, robust credit based wi-fi community network," in *INFOCOM 2009, IEEE*, pp. 1638 –1646, april 2009.

[12] International Telecommunication Union (ITU), "ICT Data and Statistics (IDS)." http://www.itu.int/ITU-D/ict/statistics/, July 2012.

[13] "GSM/3G Stats." http://www.gsacom.com/news/statistics.php4, July 2012.

[14] Bluetooth SIG, "Specification of the Bluetooth System." Specification Volume 0 - Covered Core Package Version 4.0, June 2010.

[15] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1 –314, 5 2011.

[16] R. N. Mayo and P. Ranganathan, "Energy consumption in mobile devices: why future systems need requirements–aware energy scale-down," in *Proceedings of the Third international conference on Power - Aware Computer Systems*, PACS'03, (Berlin, Heidelberg), pp. 26–40, Springer-Verlag, 2004.

[17] H. Holma and A. Toskala, *Wcdma for Umts: Hspa Evolution and Lte*. Wiley, 2007.

[18] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "Characterizing radio resource allocation for 3G networks," in *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, (New York, NY, USA), pp. 137–150, ACM, 2010.

[19] G. Castignani, N. Montavont, and A. Lampropulos, "Energy considerations for a wireless multi-homed environment," in *Energy-Aware Communications* (R. Lehnert, ed.), vol. 6955 of *Lecture Notes in Computer Science*, pp. 181–192, Springer Berlin Heidelberg, 2011.

[20] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, CODES/ISSS '10, (New York, NY, USA), pp. 105–114, ACM, 2010.

[21] Y. Xiao, P. Savolainen, A. Karppanen, M. Siekkinen, and A. Ylä-Jääski, "Practical power modeling of data transmission over 802.11g for wireless applications," in *Proceedings of the 1st International Conference on*

Energy-Efficient Computing and Networking, e-Energy '10, (New York, NY, USA), pp. 75–84, ACM, 2010.

[22] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, "802.11 power-saving mode for mobile computing in wi-fi hotspots: limitations, enhancements and open issues," *Wirel. Netw.*, vol. 14, pp. 745–768, Dec. 2008.

[23] H. Petander, "Energy-aware network selection using traffic estimation," in *Proceedings of the 1st ACM workshop on Mobile internet through cellular networks*, MICNET '09, (New York, NY, USA), pp. 55–60, ACM, 2009.

[24] Y. Xiao, R. Kalyanaraman, and A. Yla-Jaaski, "Energy consumption of mobile youtube: Quantitative measurement and analysis," in *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST '08. The Second International Conference on*, pp. 61 –69, sept. 2008.

[25] H. Haverinen, J. Siren, and P. Eronen, "Energy consumption of always-on applications in wcdma networks," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pp. 964 –968, april 2007.

[26] G. Lampropoulos, A. Kaloxylos, N. Passas, and L. Merakos, "A power consumption analysis of tight-coupled wlan/umts networks," in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pp. 1 –5, sept. 2007.

[27] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan, "Interactive wifi connectivity for moving vehicles," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 427–438, Aug. 2008.

[28] R. Mahajan, J. Zahorjan, and B. Zill, "Understanding wifi-based connectivity from moving vehicles," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, (New York, NY, USA), pp. 321–326, ACM, 2007.

[29] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1 –11, april 2006.

[30] K. Lee, I. Rhee, J. Lee, S. Chong, and Y. Yi, "Mobile data offloading: how much can WiFi deliver?," in *Co-NEXT '10*.

[31] M. Afanasyev, T. Chen, G. Voelker, and A. Snoeren, "Usage patterns in an urban WiFi network," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 1359 –1372, Oct. 2010.

[32] Tropos Networks INC., "Google WiFi Continues to be one of the most successful Wireless Broadband Networks in the world as it celebrates its third anniversary." Press Release, August 2009.

[33] A. Arjona and S. Takala, "The Google Muni Wifi Network. Can it Compete with Cellular Voice?," in *Telecommunications, 2007. AICT 2007. The Third Advanced International Conference on*, 2007.

[34] G. Castignani, A. Blanc, A. Lampropulos, and N. Montavont, "Urban 802.11 community networks for mobile users: Current deployments and prospectives," *Mobile Networks and Applications*, pp. 1–12. 10.1007/s11036-012-0402-2.

[35] G. Castignani, L. Loiseau, and N. Montavont, "An evaluation of IEEE 802.11 community networks deployments," in *Information Networking (ICOIN), 2011 International Conference on*, pp. 498 –503, jan. 2011.

[36] G. Castignani, A. Lampropulos, A. Blanc, and N. Montavont, "Wi2Me: A Mobile Sensing Platform for Wireless Heterogeneous Networks," in *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pp. 108 –113, june 2012.

[37] A. Subramanian, H. Gupta, S. Das, and J. Cao, "Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks," *Mobile Computing, IEEE Trans. on*, dec. 2008.

[38] E. Giordano, R. Frank, G. Pau, and M. Gerla, "Corner: a realistic urban propagation model for vanet," in *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*, pp. 57 –60, feb. 2010.

[39] M. Solarski, P. Vidales, O. Schneider, P. Zerfos, and J. Singh, "An experimental evaluation of urban networking using ieee 802.11 technology," in *Operator-Assisted (Wireless Mesh) Community Networks, 2006 1st Workshop on*, pp. 1 –10, sept. 2006.

[40] C. Perkins, "IP Mobility Support for IPv4," RFC 3220, IETF, february 2002.

[41] T. Goff, J. Moronski, D. Phatak, and V. Gupta, "Freeze-tcp: a true end-to-end tcp enhancement mechanism for mobile environments," in *Proceedings of INFOCOM 2000. IEEE*, vol. 3, pp. 1537 –1545 vol.3, mar 2000.

[42] P. Bahl, M. Hajiaghayi, K. Jain, S. Mirrokni, L. Qiu, and A. Saberi, "Cell breathing in wireless lans: Algorithms and evaluation," *Mobile Computing, IEEE Transactions on*, vol. 6, pp. 164 –178, feb. 2007.

[43] H. Haverinen and J. Salowey, " Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)," RFC 4186, IETF, Jan 2006.

[44] G. Castignani, A. Arcia, and N. Montavont, "A study of the discovery process in 802.11 networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 15, pp. 25–36, Mar. 2011.

[45] G. Castignani, N. Montavont, A. Arcia-Moret, M. Oularbi, and S. Houcke, "Cross-layer adaptive scanning algorithms for ieee 802.11 networks," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pp. 327 –332, march 2011.

[46] V. H. Z. Francisco A. González, Jesús A. Pérez, "HAMS: Layer 2 Accurate Measurement Strategy in WLANs 802.11," in *Proceedings of the 1st IEEE International Work- shop on Wireless Network Measurements (WiNMee)*, 2005.

[47] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 93–102, Apr. 2003.

[48] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs," in *Proceedings of the second international workshop on Mobility management and wireless access protocols*, MobiWac '04, (New York, NY, USA), pp. 19–26, ACM, 2004.

[49] V. Mhatre and K. Papagiannaki, "Using smart triggers for improved user performance in 802.11 wireless networks," in *Proceedings of the 4th international conference on Mobile systems, applications and services*, MobiSys '06, (New York, NY, USA), pp. 246–259, ACM, 2006.

[50] H. Velayos and G. Karlsson, "Techniques to reduce the IEEE 802.11b handoff time," in *IEEE International Conference on Communications*, vol. 7, pp. 3844 – 3848 Vol.7, june 2004.

[51] "IEEE Standard for Local and Metropolitan Area Networks- Part 21: Media Independent Handover," *IEEE Std 802.21-2008*, pp. c1 –301, 21 2009.

[52] S. Yoo, D. Cypher, and N. Golmie, "LMS predictive link triggering for seamless handovers in heterogeneous wireless networks," in *IEEE Military Communications Conference, 2007. MILCOM 2007*, pp. 1–7, IEEE, Oct. 2007.

[53] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive scan: Fast handoff with smart triggers for 802.11 wireless LAN," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 749–757, May 2007.

[54] Y. Liao and L. Gao, "Practical Schemes for Smooth MAC Layer Handoff in 802.11 Wireless Networks," in *Proceedings of the 2006 International*

Symposium on on World of Wireless, Mobile and Multimedia Networks, WOWMOM '06, (Washington, DC, USA), pp. 181–190, IEEE Computer Society, 2006.

[55] J. Montavont, N. Montavont, and T. Noel, "Enhanced schemes for L2 handover in IEEE 802.11 networks and their evaluations," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 3, pp. 1429 –1434, sept. 2005.

[56] I. Ramani and S. Savage, "SyncScan: practical fast handoff for 802.11 infrastructure networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, pp. 675 – 684 vol. 1, march 2005.

[57] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, no. 0, pp. –, 2012.

[58] "Channel Deployment Issues for 2.4-GHz 802.11 WLANs," tech. rep., Cisco Systems, Inc, 2004.

[59] M. Burton, "Channel Overlap Calculations for 802.11b Networks," tech. rep., Cirond Technologies, Inc, 2002.

[60] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using wifi," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, MobiCom '08, (New York, NY, USA), pp. 199–210, ACM, 2008.

[61] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee, "Diagnosing Wireless Packet Losses in 802.11: Separating Collision from Weak Signal," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 735 –743, april 2008.

[62] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs," *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)*, pp. 1 –244, 12 2008.

[63] G. Athanasiou, T. Korakis, and L. Tassiulas, "An 802.11k Compliant Framework for Cooperative Handoff in Wireless Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, p. 350643, Sept. 2009.

[64] M. Oularbi, A. Aissa-El-Bey, and S. Houcke, "Physical layer ieee 802.11 channel occupancy rate estimation," in *I/V Communications and Mobile Network (ISVC), 2010 5th International Symposium on*, pp. 1 –4, 30 2010-oct. 2 2010.

[65] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, "PISA — a platform and programming language independent interface for search algorithms," in *Evolutionary Multi-Criterion Optimization (EMO 2003)* (C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, eds.), Lecture Notes in Computer Science, (Berlin), pp. 494 – 508, Springer, 2003.

[66] K. P. Yoon and C.-L. Hwang, *Multiple Attribute Decision Making: An Introduction*. Quantitative Applications in the Social Sciences, SAGE Publications, Inc, 1995.

[67] S. Chakraborty and C.-H. Yeh, "A simulation based comparative study of normalization procedures in multiattribute decision making," in *Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases - Volume 6*, AIKED'07, (Stevens Point, Wisconsin, USA), pp. 102–109, World Scientific and Engineering Academy and Society (WSEAS), 2007.

[68] T. Saaty, *Fundamentals of Decision Making and Prority Theory with the Analytic Hierarchy Process*. Analytic hierarchy process series, Rws Publications, 1994.

[69] "Properties of a Positive Reciprocal Matrix and their Application to AHP," *Journal of the Operations Research*, vol. 41, no. 3, pp. 404 – 414, 1998.

[70] J. L. Deng, "Introduction to grey system theory," *J. Grey Syst.*, vol. 1, pp. 1–24, Nov. 1989.

[71] K. Savitha and D. C. Chandrasekar, "Network Selection Using TOPSIS in Vertical Handover Decision Schemes for Heterogeneous Wireless Networks," *International Journal of Computer Science Issues*, June 2011.

[72] F. Bari and V. Leung, "Multi-Attribute Network Selection by Iterative TOPSIS for Heterogeneous Wireless Access," in *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, pp. 808 –812, jan. 2007.

[73] E. Triantaphyllou, *Multi-Criteria Decision Making Methods: A Comparative Study*. Applied Optimization, Kluwer Academic Publishers, 2000.

[74] P. Tran and N. Boukhatem, "Comparison of MADM decision algorithms for interface selection in heterogeneous wireless networks," in *Software, Telecommunications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference on*, pp. 119 –124, sept. 2008.

[75] E. Triantaphyllou and B. Shu, "On the maximum number of feasible ranking sequences in multi-criteria decision making problems," *European Journal of Operational Research*, vol. 130, no. 3, pp. 665 – 678, 2001.

[76] J. Puttonen and G. Fekete, "Interface Selection for Multihomed Mobile Hosts," in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pp. 1–6, 2006.

[77] C. Leondes, *Fuzzy Logic and Expert Systems Applications*. Neural Network Systems Techniques and Applications, Academic Press, 1997.

[78] J. Puttonen, G. Fekete, T. Vaaramaki, and T. Hamalainen, "Multiple interface management of multihomed mobile hosts in heterogeneous wireless environments," pp. 324–331, IEEE, 2009.

[79] D. Johnson and C. Perkins, "Mobility Support in IPv6," RFC 3775, IETF, June 2004.

[80] Q. Song and A. Jamalipour, "A network selection mechanism for next generation networks," in *2005 IEEE International Conference on Communications, 2005. ICC 2005*, vol. 2, pp. 1418– 1422 Vol. 2, IEEE, May 2005.

[81] P. Zhang, W. Zhou, B. Xie, and J. Song, "A novel network selection mechanism in an integrated WLAN and UMTS environment using AHP and modified GRA," in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, pp. 104–109, IEEE, Sept. 2010.

[82] S. Balasubramaniam and J. Indulska, "Vertical handover supporting pervasive computing in future wireless networks," *Comput. Commun.*, vol. 27, pp. 708–719, May 2004.

[83] E. Stevens-Navarro and V. Wong, "Comparison between vertical handoff decision algorithms for heterogeneous wireless networks," in *Vehicular technology conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 2, pp. 947–951, 2006.

[84] L. Wang and D. Binet, "MADM-based network selection in heterogeneous wireless networks: A simulation study," in *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009*, pp. 559–564, IEEE, May 2009.

[85] R. Rojas, *Neural Networks: A Systematic Introduction*. Springer-Verlag, 1996.

[86] J. Espi, R. Atkinson, D. Harle, and I. Andonovic, "An optimum network selection solution for multihomed hosts using hopfield networks," in *Proceedings of the 2010 Ninth International Conference on Networks*, ICN '10, (Washington, DC, USA), pp. 249–254, IEEE Computer Society, 2010.

[87] Q. Guo, J. Zhu, and X. Xu, "An adaptive multi-criteria vertical handoff decision algorithm for radio heterogeneous network," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 4, pp. 2769 – 2773 Vol. 4, may 2005.

[88] J. J. Hopfield, "Neurocomputing: foundations of research," ch. Neural networks and physical systems with emergent collective computational abilities, pp. 457–464, Cambridge, MA, USA: MIT Press, 1988.

[89] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. Computational Intelligence, Mit Press, 1996.

[90] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[91] A. Schrijver, *Combinatorial Optimization*. Algorithms and combinatorics, Springer, 2003.

[92] A. Schrijver, *On the History of Combinatorial Optimization (till 1960)*, pp. 1–68. 2005.

[93] R. Ben Rayana, *A Smart Management Framework for Multihomed Mobile Nodes and Mobile Routers*. PhD thesis, TELECOM Bretagne, Université Européenne de Bretagne, 2009.

[94] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies," in *European Conference on Artificial Life*, pp. 134–142, 1991.

[95] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley-Interscience series in systems and optimization, John Wiley & Sons, 2009.

[96] K. Deb, *Optimization for Engineering Design: Algorithms and Examples*. Prentice-Hall of India, 2004.

[97] L. G. Suciu, *Profile Management and Automatic Selection mechanisms for Multi-Interface Mobile Terminals*. PhD thesis, TELECOM Bretagne, Université Européenne de Bretagne, 2005.

[98] M. Nam, N. Choi, Y. Seok, and Y. Choi, "Wise: energy-efficient interface selection on vertical handoff between 3g networks and wlans," in *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 1, pp. 692 – 698 Vol.1, sept. 2004.

[99] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, (New York, NY, USA), pp. 280–293, ACM, 2009.

[100] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, "Energy-delay tradeoffs in smartphone applications," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, (New York, NY, USA), pp. 255–270, ACM, 2010.

[101] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Bradford Books, MIT Press, 1992.

[102] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii," in *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, (London, UK, UK), pp. 849–858, Springer-Verlag, 2000.

[103] J.-H. Yeh, J.-C. Chen, and C.-C. Lee, "Comparative analysis of energy-saving techniques in 3gpp and 3gpp2 systems," *Vehicular Technology, IEEE Transactions on*, vol. 58, pp. 432 –448, jan. 2009.

[104] "Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS." ETSI, TR UMTS 30.03 ver. 3.2.0, Apr. 1998.

[105] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems* (K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, eds.), (Athens, Greece), pp. 95–100, International Center for Numerical Methods in Engineering, 2001.

[106] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb, "Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem," in *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, PPSN VII, (London, UK, UK), pp. 44–53, Springer-Verlag, 2002.

[107] L. Bui, H. Abbass, D. Essam, and D. Green, "Performance analysis of evolutionary multi-objective optimization methods in noisy environments," in *8th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, pp. 29–39, 2004.

[108] Wikipedia, "Youtube — Wikipedia, the free encyclopedia," 2012. [Online; accessed 26-July-2012].

[109] European Telecommunications Standards Institute, "Universal Mobile Telecommunications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03

version 3.1.0)," Tech. Rep. TR 101 112 V3.1.0, ETSI, Sophia Antipolis, Valbonne, FRANCE, November 1997.

[110] J. Knowles, L. Thiele, and E. Zitzler, "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers," TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2006.

RÉSUMÉ

INTRODUCTION

Dans l'environnement sans fil actuel, les utilisateurs peuvent trouver des différents types de réseaux d'accès, tels que les réseaux locaux sans fil (WLAN), les réseaux personnels (WPAN) ou les réseaux cellulaires. Ces réseaux ont été conçus pour répondre aux différents besoins en termes de performance et de couverture, orientés vers des différents marchés et des caractéristiques des utilisateurs. Par exemple, la technologie Bluetooth (WPAN) a été conçu pour les communications entre dispositifs mobiles à proximité pour le partage de données, la norme IEEE 802.11 (WLAN) pour fournir un accès à haut débit dans des petites zones de couverture et les technologies cellulaires pour couvrir des zones très larges avec un accès haut débit mobile.

Toutes ces technologies réseaux ont été intensivement déployées dans les dernières années, en particulier dans les zones urbaines. Au même temps, puisque aucune de ces technologies ne s'est imposée sur le marché, des dispositifs mobiles intègrent plusieurs technologies sans fil. Dans ce scénario, les utilisateurs pourraient désormais profiter d'une utilisation mobile, en traversant des différents points d'accès de façon transparente, et de la multi-domiciliation, en ayant la possibilité de sélectionner dynamiquement différents réseaux d'accès sur des différentes interfaces sans fil de manière alternative ou simultanée. Une telle utilisation des réseaux sans fil est décrit par le paradigme « Always Best Connected », proposé par Gustafsson et Jonsson [5].

Cependant, dans le contexte actuel ils existent plusieurs limitations qui empêchent l'exploitation de la diversité des réseaux d'une manière « Always Best Connected ». Par rapport à la gestion de la mobilité, les utilisateurs mobiles doivent découvrir des points d'accès aux réseaux dans leur voisinage et assurer une transition transparente vers un nouveau point d'accès lors d'une dégradation de la connexion avec le point d'accès courant. Concernant la gestion de la multi-domiciliation, les utilisateurs mobiles actuels se connectent typiquement à un seul réseau à tout moment, sans profiter des différents réseaux sans fils disponibles dans un même endroit. Afin de rendre possible l'utilisation de plusieurs interfaces réseaux au même temps, des protocoles de gestion de la multi-domiciliation, tels que shim6 ou Host Identity Protocol (HIP), permettent de donner un identificateur unique aux applications et des mécanismes intermédiaires pour assurer qu'un flux puise changer d'interface sans être interrompu. Cependant, même si ces protocoles peuvent assurer une continuité des flux, l'utilisateur doit décider à tout moment

quelle est l'attribution des flux aux interfaces qui s'adapte le mieux à ses besoins.

Objectifs

Dans cette thèse, nous étudions la diversité des réseaux d'accès sans fil dans le but d'exploiter l'utilisation simultanée de plusieurs interfaces dans le contexte des utilisateurs mobiles. En particulier, nous nous concentrons sur les deux limitations cités précédemment: le support de la mobilité et la prise de décision pour le support de la multi-domiciliation. Les objectifs de cette thèse sont énoncés dans la liste suivante:

- Comprendre, grâce à des études sur le terrain, la diversité des déploiements sans fils actuels et identifier le potentiel et les limitations pour une exploitation intelligente des différents réseaux, particulièrement aux problèmes liés à la gestion de la mobilité et de la multi-domiciliation.
- Analyser les problématiques liées à la mobilité dans les réseaux IEEE 802.11 en étudiant les processus de découverte des points d'accès. Ce processus, étant le processus le plus couteaux en temps, l'objectif principal est de réduire sa durée afin d'affaiblir l'impact sur les applications en cours.
- Définir des mécanismes d'adaptation des paramètres du processus de scanning des points d'accès pour garantir une découverte la plus complète possible des déploiements au même temps qu'un délai réduit pour ce processus.
- Etudier la problématique de la multi-domiciliation dans les terminaux mobiles, en particulier aux limitations liés à l'utilisation simultanée de plusieurs interfaces réseaux.
- Proposer un mécanisme d'attribution des flux aux différentes interfaces sans fils (i.e., répartition de la charge) afin de trouver le meilleur compromis entre la qualité de service et l'énergie consommée par les interfaces.

CONTRIBUTIONS

Etude d'évaluation de la diversité des réseaux

Afin de caractériser la diversité des réseaux sans fil, il y avait la nécessité d'analyser les déploiements réels. Dans un contexte urbain, nous retrouvons des déploiements cellulaires (2G, 3G et 4G), ainsi que des réseaux IEEE

802.11 (WLAN) déployés d'une façon typiquement incontrôlé. Dans ces déploiements WLAN, nous retrouvons depuis quelques années les *réseaux communautaires*, composés par des bornes IEEE 802.11 des abonnées Internet résidentiels qui acceptent de partager une partie de leur bande passante en utilisant un identifiant réseaux commun. Particulièrement en France, ils existent aujourd'hui plus de 13 millions de points d'accès communautaires déployés dans des centaines de villes, appartenant à plusieurs opérateurs, tels que Free, SFR, Bouygues Telecom et Orange.

Pour explorer ces réseaux et analyser un tel déploiement hétérogène, nous avions besoin d'une plate-forme mobile pour la capture de traces et le calcul de statistiques. Ils existent aujourd'hui un certain nombre d'outils pour collecter des traces des réseaux cellulaires et WLAN. Nous retrouvons entre autres OpenBMap, Sensorly, Wigle et OpenSignalMaps, qui fonctionnent sur des différents plateformes (e.g., Linux, Android, iOS). Cependant, aucun de ces outils n'est capable de fournir tous les mécanismes essentiels pour analyser la performance de ces réseaux, en particulier, ils ne permettent pas la connexion automatique aux réseaux communautaires et la génération de trafic pour estimer la qualité de service de ces réseaux. Pour cela, nous avons conçu et développé un nouvel outil de sondage des réseaux cellulaires et WLAN (y compris les réseaux communautaires) pour le système Android, Wi2Me, capable de collecter des traces pour le calcul de statistiques. Deux applications sont proposées: *Wi2Me-User* et *Wi2Me-Research*. *Wi2Me-User* agit comme un gestionnaire des réseaux WLAN communautaires, en fournissant un mécanisme automatique pour la connexion et l'authentification aux réseaux communautaires. Il permet de collecter des traces d'utilisation de l'interface IEEE 802.11, tels que le volume des donnés échangés par les applications ou l'évolution des connexions TCP dans le temps. D'autre part, *Wi2Me-Research* est une application de war-driving qui a été conçu pour le sondage détaillé des réseaux sans fils pour des buts spécifiques de recherche. Elle est capable de collecter des traces avec un échantillonnage plus fin que Wi2Me-User et de contrôler d'une façon précise la génération de flux de donnés pour évaluer la performance des réseaux. Dans un cas typique d'usage de Wi2Me-Research, le dispositif mobile recherche périodiquement des points d'accès IEEE 802.11 et des stations de base 2G/3G ainsi qu'il obtient des informations de géolocalisation. Dans le cas qu'un point d'accès communautaire avec une puissance raisonnable (e.g., $-80\,\mathrm{dBm}$) est trouvé, le mobile essaie de se connecter et s'authentifier à ce réseau. Puis, il envoie et reçoit des fichiers sur des différentes connexions TCP. Dans le cas où aucun point d'accès n'est disponible, le mobile essaie de se connecter au réseau cellulaire et d'envoyer et recevoir ces mêmes fichiers. Toutes les traces collectées lors de l'utilisation de Wi2Me-Recherche sont enregistrées dans une base de données SQLite en local et sont ensuite envoyées vers un serveur distant qui est responsable de ressembler les traces des différents dispositifs et de calculer des statistiques.

Dans cette thèse, nous proposons des différentes expérimentations menées avec Wi2Me-Research pour caractériser la diversité des réseaux dans un milieu urbain. Dans une première étude, nous avons fixé un parcours de mobilité dans la ville de Rennes qui a duré plus de 10 heures. Cela nous a permis de découvrir 6761 points d'accès IEEE 802.11 et 61 stations de base et de se connecter à plus de 660 points d'accès communautaires et 17 stations de base d'un seul opérateur cellulaire (SFR). Nous avons observé que dans le parcours considéré, la disponibilité des réseaux communautaires des deux opérateurs les plus importants (Free et SFR) est équivalente à celle fournie par le réseau cellulaire (i.e., 98.9 % du temps contre 99.2 % du temps respectivement) avec une densité de 15 points d'accès par scanning (dont 3.3 points d'accès communautaires). Cependant, nous avons trouvé que la puissance reçue des réseaux communautaires lors d'une connexion est en médiane $-80\,\mathrm{dBm}$. Nous avons aussi observé des potentiels interférences causées par le déploiement de points d'accès dans des canaux qui chevauchent, ce qui limite potentiellement le débit du lien. La durée médiane des connections est de 27.5 s, ce qui nous a permis dans notre expérimentation de se déplacer en médiane 26 à une vitesse d'environ $1\,\mathrm{m/s}$. Nous avons enregistré aussi dans ces expérimentations les paquets échangés entre le serveur TCP et le mobile, ce qui nous a permis d'étudier l'évolution des connexions TCP dans le temps, particulièrement des paramètres pour le contrôle de la congestion. Nous avons identifié dans ce cas que pour les réseaux communautaires il existe une limitation de bande passante (d'environ $1\,\mathrm{Mbps}$) car la fenêtre de congestion TCP (CWND) est toujours beaucoup plus réduite que la fenêtre du récepteur (RWND).

Nous avons analysé, grâce à la recollection de traces dans le serveur de fichiers, l'évolution des connexions TCP lors d'un handover, i.e., une transition entre deux points d'accès IEEE 802.11. Dans ce cas, nous observons une des limitations les plus importantes pour les réseaux communautaires, car lors d'une déconnexion d'un point d'accès, même si un nouveau point d'accès du même opérateur communautaire est disponible, la continuité des flux d'applications n'est pas garantie. Cela démontre un manque de gestion de la mobilité au niveau réseau (i.e., niveau 3) pour les réseaux communautaires. Contrairement, dans un réseau contrôlé (tel qu'un déploiement de points d'accès au sein d'une entreprise ou d'un campus) la mobilité au niveau 3 est suivant gérée, mais avec des forts impacts du handover dans la performance de la connexion courante. Nous proposons un ensemble d'expérimentations au sein du réseau WLAN de notre campus (SALSA) pour montrer que le nombre de retransmissions et le délai augmentent de façon considérable à partir de 10 s avant le handover, résultant en une réduction de la fenêtre de congestion TCP, qui se produit en médiane une secondes après le handover.

Nous remarquons l'importance pour les opérateurs communautaires d'optimiser les réseaux en faisant un contrôle plus important des aspects sans

fils, qui généralement sont gérés directement par les abonnés. Par exemple, pour éviter des interférences causées par des chevauchements des canaux, les opérateurs pourraient gérer l'allocation des canaux pour chaque point d'accès de façon centralisée, ou même éteindre des points d'accès dans des zones à très forte couverture (où plusieurs points d'accès sont disponibles das un même endroit). Aussi, une solution de contrôle de puissance pour les points d'accès pourrait être envisageable. L'impact du handover dans les communications en cours nous pousse à étudier le processus de découverte dans IEEE 802.11 à fin de réduire son délai. De plus, vu la forte densité de points d'accès que nous avons trouvé, une utilisation simultanée de différents réseaux d'accès semble possible. Nous étudions donc des mécanismes de attribution des flux aux différentes interfaces dans le contexte des mobiles multi-interfaces et multi-domiciliés.

Algorithmes adaptatifs pour la découverte des points d'accès IEEE 802.11

Lors d'un handover entre des points d'accès IEEE 802.11 plusieurs processus son concernés. D'abord, un dispositif mobile doit être capable de détecter les dégradations du lien avec son point d'accès courant et décider quand est-ce qu'une transition vers un nouveau point d'accès est nécessaire. Dans cette transition, le dispositif doit d'abord faire une recherche des points d'accès candidats pour le handover, sélectionner le meilleur d'entre eux et finalement s'authentifier et s'associer. Aussi, si le nouveau point d'accès sélectionné appartient à un réseau différent, une configuration de la couche IP est nécessaire pour garantir aux applications d'accéder au réseau. Nous faisons référence dans ce dernier cas à un handover de niveau 3, qui se déroule jusqu'à après le handover de niveau 2 (qui finit par l'association avec le nouveau point d'accès). Des solutions existent (e.g., MobileIP [40]) pour gérer le handover au niveau 3 et garantir une continuité des applications sans couper les connexions au niveau transport. Cependant, le déroulement de ces phases dans le processus de handover implique que le dispositif ne puisse pas communiquer avec le réseau pendant un certain temps. En particulier pour le handover de niveau 2, il a été démontré dans plusieurs travaux [46] [47] que la durée du processus de découverte des points d'accès (i.e., scanning) implique le 90 % du temps total du handover de niveau 2.

Ce processus de scanning est défini par le standard [8] avec deux modalités: le scanning passif et le scanning actif. Dans le scanning passif, le dispositif mobile découvre son environnement en écoutant les balises diffusées par les points d'accès voisins qui sont envoyés tous les 100 ms. Dans le scanning actif, le dispositif mobile demande des informations aux points d'accès en envoyant des Probe Requests et en attendant des Probe Responses. Cela permet de réduire le délai du processus de scanning, car dans chaque canal le dispositif mobile doit seulement attendre les réponses à sa requête, ce qui est plus rapide que la fréquence des balises dans le scanning passif. Dans

chaque canal, après l'envoi du Probe Request, le mobile attend un temps égal à *MinChannelTime* (MinCT). Si au moins un point d'accès répond avant ce temps, le mobile attend alors plus longtemps, jusqu'à *MaxChannelTime* (MaxCT). Voir donc que MinCT $\leqslant$ MaxCT. Cependant, la définition des valeurs de MinCT et MaxCT n'est pas fournie par le standard. Noter que les valeurs pour ces temps et la séquence de canaux à explorer définissent la performance du processus de scanning, qui peut être mesurée par sa durée (i.e., la latence), le taux de découverte (i.e., le nombre de points d'accès découverts dans le voisinage), le taux de échec du processus (i.e., le nombre de fois qu'aucun point d'accès est découvert à cause des mauvaises paramètres) et le temps pour découvrir le premier point d'accès (dans le cas où le dispositif ait besoin de découvrir rapidement un seul point d'accès). Dans ce cas, chaque implémentation de firmware/pilote IEEE 802.11 exige la définition des valeurs de ces temps et de la séquence de canaux. Nous avons observé que dans les dispositifs actuels, il y a un grade diversité de paramètres, ce qui donne des performances de scanning différentes. En plus, nous montrons dans cette thèse que, selon les conditions du déploiement, le délai de réception des réponses des points d'accès varie énormément, ce qui nous incite à définir des algorithmes de scanning capables d'adapter dynamiquement ses paramètres dans le but principal de minimiser la latence et de maximiser le taux de découverte de la topologie.

Dans cette thèse, nous proposons deux algorithmes pour l'adaptation des paramètres de scanning: Adaptive Discovery Algorithm (ADA) et Cross-layer Adaptive Scanning. Dans ADA, nous utilisons une séquence de canaux aléatoire que donne la priorité aux canaux 1, 6 et 11, car nous avons trouvé que dans des déploiements urbains, autour de 80 % des points d'accès sont déployés dans ces canaux [35]. Par rapport aux valeurs de MinCT et MaxCT, nous avons d'abord étudié le délai des Probe Responses pour proposer un algorithme qui ajuste les valeurs de ces temps selon les points d'accès que le mobile découvre pendant le processus de scanning. L'algorithme établi d'abord des valeurs moyens pour MinCT et MaxCT et puis il augmente ou diminue ces valeurs selon il découvre ou il ne découvre pas des points d'accès. Nous avons implémenté ADA sur le pilote sans fil MadWiFi et nous avons mis en place des scénarios de expérimentation pour évaluer sa performance et la comparer à des algorithmes à paramètres fixes. Les résultats montrent que pour une même latence de scanning, ADA est capable de découvrir plus de points d'accès et de fournir un taux d'échec presque zéro.

Nous avons observé que les conditions dans chaque canal, particulièrement le trafic généré par des stations, génèrent un fort délai des Probe Responses lors d'un scanning. Un deuxième algorithme adaptatif, Cross-layer Adaptive Scanning, a été conçu pour utiliser des informations de la couche physique, telles que le taux de charge et la puissance mesurées dans chaque canal. En utilisant ces informations, le mobile est capable de sélectionner la meilleure séquence de canaux et d'établir les valeurs de MinCT

et MaXCT les plus appropriées pour recevoir les réponses dans chaque scénario. Dans ce cas, nous implémentons l'algorithme dans le driver ath5k pour Linux et nous expérimentons dans cinq scénarios de déploiement différents, pour comparer la performance de l'algorithme adaptatif face à des stratégies de paramètres fixes. Nous observons dans ce cas qu'en utilisant des informations de la couche physique, l'algorithme adaptatif peut identifier des canaux qui ont très probablement des points d'accès déployés et utiliser des temps d'attente plus longues dans ces cas pour augmenter la probabilité de recevoir des réponses.

Mécanisme de prise de décision pour les terminaux multi-domiciliés

Tel que nous avons montré dans l'étude d'évaluation de la diversité des déploiements sans fil, il existe actuellement un potentiel pour une utilisation simultanée des interfaces réseaux. Pour cela, l'utilisateur doit définir des mécanismes de sélection des réseaux visant à exploiter la diversité des interfaces. Typiquement, ils existent deux types de mécanismes pour la sélection des réseaux, ceux orientés à déterminer le meilleur réseau pour basculer tous les flux d'application (i.e., le cas du handover vertical) et ceux orientés trouver la répartition des flux la plus optimal entre les différentes interfaces réseaux (i.e., le cas de la multi-domiciliation). Des différentes solutions ont été proposées dans la littérature qui définissent des mécanismes de décision pour le handover vertical. Ces solutions peuvent être basés sur des algorithmes de prise de décision multicritères (MADM) en utilisant des critères très divers (e.g., QoS, profil des utilisateurs, état des interfaces) ou sur des mécanismes de décisions modelés comme des problèmes d'optimisation multi-objectifs. Cependant, dans ces solutions, les problèmes sont suivant simplifiés en utilisant des informations de haut niveau visant à établir des préférences des différents critères/objectifs lors de la prise de décision. Ces informations de haut niveau sont traduites par des poids, qui indiquent une mesure subjective de la préférence. L'utilisation des poids réduit la complexité du problème à une seule dimension et empêche à l'utilisateur d'évaluer le compromis qui existe entre des différents critères/objectifs contradictoires. Nous observons aussi que dans les solutions existantes, un nombre très important de critères est utilisé pour la prise de décision, ce qui augmente la complexité du mécanisme car dans ce cas, le dispositif doit surveiller plusieurs paramètres au même temps.

En particulier, la consommation énergétique est parfois considérée comme un critère pour la prise de décision. Typiquement un coût énergétique constant est attribué à chaque interface réseaux. Cependant, la consommation d'une interface dépend fortement de la façon dans laquelle elle est utilisée, i.e., des caractéristiques des flux applicatifs transportés par chaque interface.

Nous proposons un mécanisme de sélection des réseaux visant à attribuer les différents flux sur des interfaces disponibles. Nous modélisons ce mé-

canisme comme un problème d'optimisation multi-objectifs qui considère deux aspects: l'énergie consommée par les interfaces et l'écart entre la demande de bande passante des applications et la capacité disponible des interfaces. D'autre part, pour éviter la subjectivité dans la prise de décision, nous n'utilisons aucun poids pour les objectifs mais des algorithmes génétiques capables d'obtenir des solutions optimales en évaluant chaque solution sur plusieurs objectifs au même temps grâce au critère de non-dominance.

Nous évaluons notre approche par simulation (en utilisant le framework pour optimisation multi-objectifs PISA [65]) et nous faisons des comparaisons face à des mécanismes de combinaison de poids type MADM (tels que Simple Additive Weighting ou Multiplicative Exponential Weighting). Nous observons qu'en utilisant des algorithmes génétiques, il est possible d'obtenir plusieurs solutions optimales qui mettent en évidence le compromis existant entre les deux objectifs proposés. Nous observons aussi dans nos résultats de simulation que pour les problèmes qui concernent plusieurs applications à attribuer, notre approche permet d'obtenir des solutions plus rapidement qu'en faisant une recherche complète avec des algorithmes basés sur des poids, avec une bonne approximation aux solutions optimales.

CONCLUSION

L'environnement sans fil actuel se caractérise par une diversité de technologies. Ces technologies ont été déployées dans les dernières années visant à fournir un accès Internet sans fil à haut débit pour les utilisateurs mobiles. En particulier dans ces dernières années, les réseaux communautaires basés sur IEEE 802.11 ont été déployés par des abonnes résidentielles ADSL, en créant des réseaux omniprésentes en milieu urbain. Cela ouvre la possibilité aux utilisateurs multi-domiciliés d'exploiter la diversité des réseaux d'une façon « Always Best Connected ».

Toutefois, dans un tel environnement hétérogène sans fil, il y a encore quelques limitations qui empêchent les utilisateurs mobiles d'exploiter pleinement la diversité des réseaux. Tout d'abord, comme nous l'avons montré dans l'étude d'évaluation de la diversité des réseaux réalisé avec la plateforme Wi2Me, même s'il y a une forte densité des réseaux sans fil (IEEE 802.11 et les réseaux cellulaires) dans les zones urbaines, la faible intégration des déploiements des différentes technologies empêche les opérateurs de fournir une mobilité transparente à travers les différentes technologies réseau, et plus particulièrement entre les différents points d'accès IEEE 802.11 des différents opérateurs. D'autre part, nous avons également montré dans notre étude d'évaluation qu'il existe une grande complémentarité des réseaux IEEE 802.11 et les réseaux cellulaires dans le milieu urbain, permettant à l'utilisateur mobile de communiquer avec plusieurs interfaces sans fil à tout moment. Cependant, il existe actuellement un manque de mécanisme de sélection des réseaux pour permettre l'utilisation simultanée de plusieurs in-

terfaces (généralement WLAN et les réseaux cellulaires), afin de maximiser l'exploitation de la diversité sans fil.

Dans cette thèse, nous nous sommes concentrés sur les deux limitations mentionnées ci-dessus. Après avoir fourni une étude d'évaluation des réseaux sans fil actuels dans les zons urbaines nous avons d'abord analysé les limitations de la mobilité et, en particulier, le processus de découverte dans la norme IEEE 802.11. Nous avons montré que les optimisations actuelles de l'algorithme de scanning actif dans IEEE 802.11 ne considèrent pas le compromis entre la latence du processus et les points d'accès découverts dans la topologie. Nous gérons ce compromis en proposant un ensemble d'algorithmes de scanning adaptatifs, qui visent à définir les paramètres les plus appropriés pour chaque canal. Nous avons implémenté deux algorithmes adaptatifs sur des pilotes open-source IEEE 802.11 et nous avons proposé une évaluation expérimentale dans deux bancs d'essai différents, afin de comparer les algorithmes adaptatifs contre des algorithmes à paramètres fixes. Les résultats montrent que l'utilisation d'une stratégie d'adaptation permet de mieux gérer le compromis entre la latence et la découverte de topologie, c'est à dire, pour une latence donnée, un algorithme adaptatif est capable de découvrir un plus grand nombre de points d'accès.

La deuxième contribution de cette thèse a porté sur le processus de prise de décision pour la sélection des réseaux dans un contexte multi-domicilié. Dans ce cas, nous considérons qu'un utilisateur mobile doit décider comment attribuer les flux applicatifs différents pour les interfaces disponibles. Cela diffère des mécanismes existants de sélection des réseaux, qui se concentrent principalement sur le soutien à la prise de décision pour handover vertical, afin d'optimiser un ensemble de critères (par exemple, la qualité de service, le coût, la sécurité, l'énergie). Nous considérons particulièrement la consommation d'énergie des interfaces sans fil comme un critère pour la prise de décision. A différence des mécanismes de sélection des réseaux existants, dans lesquels l'énergie est considérée comme un coût fixe associé à chaque interface, nous prenons en compte la consommation d'énergie d'une manière fine. Dans ce cas, nous utilisons des modèles de trafic pour estimer la quantité d'énergie consommée par l'assignation d'un flux sur une interface. Nous modélisons le problème d'assignation des flux comme un problème d'optimisation multi-objectif et nous cherchons des solutions en utilisant des algorithmes génétiques. En tant qu'objectifs, nous considérons la minimisation de la consommation d'énergie des interfaces et de l'insatisfaction de bande passante qui est généré lorsque la bande passante demandée par les applications dépasse la bande passante disponible des interfaces. Nous évaluons notre mécanisme par simulation, en prenant en compte des différents scénarios qui correspondent à des différents types d'applications et de caractéristiques des interfaces. Les résultats des simulations montrent que la résolution du problème avec des algorithmes géné-

tiques fournissent une vue complète du compromis entre les différents objectifs.

Il y a encore un certain nombre de questions ouvertes dans le contexte de la mobilité et de la gestion de la multi-domiciliation dans les réseaux sans fil hétérogènes. Même si nous pouvons assurer une latence de handover faible et une assignation optimale des applications, nous avons observé que dans les dispositifs actuels, il y a encore des limitations qui dégradent l'expérience des utilisateurs. Tout d'abord, les dispositifs mobiles utilisent généralement une seule interface réseaux à la fois. Toutefois, cette limitation peut être simplement remédié en modifiant les politiques implémentés dans le système d'exploitation et en mettant en œuvre un protocole de multi-domiciliation (shim6, HIP). Deuxièmement, nous avons constaté que les dispositifs actuels ne sont pas réactifs aux changements dans l'environnement sans fil, c'est à dire qu'il n'y a pas des algorithmes de déclenchement intelligents afin de déterminer si la situation actuelle se dégrade et que le dispositif a donc besoin de s'adapter à cette nouvelle situation. Dans ce cas, le mobile doit découvrir des nouveaux réseaux et calculer l'attribution des flux las plus optimal. Nous travaillons actuellement dans un algorithme d'anticipation sur Android qui surveille l'évolution à court terme du signal avec le point d'accès pour déclencher le processus adaptatif de découverte (comme ceux que nous avons proposé dans cette thèse). En ce qui concerne la sélection des réseaux, nous avons évalué les processus de décision déclenchés par l'utilisateur (par exemple, lorsque des nouvelles applications sont exécutées). Dans ce cas, nous n'avons pas considéré ce processus étant déclenchée par une modification de l'environnement sans fil. Dans le futur, cette situation pourrait être considérée afin de concevoir des déclencheurs plus efficaces pour le processus de prise de décision.

PUBLICATIONS

PEER-REVIEWED JOURNALS

1. German Castignani, Alberto Blanc, Alejandro Lampropulos and Nicolas Montavont, *Urban 802.11 community networks for mobile users: Current deployments and prospectives*, Mobile Networks and Applications (MONET), Special Issue on Mobility and Social Networks Confluence, ISSN 1383-469X, pp. 1-12, ACM/Springer, 2012.

2. German Castignani, Andrés Arcia, and Nicolas Montavont, *A study of the discovery process in 802.11 networks*. SIGMOBILE Mob. Comput. Commun. Rev. 15, 1, pp. 25-36, March 2011.

PEER-REVIEWED CONFERENCES AND WORKSHOPS

3. Yves Josse, Bruno Fracasso, German Castignani and Nicolas Montavont, *Energy Efficient Distributed Antenna Systems Deployments using Radio-over-Fiber*, Online Conference on Green Communications (GreenCom), IEEE, 2012

4. German Castignani, Pablo Negri and Nicolas Montavont, *Relevamiento de despliegues de redes IEEE 802.11: Hacia redes inalambricas comunitarias*, 41° Jornadas Argentinas de Informática (JAIIO), Argentina, 2012

5. Laudin Molina, Andrés Arcia-Moret, German Castignani and Nicolas Montavont, *Caracterización de Topologías Espontáneas IEEE 802.11: Hacia un Sistema de Redes Comunitarias*, 5° Congreso Ibero-americano de Estudiantes de Ingeniería Eléctrica (CIBELEC), Venezuela, 2012

6. German Castignani, Alejandro Lampropulos, Alberto Blanc and Nicolas Montavont, *Wi2Me: A Mobile Sensing Platform for Wireless Heterogeneous Networks*, 32th International Conference on Distributed Computing Systems Workshops (ICDCSW), 2012

7. German Castignani, Nicolas Montavont and Alejandro Lampropulos, *Energy considerations for a wireless multi-homed environment*, in Energy-Aware Communications (R. Lehnert, ed.), vol. 6955 of Lecture Notes in Computer Science, 17th Eunice Open European Summer School, pp. 181-192, Springer Berlin, 2011

8. German Castignani, Nicolas Montavont, Andrés Arcia-Moret, Mohamed Oularbi and Sebastien Houcke, *Cross-layer adaptive scanning algorithms*

for IEEE 802.11 networks, Wireless Communications and Networking Conference (WCNC), IEEE, pp. 327-332, March 2011

9. German Castignani, Lucien Loiseau and Nicolas Montavont, *An evaluation of IEEE 802.11 community networks deployments*, Information Networking (ICOIN), 2011 International Conference on, pp.498-503, 2011

10. German Castignani, Andrés Arcia-Moret and Nicolas Montavont, *An evaluation of the resource discovery process in IEEE 802.11 networks*, In Proceedings of the Second International Workshop on Mobile Opportunistic Networking (MobiOpp '10), ACM, 2010

11. German Castignani, Andrés Arcia-Moret and Nicolas Montavont, *Analysis and Evaluation of WiFi Scanning Strategies*, 4° Congreso Iberoamericano de Estudiantes de Ingeniería Eléctrica (CIBELEC), Venezuela, 2010

12. German Castignani and Nicolas Montavont, *Adaptive System for 802.11 Scanning*, INFOCOM Workshops 2009, IEEE, pp. 1-2, April 2009

13. German Castignani and Nicolas Montavont, *Adaptive discovery mechanism for wireless environments*, 14th Eunice Open European Summer School, 2008

INTERNAL RESEARCH REPORTS

14. German Castignani, Nicolas Montavont, Andrés Arcia-Moret, Mohamed Oularbi and Sebastien Houcke, *IEEE 802.11 scanning algorithms: cross-layer experiments*, Collection des rapports de re-cherche de TELECOM Bretagne, RR-2011-05-RSM, ISSN 1255-2275, 2011

ACRONYMS

2G Second Generation Wireless Network

3GPP 3rd Generation Partnership Project

3G Third Generation Wireless Network

4G Fourth Generation Wireless Network

AAA Authentication, Authorization and Accounting

ABC Always Best Connected

ACO Ant Colony Optimization

ADA Adaptive Discovery Algorithm

ADSL Asymmetric Digital Subscriber Line

AHP Analytic Hierarchy Process

AMVHO Adaptive Multi-Criteria Vertical Handover

ANN Artificial Neural Networks

AP Access Point

BBV Bi-objective Binary Value

BER Bit Error Rate

BSSID Basic Service Set Identifier

BSS Basic Service Set

CCK Complementary Code Keying

CDF Cumulative Distribution Function

CDMA Code Division Multiple Access

CGI Common Gateway Interface

CID Cell Identifier

CN Community Networks

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CWND TCP Congestion Window

DCF Distributed Coordination Function

DCH Dedicated Channel

DHCP Dynamic Host Configuration Protocol

DIFS DCF Interframe Space

DSSS Direct Sequence Spread Spectrum

DS Distribution System

E-DCH Enhanced Dedicated Channel

EAP Extensible Authentication Protocol

EAP-SIM Extensible Authentication Protocol Method for GSM Subscriber Identity Module

EDGE Enhanced Data Rates for GSM Evolution

ESS Extended Service Set

FACH Forward Access Channel

FDD Frequency-Division Duplexing

FEMO Fair Evolutionary Multiobjective Optimizer

FHSS Frequency Hopping Spread Spectrum

FIS Fuzzy Inference System

FRD First Probe Response Delay

GPRS General Packet Radio Service

GPS Global Positioning System

GRA Grey Relational Analysis

GRC Grey Relational Coefficient

GSM Global System for Mobile Communications

HIP Host Identity Protocol

HSDPA High Speed Downlink Packet Access

HSPA+ Evolved High Speed Packet Access

HSPA High Speed Packet Access

HSUPA High Speed Uplink Packet Access

HTTPS Hypertext Transfer Protocol Secure

IBSS Independent Basic Service Set

IFS Interframe-Spacing

ISM Industrial Scientific and Medical

ISP Internet Service Provider

IoT Internet of Things

LAC Location Area Code

LMS Least Mean Square

LMPA Local Maximum Precision Algorithm

LOTZ Leading Ones Trailing Zeros

LTE Long Term Evolution

MAC Medium Access Control

MADM Multi-Attribute Decision Making

MEW Multiplicative Exponential Weighting

MIMO Multiple Input Multiple Output

MOO Multi-Objective Optimization

MS Mobile Station

MVC Model View Controller

MoS Mean Opinion Score

NAT Network Address Translation

NSGA2 Non-dominated Sorting Genetic Algorithm 2

NSGA Non-dominated Sorting Genetic Algorithm

OFDMA Orthogonal Frequency-Division Multiple Access

OFDM Orthogonal Frequency Division Multiplexing

OS Operative Systems

PCH Paging Channel

PERL Practical Extraction and Report Language

PSM Power Saving Mode

QoS Quality of Service

RADIUS Remote Authentication Dial-In User Service

RNC Radio Network Controller

RRC Radio Resource Control

RSSI Received Signal Strength Indication

RTT Round Trip Time

RWND TCP Receiver Window

SAW Simple Additive Weighting

SIM Subscriber Identity Modules

SPA Simple Precision Algorithm

SPEA2 Strength Pareto Evolutionary Algorithm 2

SSID Service Set Identifier

TDD Time-Division Duplexing

TDMA Time Division Multiple Access

TIM Traffic Indication Map

TOPSIS Technique for Order Preference by Similarity to Ideal Solution

TU Time Unit

UMTS Universal Mobile Telecommunications System

USRP2 Universal Software Radio Peripheral

VoIP Voice over Internet Protocol

VPN Virtual Private Network

WCDMA Wideband Code Division Multiple Access

WEP Wired Equivalent Privacy

WLAN Wireless Local Area Networks

WPAN Wireless Personal Area Networks

WPA Wi-Fi Protected Access

WSN Wireless Sensor Networks

WiMAX Worldwide Interoperability for Microwave Access