

Metrics and non-intrusive techniques to characterize Wi-Fi networks

Laudin Alessandro Molina Troconis

▶ **To cite this version:**

Laudin Alessandro Molina Troconis. Metrics and non-intrusive techniques to characterize Wi-Fi networks. Networking and Internet Architecture [cs.NI]. Ecole nationale supérieure Mines-Télécom Atlantique, 2018. English. NNT : 2018IMTA0091 . tel-02152367

HAL Id: tel-02152367

https://theses.hal.science/tel-02152367

Submitted on 11 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THESE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE MINES-TELECOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE
COMMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité *Informatique*

Par

Laudin Alessandro MOLINA TROCONIS

Techniques et métriques non intrusives pour caractériser les réseaux Wi-Fi

Thèse présentée et soutenue à Rennes, le 5 juillet de 2018
Unité de recherche : Systèmes, réseaux, cybersécurité et droit du numérique (SRCD) / IRISA
Thèse N° : 2018IMTA0091

Rapporteurs avant soutenance :

Mme. Hakima CHAOUCHI	Professeur, Telecom Sud Paris
M. Herve RIVANO	Professeur, INSA Lyon

Composition du Jury :

Président : M. Herve RIVANO	Professeur, INSA Lyon
Examinateurs : Mme. Françoise SAILHAN	Maitre de conférences, Conservatoire National des Arts et Métiers
Dir. de thèse :M. Nicolas MONTAVONT	Professeur, IMT Atlantique Bretagne-Pays de la
Encadrant. de thèse : M. Alberto BLANC	Maitre de conférences, IMT Atlantique Bretagne-Pays de la Loire

Invités

Mme. Ljiljana SIMIĆ	Scientifique Principal, Institute for Networked Systems, RWTH Aachen University
M. Vincenzo MANCUSO	Chercheur Assistant Professor, Institute IMDEA Networks
M. Yoann CORRE	Responsable de l'équipe Wireless Research and Technology, SIRADEL

Molina Troconis Laudin Alessandro: *Metrics and non-intrusive techniques to characterize WI-Fi networks* © July 2018

To my parents, Laudin and Lady, for absolutely everything.
To Neis, for her unwavering support and endless kindness.

ACKNOWLEDGMENTS

I am truly indebted and thankful to my advisor, Dr. Alberto Blanc, who has trusted me since the beginning and has guided me throughout this research.

I am greatly thankful to my thesis director, Professor Nicolas Montavont, who trusted and guided me even before starting this dissertation. I appreciate every piece of advice he gave me.

I would like to thank the members of the jury for having accepted to examine this dissertation.

Thank you very much Dr. Ljiljana Simić, Dr. Vincenzo Mancuso and Dr. Yoann Corre for the guidance, critics and comments.

I would like to address a special thank to Dr. Andrés Arcia. Andrés has being supporting and teaching me for more than 10 years.

I want to thank specially Tanguy, Dareen, Mohamed and Federico for their collaboration, comments, and support.

Gracias a mi esposa Neismelly, por su paciencia y apoyo. Todo esto ha sido posible gracias a ella. Gracias mi hijo Lucas, la gran motivación durante la recta final.

Gracias a toda mi familia. El apoyo de mis padres, hermanos, tíos y primos es invaluable. Mi familia me inspira y me guía.

Thanks to all my friends and colleagues. You made my dissertation a nice journey.

ABSTRACT

Nowadays, smartphones and other mobile devices are present worldwide, with over 4.40 Billion devices globally. These mobile devices enable users to easily access the Internet via wireless networks. IEEE 802.11 networks use unlicensed bands and are often used to connect users to the Internet. Different actors are installing IEE 802.11 networks everywhere (e.g., home users at their houses, enterprises at their offices, universities at their campuses), without central planning or coordination, creating chaotic deployments. As a result, IEEE 802.11 networks are widely deployed all over the world, with high access point density in urban areas. In this context, end-users and network operators are trying to exploit these dense network deployments to get ubiquitous Internet connectivity, and possibly other services. However, taking advantage of these dense deployments requires strategies to gather and provide information about the available IEEE 802.11 networks.

In this dissertation, we first study the network discovery process within the context of these dense network deployments. Then, we present the Wireless Measurements Sharing Platform, a collaborative information system, where mobile stations collect simple network measurements (e.g., the presence of an access point) and send these measurements to a central system. By gathering and processing several network measurements from different mobile users, the platform provides access to valuable characteristics of the network deployment. We evaluate the usefulness of this collaborative platform thanks to two applications: first, the minimal access point set, to reduce the energy needed to offer IEEE 802.11 coverage in a given area. Second, the optimization of the scanning parameters, to reduce the time a mobile station needs for the network discovery. These two applications show that the proposed collaborative information system can solve different problems. Then, we describe a method to identify whether an access point operates in a saturated channel, by passively monitoring and analyzing the beacon arrival distribution. In an empirical evaluation, the method correctly identifies all the saturated scenarios, out of which 34 % are false positives. The classification method needs to collect Beacons during a period of about 11 s.

RÉSUMÉ

Aujourd'hui, les appareils mobiles sont présents dans le monde entier. Ces appareils permettent aux utilisateurs d'accéder à l'Internet notamment par l'intermédiaire des réseaux WiFi. La diversité et le nombre de déploiements sans coordination centrale (y compris les utilisateurs à leur domicile) conduit à des déploiements qu'on peut qualifier de chaotiques. En conséquence, les réseaux WiFi sont largement déployés, avec une forte densité dans les zones urbaines. Dans ce contexte, les utilisateurs et les opérateurs tentent d'exploiter ces déploiements pour obtenir une connectivité omniprésente, et éventuellement d'autres services. Cependant, pour tirer parti de ces déploiements, il faut des stratégies pour identifier les réseaux utilisables et choisir les plus adaptés aux besoins. Pour cela, nous étudions le processus de découverte des réseaux dans le contexte de ces déploiements. Ensuite, nous présentons une plateforme de partage de mesures sans fil, un système d'information collaboratif où les stations mobiles recueillent des mesures du réseau et les envoient à un système central. En rassemblant mesures provenant de différents utilisateurs, la plateforme donne accès à des caractéristiques du déploiement précieuses. Nous évaluons l'utilité de cette plateforme collaborative grâce à deux applications : (1) le ensemble minimal de points d'accès, afin de réduire l'énergie nécessaire pour offrir une couverture WiFi dans une zone donnée. (2) l'optimisation des paramètres de recherche de réseau, afin de réduire le temps nécessaire pour découvrir les réseaux existants. Ensuite, nous étudions une méthode passive pour déterminer si un réseaux fonctionne dans un canal saturé.

PUBLICATIONS

PEER-REVIEWED JOURNALS

[1] Laudin Molina, Tanguy Kerdoncuff, Dareen Shehadeh, Nicolas Montavont, and Alberto Blanc. "WMSP: Bringing the Wisdom of the Crowd to WiFi Networks." In: *IEEE Transactions on Mobile Computing* 16.12 (2017), pp. 3580–3591. ISSN: 1536-1233. DOI: 10.1109/TMC.2017.2694422.

PEER-REVIEWED CONFERENCES AND WORKSHOPS

[1] Andrés Arcia-Moret, Laudin Molina, Nicolas Montavont, German Castignani, and Alberto Blanc. "Acess Point Discovery in 802.11 Networks." In: *Proceedings of the 2014 IFIP Wireless Days*. WD '2014. Rio de Janeiro, Brazil: IEEE, 2014. ISBN: 978-1-4799-6606-6. DOI: 10.1109/WD.2014.7020817.

[2] Andrés Arcia-Moret, Antonio Araujo, José Aguilar, Arjuna Sathiaseelan, and Laudin Molina. *Assisted Network Discovery for Next Generation Wireless Networks*. Las Vegas, NV, USA, 2016. DOI: 10.1109/CCNC.2016.7444884.

[3] Andrés Arcia-Moret, Antonio Araujo, José Aguilar, Laudin Molina, and Arjuna Sathiaseelan. "Intelligent Network Discovery for Next Generation Community Wireless Networks." In: *12th Annual Conference on Wireless On-demand Network Systems and Services*. WONS '2016. Conference Location: Cortina d'Ampezzo, Italy: IEEE, 2016, pp. 1–7. ISBN: 978-3-9018-8279-1.

[4] Laudin Molina, Alberto Blanc, Nicolas Montavont, and Ljiljana Simić. "Identifying Channel Saturation in Wi-Fi Networks via Passive Monitoring of IEEE 802.11 Beacon Jitter." In: *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*. MobiWac '17. Miami, Florida, USA: ACM, 2017, pp. 63–70. ISBN: 978-1-4503-5163-8. DOI: 10.1145/3132062.3132069.

CONTENTS

1 INTRODUCTION 1
1.1 Thesis Overview and Contributions 3

2 ACCESS POINT DISCOVERY IN IEEE 802.11 NETWORKS 5
2.1 Scanning Process in IEEE 802.11 Networks 6
2.1.1 The Wi-Fi Scanning Implementation in the Linux Kernel 8
2.2 Literature Review 10
2.3 Evaluation of the Discovery Process 12
2.3.1 Methodology and Tools 12
2.3.2 Evolution of the Discovery Process 13
2.3.3 Dependency on the Platform 14
2.4 Analysis of Scanning Characteristics 18
2.4.1 Timer impact 18
2.4.2 Probe Responses versus Beacons for Topology Discovery 18
2.4.3 Overlapping Channels 19
2.4.4 On the Quality of Access Points 21
2.5 Concluding remarks 22

3 COLLABORATIVE INFORMATION SYSTEM 25
3.1 Related Work 26
3.1.1 Information Services 26
3.1.2 Centralized Systems 27
3.1.3 Data Collection via Crowdsourcing 28
3.2 Crowdsourcing Platform 30
3.2.1 Information Collected 31
3.2.2 Collecting Measurements 32
3.2.3 User Incentives 32
3.2.4 Wireless Measurements Sharing Platform Architecture 33
3.2.5 Data Preprocessing 34
3.3 Empirical Evaluation 36
3.3.1 Dataset Collection 36
3.3.2 Dataset Description 37
3.3.3 Wireless Measurements Sharing Platform Implementation 38
3.4 Reducing Energy Consumption 39
3.4.1 Minimal Access Point Set 39
3.4.2 Performance Evaluation of the Minimal Access Point Set Implementation 41
3.4.3 Empirical Evaluation 41

3.4.4 Subset Throughput Evaluation Through Simulation 44
3.4.5 Minimal Access Point Set Lifetime 45
3.5 Reduced Scanning Timers 46
3.5.1 Optimal Scanning Parameters 48
3.5.2 Empirical Evaluation 48
3.5.3 Performance Evaluation of the Optimal Scanning Parameters 50
3.5.4 Assisted Scanning 50
3.6 Concluding Remarks 52

4 EVALUATING THE QUALITY OF WI-FI NETWORKS 53
4.1 IEEE 802.11 Overview 54
4.1.1 Medium Access Control 54
4.1.2 Beacon Transmission 54
4.1.3 Channel Saturation 55
4.2 Metrics for Characterizing Wi-Fi Networks 55
4.3 Beacon Jitter 56
4.3.1 Experiments 58
4.3.2 Experimental Results 60
4.4 Beacon Jitter and Channel Load 63
4.4.1 The Kolgomorov-Smirnov Test to Compare Beacon Jitter Distributions 66
4.5 Identifying Wi-Fi Channel Load 67
4.5.1 Empirical Validation 67
4.5.2 Packet Sampling 70
4.5.3 Observation Window 71
4.6 Concluding Remarks 72

5 CONCLUSIONS AND PERSPECTIVES 73
5.1 Concluding Remarks 73
5.2 Future Work 75

BIBLIOGRAPHY 91

LIST OF FIGURES

Figure 1 Sequence diagram of the active scanning procedure as specified in the IEEE 802.11 standard. 7

Figure 2 Role of minimal channel time (MinCT) and maximum channel time (MaxCT). Static values for MinCT and MaxCT may cause waste of time (channel 2) and missing access point (AP) (channel 3). 8

Figure 3 Sequence diagram of the active scanning implemented in the Linux kernel. 9

Figure 4 Procedure followed to evaluate of the Wi-Fi discovery process in urban areas. 13

Figure 5 Example of the sequential scanning impact on the discovery rate. 14

Figure 6 Evolution of the cumulative discovery rate during the scanning process. 15

Figure 7 Comparison of the AP frequency between a Linux laptop and an Android smartphone. The x-axis shows individual APs, ordered from the least detected to the most detected. The y-axis shows AP frequencies, i.e., the proportion of scanning in which a given AP has been seen. For example, the AP indexed 50 in Figure (a) has been seen in 97 % of the scanning. Thicker lines represent the APs observed in all scans. 16

Figure 8 CDF of the normalized number of detected APs in each scanning. 17

Figure 9 AP discovery frequency (Linux laptop with timer 100 ms, channels 1-6-11). 17

Figure 10 AP discovery frequency vs. median of the signal strength of the Probe Response frames from a given AP (power). 18

Figure 11 AP discovery frequency. Probe timer set to 5 ms, 20 ms and 100 ms. 100 trials with each timer. The client explored channels 1 to 11 sequentially. Figures correspond to the Linux platform. 19

Figure 12 Probe Responses and Beacons count during channel probe (timer 500 ms, probing channels 1, 6 and 11 only). 20

Figure 13 Percentage of APs discovered while probing adjacent channels. Gray boxes indicate the channel where the mobile station (MS) was probing. Values above and below the gray box show the percentage of AP detected in the overlapping channels. For example, while probing channel 1, the MS discovered 70 % of the APs operating in channel 2. 21
Figure 14 Per scan AP's receive signal strength indicator (RSSI) (timer 100 ms). 22
Figure 15 Components of Wireless Measurements Sharing Platform (WMSP). 30
Figure 16 Sequence diagram describing the interaction between MSs and WMSP. 31
Figure 17 Architecture of WMSP. 33
Figure 18 Steps to associate a measurement to a geographic cell. 35
Figure 19 Example of the trace aggregation procedure. 36
Figure 20 Areas covered during the empirical evaluation of WMSP. 37
Figure 21 Distribution of the AP count during the empirical evaluation of WMSP. 38
Figure 22 WMSP software architecture. 39
Figure 23 Parallel computation of a minimal AP set. 40
Figure 24 Scalability of the minimal AP set algorithm in WMSP on Grid'5000. 41
Figure 25 Minimal AP set evaluation trajectory. 43
Figure 26 Availability of the APs in the minimal AP set. 43
Figure 27 Simulated environment used for the evaluation of the minimal AP set. 45
Figure 28 MS throughput distribution in the simulated environment after executing the minimal AP set algorithm. 46
Figure 29 Comparison of the performance of different scanning configuration. 50
Figure 30 Scalability of the optimal scanning parameters algorithm in WMSP on Grid'5000. 51
Figure 31 Beacons transmitted by one AP in an idle (left) and busy (right) channel. 55
Figure 32 Three possible cases for Beacon jitter: 1) b_i and b_{i-1} on time; 2) b_{i-1} on time and b_i late; 3) b_{i-1} late and b_i on time. 58
Figure 33 Throughput and Beacon jitter results from controlled conditions experiments while using two transmitters. 61

Figure 34 Throughput and Beacon jitter results from controlled conditions experiments while using three transmitters. 62
Figure 35 Throughput and Beacon jitter results from controlled conditions experiments while using four transmitters. 63
Figure 36 Throughput and Beacon jitter results from controlled condition experiments while using five transmitters. 64
Figure 37 Empirical distribution of the actual Beacon interval in the controlled condition experiments. 66
Figure 38 Empirical distribution of the actual Beacon interval corresponding to the saturated experiments performed in controlled and uncontrolled conditions. 67
Figure 39 Graphical representation of the matrix containing the Kolmogorov-Smirnov (KS) values for all pairs of experiments. 68
Figure 40 Performance of the channel classification method for different values of the threshold α. 70
Figure 41 Performance of the classification algorithm for various observation windows. Sampling is count-driven, i.e., number of Beacon interval. 71

LIST OF TABLES

Table 1 AP availability and handover time for the four variations of the minimal AP set. 44
Table 2 Characteristics of the city used to simulate the AP's deployment. 45
Table 3 Time validity of the minimal AP set. 47
Table 4 Optimal scanning parameters that warrant 10 ms, 20 ms, 50 ms and 100 ms delay. 48
Table 5 Comparison of the expected and actual scanning performance. 49
Table 6 Symbols used in the chapter 57
Table 7 Total offered load (e.g., the sum of over all transmitting APs) of the experiments performed under controlled conditions. 60
Table 8 Total offered load (e.g., the sum of over all transmitting APs) of the experiments performed under uncontrolled conditions. 60

Table 9 Values of α providing the best performance (highest Matthews correlation coefficient (MCC)) with different reference distributions. 69
Table 10 Performance of the proposed method for $\alpha = 0.21$. 69

1 INTRODUCTION

Today, IEEE 802.11 Wi-Fi[1] enabled devices (e.g., laptops, smartphones, tablets) are ubiquitous and widely used by a number of users accessing all sorts of applications and services. *Ericsson Mobility Report November 2017 – Ericsson* [25] indicates that there were 4.40 Billion smartphone subscriptions globally on 2017, and there will be 7.20 Billion by 2023. It is reasonable to assume that those smartphones are also potential Wi-Fi users, eager to access Wi-Fi networks. To meet the ever-increasing demand for wireless connectivity, different actors have installed IEEE 802.11 access points (APs): Internet service provider (ISP) customers in their homes for their own use; businesses in their offices for their own employees and their customers (e.g., airports, shops, malls); public institutions serving large areas (e.g., local administrations providing network coverage in a city).

IEEE 802.11 networks present different advantages, e.g., low-cost, plug-and-play, offer good throughput and use license-free bands. All these elements allow to easily deploy multiple APs in a de-centralized way, without planning or coordination. Moreover, users and institutions are using Wi-Fi networks to build the so-called *community networks* [8] (e.g., Freifunk[2], Peoples' Open[3], guiFi.net [4], Fon[4]), by allowing users to get access to the Internet via shared Wi-Fi networks. All this results in dense and unorganized deployments.

A key problem of these dense and chaotic deployments is the limited information about the networks. For instance, operators may know the postal address of the users and their APs, but do not have any information about the AP coverage and user experience. Having detailed information is important for operators, for example, in channel selection, and to place new APs to cut coverage gaps. Users do not have enough information about the neighboring Wi-Fi networks either. Traditionally, the only available information to the users is the signal strength of the neighboring networks. Additional information may allow Wi-Fi users to optimize regular procedures like network discovery and selection. Obtaining network information is non-trivial, because of the nature of Wi-Fi networks (de-centralized, operating in non-exclusive and crowded frequency bands, radio links with changing conditions, radio interference).

Researchers have highlighted the need for measurements from the user point of view [67, 69, 70]. Crowdsource [29] is one approach to

1 Through this dissertation we are going to use Wi-Fi and IEEE 802.11 interchangeably.
2 `https://freifunk.net`
3 `https://peoplesopen.net`
4 `https://network.fon.com/`

collect network information using users as probes. Proprietary solutions like OpenSignal[5] use this crowdsourcing approach to measure mobile network performance. Although there are different metrics to describe the network performance (e.g., receive signal strength indicator (RSSI), signal-to-noise ratio (SNR), network saturation, collision probability, application layer throughput), RSSI is commonly used, likely because it is readily available to the clients. RSSI is a measure of the energy observed at the physical layer [36] while receiving a frame. SNR complements the RSSI with the noise level. These two metrics only reflect the signal strength, whereas other characteristics also affect the network performance (e.g., channel load, number of users). Some studies report that the SNR and especially the RSSI may be inaccurate [32, 42] or over-optimistic for key processes like rate adaptation [87]. As a practical example, Google recently enabled Android 8.1 to tell the speed of public networks [19], in addition to the well-known signal strength bars.

The main goal of this dissertation is to propose solutions enabling users and network operators to efficiently collect, analyze, and exploit information about existing Wi-Fi networks. Our approach is to extract information from the increasing number of mobile users who, with their mobile station (MS), periodically collect network information as input for different applications, i.e., we collect measurements from the user perspective. Users and operators could collect information using non-intrusive techniques, and then combine and share it via a collaborative approach. This would allow both, users and operators, to have a better picture of the networks thanks to the contextual information from different MS.

Differently from other approaches, we take advantage of standard procedures to get network information in a non-intrusive way. By relying on standard procedures and features of the standard IEEE 802.11 protocol, there is no need to modify the hardware, allowing existing MS and AP to participate. Recording and aggregating the results of the discovery process allow us to locate Wi-Fi networks and get information about the context in which operate. By monitoring the variations on the Beacon transmission time we can identify whether the channel around APs is busy or not.

Having contextual information is key for a better use of Wi-Fi networks [24] and to enable new services, both for service providers and for users. Potential uses for operators are: (1) monitor and ensure quality of service (QoS); (2) dynamically adjust the deployment on specific parameters, like throughput, coverage, or energy efficiency to fit changes in the requirements; (3) help deciding whether use data offloading [24, 33, 44] from cellular networks to Wi-Fi; (4) planning the installation of new APs by examining the current APs [88]. Contextual Wi-Fi information is also valuable to the users, who can, for

5 https://opensignal.com

instance, optimize network procedures, like AP discovery or AP selection.

1.1 THESIS OVERVIEW AND CONTRIBUTIONS

In this dissertation, we study the Wi-Fi networks in order to propose metrics to characterize Wi-Fi networks. Particularly, we focus on solutions enabling end-user devices (e.g., smartphones, laptops) to characterize networks, without hardware modifications. In addition, we aim for passive techniques so that we do not add more traffic to the already crowded Wi-Fi networks.

Since we have observed a growing number of Wi-Fi networks in urban areas, we first study the IEEE 802.11 discovery process in these urban areas, in order to describe its characteristics and to cut its duration. In the context of the discovery process, we study the transmission of Beacon frames, looking for a technique to passively measure the network quality. Then, since user devices can only partially discover the available networks in dense environments in urban areas, we present a sharing platform to support Wi-Fi processes, like energy saving and discovery.

The main contributions of this dissertation are:

EMPIRICAL STUDY OF THE DISCOVERY PROCESS IN URBAN AREAS In order to propose metrics and techniques to characterize Wi-Fi networks, we first study the IEEE 802.11 network discovery process in urban areas. We note that: (1) there are dense Wi-Fi deployments in urban areas; (2) an MS can not fully discover all available networks in such dense Wi-Fi deployments with only one scan; (3) longer probe timers do not necessarily result in the discovery of more networks; (4) while scanning a given channel, it is possible to discover APs operating on adjacent channels. Based on these observations we argue that a collaborative information service could help the users during the network discovery by (1) storing and aggregating discovery reports; (2) providing configuration parameters tailored for particular areas and user requirements.

DESIGN AND EVALUATION OF AN ARCHITECTURE FOR A COLLABORATIVE INFORMATION SERVICE We propose Wireless Measurements Sharing Platform (WMSP), a collaborative information service, which, using a crowdsourcing approach, collects network data from APs and MSs, pre-process it, and stores it. Different applications can be developed to use the data stored. We implement Wireless Measurements Sharing Platform (WMSP) using big data technologies, and show its feasibility and usefulness by collecting network measurements from two urban areas, and using the data in two applications: minimal AP set and reducing scanning timers.

METRICS AND NON-INTRUSIVE TECHNIQUES CHARACTERIZING WI-FI NETWORKS We propose the Beacon jitter as a metric to identify whether an AP operates on a saturated channel. We also propose a non-intrusive technique to obtain the Beacon jitter. Existing solutions, like the signal strength, miss information about channel usage. Other solutions involve active measurements, increasing the load on the already saturated channel or need special hardware. Users and operators can exploit these techniques and share the results using to the collaborative information service (WMSP), to to get access to a better characterization of the networks.

2

ACCESS POINT DISCOVERY IN IEEE 802.11 NETWORKS

Given the popularity of low-cost IEEE 802.11 networks operating in the unlicensed 2.40 GHz band, it is common to find large installations of IEEE 802.11 networks deployed independently by home users and organizations. This creates spontaneous deployments with different densities and distributions [1], and with unpredictable discovery patterns, i.e., devices operating at different locations, having different performance in terms of hardware and software [74].

Discovering those networks is a pre-requisite for other processes (e.g., network connection and handover) and enable services, for example use Wi-Fi access points (APs) to geo-locate an mobile station (MS) without using GPS [10, 18]. The network connection happens when the MS initiates the network interface card (NIC). Before establishing a Wi-Fi connection, the MS needs a list with information about the available Wi-Fi networks. The MS then uses that information during the association and authentication processes. The handover happens when a MS determines the need to attach to a new AP. In this case, the MS first has to discover the surrounding APs, then it must decide to which one it should try to associate next. In the case of geo-location, one technique consists on discovering the APs in the neighborhood and then position itself by comparing the result with a database of existing Wi-Fi APs [18].

The network discovery starts with the so called scanning process, which is time-consuming [13, 55], costly in terms of the traffic generated [84], energy intensive [10], and usually incomplete, that is, it reports only a subset of the available networks (see Section 2.3).

Currently, the scanning process treats all Wi-Fi deployments equally, resulting in inappropriate probing timers (too short or too long) and incomplete results. Take for example Alice and Bob. Alice carries a smartphone while walking in an industrial area, with a relatively low Wi-Fi density and enterprise grade APs. While Bob is sitting in a dense residential area, where hundreds of families installed home Wi-Fi networks. In both cases the scanning duration would be the same (Section 2.1.1). However, Alice's smartphone may be able to discover all available networks quickly, while Bob's smartphone will only partially discover the available networks.

Inaccurate information about available networks may result in the MS associating to a sub-optimal network, for example. Additionally, the traffic generated during the scanning process consumes a non-

negligible portion of the channel capacity [84], becoming a potential problem.

Addressing the discovery problem in crowded scenarios first requires an understanding of the discovery process actually used in the devices, and its behavior in crowded scenarios. In this chapter, we start with an overview of the scanning process in Wi-Fi networks, then we present an analysis of the discovery process in crowded scenarios. For this we use empirical data, collected in the city of Rennes, France. We first describe the discovery process and review the implementation in the Linux kernel. Then, we analyze the scanning parameters and their impact on the process. Finally, we investigate the relationship between the time spent scanning for available APs and the number of APs actually detected. We show that the scanning process may not be as complete as supposed when using long values for the timers (Section 2.3). Nor is the opposite true, i.e., short timers report only a partial view of the network. In particular, we show that, in order to discover all available APs at a given location, the MSs need to combine multiple scans (Section 2.3).

Based on these results we argue that MSs could improve the discovery process by adapting the scanning parameters according to the characteristics of network deployment, supported by a central service providing information about the networks. APs and MSs collect network information and upload it to the central service. The central service then pre-processes, organizes, and makes the information available. Chapter 3 presents the central service.

Improving the discovery process allows the reduction of the time used for network discovery, reducing network interruptions. Which in turn improves users mobility thanks to faster and smoother handover between Wi-Fi APs. An optimal scanning procedure may also reduce the traffic exchanged at Layer-2, resulting in a better channel usage.

2.1 SCANNING PROCESS IN IEEE 802.11 NETWORKS

Beacon and Probe Response frames are two management frames containing information about the transmitter AP. MSs use that information during an eventual association. MSs can collect Beacons and Probe Responses using the so called scanning procedure.

The standard [36] describes two types of scanning procedures, called passive and active scanning. In passive scanning an MS hops (and listens) over the available channels, listening on each channel for Beacon frames that are periodically sent by APs. The amount of time spent on each channel is called maximum channel time (MaxCT). In active scanning, an MS actively broadcasts Probe Requests frames over the available channels, and waits for APs response frames (unicast Probe Response frames) in the same channel. In other words, in the active

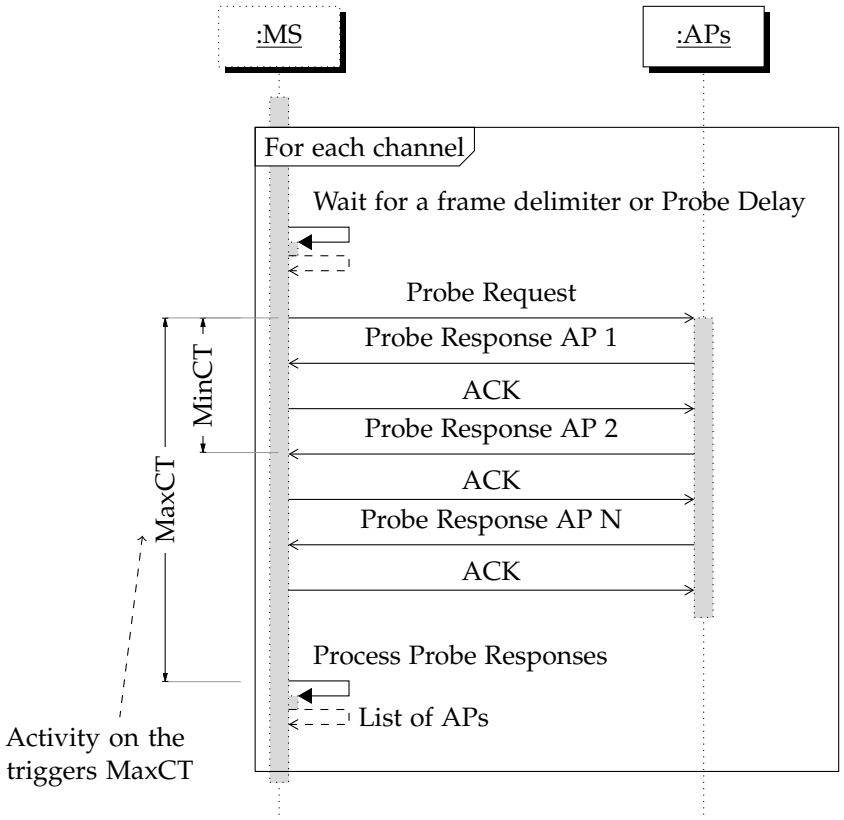


Figure 1: Sequence diagram of the active scanning procedure as specified in the IEEE 802.11 standard.

scanning process the MS triggers explicitly responses from the nearby APs.

The active scanning algorithm uses three timers, namely *Probe Delay*, minimal channel time (MinCT) and MaxCT. MSs have to update the network allocation vector (NAV) before transmitting a Probe Request. Before sending a Probe Request the MS must wait either for the reception of a frame or for the expiration of the *Probe Delay* timer.

The approach specified in the standard combines the timers MinCT and MaxCT to wait for Probe Responses. Note that an MS may also receive Beacon frames during the active scanning. To probe a channel, the MS first sends a Probe Request to the broadcast destination address, it then waits for Probe Responses for MinCT seconds. If the MS detects activity on the channel, the waiting period is increased to MaxCT. Then, the scanning procedure processes all the Probe Responses received and repeats the process in the next channel. The MS returns the list of the networks discovered after processing all channels. The diagram in Figure 1 summarizes the procedure to explore a given channel, as defined by the standard.

Figure 2 illustrates the time spent by the scanning MS on each channel together with the APs discovered. The time spent on each channel is represented by the height of the corresponding bar. In this illustra-

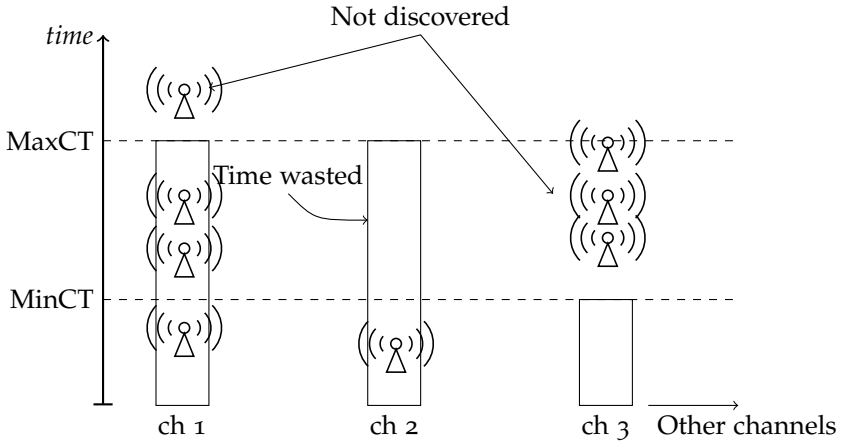


Figure 2: Role of MinCT and MaxCT. Static values for MinCT and MaxCT may cause waste of time (channel 2) and missing AP (channel 3).

tion the MS discovers the APs that are inside the bars, and misses the APs outside the bars. The total scanning time is given by the sum of the heights of the bars.

Figure 2 shows how an MS may waste time in some channels, if it uses times that are too long (e.g., channel 2 in Figure 2) and miss APs on other channels (e.g., channel 3 in Figure 2), when using timers that are too short. An efficient scanning procedure should adjust the values for MinCT and MaxCT according to the channel conditions and the network characteristics. More importantly, to select the timers the MS should take into account the user requirements. For example, the scanning process for geo-location purposes does not require to fully discover all available networks. In this case, a fast scanning, with short times, of a subset of the channels is preferable [10].

2.1.1 *The Wi-Fi Scanning Implementation in the Linux Kernel*

Since the Linux kernel is open source and widely used in smartphones it is interesting to look at. Furthermore, it allows us to set up custom modifications to evaluate different timer values and scanning algorithms. The Linux kernel implements the scanning procedure in the module `mac80211.ko`. Figure 3 summarizes the algorithm implemented to explore one channel. The algorithm corresponds to the version 4.15.7 of the Linux kernel. Notice that the algorithm differs from the one described in the standard [36] (See Figure 1) in the following points:

1. The standard defines three timers (`PROBE_DELAY`, `MinCT` and `MaxCT`) while the kernel uses two timers, `PROBE_DELAY` and `CHANNEL_TIME`. As in the standard, `PROBE_DELAY` is used for clear channel assessment (CCA) purposes. `CHANNEL_TIME` works as the only timer to wait for Probe Responses.

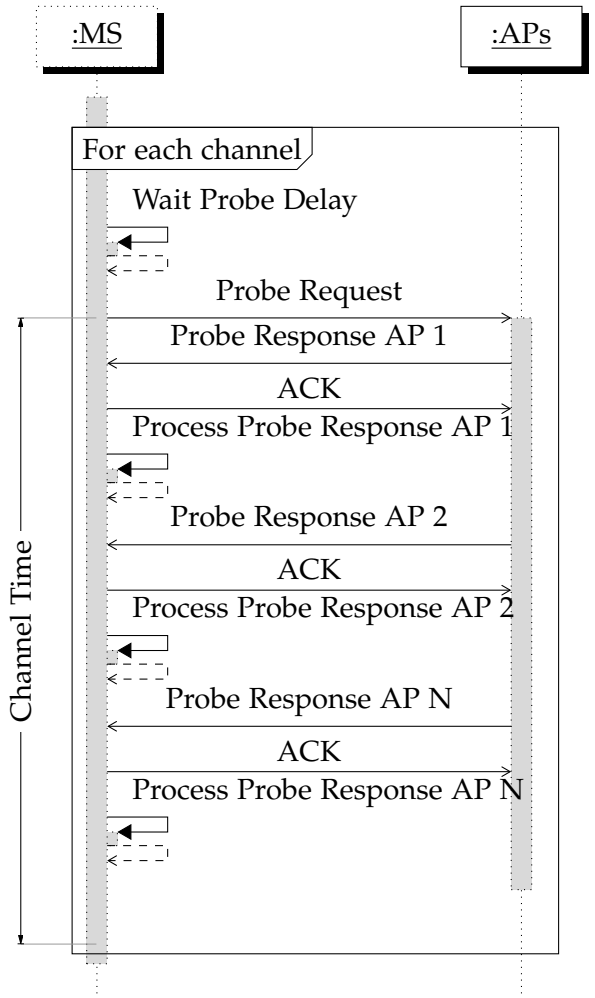


Figure 3: Sequence diagram of the active scanning implemented in the Linux kernel.

2. As mentioned above, the standard states that, before sending a Probe Request, the MS should wait to have an updated NAV. In the case of the Linux kernel the MS always waits for `PROBE_DELAY` to expire before sending a Probe Request.

3. The standard increases the time to probe a channel to MaxCT when the MS detects at least one probe response. The kernel probes each channel for a constant time (`CHANNEL_TIME`), regardless of the channel activity.

4. In the standard, the MS process all the Probe Responses received on a given channel together. The Linux kernel process the Probe Responses as they arrive.

2.2 LITERATURE REVIEW

Since MSs cannot send (and received) data frames while they are scanning other channels, most of the existing proposals on the AP discovery in IEEE 802.11 networks focus on reducing the impact of scanning procedure on other processes. For example, by reducing the duration of the scanning procedure the during the handover. Several studies analyze the scanning process from the point of view of the MS and have identified it as the largest contribution to the handover or the association delay. We identify, in the existing literature, two approaches to reduce the impact on those processes: (1) limit the number channels to probe [26, 71, 72]; and (2) reduce the channel waiting time [9, 14, 54, 63, 81].

Velayos and Karlsson [81] propose specific values for MinCT and MaxCT based on theoretical best and simulated worst case for the Probe Request–Probe Response exchange. The authors use Equation 1 to estimate MinCT, assuming that at least one of the APs operating on the channel will successfully transmit one Probe Response on the first attempt, that is, without collisions. The standard defines the values of the distributed coordination function (DCF) inter-frame space (DIFS), aCWmin and aSlotTime, these three values depend on the physical layer. The resulting MinCT is 670 µs, 163 µs and 163 µs for b, g, n on 2.40 GHz, 169 µs for n on 5 GHz and 169 µs on ac. Velayos and Karlsson [81] rounded MinCT to 1 ms.

$$\text{MinCT} = DIFS + aCWmin \times aSlotTime \tag{1}$$

Concerning MaxCT, the authors simulate different scenarios with different channel loads and conclude that 10.24 ms is a reasonable choice for MaxCT. The simulations indicated that 10.24 ms is enough for the transmission of Probe Responses in a cell that offers good throughput.

Another approach is to dynamically adapt the values of MinCT and MaxCT. Castignani et al. [14] propose to adapt the values of MinCT and MaxCT following the discovery's evolution. The goal is to increase or to reduce the timers channel by channel. The MS reduces the timers whenever it finds an AP, and increases the timers when it does not find any AP. The authors explain that the impact of failing to discover an AP is less significant than not discovering any AP. Castignani et al. [12] suggest combining information from the physical layer to adapt the channel sequence and the channel waiting time. Specifically, Castignani et al. [12] propose to sense the channel during a given time window, then adjust the channel probing time according to the number of stations, and prioritizing the channels by the total received power on that channel. They estimate the number of station on the channel using the signal-plus-noise, and measure the power from the captured signal during the sensing window.

Some proposals consider a discontinuous scanning from the MS in order to be prepared ahead of time in case of a handover event. For instance, Liao and Cao [47] consider that the MS has enough time to perform the handover in a make-before-break fashion. Nah et al. [56] propose to take into account the type of application to adapt the scanning waiting time and the number of channels to scan. The idea is to divide the scanning process into few scans, each probing a subset of the channels, in order to prevent network degradation due to long interruption. The duration of each partial scan is selected so that the MS uses the buffers in the network path to minimize the impact.

Another approach consists in scanning only selected channels based on previous experience of the MS [72]. For example, there may be a ranking based on previous signal strength experienced by the MS. Caching information about the surrounding APs is also used by the MS. This approach although suggested by [72] has been presented as a system using neighbor graphs in [71]. By using neighbor graphs the MS can compute the number of channels to scan and the timer values by exploring its directly connected nodes in the graph. The information on the graph can even be used for unicast Probe Requests as in [60] thereby causing less congestion.

As it has been pointed out by Raghavendra et al. [62], the crowded nature of existing IEEE 802.11 deployments causes several problems such as intermittent connectivity, low throughput and high packet loss. In particular, high packet loss in crowded topologies forces MSs to trigger unnecessary handovers (including scanning and reassociations), even in the absence of mobility. Mhatre and Papagiannaki [52] highlight the need for adaptive handovers to mitigate this performance degradation in crowded topologies.

Castignani et al. [14] have presented an extensive analysis of the scanning process in a controlled environment. Their results are threefold. First, they have identified the trade-offs between several metrics defining the scanning process: the full discovery latency, the failure rate and the discovery rate. Second, they have made several proposals for improving the scanning process. Third, they propose a simple adaptation method for ramping up from low timers to high timer values in order to increase the discovery rate and minimize the failure rate.

Montavont et al. [54] have derived mathematical expressions for quantifying the discovery rate, failure rate and the full scanning latency. Moreover, using genetic algorithms over a set of collected scanning traces, they have found an optimal channel sequence and corresponding timers for achieving high discovery rates and minimum failure rates in a Community Network scenario. This approach has the advantage of allowing MSs to adapt the scanning configuration to the user requirements.

2.3 EVALUATION OF THE DISCOVERY PROCESS

As mentioned earlier, IEEE 802.11 networks operates in the 2.40 GHz and 5 GHz unlicensed bands and it is common to find large installations that are unplanned and uncontrolled. In this section we are interested in an empirical evaluation of the discovery process in those unplanned and uncontrolled networks. This empirical evaluation allow us to:

- Measure the time needed to receive Probe Response and Beacon frames.
- Asses the impact of the channel probe time on the number of APs discovered.
- What is the fraction of the networks discovered during a single scan instance, and what is the contribution of multiple scan instances.
- Describe the contribution of the adjacent channels on the discovery process. That is, APs discovered because of the frames overheard on an adjacent channel.

2.3.1 *Methodology and Tools*

The data used in the evaluation came from two stations (a laptop and a smartphone) configured to continuously scan for IEEE 802.11 networks while registering the responses. The module `mac80211.ko`, which implements the scanning and medium access control layer (MAC) operations in the Linux kernel, is the same in the laptop and in the smartphone. However, our experiments show that the scanning behavior on the smartphone differs from the algorithm implemented in the Linux source code (see Section 2.3.3). Since the Linux kernel allows to delegate the scanning procedure to the NIC, we believe that the NIC handles the scanning procedure in the smartphone and not the kernel. The information collected includes: timestamps of the Probe Request, Probe Response and Beacons, basic service set (SSID), basic service set (BSS) identifier (BSSID), security mode, operating channel and signal strength.

Description of the stations:

LAPTOP Hewlett-Packard, NC-2400. NIC: Intel Corporation, model PRO/Wireless 3945ABG-Golan.

SMARTPHONE Galaxy S3 (GT-I9300) running Android OS 4.3.1.

We ran the experiments at different locations in the Rennes city center, on streets and squares surrounded by four to ten story buildings. We used eigth values for the channel probe timer: 5 ms, 10 ms,

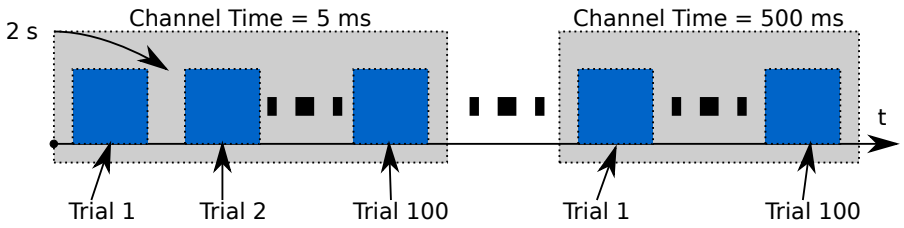


Figure 4: Procedure followed to evaluate of the Wi-Fi discovery process in urban areas.

15 ms, 20 ms, 30 ms, 50 ms, 100 ms and 500 ms. The station executed 100 scanning trials for each value of the timer, for a total of 800 trials at each location. The station triggered the trials sequentially, every two seconds (see Figure 4), taking approximately 45 min between the first and the last scan. We have gathered more than 66 000 Beacons and more than 18 000 Probe Responses.

2.3.2 *Evolution of the Discovery Process*

The network topology is, at first, unknown and it is progressively obtained through sequential scans. The more the MS scans, the more AP it discovers. Figure 5 illustrates a sample discovery process. Scan 1 discovers three APs (AP1, AP2 and AP3). Being the first scan, its results are the known topology at that time. Scan 2 discovers two APs (AP1 and AP4), AP4 appearing for the first time, increasing the known topology to four APs. Finally, Scan 3 discovers two APs: AP2, which was already in the known topology, and AP5, which appears for the first time and increases the known topology to five APs. APs may appear intermittently in successive scans, e.g., AP2, that only appears in scan 1 and scan 3. Thus, we define *AP appearance frequency* as the percentage of the scans in which a given AP appears. In Figure 5 the AP frequencies are 66 %, 66 %, 33 %, 33 % and 33 % for AP1, AP2, AP3, AP4 and AP5 respectively. It is possible that consecutive scans result in different and exclusive AP sets, this is the case of scans 2 (AP1 and AP4) and 3 (AP2 and AP5). Figure 5 suggests that a full discover likely requires several scans. We confirm this in Figure 6, which is based on the results of the aforementioned experiments in Rennes.

Figure 6 shows the evolution of the discovery of the topology for the sequence of 100 trials for different channel time values. The figure only takes into account networks discovered via Probe Responses. Observe that none of the plots reaches the discovery of 100 % of the topology, where 100 % is the set of all APs discovered after the 800 scan trials. That is, even for the largest timer (500 ms), there are some APs (up to 15 %) that were not detected after 100 scan trials, but that have been detected while using other values for the Channel Time, suggesting that further scanning will increase the number of known APs. This could also be due to new APs appearing for short peri-

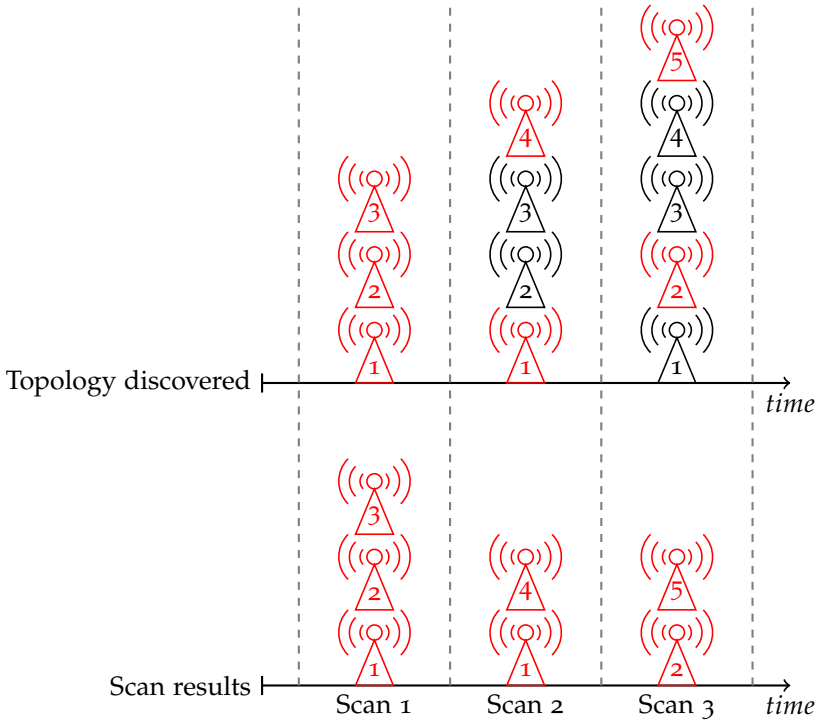


Figure 5: Example of the sequential scanning impact on the discovery rate.

ods of time (e.g., smart-phones used to share a cellular connection or public transportation APs).

Note that, after every new scanning, the discovered topology likely increases. See for example, as indicated by region *A* within Figure 6, how the known topology increases by 4 % after just a couple of scanning in the middle of the cumulative scanning process. This is the case of trials 44 and 45. During trial 44 the MS discovered seven APs, and during trial 45 it discovered six APs. Three out of those six APs were discovered for the first time, while the other three were registered during previous trials. We observe an even more dramatic result for region *B*, for the 20 ms curve, around trial numbered 60, in which the known topology increases by 10.60 % after four scans.

2.3.3 *Dependency on the Platform*

The scanning algorithm and the elements involved in the discovery process may differ from one platform to another (e.g., network card and its driver), affecting the results of the discovery process. This section compares the discovery process on two platforms, a laptop running Debian GNU/Linux and a smartphone running the Android operating system. The laptop used the same configuration described in Section 2.3. The smartphone used the application Wi2Me [16], which periodically triggers a scanning and stores the results locally.

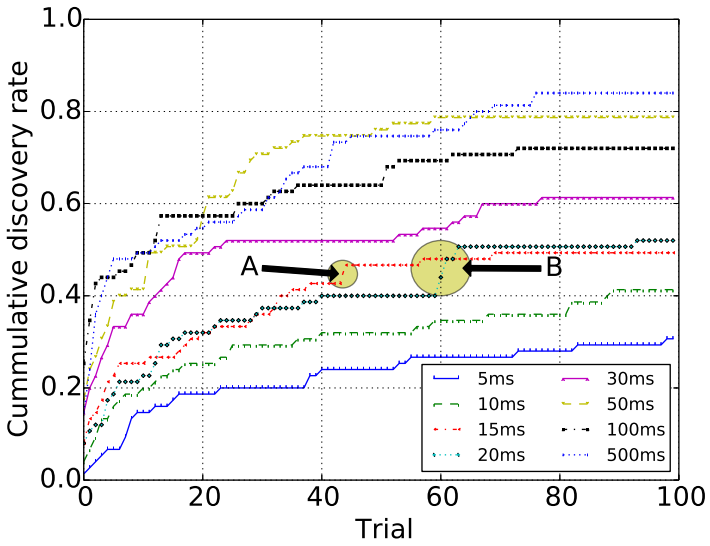
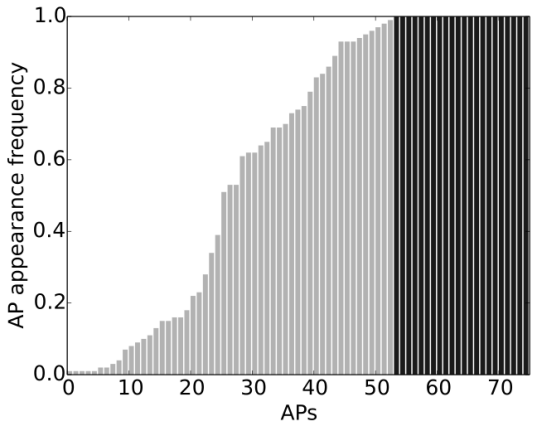


Figure 6: Evolution of the cumulative discovery rate during the scanning process.

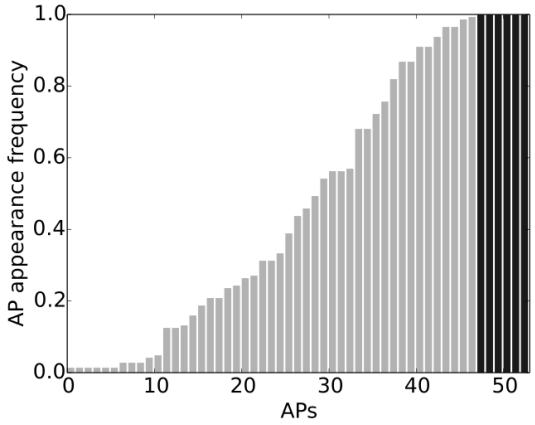
We notice that the smartphone does not follow the scanning procedure as specified in the IEEE 802.11 standard, nor the algorithm implemented in the Linux kernel. A possible explanation is that Android delegates the scanning procedure to the NIC, which implements a different scanning algorithm. Since we do not have access to the NIC implementations, we do not have details about the timers used. However, in our experiment we observe that the smartphone probes each channel for about 250 ms, resulting in a duration of 3.50 s, to probe all 13 channels. Wi2Me initiated the scanning procedure 3 s after finishing the previous scan. This implies that Wi2Me launched a new scan every 6.50 s. The smartphone sends multiple Probe Requests (four on average) per channel, while Linux algorithm sends only one Probe Request on each channel, meaning that the smartphone is more aggressive than the regular Linux implementation.

MSs transmit the Probe Requests in broadcast, i.e., the there will be no re-transmissions. Therefore, the more Probe Requests, the higher the chances for APs to receive the request. However, more Probe Requests imply more traffic in the channel because APs should reply to all Probe Request. Intuitively, the smartphone increases the chances of discovering APs by transmitting multiple Probe Requests. This may be relevant in dense deployments, where collisions are likely to happen, thus reducing the number of APs that receive the Probe Request and resulting in a partial AP discovery.

The AP discovery frequency observed in the two platforms differs. Figure 7 shows the AP discovery frequency for the two platforms in a given location. Both, Figure 7a and Figure 7b, show that most APs are intermittently discovered: only 20 % and 12 % of the APs appear in all scans for the Linux laptop and the Android smartphone



(a) Linux laptop with channel time 250 ms.



(b) Android smartphone.

Figure 7: Comparison of the AP frequency between a Linux laptop and an Android smartphone. The x-axis shows individual APs, ordered from the least detected to the most detected. The y-axis shows AP frequencies, i.e., the proportion of scanning in which a given AP has been seen. For example, the AP indexed 50 in Figure (a) has been seen in 97 % of the scanning. Thicker lines represent the APs observed in all scans.

respectively. The total number of APs discovered differs in the two platforms, Linux discovered 75 APs, while the Android smartphone discovered 53 APs. The difference is likely due to differences in hardware architecture, such as the type and location of the antenna [27, 30], or the chip set.

The Linux laptop discovered more APs than the Android smartphone. In the experiments described in this chapter, Linux discovered 75 different APs while the Android smartphone discovered 53 (see X-axis in Figure 7). Additionally, the discovery process in the Linux laptop is more stable, that is, the variance of the scan result size is lower in the laptop. Figure 8 shows the CDF of the number of detected APs, normalized with the total number of APs detected by the

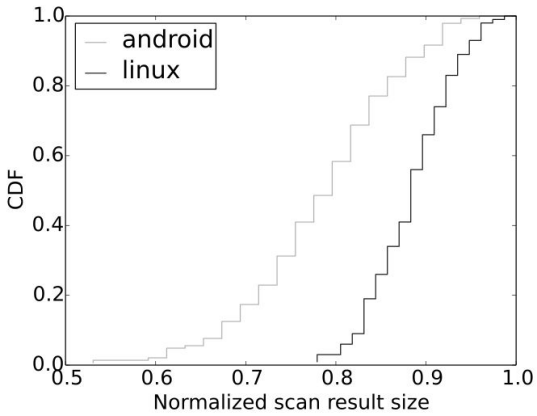


Figure 8: CDF of the normalized number of detected APs in each scanning.

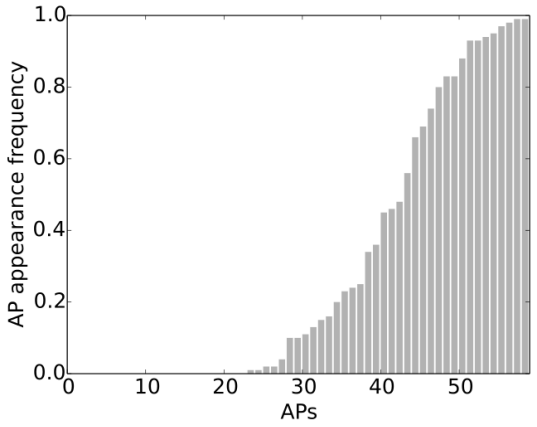
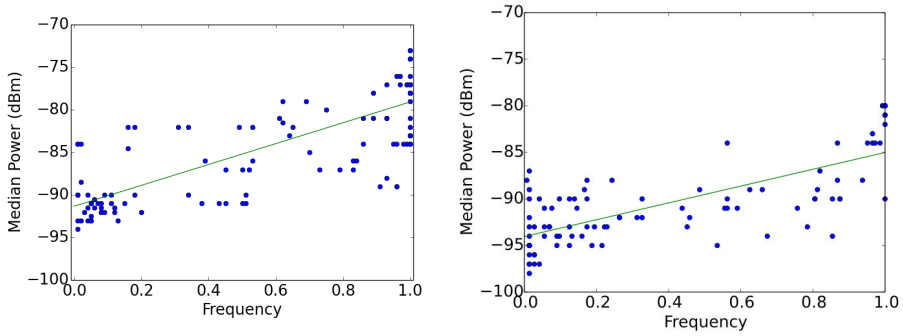


Figure 9: AP discovery frequency (Linux laptop with timer 100 ms, channels 1-6-11).

platform, i.e., 75 for the Linux laptop and 53 for the Android smartphone. In the Linux laptop, almost all scans discovered 78 % or more of the available APs, while in the Android smartphone only 50 % of the scans discovered 80 % or more of the available APs. Focusing on the variance of the number of APs discovered, Figure 8 shows that the Linux laptop has lower variance than Android smartphone, with the Linux laptop ranging between 78 % and 100 % and the Android smartphone between 53 % and 100 %.

Figures 10a and 10b present the scatter plots of the signal strength and the AP discovery frequency, the figures show that Linux laptop tends to receive frames with a stronger signal strength, with an approximate difference of 5 dBm in favor of the laptop. This stronger reception may explain the better discovery of the Linux laptop. In both, Figures 10a and 10b, the signal strength shows a positive correlation between the AP discovery frequency and the signal strength, however, some APs have relative good signal strength and still are discovered a few times, while other APs with low signal strength are often discovered. This is further discussed in Section 2.4.4.



(a) Linux laptop probing all channels with a channel time of 250 ms.

(b) Android smartphone.

Figure 10: AP discovery frequency vs. median of the signal strength of the Probe Response frames from a given AP (power).

2.4 ANALYSIS OF SCANNING CHARACTERISTICS

This section analyzes three parameters that impact the AP discovery process: the value of timer, the contribution of Beacons and Probe Responses and the overlapping channels contributions. Results in this section came from the Linux platform.

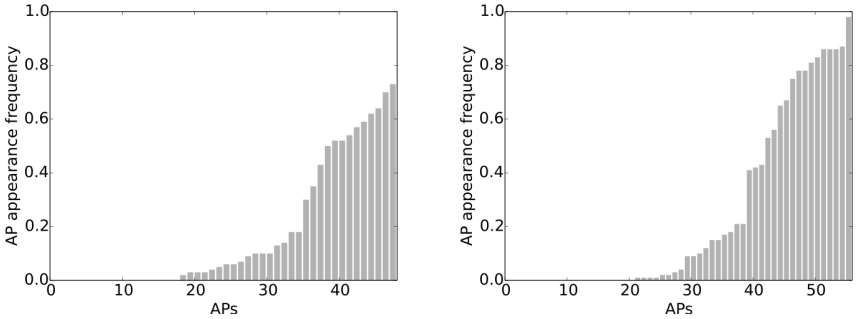
2.4.1 *Timer impact*

Figure 11 shows the AP appearance frequency when using timers 5 ms, 20 ms and 100 ms as channel probe timer. The abscissa represents the APs ordered by the AP appearance frequency, one bar represents one AP. The figures include Probe Responses only (i.e., we leave out passive Beacons), so that we focus on the discovery due to the Probe Request–Probe Response exchange.

As expected, the higher the timer, the more APs discovered. After 100 scan trials the client discovered: 47 APs with a 5 ms timer (Figure 11a), 55 APs with a 20 ms timer (Figure 11b), and 67 APs with a 100 ms timer (Figure 11c). Increasing the timers above 100 ms does not increase the number of Probe Response received (see Section 2.4.2). However, for the first time we observe that 3 APs appeared in 100 % of the scan trials (Figure 11c). Figure 11a shows that most of the APs are discovered in 72 percent or less of the scans.

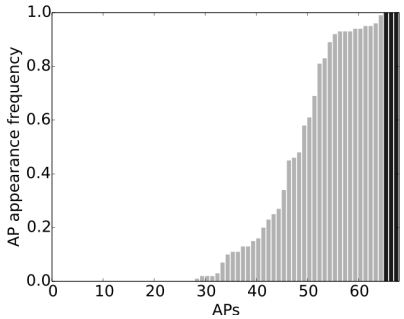
2.4.2 *Probe Responses versus Beacons for Topology Discovery*

A station discovers an AP because it received a Probe Response or a Beacon transmitted by this AP, i.e., Beacons and Probe Responses contribute to the AP discovery process. APs transmit Probe Responses as a response to a Probe Request. In the meantime, APs periodically transmit Beacons every Beacon interval. The Beacon interval is a con-



(a) Probe timer set to 5 ms.

(b) Probe timer set to 20 ms.



(c) Probe timer set to 100 ms.

Figure 11: AP discovery frequency. Probe timer set to 5 ms, 20 ms and 100 ms. 100 trials with each timer. The client explored channels 1 to 11 sequentially. Figures correspond to the Linux platform.

figurable parameter, in our measurements we observed that 102.40 ms is a commonly used value.

Figure 12 presents the number of the received Beacons (top) and Probe Responses (bottom) since the beginning of the scan. This figure corresponds to the probe timer equal to 500 ms, with the client probing the 1, 6 and 11 channels. Observe that Probe Responses appear mainly during the first 100 ms. Instead, the number of Beacons seems to be smaller during the first moments of the channel probe, likely due to collisions with the many Probe Responses that follow the Probe Request [84], then the Beacon count increases and remains relatively constant up to the end of the channel probe. To summarize, the MS receives few Beacons following a Probe Request, but large timers allows the MS to receive more Beacons and to discover more APs.

2.4.3 *Overlapping Channels*

Castignani et al. [15] show that 80 % of the APs operate on the non-overlapping channels 1, 6 and 11. Based on this observation, studies such as [54] or [72] propose to limit the scanning to those three pop-

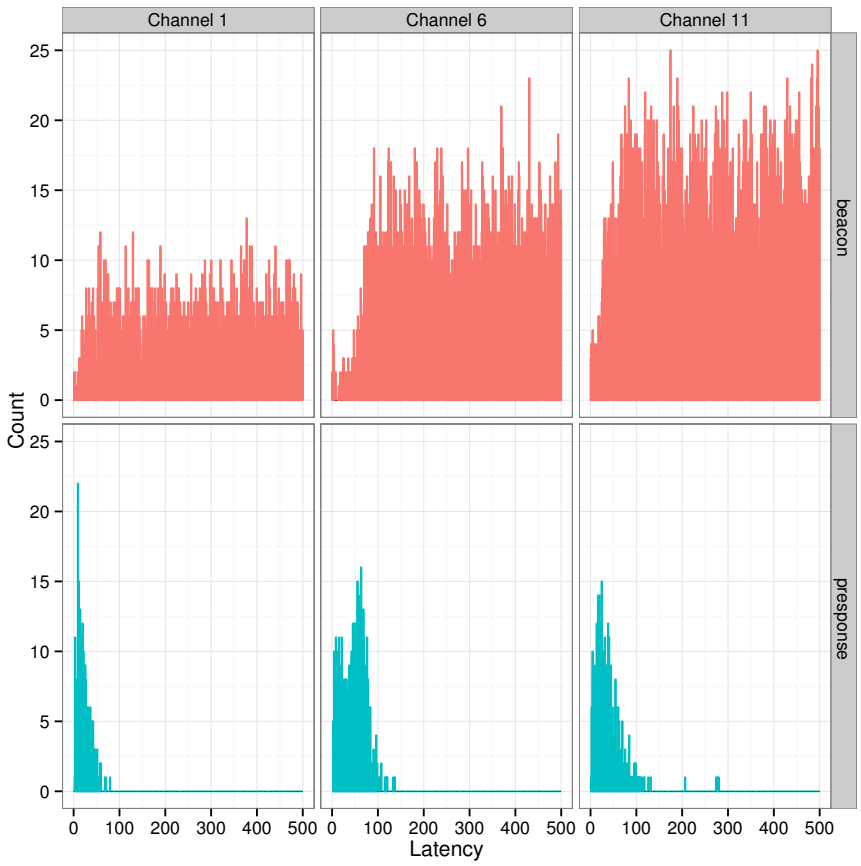


Figure 12: Probe Responses and Beacons count during channel probe (timer 500 ms, probing channels 1, 6 and 11 only).

ular channels to reduce the scanning latency, while still obtaining a reasonable discovery ratio.

A comparison of Figure 11c, where the MS probes all 13 channels, with Figure 9, where the MS only probes channels 1, 6 and 11 shows that, as expected, the MS discovers more APs when probing all 13 channels. Interestingly, more APs are present in all scanning trials. This is because, when scanning all channels, the client probes more often each channel due to the overlapping nature of channels. For example, scanning channel 2 probes the overlapping channel 1, therefore increasing the total number of APs found as it is possible to detect part of the APs operating on channel 1. The IEEE 802.11 standard divides the 2.40 GHz band into 13 channels in Europe. Channels are 20 MHz wide[1] with a separation of 5 MHz, which implies that adjacent channels partially overlap. This allow APs and MSs to "overhear" frames transmitted in adjacent channels. This is particularly true for Probe Requests, Probe Responses and Beacons, which use the strongest modulation and coding scheme (MCS). (Using the strongest

1 22 MHz in the case of IEEE 802.11b, and possibly 40 MHz in the case of IEEE 802.11n

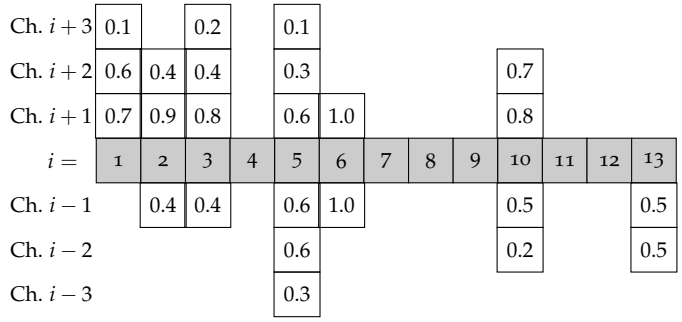


Figure 13: Percentage of APs discovered while probing adjacent channels. Gray boxes indicate the channel where the MS was probing. Values above and below the gray box show the percentage of AP detected in the overlapping channels. For example, while probing channel 1, the MS discovered 70 % of the APs operating in channel 2.

MCS increases the chances of the frame being decoded, even with a weak signal.)

Figure 13, shows the fraction of APs discovered on overlapping channels. For example, while the client was probing channel 2, it discovered 40 % of the APs operating on channel 1, 90 % of the APs operating on channel 3 and 40 % of the APs operating on channel 4. Note that some channels are empty, and channels 1, 6 and 11 are more popular than the others. Probing channels adjacent to crowded channels results in the discovery of more than 40 % of the APs on the crowded channel. For example, probing channel 5 results in the discovery of 60 % of the APs operating on channel 6. Similarly, probing channel 10 allows the discovery of 80 % of the APs on channel 11. Additionally, the client may discover APs operating up to 3 channels away. This is the case of probing channels 1, 3 and 5. Probing channel 1 results in the discovery of 10 % of channel 4, probing channel 3 results in the discovery of 20 % of channel 6 and probing channel 5 results in the discovery of 80 % of channel 8 and 30 % of channel 2.

2.4.4 *On the Quality of Access Points*

Evaluating the quality of APs is complex, especially during the discovery phase. Aspects such as the variability of radio channels, the signal strength, the channel load, the number of operating stations, the environment, and the quality of the back haul network impact the user quality of experience. receive signal strength indicator (RSSI) is a popular metric to compare networks, specially for network selection [77]. When performing an active or passive scanning, the client has access to the RSSI for the received frames.

Figures 10a and 10b show the median RSSI for the received frames from all discovered APs. Figure 14 shows the RSSI of the Probe Re-

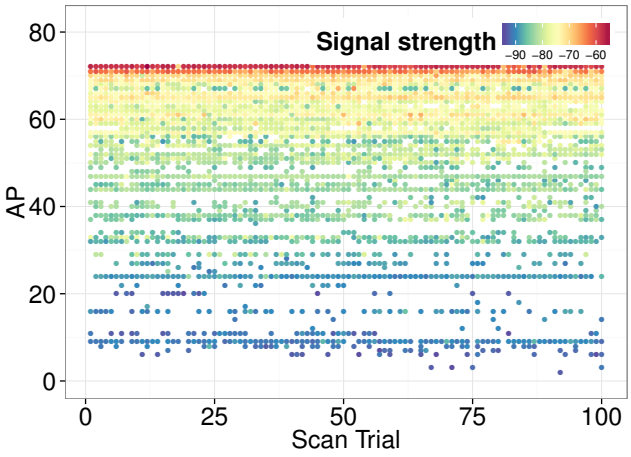


Figure 14: Per scan AP's RSSI (timer 100 ms).

sponse corresponding to all APs and scan trials. The figure corresponds to probe timer 100 ms. Figures 10a and 10b show a correlation between signal and AP discovery frequency, that is, the stronger the AP signal strength, the more popular the AP. Figure 14 shows the same trend, however, the figure also shows that the signal strength varies from one sample to another. Also, some APs with a good signal appear only a few times. Thus, there is no ironclad guarantee that a discovered AP, with strong signal strength, will provide good connection for the client. Interested reader can refer to [28] for an in depth discussion of the subject.

2.5 CONCLUDING REMARKS

In summary, discover nearby networks is a pre-requisite for network connection, handover and geo-location. Handover is important for mobile users because it enables the MS to switch from one AP to another. During the handover the MSs have to interrupt ongoing connections in order to execute the scanning, select and associate to a new, hopefully better, AP. Thus, an ideal scanning process should discover all available APs quickly, however, it is time-consuming and results in an incomplete list of available APs.

Much work has been done to study the IEEE 802.11 scanning process, mainly in the context of the handover process. MSs may adjust the scanning by changing the channel probing timer and the number of channels to probe. MSs may change the scanning parameters once and for all or dynamically, taking into account the network characteristics and the user requirements.

We observed that for different client devices, when the number of APs is high, a single scanning procedure is not enough to discover all the available APs. We have also discussed the differences between active and passive scanning. Although the former contributes the most to the discovery process it cannot detect all the available APs, only the latter can. We also observe that the active scanning can take advantage of the channel overlap, by overhearing frames on adjacent channels.

Given the results presented in this chapter, we argue that knowing few key characteristics of the network beforehand helps in the discovery process. Chapter 3 describes a collaborative solution that enables users to store and share their (partial) view of the Wi-Fi network topology. This solution may provide the above mentioned network characteristics. MSs can, for example, dynamically set the scanning parameters (channels to explore, MinCT and MaxCT) according to the expected network characteristics. Similarly, MSs may access the information in the collaborative solution to complete their scanning results.

3

COLLABORATIVE INFORMATION SYSTEM

Numerous [1, 3] studies and projects report the presence and growth of Wi-Fi networks all around the world (e.g., OpenSignal, WiGle, G-MoN, Sensorly). For instance, Achtzehn et al. [1] report the increase of the density of Wi-Fi networks in urban areas by 14 times in the past decade. By April 2018, *iPass Wi-Fi Growth Map* [90] reports more than 286 Millions hotspots worldwide, and *The Zettabyte Era* [79] predicts that there will be nearly 541.60 Million public Wi-Fi hotspots by 2021. These Wi-Fi networks are usually installed independently, without any planning are they are unmanaged, resulting in "chaotic" Wi-Fi deployments [3]. A key element for a better use of these networks is to gather contextual information (e.g., location, quality, coverage, and available throughput of the access points (APs)) from the mobile station (MS) perspective [24]. However, because of the nature of these unplanned networks (de-centralized, operating in non-exclusive and crowded frequency bands, radio interference) obtaining such information is non-trivial.

By analyzing these networks, users and network operators could improve network performance, and provide services more efficiently, for example, data offloading [24, 33, 44]. Also, having network information can potentially enable energy reduction, by dynamically turning on and off APs to better match transmission capacity and traffic demand. Operators could also improve network coverage by deploying more APs to close coverage gaps and/or to increase the network capacity in high demand areas. For instance, during a parade, operators could use the customers' APs to provide a network tailored to the event, and to identify coverage gaps to place new APs.

At the same time, there is an increasing number of mobile users who, with their MSs, periodically collect network information as input for different applications. Typically, MSs collect information about the networks during the network discovery process, whose result is a list of detected APs, including the receive signal strength indicator (RSSI), operating channel, and capabilities. In addition, current smart phones include global positioning system (GPS) receivers allowing to geolocate the results. This background process does not need additional efforts from the users. A single user has a partial view of the network (see Section 2.3). Even if she combined all the results collected by her MS over a long period, she would have only a partial view of the network, strictly limited to the locations that she has already visited. If multiple users share these results, they can benefit from the observations of other users.

This chapter presents Wireless Measurements Sharing Platform (WMSP), a collaborative information service to gather, aggregate, and exploit data collected by mobile users. WMSP solves the challenges related to the collection, data pre-processing and aggregation of partial and/or inaccurate Wi-Fi measurements. To ensure the scalability of the system, we have used Big Data and cloud-based technologies. WMSP starts by pre-processes raw measurements provided by the users. These data are further analyzed by pluggable "applications," which are an integral part of the system, to solve particular problems, for instance to facilitate cellular traffic offloading, and network planning.

We present two applications (use cases) for WMSP: *minimal AP set* [38] and *optimal scanning parameters* [54]. The *minimal AP set* application computes a subset of the existing APs that are capable of offering seamless coverage in a given area, while lowering the energy consumption. The *optimal scanning parameters* application uses a genetic algorithm to optimize the network discovery, i.e., the process that allows MSs to find available networks. This discovery phase is the dominant factor when executing a handover [53] (when an MS moves outside the coverage area of the serving AP and switches to another).

Experiments show the feasibility and scalability of WMSP. The test dataset is the result of more than 150 man-hour, covering a total distance of over 700 km.

3.1 RELATED WORK

In this section, we present the related work on collaborative information systems, including information services, centralized controllers, and data collection via crowdsourcing.

3.1.1 *Information Services*

The idea of an entity providing information about existing networks is not new. For example, the standard IEEE 802.21 [39], whose main purpose is to support the handover between heterogeneous technologies, provides a framework that enables the interaction with lower layers, that is, a "glue" between upper layers and the current and future mobile network technologies [20]. This develops the paradigm of *media independence* to offer an independent abstraction to upper layers. The standard IEEE 802.21 also introduces the media independent information service (MIIS) to provide network information within a geographical area. The information is available to the MSs regardless the point of attachment to the network (e.g., IEEE 802.11, IEEE 802.3,

3GPP). For instance, an MS may use it to get information about the available Wi-Fi or cellular networks within a given area.

De La Oliva et al. [21] advocate using of the *media independence* paradigm, developed in the standard IEEE 802.21 [39], for applications others than network handover. Particularly, the authors discuss the requirements in the context of White Spaces [37] as a possible use case. They propose an updated architecture for the IEEE 802.21 standard, adding services that allow information exchange between nodes at layer 2 or at layer 3.

3.1.2 *Centralized Systems*

Tamma et al. [78] use fixed sensor nodes, at given locations, to collect network traffic statistics. The sensor nodes send the statistics to a central controller that stores traffic records tagged with time and location information. A central controller uses these records to characterize networks in terms of space, time and frequency. This proposal relies on a set of carefully placed and dedicated sensor nodes. This is plausible for controlled and relatively small deployments, like campus networks. But it is not clear who could shoulder the cost of building such a system to monitor available APs in an urban setting.

Bi et al. [6] give a general overview of possible use cases of Cloud computing and Big Data for wireless networks. They use the term "Big Data aware wireless network," consisting of several mutually complementary components that enable data-driven services. For instances, data-aware cache management, crowd computing and mobile cloud processing. They also propose a hybrid signal processing model in the context of wireless networks, combining signal processing in the base stations and in the Cloud.

Dely et al. [22] use cloud-based technologies to partially move medium access control layer (MAC) layer procedures from the physical APs to virtual APs running on remote servers. APs forward MAC frames between virtual APs and MSs. The virtual APs handle the processing of the MAC layer. OpenFlow [58] handles the link between virtual and physical APs. This way, one can control all these virtual APs centrally, enabling, for instance, to lower the energy consumption, by dynamically switching the APs on and off, depending on the network load and on the coverage. To this end, the physical APs collect information about the channel utilization and offer this information to the network controller.

Ding et al. [24] present SoftOffload, a central and collaborative platform to offload data traffic from cellular networks to Wi-Fi networks. The platform has three main components: a central controller, local agents deployed on the APs and an extension module for the MSs. The central controller collects information from network devices. It also tracks and manages all agents in the network. The MSs collect

network and user information, such as, user movements and signal strength of neighboring APs. The information collected is then analyzed by the controller using a context-aware (i.e., available bandwidth, nearby APs information, mobility effects) decision algorithm to perform offloading decisions. Ding et al. [24] focus their proposal on the software design networking (SDN) architecture. As far as the central controller is concerned, they propose a cloud-based solution, but they do not give any detail about this element.

Finally, alternatives such as the Behop project [85], VPuN [65] or PAWS [66] offer centralized services to control wireless network deployments. The focus of those proposals is on traffic profiling.

3.1.3 *Data Collection via Crowdsourcing*

Different authors (e.g., [5, 27, 46, 57, 61, 75]) and platforms (e.g., SpeedTest.net[1], OpenSignal[2]) take advantage of the popularity of smartphones and use a crowdsourcing approach to measure wireless networks, that is, use mobile users as networks probes to perform wireless measurements. For example, Sommers and Barford [73] compare Wi-Fi and cellular performance using crowd-sourced data from SpeedTest.net, Sommers and Barford [73] indicate that Wi-Fi generally delivers better throughput, however, the latency is often better with the cellular networks. The authors highlight the lack of information about AP locations and the challenges to assemble a dataset of the Wi-Fi networks in urban areas.

The NetSense project [75] studies the wireless networks surrounding the campus of the University of Notre Dame. The study address two sides of the wireless networks: the technical side and the sociological side, that is, how does the digital world impact the social life (e.g., how we make and keep friends). The project provided a smartphone with unlimited data, unlimited texting, and unlimited mobile-to-mobile calls for free in exchange for participation. The project involved a group of roughly two hundred users. Each user carried a smartphone running a pre-installed application, which gathered all sorts of data periodically (e.g., networks, proximity with other users, device state, application usage, location, communications, contacts), and then uploaded the traces to a server, storing the data using a MySQL database.

Striegel et al. [75] focuses on discussing the challenges of the creation, instrumentation and management of the project itself. Some studies use the NetSense data to analyze the technical side (e.g., [48, 70, 76]), while others focus on the sociological aspects (e.g., [49, 51, 83]). Concerning the studies on the technical aspects of the wireless networks, Liu and Striegel [48] use data collected by the NetSense

1 http://www.speedtest.net

2 https://opensignal.com

project during a period of eight weeks to study Wi-Fi as a candidate for data offloading from cellular networks. Authors found that the cellular consumption still dominated overall data usage, and question the gains of Wi-Fi as an alternative for cellular offloading. In contrast, in a subsequent study, Striegel et al. [76] found that the data usage is roughly equal in long term evolution (LTE) and in Wi-Fi. In this second study Striegel et al. [76] also used data collected in the NetSense project, with two differences: a longer period (two and a half years) and with portion of the smartphones upgraded. Striegel et al. [76] observe that: (1) "device design plays a tremendous role in Wi-Fi offloading", that is, inexpensive handsets are related to poor Wi-Fi quality. (2) Download traffic is $4\times$ the upload traffic. This ratio remains consistent regardless of the network technology in use (cellular or WI-FI) and the user location. Shi et al. [70] evaluate the use of the smartphone scans to monitor and plan Enterprise Wi-Fi networks, that is, use clients as network probes. In this study, authors combine the data sets of two projects, NetSense and PhoneLab [57]. Shi et al. [70] concludes that smartphones provide unique insights which are difficult or impossible to obtain by other means (e.g., statically deployed sniffers and measurements from the infrastructure side).

Differently from NetSense [75], that uses a dedicated agent running on the smartphones and collecting a set of predefined metrics, PhoneLab [57] proposes an open platform for testing applications. PhoneLab consists of 288 Android smartphones used by the study participants as their primary device. PhoneLab lets interested researchers develop Android applications specific to each experiment, i.e., possibly collecting different kind of information. At boot time, the smartphone starts the "PhoneLab Conductor", which handles the configuration and data collection tool. The PhoneLab Conductor keeps track of the experiments configuration and uploads the results of the experiments running in the smartphone. Similar to PhoneLab, LiveLabs [5] is a test bed allowing *in-situ* experiments, the project seeks to include a larger community, including a university campus, an airport, a shopping mall and a resort island, across iOS and Android devices. PhoneLab's participants are limited to the community the University of Buffalo and use only Android devices.

Farshad et al. [27] use a mobile crowdsensing approach to study the Wi-Fi networks in the city of Edinburgh, where the participants carried smartphones with an application registering Wi-Fi Beacons. Authors report that, it is common to find ten APs contending on a single channel. Also, they report densities of up to 59 APs in a single spot, with roughly 70 % operating on channels 1, 6 and 11. In conclusion, Farshad et al. [27] outline a crowdsensing system to monitor Wi-Fi networks, with a cloud-based back-end that is aware of the channel conditions. Users and services providers can use the information on the cloud to better configure their Wi-Fi networks.

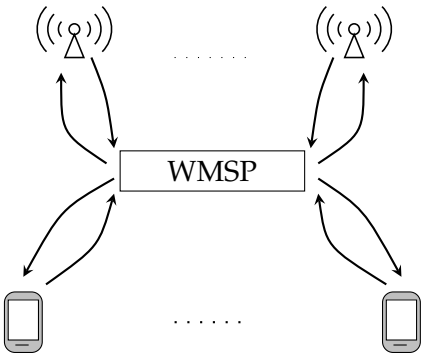


Figure 15: Components of WMSP.

Different authors [46, 61] study the variability of the cellular measurements provided by crowdsourcing participants. Li et al. [46] propose a crowdsourcing approach to monitor the quality of cellular networks. A key element of the proposal is the aggregation approach of the measurements from cellular networks. Authors assume that the measurements from different users have difference quality and precision due to the variance in user device and users habits. Thus, authors propose a weighted aggregation approach to process measurement values from users and to obtain consensual quality measurement. Peng et al. [61] considers the quality of the data to design incentive mechanism for the users. Peng et al. [61] then develop an algorithm to estimate the quality of the measurement provided by the crowdsourcing participants. The authors then offer a reward in accordance with each measurement contribution, aiming to motivate participants to contribute with higher quality measurements.

3.2 CROWDSOURCING PLATFORM

WMSP collects, stores and process measurements of IEEE 802.11 networks. Figure 15 shows the actors involved:

- MSs that collect measurements actively or passively, such as RSSI and network identification;
- APs that may be used as a relay to reach the MSs (to forward requests and responses), and/or perform network measurements themselves;
- WMSP itself that collects, stores and processes the measurements. It also offers access to the measurement database, so that specific applications can exploit the information.

Figure 16 shows the interactions between clients and WMSP. An MS contributes with raw measurements of the topology, that is, a partial view of the topology as observed by the MS. Then, WMSP preprocess and stores the information. Eventually, other MSs arriving to

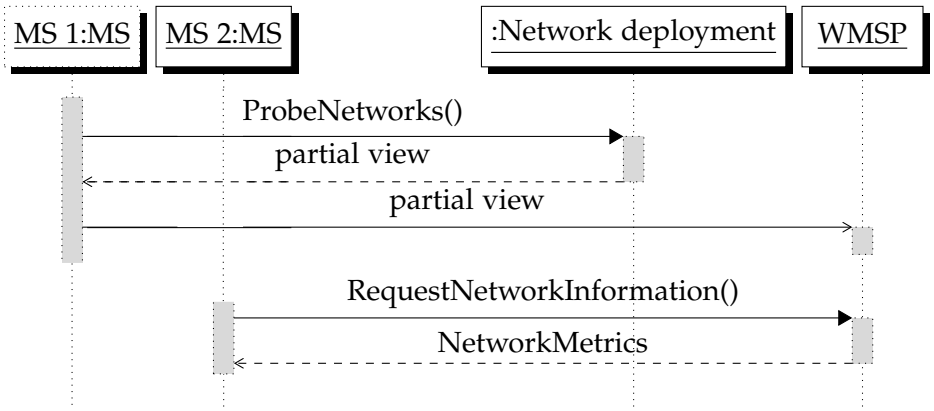


Figure 16: Sequence diagram describing the interaction between MSs and WMSP.

the same place will contribute with other views, so that WMSP will build a more accurate representation of the actual topology. WMSP makes the known technology available to the applications, such as the minimal AP set (Section 3.4) or the variable scanning parameters (Section 3.5).

3.2.1 *Information Collected*

WMSP collects a few metrics that are easily gathered by existing devices, and yet sufficient to characterize wireless network deployments. The metrics include: AP's basic service set (SSID), basic service set (BSS) identifier (BSSID), supported security protocols and operating channel, RSSI, link data rate and the GPS coordinates where the device took the measurement. This information allows WMSP to characterize the network deployments, including the AP density, the channel usage and network performance [15].

Both, MSs and APs can collect these measurements. Devices can simply report the link quality observed. A device, particularly an MS, can also gather this information after the network discovery. The network discovery is usually triggered when an MS needs to perform a handover a new AP, but it can also be triggered on request. Note that two different devices will likely observe different link quality because of their hardware and software differences. Even two consecutive measurements from the same device may differ (see Section 2.3). To cope with these variations, WMSP exploits the popularity and widespread use of smartphones. The high user density carrying smartphones and other connected devices, particularly in urban areas make possible to have an up-to-date data collection [11] and accurate view of the network topology and its performance.

3.2.2 *Collecting Measurements*

We have identified two possible methods for WMSP to gather measurements: 1) using a mobile App and 2) following the methodology described in the standard amendment IEEE 802.11k [38].

The first method is to use a mobile App on the devices. This App logs network activity and discovery, and periodically sends these logs to the WMSP front end. The App could be pre-installed by an operator, or simply downloaded by volunteers from an App store. We used the mobile App called Wi2Me [16] to collect the data presented in this work.

The second method is based on the standard IEEE 802.11k [38], which defines Radio Resource Measurements. This amendment defines the methods and the information exchange between stations (MSs and APs). Any station can trigger a radio measurement locally, request another station to perform a radio measurement or be requested to perform a radio measurement. The radio measurements contain layer 1 and layer 2 metrics, including network discovery information and channel load estimation. With IEEE 802.11k, WMSP could request periodic radio measurement to MSs and APs. Depending on the information requested, APs could perform the measurements by themselves or rely on their associated MS.

3.2.3 *User Incentives*

While the availability of extensive network measurement is essential for WMSP, how the measurements are collected, by whom and what are the associated incentives is a separate, albeit important, concern. The best strategy to achieve this goal depends on many factors, including who deploys WMSP (a network operator, a content provider or a third party, e.g., local government) and associated business models/incentives. It is worth pointing out that, first, the overhead or "cost" to the users is low, if not negligible, and, second, WMSP is application agnostic, therefore, the incentive policies can be different for each application.

The overhead to collect the measurements is negligible because MSs by default already have access to several measurements (e.g., available APs, associated RSSI). The MS will need to store them for a short time and upload them, using a network connection. One can ensure that these costs are indeed negligible by limiting the amount of measurements stored on each device and by uploading them to WMSP only when connected to a Wi-Fi network (as these usually do not count towards traffic quotas of cellular data plans). Privacy concerns can represent an additional cost [40] but they can be addressed by anonymization techniques. If the application is installed by default, either by the manufacturer or by the network operator, no user action

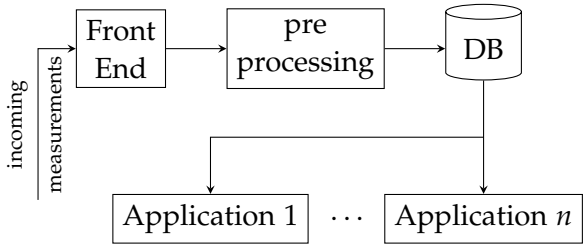


Figure 17: Architecture of WMSP.

is potentially needed [27], other than agreeing to an EULA. Otherwise, the user will need to incur the additional cost of installing a new application. In this case, users will participate only if the resulting rewards compensates the effort and resources needed [17]. This reward can be intrinsic (the participation itself is beneficial to the users) or extrinsic (the crowdsourcer pays for the participation).

3.2.4 *Wireless Measurements Sharing Platform Architecture*

Even though each AP and MS generate reasonably sized measurements every day, adding these up over the many APs and MSs currently deployed can lead to constantly growing amount of data. For instance, a provider with 20 millions subscribers, each one contributing 500 KiB worth of measurements each day would have to process 10 GiB of new data every day. Storing historical data and combining it with new measurements is essential to obtain complete and accurate results. Several solutions, usually referred to as *Big Data*, have been proposed to handle large amount of data. They include databases (e.g., Cassandra, MongoDB, HBase), as well as parallel computing frameworks (e.g., Hadhoop, Spark, Storm, Flink).

Figure 17 shows the architecture of WMSP itself. A front-end subsystem interacts with the APs and the MSs, either by polling them directly according to a given policy or simply by receiving the measurements from the APs and the MSs. In our implementation, an Apache Spark job polls periodically the front-end to retrieve the last data received. This streaming job pre-processes the data and stores the results in an Apache Cassandra database. We give a more detailed description of this step in Section 3.2.5.

One key advantage of NoSQL databases is that they natively support reliable cluster deployments, transparently partitioning and duplicating data among several nodes. Different NoSQL databases offer different consistency models. In our specific use case, we do not have strict consistency requirements: using (slightly) outdated measurements is not ideal but far from catastrophic, similarly, loosing a few measurements is acceptable. Traditional databases offer stronger consistency guarantees but the price to pay is non-negligible, in terms of performance and complexity of cluster deployments.

Parallel computing is obviously well suited to process large amount of data. They can run on commodity hardware, i.e., standard server computers, and do not need particularly powerful processors and/or large amounts of memory.

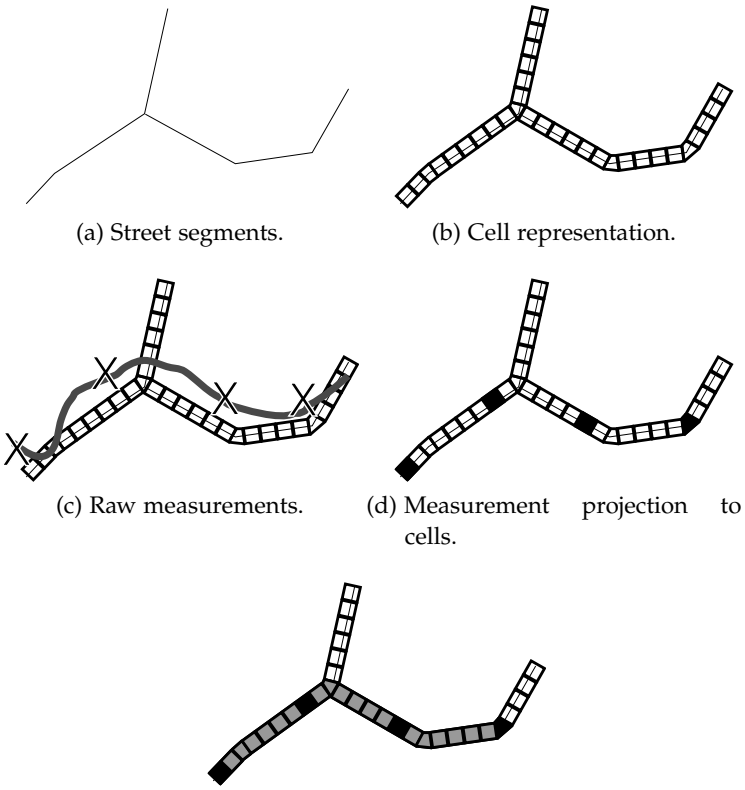
Applications can interact with both the APs and the MSs or with only one device type. Applications can use a different dissemination strategy (pushing or pulling), depending on their specific needs. Sections 3.4 and 3.5, describe two applications: one that selects a subset of the available APs in order to save energy and another that computes the optimal scanning parameters for a given geographical area.

3.2.5 *Data Preprocessing*

This section further describes the preprocessing phase mentioned above (see Figure 17). Recall that each measurement is associated with the GPS coordinates reported by the device at collection time.

The first step consists in correcting and rounding each sample's GPS coordinates. As GPS measurements are prone to error, each data sequence may follow a different geometric trajectory even if the measuring station followed the same path. To address this problem, WMSP uses the procedure shown in Figure 18. First, the system discretizes the street segments using one-meter cells (Figures 18a and 18b), each measurement (figure 18c) is associated to the closest cell using the euclidean distance, shown as black cells on Figure 18d. Finally, when estimating the coverage of an AP, it is reasonable to assume the presence of the AP between two consecutive observations of the same AP, shown as lightly shaded cells in Figure 18e.

Having a cell-based representation, it is straightforward to merge different traces, i.e., measurements collected at the same location either by different users or by the same user at different times. Figure 19 shows two sequence of measurements that observed the same AP. The measurements reporting an AP are represented with a check mark, and the ones that did not with a cross. Based on these results, it is reasonable to assume the presence of an AP between two consecutive measurements containing it (as mentioned above), as in the case of the measurements corresponding to the ones in Trace 1 and Trace 2 in Figure 19. If a measurement does not contain the AP, the algorithm assumes that it is not available in the corresponding cell, hence the zeros in Trace 1 and Trace 2. In this case as well, the algorithm assumes that an AP is not present in all the cells between two consecutive measurements that did not detect it. It is possible to have some cells between the last measurement where an AP was detected and the first one where it was not. These cells are indicated the question marks in Figure 19. It is possible to reduce the uncertainty of the coverage by merging different traces, as the measurements that detected an AP can be in different cells. Different merge strategies are



(a) Street segments.

(b) Cell representation.

(c) Raw measurements.

(d) Measurement projection to cells.

(e) AP coverage interpolation.

Figure 18: Steps to associate a measurement to a geographic cell.

possible. For the sake of simplicity, we consider an AP as available in a cell as long as the AP has a one in any of the traces. Therefore, the trace containing the final merge will replace the question marks with zeros. Other strategies may take into account additional information during the merging, for instance the age of the trace or the RSSI of the APs.

How often should WMSP update the information stored in the database and how often should the results be recalculated depend on several factors, including the sensitivity of the results to the accuracy of the input data, and how often network characteristics change. Take, for instance, the two applications described in section 3.4 and in section 3.5. For the minimal AP set, having a slightly outdated AP set might result in coverage gaps, if some APs in the set were moved. The scanning parameters are, instead, resilient to slightly outdated information because they are based on the statistics of a geographic area. A possible practical approach is to use events to trigger the analysis. For instance, WMSP could recompute the minimal AP set whenever an MS reports a coverage gap.

More generally, there is a trade-off between the frequency of the measurements, the amount of historical data that is stored, how often

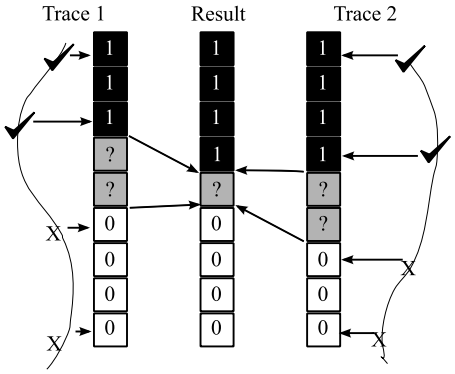


Figure 19: Example of the trace aggregation procedure.

the results are recomputed, and the relevance of the results. These trade-offs are not analyzed in this dissertation.

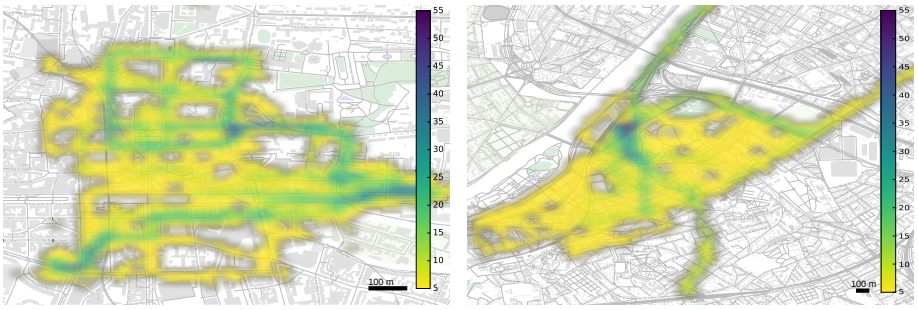
3.3 EMPIRICAL EVALUATION

The data used to evaluate the central system and the two applications (sections 3.4 and 3.5) came from several measurement campaigns performed in two cities: Rennes and Issy-les-Moulineaux. Figure 20 shows the covered areas. The color scale represents the number of traces at a given place, i.e., the number of times a volunteer walked in that area. In Rennes, we collected measurements in the city center, characterized by narrow streets and a mixture of mid-rise buildings with commercial activities as well as apartments. In Issy-les-Moulineaux the streets are wider, the building are taller, and are mostly used as offices.

3.3.1 *Dataset Collection*

Between February 2015 and February 2016, twelve volunteers walked, at different times, in the streets of the above mentioned areas. Each volunteer carried an MS running the previously mentioned Wi2Me application [16], which supports active and passive modes. In the passive mode, the application logs all scanning results but it does not issue any. The scannings are triggered by other applications running on the phone. In the active mode, Wi2Me triggers a scanning every three seconds. As a result, with the active mode, a few users can gather as many measurements (scan results) as many regular users running the passive mode. Using the active mode we collected 230 scanning traces, each trace with an average duration of 40 min and an average distance covered of 3.10 km. In total, the volunteers walked more than 700 km during more than 150 man-hour.

To estimate the amount of data that a regular user would collect, we also asked ten users to run Wi2Me in passive mode on their phones



(a) Rennes city center. (b) Issy-les-Moulineaux.

Figure 20: Areas covered during the empirical evaluation of WMSP.

for a few days. The subset represents 24 man-days, it includes a total of 36 479 scans and 54 672 network observations. Resulting in an average of 22 819 network observations per user per day and 387 unique APs observed per user per day. The average size of the daily measurement user is 500 KiB.

Each scan result includes the SSID, the BSSID, the RSSI, and the channel of each discovered AP and the MS location.

3.3.2 *Dataset Description*

Figure 20 presents the two areas studied. This figure shows the number of traces covering each cell, i.e., the number of times a volunteer visited a cell. As shown in the figure, we visited the same cells several times.

We collected multiple traces in the same areas, to make sure that we had a reasonably large number of measurements of the same APs. The amount of data needed to produce useful results depends on the specific application. For example, the optimal scanning parameter application is not sensitive to missing measurements on a single street, while the minimal set application requires a complete network view. As a future work, we plan to study how much data is needed by each application and how to decide when the system has enough data to produce reliable results.

Overall we registered 88 608 scans, involving 33 650 APs. As shown in Figure 21, we observed 50 % of the APs 13 times or more, moreover, we observed 30 % of the APs 100 times or more.

Each network measurement is treated following the pre-processing described in Section 3.2.5. The pre-processing takes advantage of the Spark streaming mode to update the cell information in the database as new metrics arrive. The database contains the raw dataset complemented with information obtained after the pre-processing.

The purpose of our dataset is not to represent the amount of data that WMSP would handle in a production setting, but to produce a set

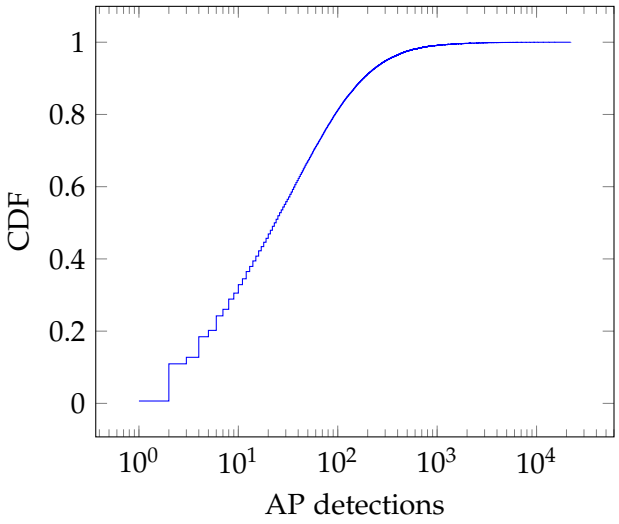


Figure 21: Distribution of the AP count during the empirical evaluation of WMSP.

of results based on real measurements, covering large enough areas, and containing enough repeated measurements of the same environment. These measurements are equivalent to what many users would have collected in the same area. We came back to the collection area and empirically verified the validity and quality of the results provided by the two WMSP use case: the minimal AP set (section 3.4) and the second one computes the optimal parameters for the scanning process (section 3.5).

3.3.3 *Wireless Measurements Sharing Platform Implementation*

Among the available parallel computing frameworks, we selected Spark (version 1.2.2) due to its support for batch and streaming computations. As shown in Figure 22, the preprocessing phase needs the streaming mode while some applications are better suited to the batch mode.

We used the Apache Cassandra (version 2.2.3) database because it is well-supported by Spark, thanks to a specific connector, and because Cassandra nodes can be co-located with Spark nodes on the same physical (or virtual) machines. The Spark master can take advantage of data locality on each node when scheduling and assigning data to each Spark worker process. As shown in Figure 22, the database contains three tables with the list of cells and the corresponding samples and AP.

We deployed WMSP, including both use cases, on the Grid'5000[3] infrastructure. All the computers involved in the experiments have the

3 https://www.grid5000.fr

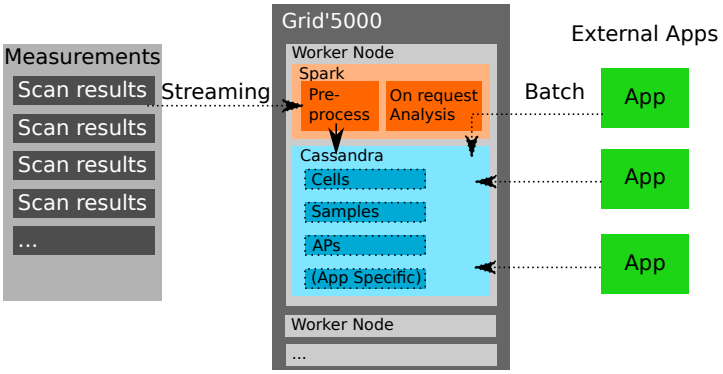


Figure 22: WMSP software architecture.

following characteristics: 2.40 GHz 8-core Intel Xeon E52630v3 CPU and 128 MiB of RAM. Each node used Debian GNU/Linux 7.60.

We configured Spark and Cassandra side-by-side in all computers. The Spark architecture has two types of nodes, one *master node* and several *worker nodes*. In our configuration, one computer performed both roles, while the rest operated as *worker nodes*. Because of its peer-to-peer model, all computers hosted one Cassandra peer.

3.4 REDUCING ENERGY CONSUMPTION

This section presents one WMSP application, the minimal AP set application, that takes the coverage information related to the APs operating in a given area and returns the subset of those APs that are sufficient to cover that area. An operator can exploit this information to save energy by turning off APs that are not in the subset, without reducing the coverage. In this initial approach, we only focus on the coverage and leave other constrains for further work, such as quality of service (QoS). This problem is an instance of the well-known set cover problem, which is NP-complete.

3.4.1 *Minimal Access Point Set*

Recall that AP information is associated to segments, i.e., streets. The analysis is divided into two different phases: processing segment intersections and processing the segments themselves. Figure 23 shows the two-phase implementation in Spark. The first phase determines the street intersections and selects one AP per intersection to be part of the minimal AP set. The selected AP is the one that covers the largest area around each intersection. In the second phase, the algorithm checks for partially covered segments. Each partially covered segment is processed following the algorithm 1, previously proposed by Shehadeh et al. [68]. We found that between 4.25 % and 10.91 % of APs are sufficient to provide the same coverage.

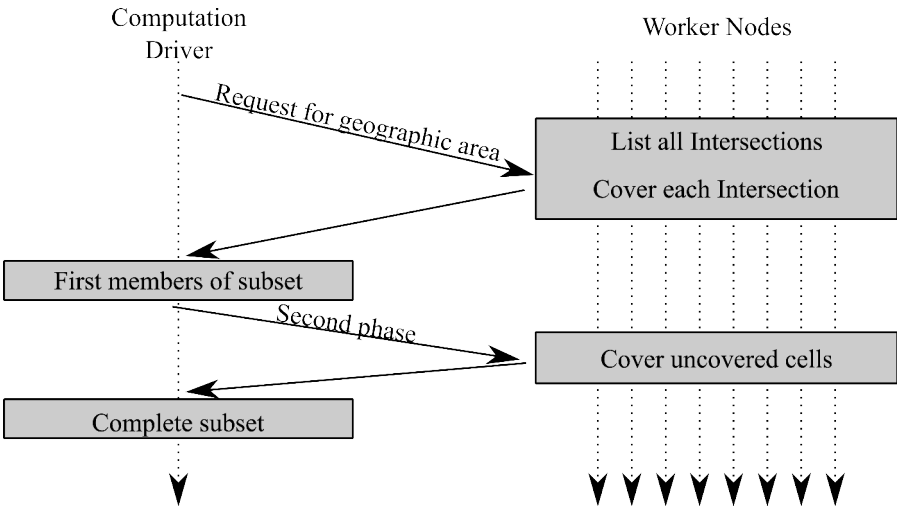


Figure 23: Parallel computation of a minimal AP set.

One significant difference with Shehadeh et al. [68] is that we do not consider a single path but rather a collection of streets (segments) intersecting each other. In other words, while previously we considered a one-dimensional problem we now consider a two-dimensional problem. One advantage of this approach is that the algorithm can now process multiple segments in parallel.

Assuming that the distance between intersections exceeds the Wi-Fi coverage, intersections can be processed in a parallel and independent manner. The same argument applies for the segments.

Algorithm 1 Minimal AP set in a segment.

```
1:  procedure MINIMAL(segment)
2:      APs ← APs in a segment
3:      calculate the APs coverage
4:      minimal_set ← {}
5:      while cells not marked do
6:          AP ← AP with the largest coverage
7:          minimal_set ← minimal_set + AP
8:          APs ← APs - AP
9:          mark the cells covered by AP
10:     end while
11:     return minimal_set
12: end procedure
```

We have considered the following three cases for the input of the minimal AP set algorithm:

Multi-trace: the regular coverage approach presented in algorithm 1, using all the available traces.

Single-trace: instead of using all available traces and collected by several users, the algorithm take a single trace as input. With this

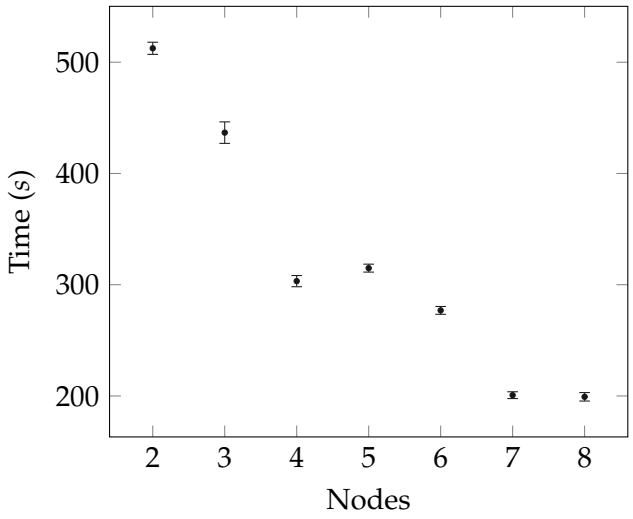


Figure 24: Scalability of the minimal AP set algorithm in WMSP on Grid'5000.

variation we emulate the use case for an area that is visited by only one MS.

Multi-channel: compute the multi-trace algorithm for the three non-overlapping channels 1, 6 and 11, then combine the results. As a result, the minimal set will provide a controlled coverage redundancy.

3.4.2 *Performance Evaluation of the Minimal Access Point Set Implementation*

We evaluate the scalability of our implementation by computing the minimal AP set for both regions: Rennes and Issy-les-Moulineaux. Figure 24 shows the execution time as a function of the number of computers involved in the computation. We observe that by increasing the computers from two to three reduces the computing time by almost 20 %, while increasing from two to eight computers implies a 60 %-reduction.

3.4.3 *Empirical Evaluation*

We evaluated the results of the minimal AP set by going back into the same streets, and monitoring the selected AP availability in the identified regions. In particular, we were interested in the resulting coverage and the impact in MS handover.

Ideally, we could verify the coverage by connecting the MS to the different APs in the minimal set. Unfortunately, this is not possible in our tests because of the presence of several private networks. Therefore, we emulated the network association by continuously monitoring the AP RSSI and availability. Algorithm 2 describes the emulation

Algorithm 2 Minimal AP set evaluation.

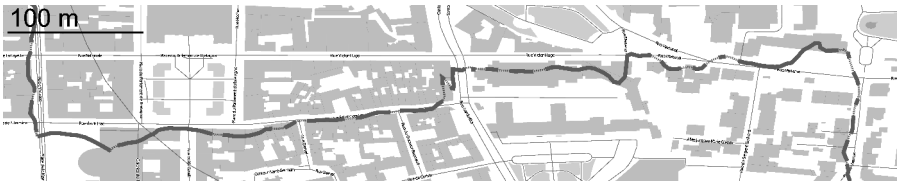
```
1:  procedure testApSet(apSet)
2:      while True do
3:          discovered ← triggerScan()
4:          available ← discovered ∩ apSet
5:          AP_x ← AP with best RSSI in available
6:          fails = 0
7:          while AP_x.RSSI > −90 dBm AND fails ≤ 5 do
8:              Assume association with AP_x
9:              Probe AP_x
10:             if no Response then
11:                 fails = fails + 1
12:             else
13:                 fails = 0
14:             end if
15:         end while
16:     end while
17: end procedure
```

methodology executed in an Android phone. The emulation begins with the execution of the default Android scanning, followed by the selection of the available AP with the higher RSSI that is present in the minimal AP set. Then, the MS sends periodic Probe Request[4] frames to that AP to check its presence. When a Probe Response is not received, or if the RSSI of the received Probe Response is below −90 dBm for five consecutive times, the whole process is repeated from the beginning, starting with the default Android scanning procedure.

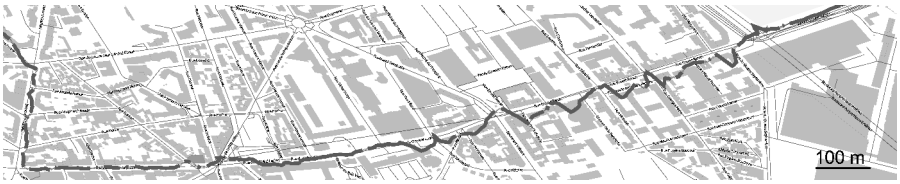
The maps in Figure 25 show the areas where we performed the empirical validation. We covered the paths in Rennes and Issy-les-Moulineaux 47 and 34 times, respectively. Each map depicts the results of a single experiment (i.e., walk) using the multi-trace minimal AP set. Solid lines correspond to locations where the MS was able to detect an AP belonging to the minimal set with an RSSI above the threshold. Dotted lines correspond to handovers, i.e., the area in which the MS was switching from one AP to another.

We evaluate the coverage of the subsets generated by the three algorithms mentioned above (Multi-trace, Single-trace, Multi-channel). As a baseline case, we consider the case where the MS is free to pick any AP without using the WMSP information. This case is called *unrestricted*. Figure 26 and Table 1 present the comparison of the AP availability time, i.e., the time the MS was connected to a given AP. In addition, Table 1 shows the average handover time.

4 APs should reply to any Probe Request frame with a Probe Response frame [36]

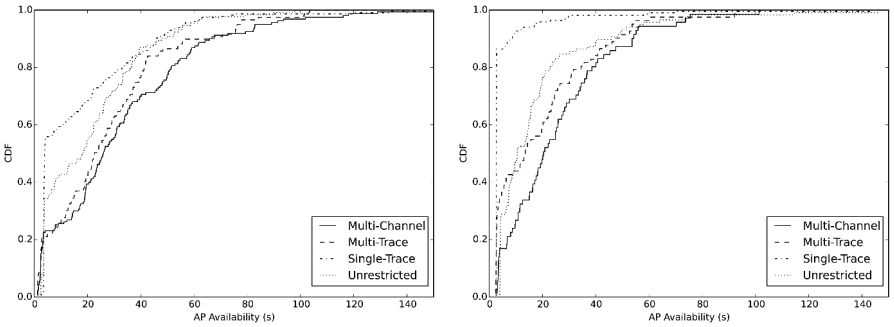


(a) Rennes city center.



(b) Issy-les-Moulineaux.

Figure 25: Minimal AP set evaluation trajectory.



(a) Rennes. (b) Issy-les-Moulineaux.

Figure 26: Availability of the APs in the minimal AP set.

We can make two main observations based on these results. First, we can see, in Figure 26, the importance of using multiple measurements, which is possible thanks to WMSP, by comparing the single-trace case with multi-trace or multi-channel. When using a single trace to compute the minimal AP set more than half of the APs have an availability of 4 s or less, compared to 20 s in the multi-trace case. This shows the importance of using multiple measurements in order to learn the topology, and thus to compute an accurate AP set. Second, we can see, in Table 1, that using the minimal AP set in the multi-trace and multi-channel cases gives results close to the unrestricted case. We conclude that the objective of having the same coverage with fewer APs is reached. In addition, we can actually see that multi-trace and multi-channel cases are even slightly better than the unrestricted case, because the minimal AP set is made of the best available APs in a given area. Such an approach could be further used for AP selection.

Table 1: AP availability and handover time for the four variations of the minimal AP set.

	Rennes				Issy-les-Moulineaux			
	Availability (s)		**Handover** (s)		**Availability** (s)		**Handover** (s)	
	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
Unrestricted	24.40	27.43	8.03	0.48	20.69	32.27	5.48	2.44
Single-trace	16.16	26.14	8.44	7.13	4.00	7.77	7.15	1.42
Multi-trace	26.76	30.03	7.83	0.14	21.50	31.77	7.80	0.43
Multi-channel	25.06	27.97	5.61	0.25	16.68	20.42	5.16	0.40

3.4.4 *Subset Throughput Evaluation Through Simulation*

Having proven through emulation that an AP subset computed offline can actually be observed in the field, we need to explore its viability in terms of throughput. As we do not have access to a collection of APs covering at least a few city blocks, we relied on simulations to study this aspect. The Network Simulator 3 (NS-3) provides the IEEE 802.11 functionalities required to reproduce both our scanning campaigns and user traffic.

The simulations use the synthetic topology shown in Figure 27, representing an urban area. Table 2 contains the parameters of the simulation. It is a square of 500 m by 500 m and contains 800 APs. This topology has the same statistical properties (in terms of street width and length, building dimensions, and AP density) as those of the city center of Rennes, where we have collected traces. We have generated a synthetic topology because it is not currently possible to use arbitrary maps and building shapes in NS-3. Furthermore, we do not know where the APs, which we detected in our traces, are actually located. We have verified that the simulations do reproduce metrics that we were able to measure from our traces, such as the CDF of the RSSI and the CDF of the AP availability (i.e., number of AP answering a probe request).

First, we simulated a MS collecting scan results while moving on each street, which is exactly what we did in order to collect the traces. Using these scan results, we computed the AP subsets using the algorithms presented in Section 3.4.1, in order to determine the throughput achievable when using only the APs in the given subset. In order to generate enough traffic over a limited area, we placed 49 MSs on seven by seven square grid whose points are 50 cm apart. The grid is centered around a random point on one of the streets. Each MS selects the member of the subset with the best RSSI as an AP and starts a UDP flow with a computer connected to the AP that sends data at

Table 2: Characteristics of the city used to simulate the AP's deployment.

Area	$500\,\text{m} \times 500\,\text{m}$
AP's density	$0.25\,\text{APs}/\text{m}^2$
Street intersection density	$3.04 \times 10^{-4}\,\text{intersections}/\text{m}^2$
Average wall length	13.91 m
Average distance between street and buildings	6.70 m
Average street length	59.59 m
Building density	$8.32 \times 10^{-4}\,\text{buildings}/\text{m}^2$

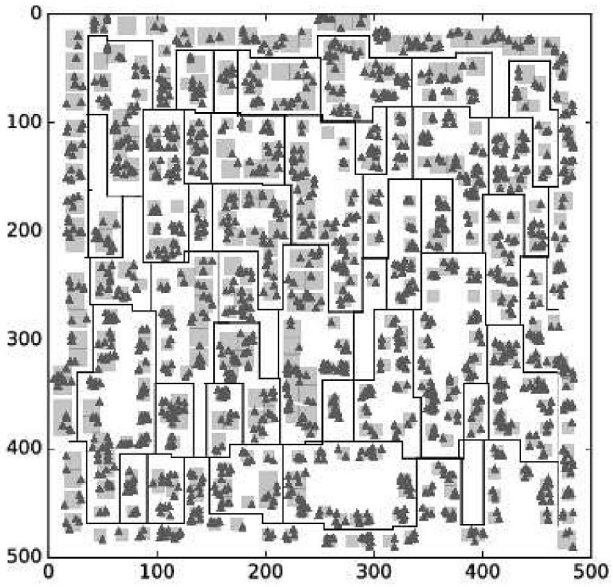


Figure 27: Simulated environment used for the evaluation of the minimal AP set.

200 kB/s. Figure 28 presents the CDF of the throughputs achieved by the stations while using two different subsets. The Multi-Channel and Unrestricted setups provide the users with similar data rates, showing that the Multi-Channel is an efficient way of reducing the number of access point while keeping the same coverage and providing an overall throughput close to the one experienced in an unmodified deployment.

3.4.5 *Minimal Access Point Set Lifetime*

In this section we evaluate the changes in the minimal AP set over time, that is, the effects of APs moved or turned off. These changes may potentially degrade the minimal AP set over time. To characterize this degradation, we performed regular measurement campaigns

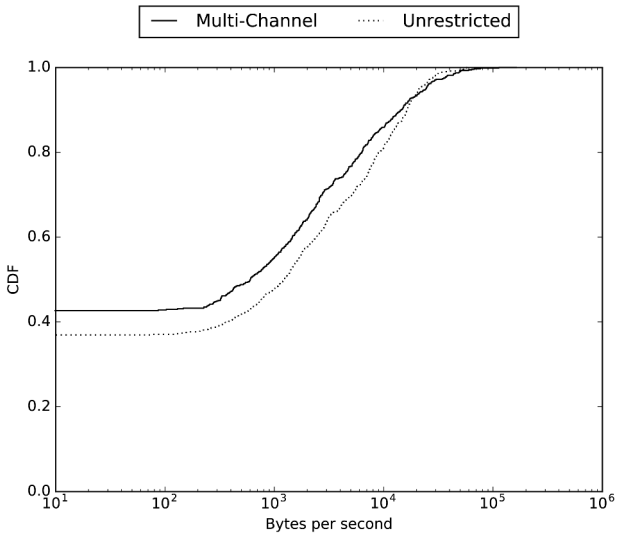


Figure 28: MS throughput distribution in the simulated environment after executing the minimal AP set algorithm.

in the same streets of the Rennes city center, over the course of 12 months. Applying the geo-localisation process presented in Figure 18, we determined the cells that were visited in at least 10 measurement campaigns and analyzed how an AP subset computed for these cells evolved over time. This process consists in iterating chronologically over our traces, when a new location is covered by a trace, its measurement data is used to compute a collection of APs to cover these new cells. Otherwise, we compare the observed APs with our previously compiled subset. Our subset is either still valid or we need to update it based on the latest information.

By processing these databases chronologically, we were able to determine the time validity of the minimal AP set for a particular location. Table 3 shows the duration between the moment a cell was assigned a covering AP, and the moment it needed to be replaced. This tells us that around 54.60 % of the cells' AP population remained stable for about 13 months. Alongside this reasonably stable cells, the lower 40.70 % of them require a replacement in less than one month. Averaged on the total duration of this experiment, 0.09 % of the covered distance would need to be replaced every week. On the region such as the one in Figure 25a, this accounts for around 140 m.

3.5 REDUCED SCANNING TIMERS

As a second use case for WMSP, we consider the problem of determining the parameters of the IEEE 802.11 network discovery procedure (see chapter 2) that satisfy specific latency constraints. Following the work presented by Montavont et al. [54], we propose to use network topology statistics to determine the best scanning parameters

Table 3: Time validity of the minimal AP set.

Validity	Cells (%)
Less than one week	22.00
Between one week and one month	18.70
Between one month and one year	4.70
More than one year	54.60

and, for the first time, we present an empirical validation. Thanks to WMSP, several users could share their partial view of the network topology, so that the WMSP application can compute up-to-date and customized efficient scanning parameters.

The use case is the following: when a MS is moving around, it may lose its radio link with its serving AP, thus triggering a handover. In a dense urban environment, we evaluated that a handover is needed every 22 s on average for pedestrian users (see table 1). In such an environment, handovers are, therefore, a common occurrence. As an MS can not exchange data frames during the handover, it is important to make this process efficient in order to minimize network interruptions on running applications. The most time-consuming phase of the handover is the discovery phase, where the MS look for candidate APs. Focusing on the 2.40 GHz band, the MS may scan 13 channels one after the other to discover the available APs.

As detailed in chapter 2, MS usually perform the network discovery following an active scanning, that is, the MS sends one or more Probe Request frames on a channel and waits for Probe Response frames sent by APs. The scanning performance is a trade-off between the latency (the time it takes to discover the available APs and select one), the discovery rate (how many of the available APs are actually discovered) and the failure rate (the probability of not finding any AP after a complete scan). Intuitively, the shorter the scanning the better, but if the time spent on each channel is too short, the MS may miss discovering relevant APs. The parameters that influence the scanning performance are the channel sequence and the timers which define the time spent on each channel waiting for AP responses. In particular, minimal channel time (MinCT) defines the time to wait before receiving the first response, and maximum channel time (MaxCT) the additional time to receive further responses. That is, an MS will use MaxCT if and only if it receives at least one Probe Response during MinCT.

Table 4: Optimal scanning parameters that warrant 10 ms, 20 ms, 50 ms and 100 ms delay.

Latency (ms)	**Channels to probe**	**Timers** (<channel, MinCT, MaxCT>)
10	1	<11,7,1>
20	2	<11,10,1> <6,4,2>
50	3	<11,15,2> <6,13,1> <1,12,4>
100	4	<11,15,10> <6,14,12> <1,14,9> <5,14,12>

3.5.1 *Optimal Scanning Parameters*

We use the network deployment statistics available through WMSP to determine the best scanning parameters. Based on these statistics, the application computes the best parameters for different latency profiles and re-distributes these profiles to the MSs, which can use them to optimize their network discovery process.

By modeling the problem of selecting the scanning parameters as a multi-objective function [54], it is possible to us a genetic algorithm to approximate the Pareto-optimal front. In this approach, an individual is a specific combination of scanning parameters (MinCT, MaxCT, and the list of channels to probe). We used traces collected in the city of Rennes to compute the fitness of each individual by computing the scan results that would have been obtained by a phone using those parameters under the same circumstances.

Table 4 presents four scanning configurations, each row corresponds to a specific latency (10 ms, 20 ms, 50 ms and 100 ms). For instance, to keep the scanning latency below 50 ms the MS should: probe channels 11, 6 and 1, using 15, 13 and 12 as the MinCT for those three channels, and 2, 1 and 4 as MaxCT.

3.5.2 *Empirical Evaluation*

In order to evaluate this solution, we configured an MS with the recommended scanning parameters, and we measured the performance by using it in the city of Rennes. We performed 24 600 scans in total, at 18 different locations, with at least 1300 scans for each tested algorithm. Note that we performed this experiment a few months after the original dataset was collected. We configured the MS to scan periodically, and we measured the scanning latency, the discovery rate and the failure probability. Table 5 compares the measured scanning performance with the expected performance. We can see that the scanning latency values are close, the difference between expected and actual values is less than 5 %. The observed failure probability is also

Table 5: Comparison of the expected and actual scanning performance.

Latency (ms)	**Channels to probe**	**Latency** (ms)		**Failure** (%)		**Discovery** (%)	
		comp.	obs.	comp.	obs.	comp.	obs.
10	1	10.80	11.60	23.80	16.70	18.70	40.80
20	2	22.00	23.70	7.90	13.10	30.60	33.70
50	3	54.60	57.60	0.50	0.80	65.70	60.10
100	4	98.80	107.40	0.20	0.30	82.10	70.00

close to the expected failure probability for a scanning latency greater or equal to 50 ms.

For scanning latencies of 10 and 20 ms, we can see that there are more scanning instances that fail after scanning all recommended channels compared to what was expected. These scanning latencies are extremely low, and the proposed scanning parameters suggest scanning only one channel (for the 10 ms latency) or two channels (for the 20 ms latency). Thus, if one Probe Request is lost the MS might end up finding no APs. This explains why there is a higher failure probability for these two configurations.

The discovery rate shows the fraction of the available APs that is actually discovered, compared to the total APs operating in the MS range. The difference between the observed metric and the one computed from the statistics is small, except the scanning at 100 ms which reaches 69 % instead of 82 %. However, determining the total number of available APs is non-trivial due to the fact that the more an MS scans, the more APs it finds, even for large MinCT and MaxCT (see section 2.3).

Figure 29 shows a comparison between seven scanning parameter combinations for a scanning latency around 50 ms, except one that is considered as a baseline. In the latter, we scan all thirteen channels for 30 ms each in order to estimate the total number of APs and compute the discovery ratio. The second setting, shown in the figure, is the one given by WMSP while the five others are scanning 1, 2, 3, 4 and 13 channels by dividing the time budget (50 ms) by the number of channels to scan. Note that we also used WMSP to select which channel to scan, by choosing the most populated channels first (in the order 11, 6, 1 and 5).

We can observe that the latency for scanning all the 13 channels is still above the targeted 50 ms because of the important overhead incurred when switching the operating channel. This notwithstanding, we can see that the recommended scanning parameters offer the best trade-off among the different tested settings.

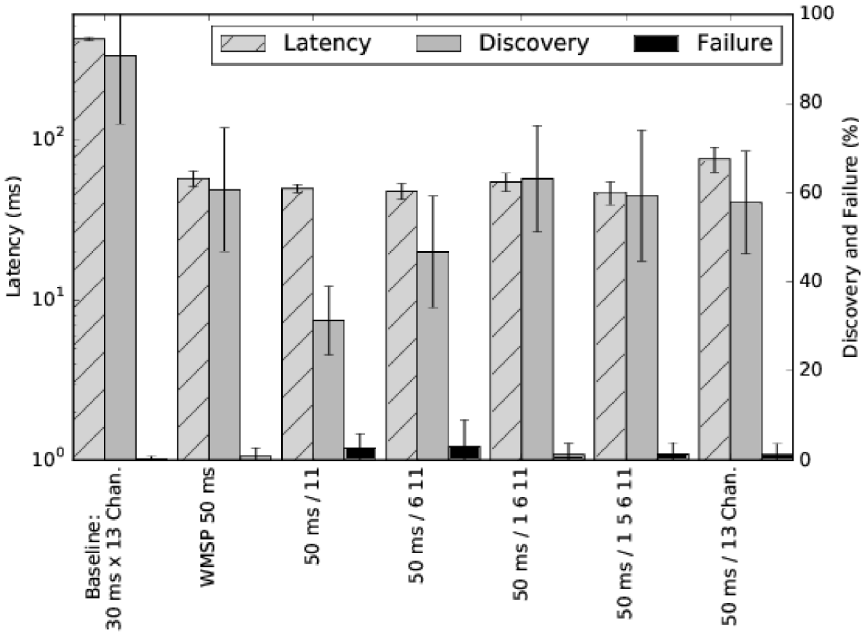


Figure 29: Comparison of the performance of different scanning configuration.

3.5.3 *Performance Evaluation of the Optimal Scanning Parameters*

We evaluated the scalability of the implementation following the same approach used for the minimal AP set implementation. Figure 30 shows the evolution of the execution time depending on the number of nodes involved in the computation. We observed that doubling the number of nodes from two to four reduce the computation time by almost a half. Moreover, an increase from two to eight nodes represents a relative time-reduction of 80 %. All this confirms that Big Data technologies can indeed be useful and efficient for these types of computations.

3.5.4 *Assisted Scanning*

In this section, we sketch the architecture for the assisted scanning, that is to say, the interaction between the previously described WMSP application and the MSs. The main elements are: the WMSP application running as a service in the front-end and the client application running in the MSs.

Client applications obtain the scanning profiles using push and pull approaches. An MS entering an unknown location will contact the WMSP application and pull the parameters corresponding to the area. These parameters are pre-computed, meaning that the time needed to retrieve them is limited by the time needed to perform the network connection plus the time to consult the database. Also, the

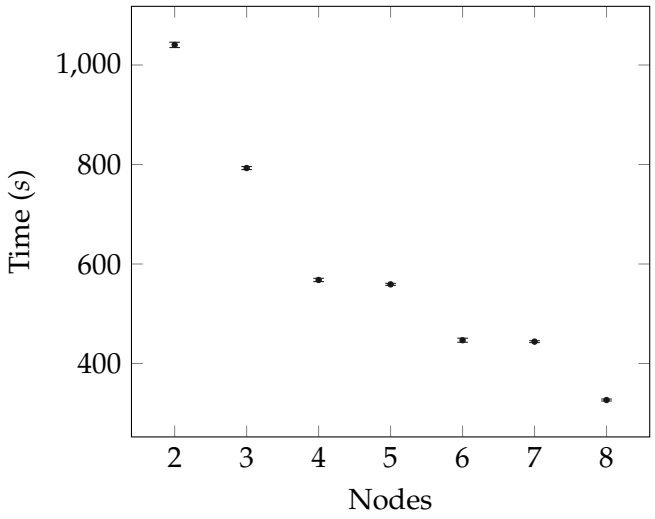


Figure 30: Scalability of the optimal scanning parameters algorithm in WMSP on Grid'5000.

client can request a subscription to a given location, in this case, each time that WMSP updates the scanning parameters the MS receives the up-to-date profiles, i.e., WMSP pushes the profiles to the subscribed MS.

The client application includes a cache system that stores the profiles related to the recent locations, plus the profiles related to the subscribed locations. This cache ensures access to profiles even if there is no network access. Profiles are updated once the network connection is established. As a fail-back, i.e., when no profile is available for a specific location, the default scanning procedure should be performed. Algorithm 3 outlines the assisted scanning procedure.

Algorithm 3 Assisted scanning.

```
1:  procedure ASSISTEDSCAN(location, latency)
2:      networks ← {}
3:      params ← consultCache(location, latency)
4:      if parameters != NULL then
5:          configure params
6:      else
7:          configure default params
8:      end if
9:      networks ← triggerScan()
10:     return networks
11: end procedure
```

3.6 CONCLUDING REMARKS

In this chapter we have presented WMSP, a cloud-based system to collect network measurements taken by existing MS and APs, and to process these data to generate a better understanding of the existing network environment. We have shown that existing hardware can be used to gather network statistics, and that it is possible to effectively combine measurements from different users to have an accurate and up-to-date view of the network environment. The two applications developed and tested showed the relevance and scalability of the system.

The approach we have presented in this chapter relies on simple network measurements passively collected. Stations (STAs) (MSs and APs) may combine the information stored in WMSP with information particular to each STA to get a more accurate description of the current network status. Chapter 4 describes non-intrusive metrics and strategies to characterize the channel quality and to forecast the link quality with the neighboring networks. Those metrics may also be collected collectively and shared via WMSP.

The deployment on the Grid'5000 platform demonstrated the performance gain achievable through the parallelization of data sample processing. This experimental set up also allowed us to identify aspects that can be improved. For example, WMSP uses the BSSID identify Wi-Fi networks, however, we have observed that some operators reuse the BSSID. The larger the area, the larger the chances of having duplicated BSSIDs. Thus, to address large areas and for longer periods of time, we need a method to uniquely name one Wi-Fi network. Another aspect that requires further study is the security. We have not considered any method to identify users and to authorize the access to WMSP. In our proposal we have taken a naive approach, we assume that all participants and measurements are reliable.

4

EVALUATING THE QUALITY OF WI-FI NETWORKS

Many studies (see, for instance [2, 64, 85]) and projects (e.g., WiGle, OpenSignal, Sensorly) have shown that, especially in urban areas, several access points (APs) can be detected at any given location. The results presented in Chapter 2 confirm that a single mobile station (MS) can discover 75 APs in a single spot.

Due to the unregulated and unplanned nature of Wi-Fi networks, APs in close proximity of each other often operate on the same channel, especially on the frequently used non-overlapping channels (1, 6 and 11 for the 2.40 GHz band). This can result in poor performance, in particular when the traffic demand exceeds the channel capacity, resulting in a *saturated channel* (i.e., where any increase in the offered load does not result in an increase of the aggregate throughput). In this case, users are often better off joining a different network, for instance another AP operating on a different, and non-saturated channel, or a network using a different access technology (e.g., cellular network, wired network).

As it is common for users to have access to multiple APs, it is important to choose the "best" AP. Although there can be different definitions of the preferred AP, depending on the specific circumstances, an AP operating in a saturated channel is definitely a non-ideal candidate. Thus, one should take into account channel saturation when selecting an AP.

In this chapter we propose a simple method for stations (STAs) (both, APs and MSs) to detect a saturated channel by passively monitoring Beacon messages, which are available to all STAs as part of the IEEE 802.11 procedures. This enables STAs to passively collect information and to determine whether a channel is saturated or not. Using Beacons to characterize the channel condition presents multiple advantages: Beacons are always present, transmitted in broadcast mode, and use the strongest modulation and coding scheme (MCS). By analyzing experiments conducted with different traffic loads, we show that it is possible to identify whether a channel is saturated based on the distribution of the Beacon jitter. Even though APs send Beacon frames periodically, they do have to wait for the channel to be idle, resulting in an additional delay that depends on the traffic intensity. Empirical results show that the Beacon jitter follows a similar distribution whenever the channel is saturated. Our solution exploits this by comparing the Beacon jitter distribution with a reference distribution, corresponding to a saturated channel. While the literature on AP selection and Wi-Fi network characterization is vast (see, for

instance, [7, 23, 34, 35, 41, 43, 45, 59, 80, 82, 86, 89]), to the best of our knowledge, no existing solution is both (1) implementable without changing the APs and (2) passive, i.e., it does not require exchanging any additional frames.

The chapter is organized as follows: Section 4.1 gives an overview of IEEE 802.11; Section 4.2 reviews popular metrics to characterize Wi-Fi networks and the relevant literature; Section 4.3 defines *Beacon jitter*; describes the experimental setup and presents the results; Section 4.4 discusses the relation between Beacon jitter and channel load; Section 4.5 presents a proposal for detecting saturated channels and discusses its performance; Section 4.6 presents concluding remarks and discusses possible extensions.

4.1 IEEE 802.11 OVERVIEW

This section presents a study of the channel saturation and its consequences on the Beacons inter-arrival time. The study relies on three main elements of the IEEE 802.11 networks: Medium Access Control, Beacon transmission and channel saturation. In the remainder of the section we briefly describe these three elements.

4.1.1 *Medium Access Control*

CSMA/CA is the fundamental medium access control layer (MAC) mechanism of the IEEE 802.11 networks [36]. Whenever a device has a frame to send, it must first sense the channel to determine if there is an ongoing transmission. If the medium is busy, the device must defer the transmission and perform a random backoff. A device must decrement the backoff counter only while the medium is idle.

4.1.2 *Beacon Transmission*

In an infrastructure basic service set (BSS) the AP must periodically broadcast Beacon frames. APs must schedule Beacons for transmission every Beacon interval, which is a configurable parameter. Beacon frames include the Beacon interval. Like other frames, a Beacon transmission follows the MAC procedure, meaning that the actual Beacon interval, Δ, and the nominal Beacon interval, T, might differ. Moreover, the actual Beacon interval may also vary over time according to the fluctuations of the channel conditions and of the offered load. APs must schedule subsequent Beacons at the undelayed nominal interval [36].

Figure 31 illustrates two possible cases: in the first scenario the medium is idle and the nominal and the actual Beacon interval are the same. In the second scenario, with a busy channel, Beacons are

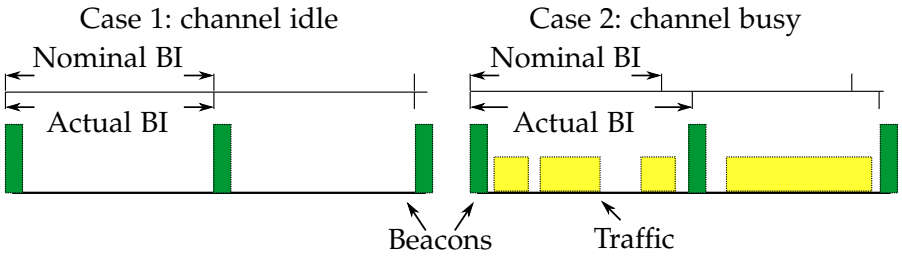


Figure 31: Beacons transmitted by one AP in an idle (left) and busy (right) channel.

delayed due to frames transmitted by other devices, resulting in a difference between the nominal and the actual Beacon interval.

4.1.3 *Channel Saturation*

It is well-known that in the random access CSMA/CA MAC protocol, as the offered load increases, the aggregate throughput (i.e., the sum of the throughput of all the transmitters in the same collision domain) increases until the offered load reaches its maximum stable value, often called the *saturation throughput* [7]. Further increases in the offered load result in the same (or lower) aggregate throughput. In this dissertation, we use the term *saturated channel* to indicate that the offered load is greater than or equal to the saturation throughput.

4.2 METRICS FOR CHARACTERIZING WI-FI NETWORKS

Over the last couple of decades, since the appearance of Wi-Fi networks, researchers have proposed different metrics to characterize these networks. In this section, we review the literature most closely related to our work.

One metric is channel delay, which measures the time needed to send a frame. Haghani et al. [31] estimate the distribution of the access delay, from the moment the MAC frame is ready for transmission (i.e., the first in the transmission queue) until the corresponding acknowledgment is received. For this, Haghani et al. [31] developed a theoretical model that takes as input the busy/idle periods in the channel, as this factor affects the channel delay. Kajita et al. [41] estimate the channel delay by using a machine learning approach, where the input parameters come from traffic around the AP. This implies that the AP can decode all frames in the neighborhood, which is not always possible, especially those frames encoded using high MCS. The higher the MCS, the more difficult is for a receiver to correctly decode the frame. This limitation became particularly relevant with recent standard amendments (e.g., IEEE 802.11n, IEEE 802.11ac),

which allow APs to use beamforming, multiple input/multiple output (MIMO) and multi user MIMO (MU-MIMO) to enhance the link quality with the receiver MS, reducing the chance that other STAs would be able to decode those frames.

Achievable throughput is another widely studied metric. Following Bianchi's seminal paper [7], several authors [34, 43] have proposed extensions and refinements based on this model to estimate the achievable throughput. They all take the collision probability as an input parameter. It is non-trivial to estimate this parameter, as it depends on several independent factors, such as the number of active transmitters, offered load, and collision domains (i.e., which nodes are within carrier sensing range) just to name a few. These factors cannot be precisely known by a single node. A common approach is to approximate the collision probability with the *retransmission ratio* [34, 43], sometimes called loss probability [80]. The STA sniffs the medium, counting the number of transmitted frames and the corresponding retransmissions. This approach is appropriate if the STA is able to capture all frames in the medium and all retransmissions are due to collisions. However, these assumptions do not always hold. As previously mentioned, the STA may not be able to decode all frames. In addition, some retransmissions may be caused by bad channel conditions and not by collisions. Hong et al. [35] use a collaborative approach to estimate the available throughput in which the MS requests information from the APs, including channel utilization and number of frames sent. Therefore, the estimation depends on the information collected by the AP and the willingness of the AP to share that information.

While these metrics are related to channel saturation, the relationship is not always straightforward. Our approach is to use the distribution of the Beacon jitter to identify saturated channels. This has multiple advantages: (1) STAs can perform the measurements independently, without requiring support from other STAs; (2) Beacons are transmitted periodically by all APs, i.e., they are always present; (3) Beacons are broadcast, therefore APs do not use beamforming, MIMO or MU-MIMO; (4) Beacons are encoded using the strongest MCS, so that they are more easily decoded.

4.3 BEACON JITTER

Beacon jitter, ρ, is defined as the difference between the nominal, T, and the actual, Δ, Beacon interval:

$$\rho_i = \Delta_i - T. \tag{2}$$

Table 6: Symbols used in the chapter

Symbol	Description
s	Slot time (µs)
t_i	Timestamp on the Beacon. Corresponds to the time at which the AP sent the $i-th$ Beacon, i.e., the time at which the $i-th$ Beacon reached the channel (µs)
T	Nominal Beacon interval (µs)
Δ_i	Actual interval between Beacons $i-th$ and $i-1-th$
ρ	Difference between the nominal (T) and the actual Beacon interval (Δ)
N	Number of samples
B	Expected backoff time (µs)
b_i	$i-th$ Beacon
c	Channel
r	Beacon jitter distribution used as reference
d	Kolmogorov-Smirnov (KS) distance between a given distribution and the reference distribution, r
α	Threshold to consider two given distributions as sufficiently close.

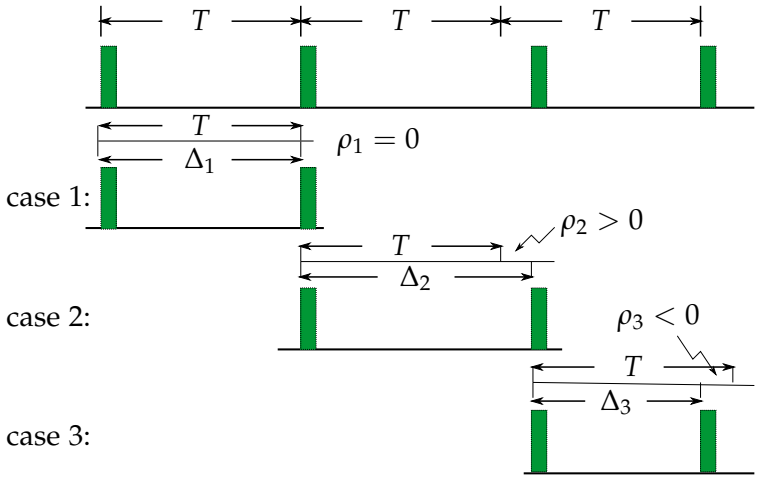


Figure 32: Three possible cases for Beacon jitter: 1) b_i and b_{i-1} on time; 2) b_{i-1} on time and b_i late; 3) b_{i-1} late and b_i on time.

.

The actual Beacon interval is defined as the time elapsed between two consecutive Beacons, b_i and b_{i-1}, where t_i and t_{i-1} represent the timestamp corresponding to the Beacons i and $i-1$, respectively:

$$\Delta_i = t_i - t_{i-1}. \tag{3}$$

We study the distribution of the Beacon jitter distribution as a proxy for detecting a saturated channel. Figure 32 shows the three possible cases for the Beacon jitter:

1. $\rho_i = 0$ indicates no jitter. This occurs when the two Beacons experienced the same delay during the channel access; this usually happens when the channel load is low.
2. $\rho_i > 0$ means that the delay of b_i is larger than the delay of b_{i-1}.
3. $\rho_i < 0$ means that the delay of b_{i-1} is larger than the delay of b_i. This happens because APs must transmit their Beacons at the nominal time, regardless of previously delayed Beacons.

4.3.1 *Experiments*

We conducted two sets of experiments to gather data to analyze the distribution of the Beacon jitter in real IEEE 802.11 networks. We conducted the first set of experiments in a dedicated test bed, where we could control all the input parameters of the experiments. We refer to these experiments as *controlled condition experiments* in the reminder of this chapter. Using these experiments, we verify that the distribution

of the Beacon jitter varies consistently with the channel load. We carried out the second set of experiments using a different test bed that shared a channel with a production IEEE 802.11 network. We refer to this second set of experiments as *uncontrolled condition experiments*. Tables 7 and 8 summarizes the configuration of the experiments.

4.3.1.1 *Controlled Condition Experiments*

The controlled condition experiments consist of six APs and six MSs. MSs are Linux (Ubuntu 14.04) computers equipped with network interface cards (NICs) BCM4360. APs are Asus RT-AC87U. Two servers are connected via an Ethernet cable to the six APs, three APs per server. All MSs and servers are further connected to an Ethernet control network that is used to coordinate the experiments. Devices are located in two rooms, with three APs, and three MSs in each room (one MS next to its corresponding AP). The receive signal strength indicator (RSSI) detected by each MS was above −70 dBm, ensuring that all MSs are within range of each other. All devices operate using IEEE 802.11ac on channel 40 of the 5 GHz band. During the execution of the experiments there were no other devices using this channel.

Table 7 summarizes the configurations for the experiments. APs transmit UDP traffic at a constant rate to the MSs. We vary the number of transmitter APs (between two and five) and the offered load per transmitter (between 0 Mbit/s and 4000 Mbit/s). In each experiment the offered load is the same for all the active transmitters. Each experiment lasts 60 s and is repeated ten times. In all configurations there is only one MS per AP. One of the six APs is idle to avoid biases due to the hardware load; Beacons used in our analysis come from this AP. One of the six MSs operates in monitoring mode; this MS captures the frames used our study.

4.3.1.2 *Uncontrolled Condition Experiments*

The uncontrolled condition experiments consist of five APs and five MSs in a single room. APs are Alix-2d2, MSs are Linux (Debian 8.7) computers equipped with IEEE 802.11g NICs. One server connects all ten devices via an Ethernet network. All devices operate on channel 1 of the 2.40 GHz band using IEEE 802.11g. An unknown number of other networks and users also operate in the surroundings on the same and adjacent channels.

We use these experiments to evaluate networks under real and uncontrolled conditions, as networks in the vicinity serve regular users. Table 8 summarizes the configurations, with a different number of transmitters (two or four) and a different offered load (between 30 Mbit/s and 60 Mbit/s). As with the experiments in controlled conditions, the transmitters use *iperf* to generate UDP traffic at a constant bit rate. Each experiment lasts 60 s and is repeated ten times. One of

Table 7: Total offered load (e.g., the sum of over all transmitting APs) of the experiments performed under controlled conditions.

Number of Transmitters	Non-Saturated (Mbit/s)	Saturated (Mbit/s)
2	0, 0.2 0.5, 1, 1.5, 2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 80, 100, 120, 160, 200	1600
3	0, 0.3, 0.75, 1.5, 2.25, 3, 6, 9, 12, 15, 30, 45, 60, 75, 90, 120, 150, 180, 240, 300	2400
4	0, 0.4, 1, 2, 3, 4, 8, 12, 16, 20, 40, 60, 80, 100, 120, 160	320, 3200
5	0, 0.5, 1.25, 2.5, 3.75, 5, 10, 15, 20, 25, 50, 75, 100, 125, 150, 200, 400, 4000	4500, 8000, 20000

Table 8: Total offered load (e.g., the sum of over all transmitting APs) of the experiments performed under uncontrolled conditions.

Number of Transmitters	Saturated (Mbit/s)
2	40, 60, 80, 100, 120
4	30, 60, 90, 120, 150, 180

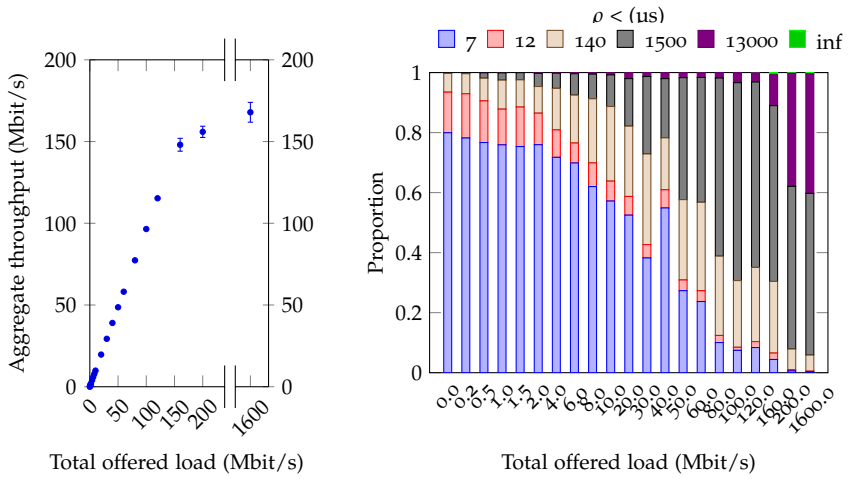
the five APs is idle to avoid biases due to the hardware load; Beacons used in our analysis come from this AP. One of the five MSs operates in monitoring mode; this MS captures the frames used in our study.

In the uncontrolled condition experiments we only have access to our five APs and five MSs. Therefore, we do not know the actual channel load and thus, we only consider experiments under saturated conditions. We have empirically verified this condition by increasing the total offered load until the aggregate throughput of the nodes under our control did not increase any further.

4.3.2 *Experimental Results*

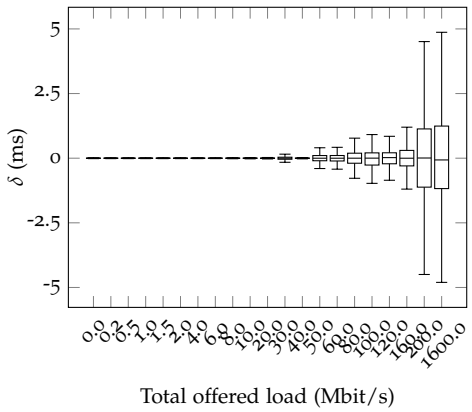
Figures 33 to 36 show the aggregate throughput and the Beacon jitter, ρ, for the controlled condition experiments, for two, three, four, and five transmitting APs, respectively. The Beacon jitter corresponds to the Beacons transmitted by the idle AP and measured by the monitoring MS.

Figures 33a, 34a, 35a, and 36a show the aggregate throughput vs. the total offered load (i.e., the sum over all transmitting APs). To



(a) Aggregate throughput vs. total offered load.

(b) Barplot of Beacon jitter vs. total offered load.

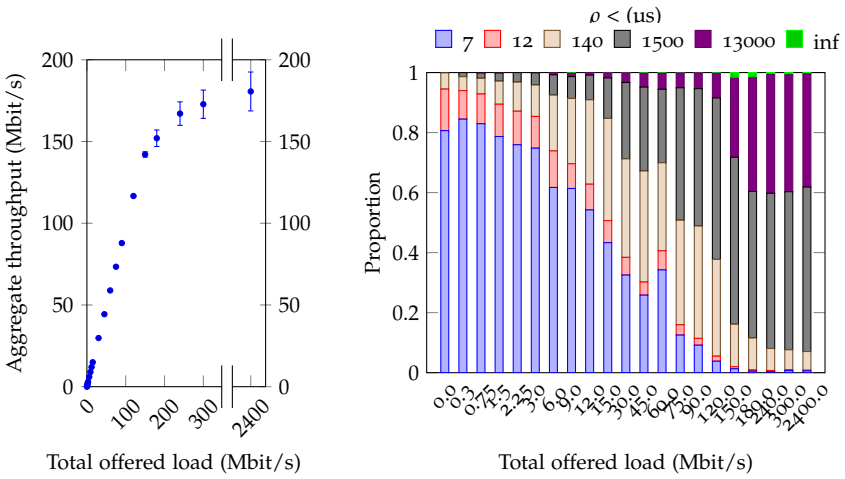


(c) Beacon jitter box plots vs. total offered load.

Figure 33: Throughput and Beacon jitter results from controlled conditions experiments while using two transmitters.

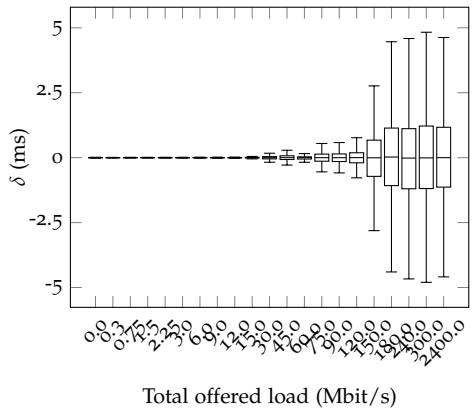
improve the readability of the plots, the x-axis is broken in multiple segments due to the large range spanned by the values of the total offered load. In each case, at first, the aggregate throughput increases linearly as the total offered load increases. It then flattens as the total offered load approaches the saturation throughput.

In the remainder of the dissertation we consider as saturated all the experiments with an offered load larger or equal to the one that gives the largest aggregate throughput. For example, in the case of five transmitters (Figure 36a) the maximum aggregate throughput observed is 161 Mbit/s, corresponding to a total offered load of 4500 Mbit/s. This is, a configuration in which each of the five transmitters have configured *iperf* to generate user datagram protocol (UDP) traffic at 900 Mbit/s. Therefore, we consider as saturated all the experiments with a total offered load larger than this value. Note that the number of transmitters changes the saturation point. The



(a) Aggregate throughput vs. total offered load.

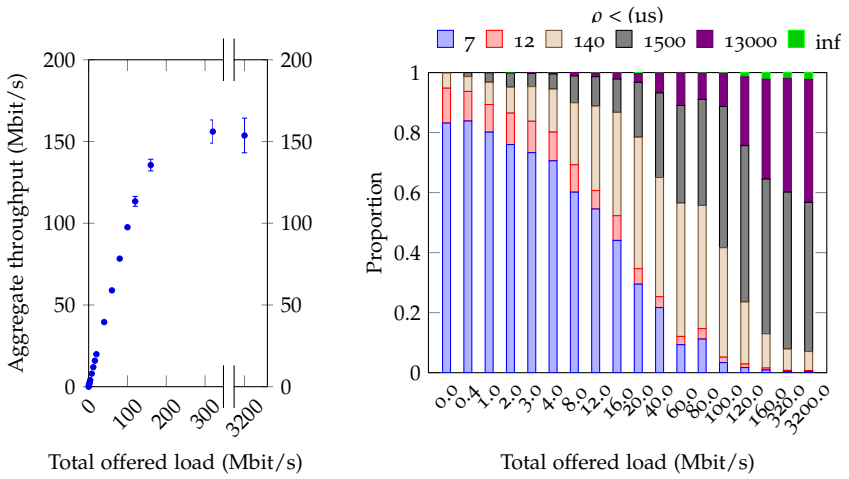
(b) Barplot of Beacon jitter vs. total offered load.



(c) Beacon jitter box plots vs. total offered load.

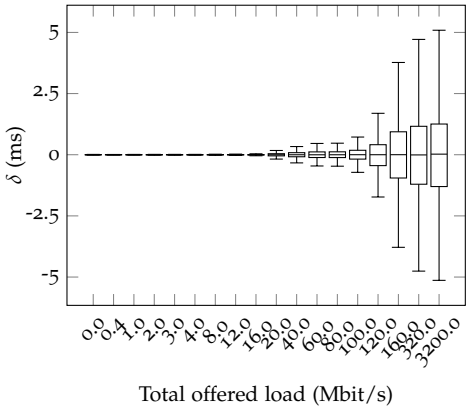
Figure 34: Throughput and Beacon jitter results from controlled conditions experiments while using three transmitters.

more transmitters competing to access the channel, the bigger the collision probability. In the case of two and three transmitters the maximum observed aggregate throughput corresponds to the experiments with the largest total offered load (Figures 33a, 34a); in each of these cases we consider as saturated only one experiment, namely the one with the largest aggregate throughput (1600 Mbit/s and 2400 Mbit/s). We take a pessimistic approach to mark an experiment as saturated. That is, we consider an experiment as saturated only if it is at the saturation point or to the right of the saturation point, in the plots showing the total offered load vs. the aggregate throughput (Figures 33a, 34a, 35a, and 36a). A more conservative approach would be to consider as saturated any experiment with an aggregate throughput within a confidence interval of the saturation point.



(a) Aggregate throughput vs. total offered load.

(b) Barplot of Beacon jitter vs. total offered load.



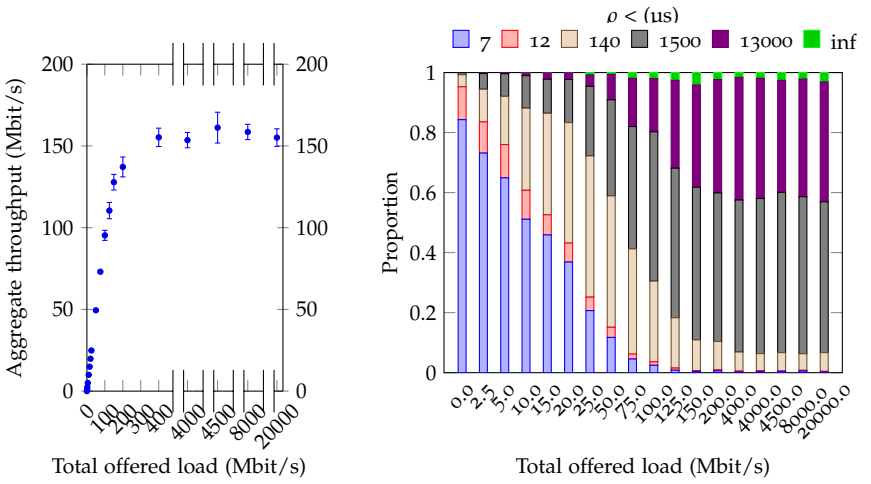
(c) Beacon jitter box plots vs. total offered load.

Figure 35: Throughput and Beacon jitter results from controlled conditions experiments while using four transmitters.

4.4 BEACON JITTER AND CHANNEL LOAD

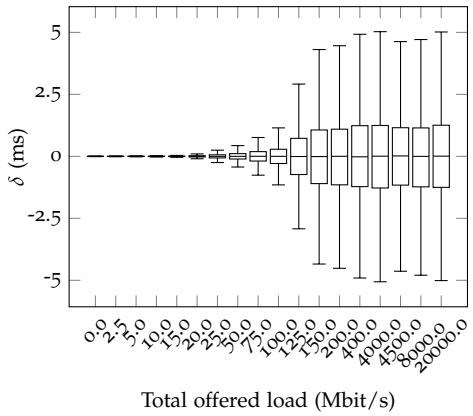
Figures 33c, 34c, 35c, and 36c show box plots of the Beacon jitter for the controlled condition experiments. The box plots indicate that the median Beacon jitter remains close to zero regardless of the number of transmitters and the offered load. This is explained by the fact that the jitter can be positive as well as negative. However, as the total offered load increases, the variability of the Beacon jitter increases, as more and more Beacons arrive late compared to the nominal Beacon interval.

Beacon jitter variability increases until the channel reaches the saturation throughput, at which point the Beacon jitter distribution remains constant. For example, Figure 36c, corresponding to experiments with five transmitters, shows that in the non-saturated experiments the interquartile range (IQR) of the Beacon jitter increases



(a) Offered load vs. total offered load.

(b) Barplot of Beacon jitter vs. total offered load.



(c) Beacon jitter box plots vs. total offered load.

Figure 36: Throughput and Beacon jitter results from controlled condition experiments while using five transmitters.

monotonically, from zero milliseconds when the total offered load is zero to 2.20 ms when the total offered load is 200 Mbit/s. In the saturated experiments the IQR of the Beacon jitter remains around 2.40 ms, regardless of the increases in the total offered load.

In the case of saturated experiments, regardless of the number of transmitters, the Beacon jitter distribution is similar. Particularly, the IQR is roughly 2.40 ms in each case. This is evident when comparing the last box plots of Figures 33c and 34c, the last two box plots of Figure 35c and the last three box plots of Figure 36c.

Figures 33b, 34b, 35b, and 36b show a different representation of the jitter distribution: each segment is the fraction of Beacon jitter smaller than 7 µs, 12 µs, 140 µs, 1500 µs and 1300 µs. Regardless of the number of transmitters, the fraction of Beacons whose jitter is less than 7 µs decreases as the total offered load increases (from around

80 % for an idle channel to a negligible value for saturated channels). Conversely, the fraction between 1500 µs and 1300 µs increases as the load increases, from a negligible value for an idle channel to roughly 60 % for a saturated one.

Figure 37 shows the empirical distribution of the absolute actual Beacon intervals Δ, given by (3). The top pane shows the CDF in log-linear scale to highlight the differences in the values between 1 µs and 1 ms. The bottom pane shows the complimentary CDF, in log-log scale, to better show the tail of the distribution, i.e., values larger than 1 ms. Solid lines correspond to experiments in a saturated channel, dashed lines represent the remaining experiments. We note that non-saturated experiments are spread on the mid-to-left side of the figure, while saturated experiments are concentrated on the right. Figure 37 shows that, as the offered load in the experiments gets closer to the saturation condition, the Beacon jitter distribution moves to the right, and confirms that, after the channel is saturated, the distribution of the Beacon jitter does not change significantly even as the total offered load increases. This can be explained by the fact that, when the channel is saturated, further increasing the offered load only increases the number of frames in the transmitter queue, without a direct impact on the traffic seen on the channel. Based on these observations, we can distinguish two cases:

1. The channel is saturated or close to saturation: Beacon jitter distributions are similar to each other.

2. The channel is not saturated: Beacon jitter distributions vary, but they are different from the distributions typical of saturated channels, that is.

Figure 38 shows the actual Beacon intervals for the saturated experiments with controlled conditions together with the saturated experiments with uncontrolled conditions. Note that the solid curves corresponding to the controlled condition experiments are the same in Figures 37 and 38. As previously mentioned, the uncontrolled condition experiments share the channel with a production IEEE 802.11 network, limiting our ability to identify the actual aggregate throughput. Thus, in uncontrolled condition experiments we only consider saturated experiments. Compared to the unsaturated experiments represented in Figure 37, Figure 38 shows that Beacon jitter distributions for saturated uncontrolled condition experiments are similar to each other and are concentrated on the right side of the figure. Additionally, Figure 38 shows that saturated controlled conditions experiments and saturated uncontrolled conditions experiments have similar Beacon jitter distributions.

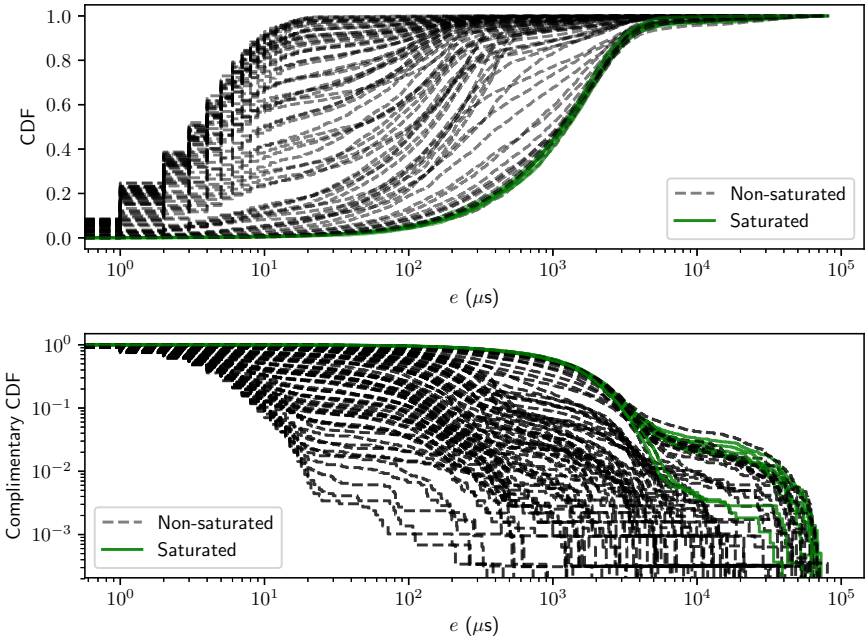


Figure 37: Empirical distribution of the actual Beacon interval in the controlled condition experiments.

4.4.1 *The Kolgomorov-Smirnov Test to Compare Beacon Jitter Distributions*

We use the Kolmogorov-Smirnov (KS) test [50] to compare Beacon jitter distributions. The KS test is based on the maximum distance between two distributions, so that similar distributions have smaller KS values.

Figure 39 is a graphical representation of a matrix containing the KS value corresponding to all pairs of experiments. Experiments are ordered, in the same way for rows and columns, by increasing normalized aggregate throughput, i.e., the aggregate throughput of an experiment divided by the maximum aggregate throughput for the corresponding number of transmitters[1]. Experiments with low normalized throughput are in the upper-left corner, while experiments with high normalized throughput are in the lower-right corner. The diagonal corresponds to the experiments compared to themselves, therefore the KS value is zero. Areas close to the diagonal show low KS values (e.g., below 0.20), indicating that experiments with a similar normalized throughput have similar Beacon jitter distribution. Note that experiments in the lower-right corner also have KS values below 0.20, which are the saturated experiments performed in both con-

1 For instance, an experiment with four transmitters and an aggregate throughput of 100 Mbit/s has a normalized (aggregate) throughput of 100/156 Mbit/s. Where 156 Mbit/s corresponds to the throughput at the saturation point for the configuration with four transmitters.

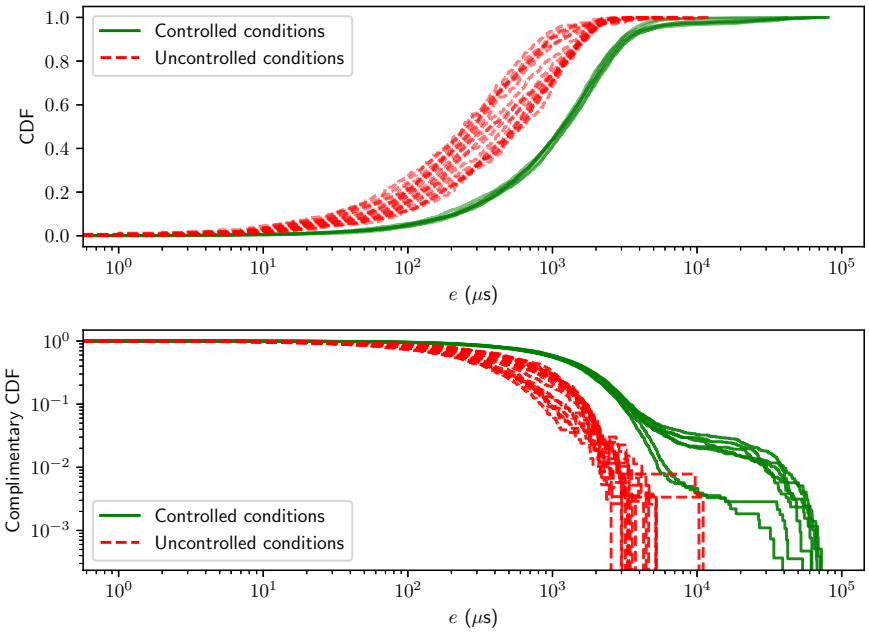


Figure 38: Empirical distribution of the actual Beacon interval corresponding to the saturated experiments performed in controlled and uncontrolled conditions.

trolled and uncontrolled conditions. This confirms that the Beacon jitter distribution is similar in all of our saturated experiments.

4.5 IDENTIFYING WI-FI CHANNEL LOAD

Section 4.4 shows that the distribution of the Beacon jitter is similar whenever the channel is saturated. Based on this empirical observation, we propose the following passive method for identifying whether a channel is saturated: we first compute the Beacon jitter distribution (ρ) from a given AP, then we compare this distribution with a reference one (r) corresponding to a saturated channel. If the KS value resulting from the comparison of ρ and r is below a given threshold α, we conclude that the channel is saturated as the two distributions are sufficiently close to each other. Algorithm 4 details the steps of the proposed method.

4.5.1 *Empirical Validation*

In this section we assess the performance of the method to classify a channel as saturated, as proposed in Section 4.5. First, we compute the KS threshold value α, then, with the selected α value, we assess the sensitivity of our method with respect to the reference distribution.

To pick α we evaluate the performance of the method for values of α in the range $[0, 1]$. For the evaluation we use 10-fold cross-validation.

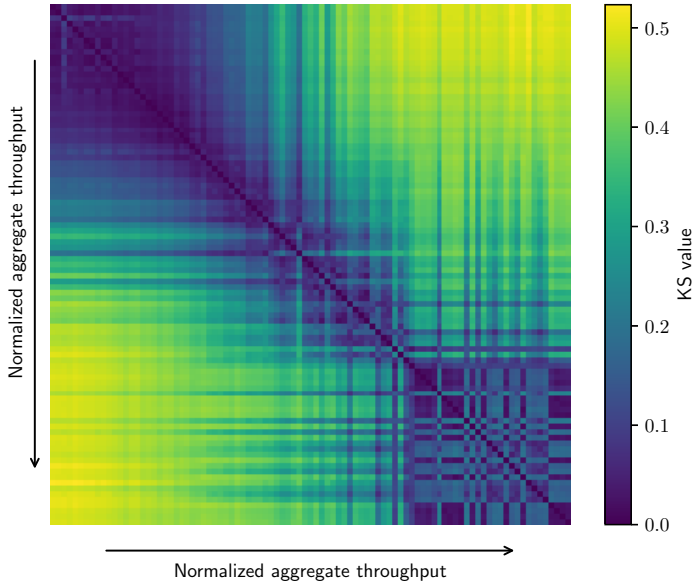


Figure 39: Graphical representation of the matrix containing the KS values for all pairs of experiments.

Algorithm 4 Channel classification

```
α ← threshold
r ← Beacon jitter distribution for the reference experiment
ρ ← Beacon jitter distribution for the target experiment
d ← KS(ρ, r)
if d < α then
    Saturated
else
    Non-saturated
end if
```

The data came from the controlled and the uncontrolled conditions experiments. To compare the performance of the different values of α we use the median Matthews correlation coefficient (MCC), which varies in the range $[-1, 1]$, where -1 means complete disagreement, 1 means perfect classification, and 0 is no better than random. Table 9 show the values of α that result in the best performance for 18 reference distribution that we evaluated.

Figure 40 shows the performance of the method with respect to α. The figure shows the MCC, precision, and recall scores when taking as reference (r) the distribution named one in Table 9. The best performance corresponds to $\alpha = 0.22$, with an average $MCC = 0.73$. The method correctly classified all saturated experiments (recall $= 1$). Among the experiments classified as saturated, 66 % are saturated experiments ($precision = 0.66$) and 34 % are non-saturated experiments.

Table 9: Values of α providing the best performance (highest MCC) with different reference distributions.

	1	2	3	4	5	6	7	8
α	0.22	0.21	0.21	0.23	0.23	0.22	0.22	0.18
MCC	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.70
Precision	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.64
Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

	9	10	11	12	13	14	15	16
α	0.19	0.17	0.19	0.17	0.21	0.17	0.19	0.23
MCC	0.71	0.73	0.74	0.74	0.71	0.73	0.72	0.58
Precision	0.64	0.67	0.68	0.68	0.64	0.67	0.66	0.53
Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

	17	18	Mean	Std	Median
α	0.20	0.20	0.21	0.03	0.21
MCC	0.64	0.66	0.71	0.04	0.73
Precision	0.60	0.62	0.65	0.04	0.66
Recall	1.00	1.00	1.00	0.00	1.00

Table 10: Performance of the proposed method for $\alpha = 0.21$.

	Mean	Std	Median	Min	Max
MCC	0.67	0.07	0.70	0.45	0.73
Precision	0.62	0.04	0.64	0.52	0.66
Recall	0.97	0.06	1.00	0.75	1.00

Note that, out of the false positives (i.e., experiments incorrectly classified as saturated), 38 % are within 90 % of the saturation throughput.

Table 10 allows to observe the robustness of the method with respect to the reference distribution. The table summarizes the performance of the proposed method when evaluating the 18 reference distributions with $\alpha = 0.21$. 0.21 is the median of the different α values shown in table 9. We note that the median for the MCC, the precision and the recall are 0.70, 0.64 and 1, respectively. Moreover, the standard deviation for MCC, precision and recall are 0.07, 0.04 and 0.06, respectively. This shows that the method is robust with respect to the selection of reference distribution.

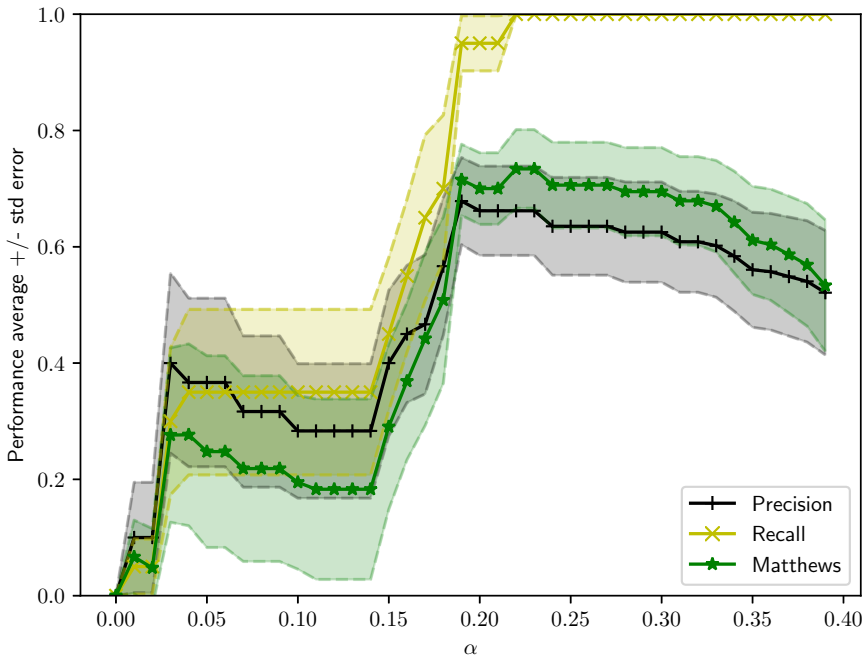


Figure 40: Performance of the channel classification method for different values of the threshold α.

4.5.2 *Packet Sampling*

When sampling, the monitoring station only considers a fraction of the packets captured during the observation window. Sampling is useful in the following cases: (1) to reduce the energy consumed by switching off the radio of the monitoring MS; (2) to reduce the amount of data that to store and process; (3) to allow the monitoring station to monitor multiple channels using a single NIC.

The observation window could be count-driven (e.g., k packets out of all packets) or timer-driven (e.g., packets captured during 10 s) [78]. The count-driven observation window is independent of the nominal Beacon interval and does not need to take into account the quality of the channel (i.e., collisions and noise). However, a downside of the count-driven observation window is that the actual sniffing time is variable. The following elements affect the duration of the observation window: (1) Beacons missed due to collisions and noise, (2) different Beacon intervals for different networks. Nevertheless, the timer-driven observation window guarantees a duration, regardless of the networks configuration and the channel conditions. A disadvantage is that the number of Beacons collected is variable, which affects the performance of the method.

The proposed classification method relies on the time interval between two consecutive Beacons, thus we consider Beacon intervals as samples, not individual Beacons. For example, three consecutive Beacons result in two samples, however, three non-consecutive Bea-

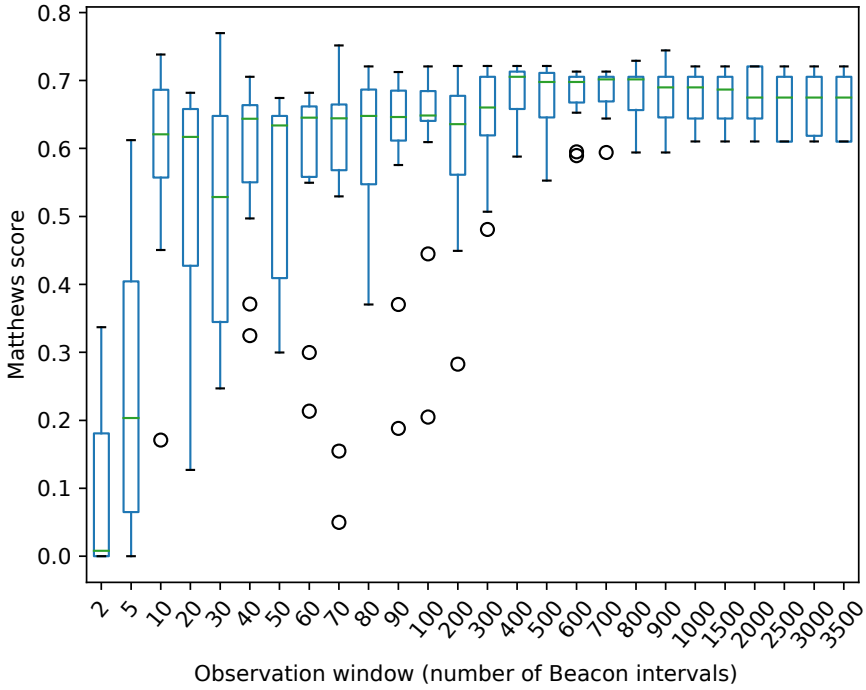


Figure 41: Performance of the classification algorithm for various observation windows. Sampling is count-driven, i.e., number of Beacon interval.

cons are discarded because they do not result in any Beacon interval. Hence, the count-driven approach for counting Beacon intervals results more natural. In the Section 4.5.3 we discuss the sampling window using a count-driven approach.

4.5.3 *Observation Window*

This section describes the trade-off between the number of samples and the performance of the classification method. Figure 41 shows the performance of the classification method as a function of the sample window. The x-axis shows the observation window, y-axis shows the MCC. Starting with an observation window of 10 Beacon intervals, the method maintains an MCC close to 0.60, and remains between 0.60 and 0.70 as the observation window increases. However, the performance has bigger variability for small observation windows. For example, for an observation window of 100 Beacon intervals, the median is 0.65 but there was one algorithm execution that register a MCC 0.17. The performance stabilizes between for an observation window larger than 800 samples, where the MCC remains between 0.60 and 0.70, with the IQR is about 0.05.

Figure 41 shows an increase in the classification performance with the increase in the sample size up to 100 samples, from that point the

performance does not increase significantly. To have those 100 Beacon intervals, an MS have to collect 101 Beacons in the ideal scenario, i.e., without missing any Beacon. An MS, monitoring an AP transmitting Beacons every 102.40 ms, needs 10.34 s to capture the 101 Beacons[2].

4.6 CONCLUDING REMARKS

In this chapter we studied the distribution of IEEE 802.11 Beacon jitters as a proxy to identify saturated channels. Using an extensive set of experiments we verified that the distribution of the Beacon jitter varies consistently with the aggregate throughput. Specifically, we found that the Beacon jitter distribution varies until the channel reaches the saturation throughput, at which point the distribution remains constant. Empirical results show that Beacon jitter distributions for saturated channels are similar to each other and that they are different from those of non-saturated channels. On the contrary, non-saturated channels present variable Beacon jitter distributions that are different from the distribution typical of saturated channels.

Based on the insight gained from our experiments, we proposed a method for non-intrusive classification of Wi-Fi channels. The method takes as input the Beacons transmitted by APs and captured by a monitoring MS. It correctly identifies 100 % of the saturated experiments, out of which 34 % are false positives. The experiments also demonstrated that regular Wi-Fi devices can passively collect the input needed, making our method a candidate strategy to be used for network and channel selection.

Our current proposal is limited to a binary classification. An extension would be by using clustering techniques, to detect patterns in the distributions that could lead to a more fine-grained classification.

2 We have observed that most vendors use 100 time unit (TU) as the default configuration, where 1 TU is 1024 µs

5

CONCLUSIONS AND PERSPECTIVES

5.1 CONCLUDING REMARKS

Nowadays, many users carry around network devices, some of which are always on (e.g., smartphones, smartwatches, and other wearables). These users and devices, eager to have access to the Internet, are pushing for performant networks everywhere. To satisfy this demand, users, institutions and service providers are installing Wi-Fi networks in more and more places, and particularly in urban areas.

The current Wi-Fi networks in urban areas are characterized by chaotic and dense deployments, with multiple access points (APs) available in a given spot. Having information about these networks is important, both for users and for Internet service providers. Internet service providers may use the information to better configure their networks to fulfill the user needs, dynamically adapting current configurations and deploying new networks. Users could exploit the network information to optimize Wi-Fi procedures. For example, to cut the network discovery time, to lower the energy consumption, or to help in the selection of the Internet access among the available networks, where the available networks could be different technologies (e.g., Wi-Fi and cellular network) or different Wi-Fi APs.

In order to address the discovery of Wi-Fi networks in urban areas, Chapter 2 provides an empirical study of the discovery process, particularly focusing on the impact of the scanning parameters (channels to probe and probing time). We have gathered more than 66 000 Beacons and more than 18 000 Probe Responses. The data show that Probe Responses contribute to the active scanning during the first 100 ms, from that point on only the Beacons contribute to the network discovery, this suggests that active scanning should use probing timers below 100 ms. The experiments have also shown that an mobile station (MS) can discover APs on the adjacent channels, this insight may be of assistance to further optimize the scanning algorithm proposed in Section 3.5. The study has shown that a single scan does not discover all available networks, not even with long probing timers. Instead, these data show that it is necessary to execute multiple scans to obtain a complete list of the available networks. In addition, different scans likely report different results. Hence, we argue that a collaborative service could opportunistically assist mobile users in the network discovery. This collaborative service enables MSs and APs to aggregate and share results from multiple scans, from the same or different MSs.

We have proposed such a collaborative service (Wireless Measurements Sharing Platform (WMSP)), a cloud-based system to collect network measurements taken by existing MS and APs, and to process these data to generate a better understanding of the existing network environment. Wireless Measurements Sharing Platform (WMSP) relies on simple network measurements passively collected by APs and MSs, namely scan results and MS location. We have shown that existing commercial off-the-shelf (COTS) MS can gather network measurements passively and that it is possible to effectively combine the measurements from different users to have an accurate and up-to-date view of the network environment. For merging multiple traces we have proposed an algorithm that, in spite of its limitation and simplicity, shows the benefits of aggregating traces containing partial views of the Wi-Fi network topology. We have presented two applications of WMSP, the minimal set of APs and the reduction of the scanning timers. These applications propose two alternatives to improve the performance of Wi-Fi processes from the network side and from the user side. Whilst the approach used in the minimal AP set use case did not take into account the quality of service (QoS) , it did substantiate the possibility of save energy by turning off some APs while maintaining the Wi-Fi coverage. The results show that it is possible to turn off about 90 % of the available APs while maintaining the same coverage. Concerning the reduced scanning timers, we have used traces collected in the city of Rennes to compute scanning parameters that satisfy specific latency constraints, while maximizing the number of APs discovered. The empirical validation demonstrates that MSs can discover 60 % of the available APs in about 50 ms.

In Chapter 4 we studied the distribution of IEEE 802.11 Beacon jitters as a proxy to identify saturated channels. Using an extensive set of experiments we have verified that the distribution of the Beacon jitter varies consistently with the aggregate throughput. Specifically, we have found that the Beacon jitter distribution varies until the channel reaches the saturation throughput, at which point the distribution remains constant. Empirical results show that the Beacon jitter distribution is similar and typical of saturated channels. On the contrary, non-saturated channels present a variable Beacon jitter distribution that is different from the distribution typical of saturated channels.

Based on the insight gained from our experiments, we proposed a method for non-intrusive classification of Wi-Fi channels. The method takes as input the Beacons transmitted by APs and captured by a monitoring station (STA). It correctly identifies 100 % of the saturated experiments, out of which 34 % are false positives. The experiments also demonstrated that regular Wi-Fi devices can passively collect the input needed, making our method a candidate strategy for network and channel selection. The main limitation of this method is that it only allows a binary classification.

5.2 FUTURE WORK

In order to extend the contributions of this dissertation, the following issues may be considered for the future work. Regarding the Wi-Fi discovery on urban areas, a further study could assess the characteristics of the discovery in suburban and rural areas. Achtzehn et al. [1] show that the "household density is the best single predictor for the Wi-Fi AP density", therefore the Wi-Fi discovery process on these areas will face an AP density that is, most likely, different from the AP density found in urban areas. Also, characteristics of the environment (e.g., construction materials, building size and, distribution) also affects the radio waves differently. It should be possible to build a model of the discovery process that takes into account all these elements. This model would be valuable for designing scanning algorithms and handovers.

Further work may also look at the effects of the discovery process on the existing traffic on adjacent channels. Scanning procedures can greatly impact the energy consumption and the throughput due to the overhead introduced in the channel [84], that is, Probe Requests triggers, in the best scenario, one Probe Response from each AP. Since APs can overhear Probe Requests on adjacent channels, the scanning procedure likely impacts MSs and APs operating on the adjacent channels. Xueheng Hu et al. [84] study the effects on Wi-Fi performance in general, and the precise effects of the scanning on adjacent channels remain unclear.

Withing urban areas, Wi-Fi networks in the context of public transportation deserves a dedicated study. In this context we observe scenarios challenging the APs and the MSs, e.g., Wi-Fi networks around a bus stop and Wi-Fi networks inside buses. Networks around a bus stop have to deal with a number of MSs devices arriving simultaneously. While the bus is at the stop, each arriving MSs will likely trigger the scanning procedure. Then, nearby APs will transmit individual Probe Responses. As a result, the collision probability in the channel will probably increase. Regarding the Wi-Fi networks inside buses, these networks have to deal with a frequent change in the channel conditions. For example, at one bus stop channel 1 may have the lower load, and therefore the best option. Few seconds later, at the following bus stop, channel 1 may be fully saturated, thus a bad choice. An evaluation of these scenarios may reveal issues and potential improvements, for example, by taking advantage of the particularities of the public transportation, e.g., regular schedules and regular routes.

The usefulness of WMSP greatly depends on the aggregation of measurements collected by the contributors. A natural progression would be to evaluate merging strategies that take into account information like the age and the quality of the traces, so that WMSP provides more accurate information and can effectively react to changes

in the networks. In the same direction, further research should study the trade-off between the frequency of the measurements, how often the results of the different applications are recomputed, and the relevance of the results. Note that this trade-off likely depends on the type of application. We have only evaluated WMSP on outdoor scenarios, further experiments should address indoor scenarios. For this, it is essential for WMSP to perform indoor localization. Since we rely on global positioning system (GPS) to locate the traces, WMSP is currently limited to outdoor scenarios.

In WMSP we have taken a naive approach, we assume that all participants and measurements are reliable. That is, we take for granted that there are not attackers. Future works should include authentication and authorization mechanisms. Also, further work is necessary to evaluate the effects of malicious measurements. A simple attack consists in one user transmitting malicious measurements, indicating the presence of an APs in a given area. This attack could result in a minimal AP set that shuts down all networks in that area.

Regarding the method discussed in Chapter 4 to identify saturated channels, the current proposal is limited to a binary classification. An extension would be by using clustering techniques, to detect patterns in the distributions that could lead to a more fine-grained classification. Additionally, Beacons may give information about the channel busyness, that is, the time that the channel is busy. Future work may model the Beacon transmission using queuing theory, and approximate the channel busyness as the channel waiting time, that is, the time that Beacons wait for transmission. Future work may also explore practical applications of the method to identify saturated channels, for example, an algorithm for automatic channel selection could give preference to non-saturated channels. Similarly, MSs may avoid networks operating on a saturated channel.

More broadly, research is also needed to determine and exploit the data available in the channel. That is, since Wi-Fi uses a broadcast medium, stations can decode the frame headers transmitted by the nearby stations (e.g., physical layer convergence procedure (PLCP) and medium access control layer (MAC) header). By analyzing the headers, a monitoring station has access to considerable information. For example, a monitoring station could potentially infer on the quality of a network by analyzing the modulation and coding scheme (MCS) used to encode the frames transmitted by the AP.

GLOSSARY

AP	access point. vii, xv–xvii, 1–8, 10–23, 25–35, 37–47, 49, 50, 52–60, 67, 72–76
AP	point d'access. 79, 81–89
BSS	basic service set. 12, 31, 54, 77
BSSID	basic service set (BSS) identifier. 12, 31, 37, 52
CCA	clear channel assessment. 8
COTS	commercial off-the-shelf. 74
CSMA/CA	carrier sense multiple access with collision avoidance. 54, 55
DCF	distributed coordination function. 10, 77
DIFS	distributed coordination function (DCF) interframe space. 10
GPS	global positioning system. 25, 34, 76
IQR	interquartile range. 63, 64, 71
ISP	Internet service provider. 1
KS	Kolmogorov-Smirnov. xvii, 57, 66–68
LTE	long term evolution. 29
MAC	medium access control layer. 12, 27, 54, 55, 76
MaxCT	maximum channel time. xv, 6–10, 23, 47–49
MCC	Matthews correlation coefficient. xviii, 68, 69, 71
MCS	modulation and coding scheme. 20, 21, 53, 55, 56, 76
MIIS	media independent information service. 26
MIMO	multiple input/multiple output. 56, 78
MinCT	minimal channel time. xv, 7, 8, 10, 23, 47–49
MS	mobile station. xvi, 2, 3, 5–11, 13–15, 19–23, 25–27, 30–37, 41, 42, 44, 46–53, 56, 59, 60, 70, 72–76

MS	station mobile. 80–86, 88–90
MU-MIMO	multi user multiple input/multiple output (MIMO). 56
NAV	network allocation vector. 7, 9
NIC	network interface card. 5, 12, 15, 59, 70
PLCP	physical layer convergence procedure. 76
QoS	quality of service. 2, 39, 74
RSSI	receive signal strength indicator. xvi, 2, 21, 22, 25, 30–32, 35, 37, 41, 42, 44, 59
SDN	software design networking. 28
SNR	signal-to-noise ratio. 2
SSID	basic service set. 12, 31, 37
STA	station. 52, 53, 56, 74
TU	time unit. 72
UDP	user datagram protocol. 61
WMSP	Wireless Measurements Sharing Platform. vii, xiii, xvi, 3, 4, 26, 30–35, 37–39, 41–43, 46–52, 74–76, 82, 85

RÉSUMÉ

INTRODUCTION

Aujourd'hui, les appareils compatibles IEEE 802.11 Wi-Fi[1] comme les ordinateurs portables, smartphones ou tablettes sont omniprésents et largement utilisés par un grand nombre d'utilisateurs accédant à toutes sortes d'applications et de services. *Ericsson Mobility Report November 2017 – Ericsson* [25] indique qu'il y avait 4,40 milliards d'abonnements de téléphonie mobile dans le monde en 2017 et qu'il y en aura 7,20 milliards d'ici 2023. Il est raisonnable de supposer que ces téléphones – ou smartphones sont aussi des clients Wi-Fi potentiels, désireux d'accéder aux réseaux Wi-Fi. Pour répondre à la demande croissante de connectivité sans fil, différents acteurs ont installé des point d'accesss (APs) IEEE 802.11 : Les clients des fournisseurs de services Internet à leur domicile pour leur propre usage ; les entreprises dans leurs bureaux pour leurs propres employés et leurs clients (par exemple aéroports, magasins, centres commerciaux) ; les institutions publiques desservant de grandes régions (par exemple, des administrations fournissant une couverture réseau dans une ville).

Les réseaux IEEE 802.11 présentent différents avantages, comme par exemple, un faible coût, un bon débit ou l'usage des bandes de fréquence radio sans licence. Tous ces éléments permettent de déployer facilement plusieurs APs de manière décentralisée, sans planification ni coordination. De plus, les utilisateurs et les institutions utilisent les réseaux Wi-Fi pour construire ce qu'on nomme les *réseaux communautaires* [8] (par exemple, Freifunk[2], Peoples' Open[3], guiFi.net [4], Fon[4]), en permettant aux utilisateurs d'accéder à l'Internet via des réseaux Wi-Fi partagés. Tout cela se traduit par des déploiements denses et non organisés.

Un problème clé de ces déploiements denses et chaotiques est le manque d'informations sur les réseaux. Par exemple, les opérateurs peuvent connaître l'adresse postale des utilisateurs et leurs PA, mais ne disposent d'aucune information sur la couverture des PA et l'expérience de l'utilisateur. Il est important que les opérateurs disposent d'informations détaillées, par exemple pour la sélection des canaux et pour positionner les nouveaux points d'accès afin d'éliminer les zones blanches. Les utilisateurs n'ont pas suffisamment d'informations sur les réseaux Wi-Fi voisins non plus. Traditionnellement, la seule infor-

1 À travers ce document, nous utilisons le Wi-Fi et l'IEEE 802.11 de manière interchangeable.

2 `https://freifunk.net`

3 `https://peoplesopen.net`

4 `https://network.fon.com/`

mation disponible pour les utilisateurs est l'intensité du signal des réseaux voisins. Des informations supplémentaires peuvent permettre aux utilisateurs Wi-Fi d'optimiser les procédures de connexion, telles que la découverte et la sélection du réseau. L'obtention d'informations sur le réseau n'est pas triviale, étant donné la nature des réseaux Wi-Fi (décentralisés, fonctionnant dans des bandes de fréquences non exclusives et encombrées, liaisons radio dans des conditions changeantes, interférences radio).

Les chercheurs ont souligné le besoin de mesures du point de vue de l'utilisateur [67, 69, 70]. Le *crowd source sensing* [29] est une approche pour collecter des informations sur les réseaux en se servant des téléphones des utilisateurs comme sondes. Les solutions propriétaires comme OpenSignal[5] utilisent cette approche *crowd source* pour mesurer la performance des réseaux mobiles. Bien qu'il existe différentes mesures pour décrire la performance du réseau (par exemple, l'indicateur de puissance du signal reçu (RSSI), le Signal-to-Noise Ratio (SNR), la saturation du réseau, la probabilité de collision, le débit de la couche d'application), le RSSI est généralement utilisé, probablement parce qu'il est facilement disponible pour les clients. Le RSSI est une mesure de l'énergie observée au niveau de la couche physique [36] lors de la réception d'une trame Wi-Fi. Le SNR complète le RSSI avec le niveau de bruit. Ces deux mesures ne reflètent que la puissance du signal, alors que d'autres caractéristiques affectent également la performance de la connexion (par exemple, la charge du canal ou le nombre d'utilisateurs). Certaines études rapportent que le SNR et particulièrement le RSSI peuvent être inexacts [32, 42] ou trop optimistes pour des processus clés comme l'adaptation de la modulation [87]. A titre d'exemple pratique, Google a récemment ajouté a Android 8.1 la possibilité de visualiser la vitesse des réseaux publics [19], en plus des barres d'intensité de signal bien connues.

L'objectif principal de cette thèse est de proposer des solutions permettant aux utilisateurs et aux opérateurs de réseaux de collecter, d'analyser et d'exploiter efficacement les informations sur les réseaux Wi-Fi existants. Notre approche consiste à extraire des informations du nombre croissant d'utilisateurs mobiles qui, avec leur station mobile (MS), collectent périodiquement des informations réseau. Ces informations pourront alors être utilisées pour différentes applications indépendantes de la capture de données. Les utilisateurs et les opérateurs peuvent collecter des informations en utilisant des techniques non intrusives, puis les combiner et les partager par le biais d'une approche participative. Cela permet aux utilisateurs et aux opérateurs d'avoir une meilleure description des réseaux grâce aux informations contextuelles provenant de différents MS et qui reflètent bien l'expérience utilisateur.

5 `https://opensignal.com`

Différemment des autres approches, nous nous servons des procédures standard pour obtenir des informations réseau de manière non intrusive. En utilisant les procédures standard et les caractéristiques du protocole IEEE 802.11, il n'est pas nécessaire de modifier les équipements, ce qui permet aux MS et AP existants de participer. En enregistrant et en agrégeant les résultats du processus de découverte, nous sommes en mesure de localiser les couvertures des réseaux Wi-Fi et d'obtenir des informations sur le contexte dans lequel ils fonctionnent. En surveillant les variations du temps de transmission de la Beacon, nous pouvons identifier si le canal autour des AP est occupé ou non.

Disposer d'informations contextuelles est essentiel pour une meilleure utilisation des réseaux Wi-Fi [24] et pour permettre de nouveaux services, tant pour les fournisseurs de services que pour les utilisateurs. Les utilisations potentielles pour les opérateurs sont : (1) surveiller et assurer la qualité de service ; (2) ajuster dynamiquement le déploiement sur des paramètres spécifiques, comme le débit, la couverture ou l'efficacité énergétique pour s'adapter aux changements ; (3) aider à la décision d'utiliser ou non le transfert de données [24, 33, 44] des réseaux cellulaires vers le Wi-Fi ; (4) planifier l'installation de nouveaux points d'accès en examinant les points d'accès actuels [88]. Les informations Wi-Fi contextuelles sont également précieuses pour les utilisateurs, qui peuvent, par exemple, optimiser les procédures réseau, comme la découverte ou la sélection des AP.

Contributions

Dans cette thèse, nous étudions les réseaux Wi-Fi afin de proposer des métriques pour caractériser les réseaux Wi-Fi. En particulier, nous nous concentrons sur les solutions permettant aux utilisateurs finaux (par exemple, les smartphones, les ordinateurs portables) de caractériser les réseaux, sans modifications matérielles. De plus, nous visons des techniques passives afin de ne pas ajouter de trafic supplémentaire aux réseaux Wi-Fi déjà surchargés.

Puisque nous avons observé un nombre croissant de réseaux Wi-Fi dans les zones urbaines, nous étudions d'abord le processus de découverte IEEE 802.11 dans ces zones urbaines, afin de décrire ses caractéristiques et de réduire sa durée. Dans le cadre du processus de découverte, nous étudions la transmission des trames Beacon, à la recherche d'une technique de mesure passive de la qualité des réseaux. Ensuite, comme les appareils utilisateurs ne peuvent découvrir que partiellement les réseaux disponibles dans les environnements denses des zones urbaines, nous présentons une plateforme de partage pour améliorer les processus Wi-Fi, comme l'économie d'énergie et la découverte de voisinage.

Les principales contributions de cette thèse sont les suivantes :

ÉTUDE EMPIRIQUE DU PROCESSUS DE DÉCOUVERTE EN MILIEU URBAIN Afin de proposer des métriques et des techniques pour caractériser les réseaux Wi-Fi, nous étudions d'abord le processus de découverte du réseau IEEE 802.11 dans les zones urbaines. Nous notons que : (1) il y a des déploiements Wi-Fi denses dans les zones urbaines ; (2) un MS ne peut pas découvrir tous les réseaux disponibles dans de tels déploiements Wi-Fi denses avec une seule phase de découverte ; (3) des temporisations de sonde plus longues n'entraînent pas nécessairement la découverte de plus de réseaux ; (4) lors du sondage d'un canal donné, il est possible de découvrir des AP fonctionnant sur des canaux adjacents. Sur la base de ces observations, nous soutenons qu'un service d'information participatif pourrait aider les utilisateurs pendant la découverte du réseau en (1) stockant et agrégeant les rapports de découverte ; (2) fournissant des paramètres de configuration adaptés à des domaines particuliers et aux besoins des utilisateurs.

CONCEPTION ET ÉVALUATION D'UNE ARCHITECTURE POUR UN SERVICE D'INFORMATION PARTICIPATIF Nous proposons *Wireless Measurements Sharing Platform (WMSP)*, un service d'information participatif qui, en utilisant une approche de crowd sourcing, collecte les données réseau des APs et des MS, les pré-traite et les stocke. Différentes applications peuvent être développées pour utiliser les données stockées. Nous mettons en œuvre la plateforme de partage de mesures sans fil (WMSP) en utilisant les technologies Big Data, et montrons sa faisabilité et son utilité en collectant des mesures de réseau dans deux zones urbaines différentes, et en utilisant les données dans deux applications : ensemble minimum de APs et réduction des temps de sondage.

MÉTRIQUES ET TECHNIQUES NON INTRUSIVES CARACTÉRISANT LES RÉSEAUX WI-FI Nous proposons d'utiliser la gigue de la Beacon Wi-Fi comme mesure permettant d'identifier si un APs fonctionne ou non sur un canal saturé. Nous proposons également une technique non intrusive pour obtenir la gigue de la Beacon. Les solutions existantes, comme l'intensité du signal, manquent d'informations sur l'utilisation des canaux. D'autres solutions impliquent des mesures actives, augmentant la charge sur le canal déjà saturé ou nécessitant un matériel dédié. Les utilisateurs et opérateurs peuvent exploiter ces techniques et partager les résultats grâce au service d'information participatif (WMSP), afin d'avoir accès à une meilleure caractérisation des réseaux.

La suite détaille chacune de ces contributions.

DÉCOUVERTE DE POINTS D'ACCÈS DANS LES RÉSEAUX IEEE 802.11

Le faible coût des réseaux IEEE 802.11 et son déploiement possible dans des domaines fréquentiels libre de licence ont fortement contribué à sa popularité et donc à des réseaux denses, non organisés. Cela crée des déploiements spontanés avec des densités et des distributions différentes[1], et avec des modèles de découverte imprévisibles, avec des dispositifs ayant des performances différentes en termes de matériel et de logiciels [74].

La découverte de ces réseaux est une condition préalable à d'autres processus comme l'obtention de la connectivité IP Lorsque le MS initie une nouvelle connexion au réseau, ou lorsqu'il doit changer de AP au cours du temps (processus dit de "handover"), le MS doit commencer par découvrir les AP qui l'entourent, puis il doit décider auquel il doit essayer de s'associer. Le processus de découverte peut également être utilisé pour d'autres applications, comme la géolocalisation sans GPS [10, 18]. Dans ce cas, une technique consiste à découvrir les AP voisins et à se positionner en comparant les résultats de découverte avec un des AP Wi-Fi existants [18].

La découverte des réseaux consiste en ce qu'on appelle en Anglais un *scanning process*, qui est long [13, 55], coûteux en termes de trafic généré [84], énergétiquement intensif [10], et généralement incomplet, c'est-à-dire qu'il ne rapporte qu'un sous-ensemble des réseaux disponibles.

Actuellement, le processus de découverte traite tous les déploiements Wi-Fi de la même manière, ce qui entraîne des délais de découverte inadaptés (trop courts ou trop longs) et des résultats incomplets. Prenons l'exemple d'Alice et Bob. Alice se déplace avec un smartphone dans une zone industrielle, avec une densité Wi-Fi relativement faible et des AP de classe entreprise. Alors que Bob est assis dans un quartier résidentiel dense, où des centaines de familles ont installé des réseaux Wi-Fi domestiques. Dans les deux cas, la durée de découverte serait la même. Cependant, le smartphone d'Alice peut être capable de découvrir rapidement tous les réseaux disponibles, tandis que le smartphone de Bob ne découvrira que partiellement les réseaux disponibles.

Par ailleurs, des informations inexactes sur les réseaux disponibles peuvent conduire le MS à s'associer à un réseau sous-optimal, par exemple. De plus, le trafic généré pendant le processus de découverte consomme une partie non négligeable de la capacité du canal [84], devenant ainsi un problème potentiel.

De nombreux travaux ont été réalisés pour étudier le processus de découverte IEEE 802.11, principalement dans le contexte de la mobilité du terminal entre AP. Les MS peuvent ajuster leur processus de découverte en modifiant la durée de sondage des canaux et le nombre de canaux à sonder. Les MS peuvent modifier ces paramètres

une fois pour toutes ou dynamiquement, en tenant compte des caractéristiques du réseau et des besoins des utilisateurs.

Nous utilisons des données empiriques pour étudier le processus de découverte des réseaux Wi-Fi dans des scénarios réalistes. Nous avons observé que pour différents MS, lorsque le nombre de AP est élevé, une seule procédure de découverte n'est pas suffisante pour découvrir tous les AP disponibles. Nous discutons également des différences entre la découverte active et celle passive. Bien que la première contribue le plus au processus de découverte, elle ne peut pas détecter tous les AP disponibles, seul le second peut le faire. Nous observons également que la découverte active peut profiter du chevauchement des canaux, en recevant des trames envoyées sur des canaux adjacents.

L'amélioration du processus de découverte vise à réduire le temps utilisé pour la découverte du réseau, pour minimiser les interruptions de connectivité au réseau. Ceci contribue donc à améliorer la mobilité des utilisateurs grâce à un basculement plus rapide et plus fluide entre AP. Les améliorations du processus peuvent également porter sur la réduction du trafic échangé au niveau de la couche 2, ce qui permet une meilleure utilisation des canaux.

Compte tenu de ces résultats, nous soutenons que la connaissance préalable de quelques caractéristiques clés du réseau aide dans le processus de découverte. La section suivante décrit une solution participative qui permet aux utilisateurs de stocker et de partager leur vue (partielle) de la topologie du réseau Wi-Fi. Cette solution permet d'obtenir les caractéristiques de réseau mentionnées ci-dessus. Les MS peuvent, par exemple, régler dynamiquement les paramètres du processus de découverte (canaux à explorer, temps de canal minimal (MinCT) et temps de canal maximal (MaxCT)) en fonction des caractéristiques attendues du réseau. De même, les MS peuvent accéder aux informations de la solution participative pour compléter leurs résultats de découverte.

système d'information participatif

Plusieurs études [1, 3] et projets (par exemple, OpenSignal, WiGle, GMoN, Sensorly) signalent la présence et la croissance des réseaux Wi-Fi dans le monde entier. Par exemple, Achtzehn et al. [1] rapporte l'augmentation de la densité des réseaux Wi-Fi dans les zones urbaines de 14 fois au cours de la dernière décennie. En avril 2018, *iPass Wi-Fi Growth Map* [90] rapporte plus de 286 millions de hotspots dans le monde, et *The Zettabyte Era* [79] prévoit qu'il y aura près de 541,60 millions de hotspots Wi-Fi publics d'ici 2021. Ces réseaux Wi-Fi sont généralement installés de manière indépendante, sans aucune planification et sans être gérés non plus, ce qui entraîne des déploiements Wi-Fi "chaotiques" [3]. Un élément clé pour une meilleure utili-

sation de ces réseaux est la collecte d'informations contextuelles (par exemple, l'emplacement, la qualité, la couverture et le débit disponible des AP) du point de vue des MS [24]. Toutefois, en raison de la nature de ces réseaux non planifiés (décentralisés, fonctionnant dans des bandes de fréquences non exclusives et encombrées, interférences radio), l'obtention de ces informations n'est pas triviale.

En analysant ces réseaux, les utilisateurs et les opérateurs de réseau pourraient en améliorer la performance et fournir des services plus efficacement, comme promouvoir du *data offloading* efficacement [24, 33, 44]. De plus, le fait de disposer d'informations sur le réseau peut potentiellement permettre de réduire la consommation d'énergie, en activant et désactivant dynamiquement les AP pour mieux faire correspondre la capacité du réseau et la demande de trafic. Les opérateurs pourraient également améliorer la couverture du réseau en déployant davantage de AP pour combler les trous de couverture et/ou augmenter la capacité du réseau dans les zones à forte demande. Par exemple, lors d'un défilé, les opérateurs pourraient utiliser les AP des clients pour fournir un réseau adapté à l'événement et identifier les zones blanches dans la couverture pour placer de nouveaux AP.

Dans le même temps, un nombre croissant d'utilisateurs collectent déjà périodiquement des informations sur le réseau pour différentes applications. Généralement, les MS collectent des informations sur les réseaux au cours du processus de découverte du réseau, dont le résultat est une liste des AP détectés, y compris le RSSI et le canal sur lequel le AP opère. De plus, les téléphones actuels incluent des récepteurs GPS permettant de géolocaliser les mesures. En faisant tourner ce processus en arrière-plan, ce service ne requiert aucun effort supplémentaire de la part des utilisateurs. Cependant, un seul utilisateur n'a qu'une vue partielle du réseau. Même s'il combine tous les résultats recueillis par son MS sur une longue période de temps, il n'aurait qu'une vue partielle du réseau, strictement limitée aux endroits qu'il a déjà visités. Par contre, si plusieurs utilisateurs partagent ces résultats, ils peuvent bénéficier des observations d'autres utilisateurs.

Nous présentons la plate-forme *Wireless Measurements Sharing Platform (WMSP)*, un service d'information participatif pour recueillir, agréger et exploiter les données recueillies par les utilisateurs mobiles. WMSP résout les défis liés à la collecte, au prétraitement des données et à l'agrégation de mesures Wi-Fi partielles et/ou imprécises. Pour assurer le passage à l'échelle du système, nous avons utilisé les technologies de Big Data et de cloud computing. WMSP commence par le prétraitement des mesures brutes fournies par les utilisateurs. Ces données sont ensuite analysées par des "applications", qui font partie intégrante du système, pour résoudre des problèmes particuliers, par exemple pour faciliter la décharge du trafic cellulaire et la planification du réseau.

Nous présentons deux applications ou cas d'usage pour WMSP : le calcul de l'ensemble minimal des AP [38] et la détermination des paramètres de scanning optimaux [54]. L'application de l'ensemble des AP minimum calcule un sous-ensemble des AP existants qui sont capables d'offrir la même couverture dans une zone donnée, tout en réduisant la consommation d'énergie en réduisant le nombre de AP qui restent allumés. La deuxième application utilise un algorithme génétique pour optimiser la découverte du réseau, c'est-à-dire le processus qui permet aux MS de trouver les réseaux Wi-FI disponibles. Cette phase de découverte est le facteur dominant lors de l'exécution d'un transfert [53] (lorsqu'un MS sort de la zone de couverture de la AP desservie et passe à un autre).

Les expériences montrent la faisabilité et l'extensibilité du WMSP. Les résultats présentés portent sur plus de 150 heures-homme, couvrant une distance totale de plus de 700 km.

EVALUER LA QUALITÉ DES RÉSEAUX WI-FI

De nombreux travaux (voir, par exemple [2, 64, 85]) et projets (par exemple, WiGle, OpenSignal, Sensorly) ont montré que, surtout dans les zones urbaines, plusieurs PA peuvent être détectés à un même endroit.

En raison de la nature non régulée et non planifiée des réseaux Wi-Fi, les AP à proximité les uns des autres opèrent souvent sur le même canal, en particulier sur les canaux non chevauchants fréquemment utilisés (1, 6 et 11 pour la bande des 2,40 GHz). Cela peut entraîner une baisse des performances, en particulier lorsque la demande de trafic dépasse la capacité du canal, ce qui se traduit par un canal saturé (c'est-à-dire que toute augmentation de la demande ne se traduit pas par une augmentation du débit global). Dans ce cas, il est souvent préférable pour les utilisateurs de rejoindre un réseau différent, par exemple un autre AP fonctionnant sur un canal différent et non saturé, ou un réseau utilisant une technologie d'accès différente (par exemple, réseau cellulaire, réseau câblé).

Comme un utilisateur aura souvent le choix entre plusieurs AP, il est important de choisir le "meilleur" AP. Bien qu'il puisse y avoir différentes définitions du terme meilleur, selon les circonstances spécifiques, un AP opérant dans un canal saturé reste un candidat à éviter. Il faut donc tenir compte de la saturation des canaux lors de la sélection d'un AP.

Dans cette section, nous proposons une méthode simple pour les stations mobiles (STA) pour détecter un canal saturé en surveillant passivement les trames Beacon, qui sont disponibles pour toutes les STA dans le cadre des procédures IEEE 802.11. Cela permet aux STA de collecter passivement des informations et de déterminer si un canal est saturé ou non. L'utilisation de Beacons pour caractériser l'état

du canal présente de multiples avantages : les Beacons sont toujours présents, transmis en mode broadcast et utilisent l'ensemble de modulation et de codage le plus puissant (MCS). En analysant des expériences menées avec différentes charges de trafic, nous démontrons qu'il est possible d'identifier si un canal est saturé en fonction de la distribution de la gigue des Beacons. Même si les AP envoient périodiquement des Beacons, ils doivent attendre que le canal soit inutilisé, ce qui entraîne un délai supplémentaire qui dépend de l'utilisation du trafic. Les résultats empiriques montrent que la gigue des Beacons suit une distribution similaire chaque fois que le canal est saturé. Notre solution exploite cela en comparant la distribution de la gigue des Beacons avec une distribution de référence, correspondant à un canal saturé. La méthode prend comme entrée les Beacons transmis par les AP et capturées par une STA de surveillance. Il identifie correctement 100 % des expériences saturées, dont 34 % sont des faux positifs. Les expériences démontrent également que les appareils Wi-Fi ordinaires peuvent recueillir passivement les données nécessaires, ce qui fait de notre méthode une stratégie candidate à utiliser pour la sélection des réseaux et des canaux.

Si la littérature sur la sélection des AP et la caractérisation des réseaux Wi-Fi est vaste (voir, par exemple, [7, 23, 34, 35, 41, 43, 45, 59, 80, 82, 86, 89]), à notre connaissance, aucune solution existante n'est à la fois (1) réalisable sans changer les AP et (2) passive, c'est-à-dire qu'elle n'exige pas l'échange de trames supplémentaires.

CONCLUSIONS

Aujourd'hui, de nombreux utilisateurs transportent des appareils connectés (par exemple, des smartphones et des ordinateurs portables), ces utilisateurs, ayant besoin d'avoir accès à l'Internet, demandent des des réseaux de plus en plus performants partout. Pour satisfaire cette demande, les utilisateurs, les institutions et les fournisseurs de services installent des réseaux Wi-Fi dans plusieurs endroits, en particulier dans les zones urbaines.

Les réseaux Wi-Fi qui se trouvent dans les zones urbaines sont caractérisés par des déploiements denses, avec de multiples points d'accès disponibles dans un même endroit. Il est important de disposer d'informations sur les réseaux, tant pour les utilisateurs que pour les fournisseurs de services Internet. Les fournisseurs de services Internet peuvent utiliser Cette information pour mieux configurer leurs réseaux afin de répondre aux besoins des utilisateurs, en adaptant dynamiquement les configurations actuelles et en déployant de nouveaux AP. Les utilisateurs peuvent utiliser l'information sur le réseau pour optimiser les procédures Wi-Fi (p. ex., réduire le temps de découverte des réseaux Wi-Fi) et pour aider à la sélection de l'accès Internet parmi les réseaux disponibles, où les réseaux disponibles

pourraient être des technologies différentes (p. ex., Wi-Fi et réseau cellulaire) ou différents réseaux Wi-Fi.

Afin de traiter la découverte des réseaux Wi-Fi en milieu urbain, nous avons capturé et analysé plus de 66 000 Beacons et plus de 18 000 Probe Responses. Les données montrent que les Probe Responses contribuent au scanning actif pendant les 100 premières millisecondes, et à partir de ce point, seuls les Beacons contribuent à la découverte du réseau, ce qui suggère que le scanning actif devrait utiliser des temporisateurs en dessous de 100 ms. Les expériences ont également montré qu'un MS peut découvrir des AP sur les canaux adjacents, ce qui peut être utile pour optimiser davantage l'algorithme de scanning proposé dans les paramètres de scanning optimaux de l'application WMSP. L'étude a ainsi montré qu'un seul scanning ne permet pas de découvrir tous les réseaux disponibles, même avec de longs temps de scanning. Au lieu de cela, il est probable que des scans différents rapportent des résultats différents. Ces données montrent qu'il est nécessaire d'exécuter plusieurs scannings pour obtenir une liste complète des réseaux disponibles, c'est pourquoi nous soutenons qu'un service participatif pourrait opportunément aider les utilisateurs mobiles dans la découverte du réseau. Ce service participatif permet aux MS et aux AP d'agréger et de partager les résultats de plusieurs scannings, d'un même MS ou de MS différents.

Nous avons proposé un tel service collaboratif (WMSP) : un système cloud pour collecter des données de réseau prises par les MS et les AP existants, et pour traiter ces données afin de générer une meilleure compréhension des réseaux existants. Le WMSP repose sur des mesures simples du réseau recueillies passivement par les MS et les AP. Nous avons constaté que le hardware existant peut être utilisé pour collecter les statistiques du réseau et qu'il est possible de combiner efficacement les mesures faites par les différents utilisateurs afin d'avoir une vue plus précise et à jour des réseaux Wi-Fi. Pour combiner plusieurs traces, nous avons proposé un algorithme qui, malgré ses limites et sa simplicité, illustre les avantages de l'agrégation de traces contenant des vues partielles de la topologie du réseau Wi-Fi. Avec l'ensemble minimal de AP et la réduction des temps de scanning, nous avons ainsi montré qu'il est possible d'améliorer les performances des processus Wi-Fi du côté réseau et du côté utilisateur. Alors que l'approche utilisée dans le cas d'utilisation de l'ensemble minimal de AP ne tenait pas compte de la qualité de service (QoS), elle confirmait la possibilité d'économiser de l'énergie en éteignant certains AP, tout en maintenant la couverture Wi-Fi.

Dans la section 5.2, nous utilisons la distribution De la gigue de Beacons comme moyen pour identifier les canaux saturés. À l'aide d'un riche ensemble d'expériences, nous avons vérifié que la distribution de la gigue des Beacons varie en fonction du trafic global. Plus précisément, nous avons constaté que la distribution varie jusqu'à ce que

le canal atteigne sa saturation et que la distribution reste constante en ce point. Les résultats empiriques montrent que la distribution de la gigue de Beacons est similaire et typique pour les canaux saturés. Au contraire, les canaux non saturés présentent une distribution variable et différente de la distribution typique des canaux saturés.

Sur la base des leçons apprises lors de nos expériences, nous avons proposé une méthode de classification non intrusive des canaux Wi-Fi. La méthode prend en entrée les Beacons transmis par les AP et capturés par une STA intéressée. Le méthode identifie correctement 100 % des expériences saturées, dont 34 % sont des faux positifs. Les expériences ont également démontré que les appareils Wi-Fi courants peuvent recueillir passivement les données nécessaires, ce qui fait de notre méthode une stratégie candidate à utiliser pour la sélection des réseaux et des canaux. La principale limitation de cette méthode est qu'elle ne permet qu'une classification binaire.

Travaux Futurs

Afin d'étendre les contributions de cette thèse, les questions suivantes peuvent être prises en considération pour les travaux futurs. En ce qui concerne la découverte du Wi-Fi dans les zones urbaines, une autre étude pourrait évaluer les caractéristiques de la découverte dans les zones suburbaines et rurales. Achtzehn et al. [1] indiquent que "la densité des ménages est le meilleur prédicteur de la densité des AP Wi-Fi", donc le processus de découverte Wi-Fi sur ces zones sera confronté à une densité AP qui est, très probablement, différente de la densité AP trouvée dans les zones urbaines. De plus, les caractéristiques de l'environnement (p. ex. matériaux de construction, taille du bâtiment et distribution) influent sur les ondes radio de façon différente. Il devrait être possible de construire un modèle du processus de découverte qui tienne compte de tous ces facteurs. Ce modèle serait utile pour la conception d'algorithmes de *scanning* et de *handover*.

D'autres travaux pourraient également examiner les effets du processus de découverte sur le trafic existant dans les canaux adjacents. Les procédures de scanning peuvent avoir un impact considérable sur la consommation d'énergie et le débit dû à la surcharge introduite dans les canaux [84], c'est-à-dire que les Probe Request déclenchent, dans le meilleur scénario, une Probe Response pour chaque AP. Les MS et les AP peuvent entendre des trames Probe Request et Probe Response plus loin et sur des canaux adjacents, puisque les deux types de trames utilisent une forte MCS, ce qui entraîne une augmentation de la contention des canaux. Xueheng Hu et al. [84] étudient les effets sur les performances Wi-Fi en général, et les effets précis du scanning sur les canaux adjacents restent flous.

L'utilité du WMSP dépend grandement de l'agrégation des données recueillies par les contributeurs. Une progression naturelle serait

d'évaluer des stratégies de fusion qui prennent en compte des informations comme l'âge et la qualité des traces, afin que WMSP fournisse des informations plus précises et puisse réagir efficacement aux changements dans les réseaux. Dans le même sens, des recherches plus poussées pourraient étudier le compromis entre la fréquence des mesures, la fréquence à laquelle les résultats des différentes applications sont recalculés et la pertinence des résultats. Il est à noter que ce compromis dépend probablement du type d'application. Nous n'avons évalué le WMSP que sur des scénarios extérieurs, d'autres expériences pourraient aborder les scénarios intérieurs. Pour cela, il est essentiel que WMSP effectue la localisation à l'intérieur. Comme nous comptons sur le GPS pour localiser les traces, nous sommes actuellement limités aux scénarios extérieurs.

En ce qui concerne la méthode discutée dans la section 5.2 pour identifier les canaux saturés, la proposition actuelle se limite à une classification binaire. Une extension serait d'utiliser des techniques de clustering, pour détecter des modèles dans les distributions qui pourraient conduire à une classification plus détaillée. De plus, les Beacons peuvent fournir des informations supplémentaires, par exemple sur l'activité du canal. C'est-à-dire le temps pendant lequel le canal est occupé. Les travaux futurs pourraient modéliser la transmission des Beacons en utilisant la théorie de la file d'attente, et approximer l'occupation du canal comme temps d'attente du canal, c'est-à-dire le temps que les Beacons attendent pour la transmission. Les travaux futurs pourraient également explorer les applications pratiques de la méthode pour identifier les canaux saturés, par exemple, un algorithme de sélection automatique des canaux pourrait donner la préférence aux canaux non saturés. De même, les MS peuvent éviter les réseaux fonctionnant sur un canal saturé.

BIBLIOGRAPHY

[1] A. Achtzehn, L. Simić, P. Gronerth, and P. Mähönen. "Survey of IEEE 802.11 Wi-Fi Deployments for Deriving the Spatial Structure of Opportunistic Networks." In: *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). 00017. Sept. 2013, pp. 2627–2632. DOI: 10.1109/PIMRC.2013.6666591.

[2] Andreas Achtzehn, Ljiljana Simic, Marina Petrova, and Petri Mahonen. "IEEE 802.11 Wi-Fi Access Point Density Estimation with Capture-Recapture Models." In: 00001. IEEE, Feb. 2015, pp. 153–159. ISBN: 978-1-4799-6959-3. DOI: 10.1109/ICCNC.2015.7069333.

[3] Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. "Self-Management in Chaotic Wireless Deployments." In: *Wirel. Netw.* 13.6 (Dec. 2007). 00536, pp. 737–755. ISSN: 1022-0038. DOI: 10.1007/s11276-006-9852-4. URL: http://dx.doi.org/10.1007/s11276-006-9852-4 (visited on 05/07/2018).

[4] Roger Baig, Ramon Roca, Felix Freitag, and Leandro Navarro. "Guifi.Net, a Crowdsourced Network Infrastructure Held in Common." In: *Computer Networks* 90 (Oct. 2015). 00021, pp. 150–165. ISSN: 13891286. DOI: 10.1016/j.comnet.2015.07.009. URL: http://linkinghub.elsevier.com/retrieve/pii/S1389128615002327 (visited on 04/23/2018).

[5] Rajesh Krishna Balan, Archan Misra, and Youngki Lee. "LiveLabs: Building an In-Situ Real-Time Mobile Experimentation Testbed." In: *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. HotMobile '14. 00024. New York, NY, USA: ACM, 2014, 14:1–14:6. ISBN: 978-1-4503-2742-8. DOI: 10.1145/2565585.2565597. URL: http://doi.acm.org/10.1145/2565585.2565597 (visited on 05/08/2018).

[6] S. Bi, R. Zhang, Z. Ding, and S. Cui. "Wireless Communications in the Era of Big Data." In: *IEEE Communications Magazine* 53.10 (Oct. 2015), pp. 190–199. ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7295483.

[7] G. Bianchi. "Performance Analysis of the IEEE 802.11 Distributed Coordination Function." In: *IEEE Journal on Selected Areas in Communications* 18.3 (Mar. 2000). 08828, pp. 535–547. ISSN: 0733-8716. DOI: 10.1109/49.840210.

[8] Bart Braem et al. "A Case for Research with and on Community Networks." In: *SIGCOMM Comput. Commun. Rev.* 43.3 (July 2013). 00095, pp. 68–73. ISSN: 0146-4833. DOI: 10.1145/2500098.2500108. URL: http://doi.acm.org/10.1145/2500098.2500108 (visited on 04/23/2018).

[9] Vladimir Brik, Arunesh Mishra, and Suman Banerjee. "Eliminating Handoff Latencies in 802.11 WLANs Using Multiple Radios: Applications, Experience, and Evaluation." In: *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*. IMC '05. 00209. Berkeley, CA, USA: USENIX Association, 2005, pp. 27–27. URL: http://dl.acm.org/citation.cfm?id=1251086.1251113 (visited on 05/06/2018).

[10] N. Brouwers, M. Zuniga, and K. Langendoen. "Incremental Wi-Fi Scanning for Energy-Efficient Localization." In: *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2014 IEEE International Conference on Pervasive Computing and Communications (PerCom). 00027. Mar. 2014, pp. 156–162. DOI: 10.1109/PerCom.2014.6813956.

[11] Zuhal Can and Murat Demirbas. "Smartphone-Based Data Collection from Wireless Sensor Networks in an Urban Environment." In: *Journal of Network and Computer Applications* 58 (Supplement C Dec. 1, 2015). 00009, pp. 208–216. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2015.08.013. URL: http://www.sciencedirect.com/science/article/pii/S108480451500199X (visited on 01/02/2018).

[12] G. Castignani, N. Montavont, A. Arcia-Moret, M. Oularbi, and S. Houcke. "Cross-Layer Adaptive Scanning Algorithms for IEEE 802.11 Networks." In: *2011 IEEE Wireless Communications and Networking Conference*. 2011 IEEE Wireless Communications and Networking Conference. 00007. Mar. 2011, pp. 327–332. DOI: 10.1109/WCNC.2011.5779152.

[13] German Castignani. "Exploiting Network Diversity." 00002. PhD thesis. Télécom Bretagne, Université de Rennes 1, Nov. 7, 2012. URL: https://tel.archives-ouvertes.fr/tel-00776306/document (visited on 04/24/2018).

[14] German Castignani, Andrés Arcia, and Nicolas Montavont. "A Study of the Discovery Process in 802.11 Networks." In: *SIGMOBILE Mob. Comput. Commun. Rev.* 15.1 (Mar. 2011). 00014, pp. 25–36. ISSN: 1559-1662. DOI: 10.1145/1978622.1978626. URL: http://doi.acm.org/10.1145/1978622.1978626 (visited on 04/06/2016).

[15] German Castignani, Alberto Blanc, Alejandro Lampropulos, and Nicolas Montavont. "Urban 802.11 Community Networks for Mobile Users: Current Deployments and Prospectives." In: *Mobile Networks and Applications* 17.6 (Aug. 23, 2012). 00015, pp. 796–

807. ISSN: 1383-469X, 1572-8153. DOI: 10.1007/s11036-012-0402-2. URL: http://link.springer.com/article/10.1007/s11036-012-0402-2 (visited on 06/10/2015).

[16] German Castignani, Alejandro Lampropulos, Alberto Blanc, and Nicolas Montavont. "Wi2Me: A Mobile Sensing Platform for Wireless Heterogeneous Networks." In: 00019. IEEE, June 2012, pp. 108–113. ISBN: 978-1-4673-1423-7 978-0-7695-4686-5. DOI: 10.1109/ICDCSW.2012.36. URL: http://ieeexplore.ieee.org/document/6258143/ (visited on 04/24/2018).

[17] Y. Chen, B. Li, and Q. Zhang. "Incentivizing Crowdsourcing Systems with Network Effects." In: *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications. Apr. 2016, pp. 1–9. DOI: 10.1109/INFOCOM.2016.7524546.

[18] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm. "Accuracy Characterization for Metropolitan-Scale Wi-Fi Localization." In: *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*. MobiSys '05. 00558. New York, NY, USA: ACM, 2005, pp. 233–245. ISBN: 978-1-931971-31-7. DOI: 10.1145/1067170.1067195. URL: http://doi.acm.org/10.1145/1067170.1067195 (visited on 05/04/2018).

[19] *Connect to Wi-Fi Networks - Pixel Phone Help*. 00000. URL: https://support.google.com/pixelphone/answer/2819519#strength_speed (visited on 04/23/2018).

[20] Antonio De La Oliva, Albert Banchs, Ignacio Soto, Telemaco Melia, and Albert Vidal. "An Overview of IEEE 802.21: Media-Independent Handover Services." In: *IEEE Wireless Communications* 15.4 (Aug. 2008), pp. 96–103. ISSN: 1536-1284. DOI: 10.1109/MWC.2008.4599227.

[21] Antonio De La Oliva, Lucas Eznarriaga, Carlos J. Bernardos, Pablo Serrano, and Albert Vidal. "IEEE 802.21: A Shift in the Media Independence." In: *Future Network Mobile Summit (FutureNetw), 2011*. Future Network Mobile Summit (FutureNetw), 2011. June 2011, pp. 1–8.

[22] P. Dely, J. Vestin, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo. "CloudMAC: An OpenFlow Based Architecture for 802.11 MAC Layer Processing in the Cloud." In: *2012 IEEE Globecom Workshops (GC Wkshps)*. 2012 IEEE Globecom Workshops (GC Wkshps). Dec. 2012, pp. 186–191. DOI: 10.1109/GLOCOMW.2012.6477567.

[23] A. Dhananjay and L. Ruan. "PigWin: Meaningful Load Estimation in IEEE 802.11 Based Wireless LANs." In: *2008 IEEE International Conference on Communications*. 2008 IEEE International

Conference on Communications. 00004. May 2008, pp. 2541–2546. DOI: 10.1109/ICC.2008.481.

[24] Aaron Yi Ding, Yanhe Liu, Sasu Tarkoma, Hannu Flinck, Henning Schulzrinne, and Jon Crowcroft. "Vision: Augmenting WiFi Offloading with An Open-Source Collaborative Platform." In: *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*. MCS '15. New York, NY, USA: ACM, 2015, pp. 44–48. ISBN: 978-1-4503-3545-4. DOI: 10.1145/2802130.2802135. URL: http://doi.acm.org/10.1145/2802130.2802135 (visited on 09/18/2015).

[25] *Ericsson Mobility Report November 2017 – Ericsson*. 00000. 11/7/2017 9:32:00 AM. URL: https://www.ericsson.com/en/mobility-report/reports/november-2017 (visited on 04/23/2018).

[26] Jakob Eriksson, Hari Balakrishnan, and Samuel Madden. "Cabernet: Vehicular Content Delivery Using WiFi." In: *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*. MobiCom '08. 00503. New York, NY, USA: ACM, 2008, pp. 199–210. ISBN: 978-1-60558-096-8. DOI: 10.1145/1409944.1409968. URL: http://doi.acm.org/10.1145/1409944.1409968 (visited on 03/28/2018).

[27] Arsham Farshad, Mahesh K. Marina, and Francisco Garcia. "Urban WiFi Characterization via Mobile Crowdsensing." In: *Proc. Network Operations and Management Symposium (NOMS), 2014*. Network Operations and Management Symposium (NOMS), 2014. 00018. IEEE, May 2014, pp. 1–9. ISBN: 978-1-4799-0913-1. DOI: 10.1109/NOMS.2014.6838233. URL: http://ieeexplore.ieee.org/document/6838233/ (visited on 10/20/2016).

[28] P. Fuxjager, D. Valerio, and F. Ricciato. "The Myth of Non-Overlapping Channels: Interference Measurements in IEEE 802.11." In: *2007 Fourth Annual Conference on Wireless on Demand Network Systems and Services*. 2007 Fourth Annual Conference on Wireless on Demand Network Systems and Services. 00094. Jan. 2007, pp. 1–8. DOI: 10.1109/WONS.2007.340486.

[29] R. K. Ganti, F. Ye, and H. Lei. "Mobile Crowdsensing: Current State and Future Challenges." In: *IEEE Communications Magazine* 49.11 (Nov. 2011). 01017, pp. 32–39. ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.6069707.

[30] J. Guterman, A. A. Moreira, C. Peixeiro, and Y. Rahmat-Samii. "Wrapped Microstrip Antennas for Laptop Computers." In: *IEEE Antennas and Propagation Magazine* 51.4 (Aug. 2009). 00024, pp. 12–39. ISSN: 1045-9243. DOI: 10.1109/MAP.2009.5338680.

[31] E. Haghani, M. N. Krishnan, and A. Zakhor. "A Method for Estimating Access Delay Distribution in IEEE 802.11 Networks." In: *2011 IEEE Global Telecommunications Conference - GLOBECOM*

2011. 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011. 00003. Dec. 2011, pp. 1–6. DOI: 10.1109/GLOCOM.2011.6134235.

[32] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. "Predictable 802.11 Packet Delivery from Wireless Channel Measurements." In: *Proceedings of the ACM SIGCOMM 2010 Conference*. SIGCOMM '10. 00366. New York, NY, USA: ACM, 2010, pp. 159–170. ISBN: 978-1-4503-0201-2. DOI: 10.1145/1851182.1851203. URL: http://doi.acm.org/10.1145/1851182.1851203 (visited on 06/25/2015).

[33] Bo Han, Pan Hui, V.S. Anil Kumar, Madhav V. Marathe, Jianhua Shao, and Aravind Srinivasan. "Mobile Data Offloading through Opportunistic Communications and Social Participation." In: *IEEE Transactions on Mobile Computing* 11.5 (May 2012). 00476, pp. 821–834. ISSN: 1536-1233. DOI: 10.1109/TMC.2011.101. URL: http://ieeexplore.ieee.org/document/5765984/ (visited on 04/24/2018).

[34] Jung-Han Han and Seung-Jae Han. "Nonintrusive Estimation of Expected Throughput for IEEE 802.11 Devices." In: *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*. Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th. 00000. Jan. 2014, pp. 275–280. DOI: 10.1109/CCNC.2014.6866583.

[35] Kunho Hong, Jun Pyo Kim, Mun-Suk Kim, and SuKyoung Lee. "Channel Measurement-Based Access Point Selection in IEEE 802.11 WLANs." In: *Pervasive and Mobile Computing* 30 (Aug. 2016). 00001, pp. 58–70. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2015.10.018. URL: https://www.sciencedirect.com/science/article/pii/S1574119215002023 (visited on 01/30/2017).

[36] IEEE Commite. "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." In: *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)* (Dec. 2016). 00000, pp. 1–3534. DOI: 10.1109/IEEESTD.2016.7786995.

[37] "IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks Amendment 1: Architecture and Interfaces for Dynamic Spectrum Access Networks in White Space Frequency Bands." In: *IEEE Std 1900.4a-2011 (Amendment to IEEE Std 1900.4-2009)* (Sept. 2011). 00000, pp. 1–99. DOI: 10.1109/IEEESTD.2011.6022707.

[38] "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." In: *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)* (June 2008). 00000, pp. 1–244. DOI: 10.1109/IEEESTD.2008.4544755.

[39] "IEEE Standard for Local and Metropolitan Area Networks–Part 21: Media Independent Services Framework." In: *IEEE Std 802.21-2017 (Revision of IEEE Std 802.21-2008 as amended by IEEE Std 802.21a-2012, IEEE Std 802.21b-2012, IEEE Std 802.21c-2014, and IEEE Std 802.21d-2015)* (Apr. 2017). 00000, pp. 1–314. DOI: 10.1109/IEEESTD.2017.7919341.

[40] Haiming Jin, Lu Su, Houping Xiao, and Klara Nahrstedt. "INCEPTION: Incentivizing Privacy-Preserving Data Aggregation for Mobile Crowd Sensing Systems." In: *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc '16. 00002. New York, NY, USA: ACM, 2016, pp. 341–350. ISBN: 978-1-4503-4184-4. DOI: 10.1145/2942358.2942375. URL: http://doi.acm.org/10.1145/2942358.2942375 (visited on 10/20/2016).

[41] S. Kajita, H. Yamaguchi, T. Higashino, H. Urayama, M. Yamada, and M. Takai. "Throughput and Delay Estimator for 2.4GHz WiFi APs: A Machine Learning-Based Approach." In: *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)*. 2015 8th IFIP Wireless and Mobile Networking Conference (WMNC). 00002. Oct. 2015, pp. 223–226. DOI: 10.1109/WMNC.2015.30.

[42] Katrina LaCurts and Hari Balakrishnan. "Measurement and Analysis of Real-World 802.11 Mesh Networks." In: *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*. IMC '10. 00043. New York, NY, USA: ACM, 2010, pp. 123–136. ISBN: 978-1-4503-0483-2. DOI: 10.1145/1879141.1879158. URL: http://doi.acm.org/10.1145/1879141.1879158 (visited on 04/23/2018).

[43] Heeyoung Lee, Seongkwan Kim, Okhwan Lee, Sunghyun Choi, and Sung-Ju Lee. "Available Bandwidth-Based Association in IEEE 802.11 Wireless LANs." In: *Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. MSWiM '08. 00071. New York, NY, USA: ACM, 2008, pp. 132–139. ISBN: 978-1-60558-235-1. DOI: 10.1145/1454503.1454529. URL: http://doi.acm.org/10.1145/1454503.1454529 (visited on 03/30/2015).

[44] Kyunghan Lee, Joohyun Lee, Yung Yi, Injong Rhee, and Song Chong. "Mobile Data Offloading: How Much Can WiFi Deliver?" In: *Proceedings of the 6th International COnference*. Co-NEXT

'10. 00450. New York, NY, USA: ACM, 2010, 26:1–26:12. ISBN: 978-1-4503-0448-1. DOI: 10.1145/1921168.1921203. URL: http://doi.acm.org/10.1145/1921168.1921203 (visited on 04/23/2018).

[45] Mingzhe Li, Mark Claypool, and Robert Kinicki. "WBest: A Bandwidth Estimation Tool for IEEE 802.11 Wireless Networks." In: *33rd IEEE Conference on Local Computer Networks, 2008. LCN 2008*. 33rd IEEE Conference on Local Computer Networks, 2008. LCN 2008. 00153. Oct. 2008, pp. 374–381. DOI: 10.1109/LCN.2008.4664193.

[46] Y. Li, J. Gao, P. P. C. Lee, L. Su, C. He, C. He, F. Yang, and W. Fan. "A Weighted Crowdsourcing Approach for Network Quality Measurement in Cellular Data Networks." In: *IEEE Transactions on Mobile Computing* PP.99 (2016). 00000, pp. 1–1. ISSN: 1536-1233. DOI: 10.1109/TMC.2016.2546900.

[47] Yong Liao and Lixin Cao. "Practical Schemes for Smooth MAC Layer Handoff in 802.11 Wireless Networks." In: *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM'06)*. 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM'06). 00109. 2006, 10 pp.–190. DOI: 10.1109/WOWMOM.2006.90.

[48] Shu Liu and Aaron Striegel. "Casting Doubts on the Viability of WiFi Offloading." In: *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*. CellNet '12. 00039. New York, NY, USA: ACM, 2012, pp. 25–30. ISBN: 978-1-4503-1475-6. DOI: 10.1145/2342468.2342475. URL: http://doi.acm.org/10.1145/2342468.2342475 (visited on 05/16/2018).

[49] Omar Lizardo, Michael Penta, Matthew Chandler, Casey Doyle, G. Korniss, Boleslaw K. Szymanski, and Jonathan Z. Bakdash. "Analysis of Opinion Evolution in a Multi-Cultural Student Social Network." In: *Procedia Manufacturing* 3 (2015). 00006, pp. 3974–3981. ISSN: 23519789. DOI: 10.1016/j.promfg.2015.07.938. URL: http://linkinghub.elsevier.com/retrieve/pii/S2351978915009397 (visited on 05/17/2018).

[50] Frank J. Massey. "The Kolmogorov-Smirnov Test for Goodness of Fit." In: *Journal of the American Statistical Association* 46.253 (1951). 03000, pp. 68–78. ISSN: 0162-1459. DOI: 10.2307/2280095. JSTOR: 2280095.

[51] L. Meng, Y. Hulovatyy, A. Striegel, and T. Milenković. "On the Interplay Between Individuals' Evolving Interaction Patterns and Traits in Dynamic Multiplex Social Networks." In: *IEEE Transactions on Network Science and Engineering* 3.1 (Jan. 2016). 00006, pp. 32–43. ISSN: 2327-4697. DOI: 10.1109/TNSE.2016.2523798.

[52] Vivek Mhatre and Konstantina 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. 00176. New York, NY, USA: ACM, 2006, pp. 246–259. ISBN: 978-1-59593-195-5. DOI: 10.1145/1134680.1134706. URL: http://doi.acm.org/10.1145/1134680.1134706 (visited on 03/27/2018).

[53] Arunesh Mishra, Minho Shin, and William Arbaugh. "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process." In: *SIGCOMM Comput. Commun. Rev.* 33.2 (Apr. 2003). 01072, pp. 93–102. ISSN: 0146-4833. DOI: 10.1145/956981.956990. URL: http://doi.acm.org/10.1145/956981.956990 (visited on 03/13/2016).

[54] Nicolas Montavont, Andres Arcia-Moret, and German Castignani. "On the Selection of Scanning Parameters in IEEE 802.11 Networks." In: *2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. 2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC). 00003. Sept. 2013, pp. 2137–2141. DOI: 10.1109/PIMRC.2013.6666497.

[55] D. Murray, M. Dixon, and T. Koziniec. "Scanning Delays in 802.11 Networks." In: *The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)*. The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007). 00053. Sept. 2007, pp. 255–260. DOI: 10.1109/NGMAST.2007.4343430.

[56] J. W. Nah, S. M. Chun, S. Wang, and J. T. Park. "Adaptive Handover Method with Application-Awareness for Multimedia Streaming Service in Wireless LAN." In: *2009 International Conference on Information Networking*. 2009 International Conference on Information Networking. 00009. Jan. 2009, pp. 1–7.

[57] Anandatirtha Nandugudi, Anudipa Maiti, Taeyeon Ki, Fatih Bulut, Murat Demirbas, Tevfik Kosar, Chunming Qiao, Steven Y. Ko, and Geoffrey Challen. "PhoneLab: A Large Programmable Smartphone Testbed." In: *Proceedings of First International Workshop on Sensing and Big Data Mining*. SENSEMINE'13. 00065. New York, NY, USA: ACM, 2013, 4:1–4:6. ISBN: 978-1-4503-2430-4. DOI: 10.1145/2536714.2536718. URL: http://doi.acm.org/10.1145/2536714.2536718 (visited on 05/08/2018).

[58] Nick MacKeown, Tom Anderson, Hari Balikrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. *OpenFlow: Enabling Innovation in Campus Networks*. Mar. 14, 2008.

[59] Qixiang Pang, S. C. Liew, and V. C. M. Leung. "Design of an Effective Loss-Distinguishable MAC Protocol for 802.11 WLAN." In: *IEEE Communications Letters* 9.9 (Sept. 2005). 00072, pp. 781–783. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2005.1506701.

[60] Sang-Hee Park, Hye-Soo Kim, Chun-Su Park, Jae-Won Kim, and Sung-Jea Ko. "Selective Channel Scanning for Fast Handoff in Wireless LAN Using Neighbor Graph." In: *Personal Wireless Communications*. IFIP International Conference on Personal Wireless Communications. Lecture Notes in Computer Science. 00184. Springer, Berlin, Heidelberg, Sept. 21, 2004, pp. 194–203. ISBN: 978-3-540-23162-2 978-3-540-30199-8. DOI: 10.1007/978-3-540-30199-8_16. URL: https://link.springer.com/chapter/10.1007/978-3-540-30199-8_16 (visited on 03/27/2018).

[61] Dan Peng, Fan Wu, and Guihai Chen. "Pay As How Well You Do: A Quality Based Incentive Mechanism for Crowdsensing." In: *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc '15. New York, NY, USA: ACM, 2015, pp. 177–186. ISBN: 978-1-4503-3489-1. DOI: 10.1145/2746285.2746306. URL: http://doi.acm.org/10.1145/2746285.2746306 (visited on 10/20/2016).

[62] Ramya Raghavendra, Elizabeth M. Belding, Konstantina Papagiannaki, and Kevin C. Almeroth. "Understanding Handoffs in Large Ieee 802.11 Wireless Networks." In: *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. IMC '07. 00022. New York, NY, USA: ACM, 2007, pp. 333–338. ISBN: 978-1-59593-908-1. DOI: 10.1145/1298306.1298353. URL: http://doi.acm.org/10.1145/1298306.1298353 (visited on 09/22/2016).

[63] I. Ramani and S. Savage. "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks." In: *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.* Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Vol. 1. 00665. Mar. 2005, 675–684 vol. 1. DOI: 10.1109/INFCOM.2005.1497933.

[64] Piotr Sapiezynski, Arkadiusz Stopczynski, Radu Gatej, and Sune Lehmann. "Tracking Human Mobility Using WiFi Signals." In: *PLoS ONE* 10.7 (July 1, 2015). 00013, e0130824. DOI: 10.1371/journal.pone.0130824. URL: http://dx.doi.org/10.1371/journal.pone.0130824 (visited on 01/26/2016).

[65] A. Sathiaseelan, C. Rotsos, C. S. Sriram, D. Trossen, P. Papadimitriou, and J. Crowcroft. "Virtual Public Networks." In: *2013 Second European Workshop on Software Defined Networks*. 2013 Second European Workshop on Software Defined Networks. 00015. Oct. 2013, pp. 1–6. DOI: 10.1109/EWSDN.2013.7.

[66] Arjuna Sathiaseelan, Jon Crowcroft, Mutray Goulden, Christian Greiffenhagen3, Richard Mortier, Gorry Fairhurst, and Derek McAuley. "PAWS: Public Access WiFi Service." In: Third Annual Digital Economy All Hands Conference – Digital Futures. 00012. Oct. 23, 2012. URL: https://abdn.pure.elsevier.com/en/publications/paws-public-access-wifi-service (visited on 03/29/2018).

[67] Sayandeep Sen, Jongwon Yoon, Joshua Hare, Justin Ormont, and Suman Banerjee. "Can They Hear Me Now?: A Case for a Client-Assisted Approach to Monitoring Wide-Area Wireless Networks." In: *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. IMC '11. 00044. New York, NY, USA: ACM, 2011, pp. 99–116. ISBN: 978-1-4503-1013-0. DOI: 10.1145/2068816.2068827. URL: http://doi.acm.org/10.1145/2068816.2068827 (visited on 04/23/2018).

[68] D. Shehadeh, N. Montavont, T. Kerdoncuff, and A. Blanc. "Minimal Access Point Set in Urban Area Wifi Networks." In: *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt). 00000. May 2015, pp. 221–228. DOI: 10.1109/WIOPT.2015.7151076.

[69] Jinghao Shi, Zhangyu Guan, Chunming Qiao, Tommaso Melodia, Dimitrios Koutsonikolas, and Geoffrey Challen. "Crowdsourcing Access Network Spectrum Allocation Using Smartphones." In: *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*. HotNets-XIII. 00016. New York, NY, USA: ACM, 2014, 17:1–17:7. ISBN: 978-1-4503-3256-9. DOI: 10.1145/2670518.2673866. URL: http://doi.acm.org/10.1145/2670518.2673866 (visited on 04/20/2018).

[70] Jinghao Shi, Lei Meng, Aaron Striegel, Chunming Qiao, Dimitrios Koutsonikolas, and Geoffrey Challen. "A Walk on the Client Side: Monitoring Enterprise {Wifi} Networks Using Smartphone Channel Scans." In: Proceedings of INFOCOM 2016. 00001. 2016. URL: https://blue.cse.buffalo.edu/papers/2016/infocom2016-scans/infocom2016-scans.pdf (visited on 05/15/2016).

[71] Minho Shin, Arunesh Mishra, and William A. Arbaugh. "Improving the Latency of 802.11 Hand-Offs Using Neighbor Graphs." In: *Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services*. MobiSys '04. 00399. New York, NY, USA: ACM, 2004, pp. 70–83. ISBN: 978-1-58113-793-4. DOI: 10.1145/990064.990076. URL: http://doi.acm.org/10.1145/990064.990076 (visited on 03/27/2018).

[72] Sangho Shin, Andrea G. Forte, Anshuman Singh Rawat, and Henning Schulzrinne. "Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs." In: *Proceedings of the Second International Workshop on Mobility Management & Wireless Access Protocols*. MobiWac '04. 00386. New York, NY, USA: ACM, 2004, pp. 19–26. ISBN: 978-1-58113-920-4. DOI: 10.1145/1023783.1023788. URL: http://doi.acm.org/10.1145/1023783.1023788 (visited on 03/27/2018).

[73] Joel Sommers and Paul Barford. "Cell vs. WiFi: On the Performance of Metro Area Mobile Connections." In: *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*. IMC '12. 00098. New York, NY, USA: ACM, 2012, pp. 301–314. ISBN: 978-1-4503-1705-4. DOI: 10.1145/2398776.2398808. URL: http://doi.acm.org/10.1145/2398776.2398808 (visited on 10/20/2016).

[74] A. Di Stefano, A. Scaglione, G. Terrazzino, I. Tinnirello, V. Ammirata, L. Scalia, G. Bianchi, and C. Giaconia. "On the Fidelity of IEEE 802.11 Commercial Cards." In: *First International Conference on Wireless Internet (WICON'05)*. First International Conference on Wireless Internet (WICON'05). 00024. July 2005, pp. 10–17. DOI: 10.1109/WICON.2005.23.

[75] Aaron Striegel, Shu Liu, Lei Meng, Christian Poellabauer, David Hachen, and Omar Lizardo. "Lessons Learned from the Netsense Smartphone Study." In: *Proceedings of the 5th ACM Workshop on HotPlanet*. HotPlanet '13. New York, NY, USA: ACM, 2013, pp. 51–56. ISBN: 978-1-4503-2177-8. DOI: 10.1145/2491159.2491171. URL: http://doi.acm.org/10.1145/2491159.2491171 (visited on 01/26/2016).

[76] Aaron Striegel, Shu Liu, Xueheng Hu, and Lei Meng. "LTE and WiFi: Experiences with Quality and Consumption." In: *Procedia Computer Science* 34 (2014). 00004, pp. 418–425. ISSN: 18770509. DOI: 10.1016/j.procs.2014.07.048. URL: http://linkinghub.elsevier.com/retrieve/pii/S187705091400903X (visited on 05/16/2018).

[77] Tingting Sun, Wade Trappe, and Yanyong Zhang. "Improved AP Association Management Using Machine Learning." In: *SIGMOBILE Mob. Comput. Commun. Rev.* 14.4 (Nov. 2010). 00003, pp. 4–6. ISSN: 1559-1662. DOI: 10.1145/1942268.1942271. URL: http://doi.acm.org/10.1145/1942268.1942271 (visited on 08/11/2015).

[78] Bheemarjuna Reddy Tamma, B. S. Manoj, and Ramesh R. Rao. "Traffic Sensing and Characterization in Multi-Channel Wireless Networks for Cognitive Networking." In: *Computer Networks* 56.7 (May 3, 2012). 00001, pp. 1968–1982. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2011.12.002. URL: http://www.

sciencedirect.com/science/article/pii/S1389128611004166 (visited on 02/10/2015).

[79] *The Zettabyte Era: Trends and Analysis*. 00063. URL: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html (visited on 04/23/2018).

[80] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley. "Facilitating Access Point Selection in IEEE 802.11 Wireless Networks." In: *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*. IMC '05. 00198. Berkeley, CA, USA: USENIX Association, 2005, pp. 26–26. URL: http://dl.acm.org/citation.cfm?id=1251086.1251112 (visited on 06/17/2016).

[81] H. Velayos and G. Karlsson. "Techniques to Reduce the IEEE 802.11b Handoff Time." In: *2004 IEEE International Conference on Communications*. 2004 IEEE International Conference on Communications. Vol. 7. 00305. June 2004, 3844–3848 Vol.7. DOI: 10.1109/ICC.2004.1313272.

[82] Hai L. Vu and Taka Sakurai. "Collision Probability in Saturated IEEE 802.11 Networks." In: *In Australian Telecommunication Networks and Applications Conference*. 00050. 2006.

[83] Cheng Wang, David S. Hachen, and Omar Lizardo. "The Co-Evolution of Communication Networks and Drinking Behaviors." In: *2013 AAAI Fall Symposium Series*. 2013 AAAI Fall Symposium Series. 00007. Nov. 12, 2013. URL: https://www.aaai.org/ocs/index.php/FSS/FSS13/paper/view/7404 (visited on 05/17/2018).

[84] Xueheng Hu, Lixing Song, Dirk Van Bruggen, and Aaron Striegel. "Is There WiFi Yet?: How Aggressive Probe Requests Deteriorate Energy and Throughput." In: *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*. IMC '15. 00006. New York, NY, USA: ACM, Oct. 2015, pp. 317–323. ISBN: 978-1-4503-3848-6. DOI: 10.1145/2815675.2815709. URL: http://doi.acm.org/10.1145/2815675.2815709 (visited on 02/04/2016).

[85] Yiannis Yiakoumis, Manu Bansal, Adam Covington, Johan van Reijendam, Sachin Katti, and Nick McKeown. "BeHop: A Testbed for Dense WiFi Networks." In: *SIGMOBILE Mob. Comput. Commun. Rev.* 18.3 (Jan. 2015). 00026, pp. 71–80. ISSN: 1559-1662. DOI: 10.1145/2721896.2721912. URL: http://doi.acm.org/10.1145/2721896.2721912 (visited on 01/23/2015).

[86] Ji-Hoon Yun and Seung-Woo Seo. "Collision Detection Based on Transmission Time Information in IEEE 802.11 Wireless LAN." In: *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*. Fourth Annual IEEE International Conference on Pervasive Comput-

ing and Communications Workshops (PERCOMW'06). 00019. Mar. 2006, 5 pp.–414. DOI: 10.1109/PERCOMW.2006.29.

[87] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. "A Practical SNR-Guided Rate Adaptation." In: *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*. IEEE INFOCOM 2008 - The 27th Conference on Computer Communications. 00223. Apr. 2008. DOI: 10.1109/INFOCOM.2008.274.

[88] L. Zhang, L. Zhao, Z. Wang, and J. Liu. "WiFi Networks in Metropolises: From Access Point and User Perspectives." In: *IEEE Communications Magazine* 55.5 (May 2017). 00000, pp. 42–48. ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600262.

[89] Chen Zhao and Cunqing Hua. "Traffic-Load Aware User Association in Dense Unsaturated Wireless Networks." In: *2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*. 2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP). 00001. Oct. 2014, pp. 1–6. DOI: 10.1109/WCSP.2014.6992149.

[90] *iPass Wi-Fi Growth Map*. 00002. URL: https://www.ipass.com/wifi-growth-map/ (visited on 04/23/2018).

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Techniques et métriques non intrusives pour caractériser les réseaux Wi-Fi

Mots clés : Caractérisation des réseaux WiFi, Ecoute passive, Fingerprinting, Crowd-sourcing, Sélection de points d'accès, Economie d'énergie

Résumé : Aujourd'hui, les appareils mobiles sont présents dans le monde entier. Ces appareils permettent aux utilisateurs d'accéder à l'Internet notamment par l'intermédiaire des réseaux WiFi. La diversité et le nombre de déploiements sans coordination centrale (y compris les utilisateurs à leur domicile) conduit à des déploiements qu'on peut qualifier de chaotiques. En conséquence, les réseaux WiFi sont largement déployés avec une forte densité dans les zones urbaines. Dans ce contexte, les utilisateurs et les opérateurs tentent d'exploiter ces déploiements pour obtenir une connectivité omniprésente, et éventuellement d'autres services. Cependant, pour tirer parti de ces déploiements, il faut des stratégies pour identifier les réseaux utilisables et choisir les plus adaptés aux besoins. Pour cela, nous étudions le processus de découverte des réseaux dans le contexte de ces déploiements. Ensuite, nous présentons une plateforme de partage de mesures sans fil, un système d'information collaboratif où les stations mobiles recueillent des mesures du réseau et les envoient à un système central. En rassemblant mesures provenant de différents utilisateurs, la plateforme donne accès à des caractéristiques du déploiement précieuses. Nous évaluons l'utilité de cette plateforme collaborative grâce à deux applications : (1) l'ensemble minimal de points d'accès, afin de réduire l'énergie nécessaire pour offrir une couverture WiFi dans une zone donnée. (2) l'optimisation des paramètres de recherche de réseau, afin de réduire le temps nécessaire pour découvrir les réseaux existants. Ensuite, nous étudions une méthode passive pour déterminer si un réseau fonctionne dans un canal saturé.

Title : Metrics and non-intrusive techniques to characterize Wi-Fi networks

Keywords: Wi-Fi characterization, Passive monitoring, Fingerprinting, Crowd-sourcing, AP selection, Energy saving

Abstract: Nowadays, mobile devices are present worldwide, with over 4.40 Billion devices globally. These devices enable users to access the Internet via wireless networks. Different actors (e.g., home users, enterprises) are installing WiFi networks everywhere, without central coordination, creating chaotic deployments. As a result, WiFi networks are widely deployed all over the world, with high access point (AP) density in urban areas. In this context, end-users and operators are trying to exploit these dense network deployments to obtain ubiquitous Internet connectivity, and possibly other services. However, taking advantage of these deployments requires strategies to gather and provide information about the available networks. In this dissertation, we first study the network discovery process within the context of these deployments. Then, we present the Wireless Measurements Sharing Platform, a collaborative information system, to which mobile stations send simple network measurements that they collected. By gathering and processing several network measurements from different users, the platform provides access to valuable characteristics of the deployment. We evaluate the usefulness of this collaborative platform thanks to two applications: (1) the minimal access point set, to reduce the energy needed to offer WiFi coverage in a given area. (2) The optimization of the scanning parameters, to reduce the time a mobile station needs for the network discovery. Finally, we describe a method to identify whether an AP operates in a saturated channel, by passively monitoring beacon arrival distribution.