Wireless passive measurements: tool, redundancy, measurements, and analyses

Mohammad Imran Syed

École doctorale Informatique, Télécommunications et Électronique de Paris

A dissertation submitted in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Computer Science**

Wireless passive measurements: tool, redundancy, measurements, and analyses

presented by :

Mohammad Imran SYED

to the committee composed of :

Committee president : Olivier Fourmaux, Prof. Sorbonne Université

Luís Henrique M. K. Costa, Prof. Univ. Fed. do Rio de Janeiro	Reviewer
Thi-Mai-Trang Nguyen, Prof. Université Sorbonne Paris Nord	Reviewer
Lila Boukhatem, Assoc. Prof. Université Paris-Saclay	Examiner
Olivier Fourmaux, Prof. Sorbonne Université	Examiner
Emmanuel Lochin, Prof. École Nationale de l'Aviation Civile	Examiner
Marcelo Dias de Amorim, DR CNRS	Supervisor
Anne Fladenmuller, Prof. Sorbonne Université	Supervisor

Paris, 5th September 2023

Ph.D. Thesis of Sorbonne Université

This page is intentionally left blank

École doctorale Informatique, Télécommunications et Électronique de Paris

Thèse présentée en vue de l'obtention du diplôme de
Docteur en **Informatique**

Mesures passives sans fil : outil, redondance, mesures et analyses

présenté par :

Mohammad Imran SYED

devant le jury composé de :

Président du jury : Olivier Fourmaux, Prof. Sorbonne Université

Luís Henrique M. K. Costa, Prof. Univ. Fed. do Rio de Janeiro	Rapporteur
Thi-Mai-Trang Nguyen, Prof. Université Sorbonne Paris Nord	Rapporteuse
Lila Boukhatem, Assoc. Prof. Université Paris-Saclay	Examinatrice
Olivier Fourmaux, Prof. Sorbonne Université	Examinateur
Emmanuel Lochin, Prof. École Nationale de l'Aviation Civile	Examinateur
Marcelo Dias de Amorim, DR CNRS	Directeur
Anne Fladenmuller, Prof. Sorbonne Université	Directrice

Paris, le 5 septembre 2023

Thèse de doctorat de Sorbonne Université

Supervisors
Dr. Marcelo Dias de Amorim (Director of Research, CNRS, France)
Dr. Anne Fladenmuller (Professor, Sorbonne Université, France)

Members of the Committee
Dr. Lila Boukhatem (Associate Professor, Université Paris-Saclay, France)
Dr. Olivier Fourmaux (Professor, Sorbonne Université, France)
Dr. Emmanuel Lochin (Professor, Ecole Nationale de l'Aviation Civile, Toulouse, France)
Dr. Luís Henrique Maciel Kosmalski Costa (Professor, Universidade Federal do Rio de Janeiro, Brazil)
Dr. Thi-Mai-Trang Nguyen (Professor, Université Sorbonne Paris Nord, France)

Printing and binding : Sorbonne Université

Acknowledgement

This work has been partially funded by the ANR MITIK project, French National Research Agency (ANR), PRC AAPG2019.

Failure is simply the opportunity to begin again, this time more intelligently.

Henry Ford

A pessimist sees the difficulty in every opportunity ; an optimist sees the opportunity in every difficulty.

Winston Churchill

Do not waste water even if you were at a running stream.

Prophet Mohammad (peace be upon him)

Seek knowledge from the cradle to the grave.

Prophet Mohammad (peace be upon him)

Remerciements

Je voudrais exprimer ma gratitude la plus profonde à mes encadrants Anne FLADENMULLER et Marcelo DIAS DE AMORIM pour leur soutien, leur guidance et leur encouragement indéfectibles tout au long de ma thèse. Leurs compétences et leurs idées ont été inestimables pour orienter la direction de mes recherches. Les réunions remontant le moral après les refus de publications en cascade, en raison de la confiance qu'ils avaient en moi et de la qualité de notre travail, m'ont énormément aidé. Ce sont les meilleurs codirecteurs de thèse qu'un doctorant puisse souhaiter, mais ce sont également de très bonnes personnes. L'aide qu'ils m'ont apportée pour la recherche d'un appartement au milieu de ma thèse, le temps qu'ils ont pris pendant leurs vacances pour prendre de mes nouvelles de ma famille au Pakistan lors de la canicule intense et des terribles inondations de 2022, leur côté humain m'a touché profondément.

Je voudrais également remercier les membres de mon jury de thèse, Lila BOUKHATEM, Olivier FOURMAUX, Emmanuel LOCHIN, Luís Henrique MACIEL KOSMALSKI COSTA et Thi-Mai-Trang NGUYEN pour leurs commentaires réfléchis et leurs critiques constructives. Leurs contributions m'ont aidé à améliorer mon travail et à le rendre plus impactant.

J'ai de très bons souvenirs de mon séjour au LIP6. Je remercie Maria POTOP-BUTUCARU pour son accueil au sein de l'équipe NPA. Je suis reconnaissant envers tous mes collègues et aussi les doctorants, Sadia KHIZAR, Gabriel Antonio FONTES REBELLO, Ricardo LOPEZ DAWN, Effrosyni PAPANASTASIOU, Nabil MAKAREM, Nouamane ARHACHOUI, Tengfei AN, Alexandre PHAM et Boris CHAN YIP HON, pour leur camaraderie, leur stimulation intellectuelle et leur soutien. Je dois mentionner le soutien de Sadia et de Gabriel pour la relecture de mes articles. Je suis redevable à Anne FLADENMULLER et Prométhée SPATHIS d'avoir réalisé mon rêve en me donnant l'opportunité d'enseigner et à Maria POTOP-BUTUCARU de me donner l'opportunité de

co-organiser la journée des doctorant(e)s/postdoctorant(e)s/stagiaires NPA. Je suis reconnaissant envers nos ingénieurs réseau du laboratoire, Pierre-Emmanuel LE ROUX et Konstantin KABASSANOV, qui ont été d'une aide immense pour configurer mon ordinateur portable, le réparer rapidement, acheter l'équipement pour les expérimentations et surtout pour l'utilisation du cluster et la restauration des fichiers d'analyse à partir de la base de données vers le cluster. Je suis également reconnaissant envers l'équipe administrative du LIP6 qui a toujours été sympa et de soutien.

Je suis reconnaissant envers mes collègues du projet ANR-MITIK pour leurs commentaires constants et leur appréciation de notre travail. Je suis particulièrement reconnaissant envers mon ami et collègue de projet Abhishek Kumar MISHRA ; les conversations que nous avons eues, les idées que nous avons partagées et les défis que nous avons relevés ensemble ont rendu ce voyage plus significatif et agréable, sans oublier notre envie de nourriture indienne et pakistanaise après le travail.

Je dois également remercier beaucoup d'autres amis : Habeeb, Saad, Tahseen, Abid, Adeel, Ejaz, Hasnain, Zain, Ali Tahir, Farooque, Ghasem, Hasnat, Karine, Emmanuel, Arsalan, Waleed, Mohsin, Danial, Rohan, Ali Qazi. Un grand merci à beaucoup de personnes et je suis sûr que j'en oublie quelques-unes.

Enfin, je voudrais remercier ma famille pour leur amour et leur soutien inconditionnels. Je suis éternellement reconnaissant envers mes parents, M. Aqeel Hussain SYED et Mme. Andleeb Fatima SYEDA, ainsi que mon frère Mohammad Asghar SYED, mes sœurs, Asma Batool SYED et Saima Batool SYED, et également à mon beau-frère Qasim Mehdi SHAH. Leur confiance en moi et leur encouragement m'ont donné la force de persévérer face aux obstacles. Le dernier, mais non le moindre, je suis extrêmement reconnaissant envers ma charmante épouse, Mme. Noor Narjis BATOOL SYED. Que ce soit pour la nourriture fraîche et le thé les jours fatigants ou pour l'aide à l'assemblage de l'équipement pour l'expérimentation à la maison, il n'aurait vraiment pas été possible de traverser ces trois années sans son soutien moral et émotionnel.

À tous ceux qui m'ont aidé en cours de route, je vous offre mes sincères remerciements. Cette réalisation n'aurait pas été possible sans vous.

Résumé

La compréhension du trafic sans fil est fondamentale pour améliorer les réseaux et concevoir des algorithmes et des protocoles avancés. Dans ce contexte, les mesures passives ont l'avantage sur les mesures actives, car elles ne dépendent d'aucune modification des équipements réseau existants. Elles sont souvent moins coûteuses et plus faciles à déployer que d'autres méthodes. Cette approche consiste à surveiller le support sans fil et à collecter des données sur divers paramètres de réseau, tels que la force du signal, l'occupation des canaux et la perte de paquets. Elle consiste à déployer plusieurs sniffeurs dans la zone cible (les sniffeurs sont des dispositifs fonctionnant en « monitor mode » qui collectent les paquets sans fil indépendamment de leur nature). Cependant, l'un des principaux défis des mesures passives est d'assurer la complétude de la trace, c'est-à-dire la capacité à collecter un ensemble de données complet et précis. Nous montrons qu'un seul sniffeur ne peut pas capturer tout le trafic en raison des caractéristiques inhérentes du support sans fil, où l'environnement peut être hautement dynamique et imprévisible.

Il existe plusieurs facteurs qui peuvent affecter la complétude de la trace dans les mesures passives sans fil. Celles-ci incluent des facteurs environnementaux, tels que les interférences provenant d'autres dispositifs sans fil, les changements dans l'environnement physique (comme les objets en mouvement) et les variations de propagation du signal sans fil dues aux changements des conditions atmosphériques. De plus, des problèmes avec l'équipement de mesure lui-même, tels que des erreurs de calibration ou des problèmes de traitement des données, peuvent également affecter la complétude de la trace.

L'importance de la complétude de la trace dans les mesures passives sans fil ne peut être surestimée. Des données inexactes ou incomplètes peuvent conduire à des conclusions incorrectes sur les performances du réseau, ce qui peut avoir des implications significatives pour la planification, l'optimisation et le dépannage du réseau. Par exemple, des données incomplètes peuvent entraîner des

opportunités manquées pour identifier et résoudre des problèmes de réseau, ainsi qu'une reconstruction de trajectoire incorrecte ou incomplète.

Dans cette thèse, nous étudions la qualité des traces capturées par des sniffeurs et examinons les améliorations résultantes en introduisant de la redondance dans le nombre de sniffeurs. Nous étudions l'impact des deux aspects suivants sur la qualité des traces sans fil : le nombre de sniffeurs et le type de matériel utilisé. Nous étudions la variation de l'indicateur de force du signal reçu (RSSI) et son impact sur l'estimation de la distance. L'analyse est facilitée par le développement d'un outil facilement utilisable et disponible appelé PyPal pour la synchronisation et la fusion de traces Wi-Fi collectées simultanément.

Mots-clefs

Mesures passives, réseaux sans fil, Wi-Fi, complétude de traces, fusion de traces, redondance, IEEE 802.11, estimation de la distance, RSSI.

Abstract

Understanding wireless traffic is fundamental for improving networks and designing advanced algorithms and protocols. In this context, passive measurements have the edge over active measurements, as there is no requirement for any modification in existing network devices. Passive measurements are often less expensive and easier to deploy than other methods. This approach involves monitoring the wireless medium and collecting data on various network parameters, such as signal strength, channel occupancy, and packet loss. It consists of deploying multiple sniffers throughout the target area (sniffers are devices operating in monitor mode that collect the wireless packets regardless of their nature). However, one of the main challenges with passive measurements is ensuring trace completeness, or the ability to collect a complete and accurate dataset. We know that a single sniffer cannot capture all the traffic due to the inherent characteristics of the wireless medium where the environment can be highly dynamic and unpredictable.

Several factors can impact trace completeness in wireless passive measurements. These include environmental factors, such as interference from other wireless devices, changes in the physical environment (such as moving objects), and variations in wireless signal propagation due to changes in atmospheric conditions. Additionally, issues with the measurement equipment itself, such as calibration errors or data processing issues, can also impact trace completeness.

The importance of trace completeness in wireless passive measurements cannot be overstated. Inaccurate or incomplete data can lead to incorrect conclusions about network performance, which can have significant implications for network planning, optimization, and troubleshooting. For example, incomplete data can result in missed opportunities to identify and address network issues, and incorrect or incomplete trajectory reconstruction.

ABSTRACT

In this thesis, we study the quality of traces captured by a sniffer and investigate the resulting improvements by introducing redundancy in the number of sniffers. We explore the impact of the following two aspects on the quality of wireless traces: the number of sniffing devices and the type of hardware used. We study the variation in the Received Signal Strength Indicator (RSSI) and its impact on distance estimation. The analysis is helped by the development of a readily-usable and easily-available tool, called PyPal, for the synchronization and merging of Wi-Fi traces collected simultaneously.

Keywords

Passive measurements, wireless networks, Wi-Fi, trace completeness, trace merging, redundancy, IEEE 802.11, distance estimation, RSSI.

Overview

Contents

List of Figures

List of Tables

List of Definitions

1

Introduction

The number of smartphone users is an ever-increasing figure, and it is expected to reach 7,516 billion by 2026 [1]. Furthermore, as more and more devices connect to the Internet [2, 3], forecasts suggest that, by the end of 2023, there will be approximately 628 million public Wi-Fi hotspots available worldwide [4]. These numbers lead to an amplification of the topology dynamics and more challenging network management issues. Consequently, our dependence on efficient measurement techniques to precisely characterize the network and understand the mobility of users also increases.

1.1 Context

Air is the preferred and winning medium of choice because of portability, affordability, and ever-increasing data rates. Wireless networks are everywhere, and understanding their behavior to improve their performance is of utmost importance [5, 6, 7, 8, 9]. Nevertheless, measuring wireless traffic (wirelessly) is challenging because of the intrinsic volatile nature of the wireless links [10]. Although cellular operators produce a lot of location data, it is not publicly available. As a result, the research community still relies on a limited set of traces, which restrains the universe of possible observations. There is a need for wireless measurements to build traces that researchers can use to evaluate and improve networking approaches.

1.2 PROBLEM STATEMENT AND APPROACH

Actively collecting traffic is burdensome because it requires deploying tools (software or application) at different nodes, which are likely under the control of various administrative entities. For example, one has to either request permissions for deployment on access points in the target area or to gather traffic from smartphones, one can either deploy probes at all access points the user associates with or create a measurement application and ask users to install it on their devices [11]. The users that volunteer might be an insufficient sample of the population, leading to unreliable, biased, or inaccurate results [12].

An efficient alternative is to run passive measurements by deploying several *sniffers* (devices collecting wireless packets in monitor mode) throughout the target area [13, 14, 15] [1]. It is a low-cost and scalable measurement strategy that does not require bothering users with intrusive services. The concept of the ANR MITIK project, which we are part of, is to infer contact traces through non-intrusive methods like passive sniffing [16]. Nevertheless, due to wireless transmission constraints like multi-path, fading effects, or collisions, there is no guarantee a single sniffer can capture all the packets, therefore, leading to incomplete traces. In Figure 1.1, we illustrate a typical scenario where four sniffers ($s_1, \ldots, s_4$) do not have the same "view" of the wireless traffic because of capture misses. It leads to discrepancies in the measurements, and further analyses relying on such incomplete traces are likely to be flawed.

The solution to circumvent the problem relies on the use of *super-sniffers*. It consists of introducing redundancy in the system by tying two or more sniffers together to increase the probability that at least one of the sniffers captures a packet. In Figure 1.2, we show a three-redundant super-sniffer. The main question that we address is *how the level of redundancy helps improve the quality of the measure*. To answer this question, we propose a definition of a trace's

1. It is, however, essential to know which data one can sniff depending on the location of the measurement campaign while preserving the privacy of the users.

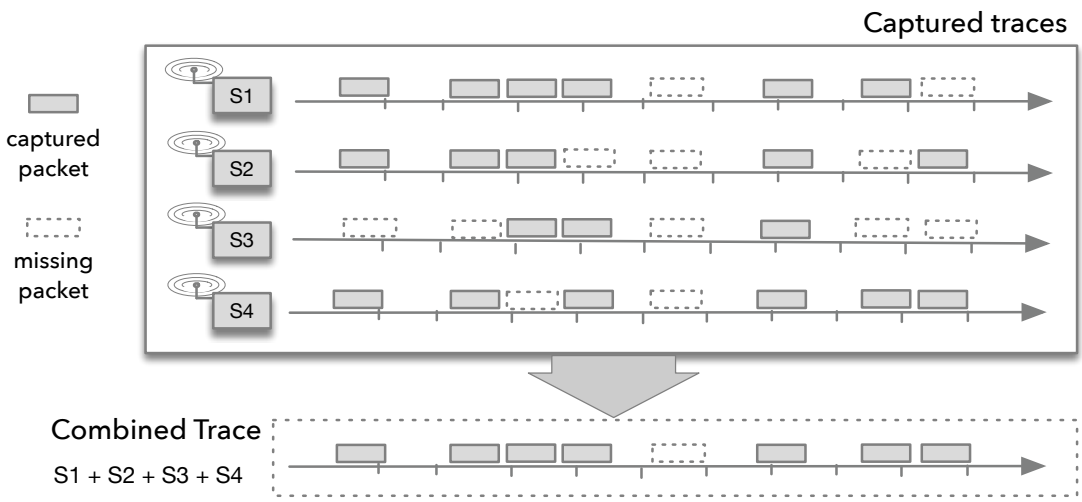


Figure 1.1. – Trace completeness. Sniffers may miss several packets because of the nature of the wireless medium. We need to combine individual traces to get as close as possible to the complete trace.

completeness and evaluate it through real-world experiments. Although we focus on Wi-Fi traces, our methodology is general and can be applied to other technologies.

1.3 WHY DID WE ADOPT AN EXPERIMENTAL METHODOLOGY?

On the one hand, there are datasets available on platforms like Crawdad [17] (now part of IEEE DataPort), but there is little knowledge of potentially what post-processing could have been performed before the dataset is made available publicly. There can also be gaps among these datasets that can unknowingly result in incomplete results and analyses [18]. On the other hand, there are quite a few network simulators [19], OMNET++. The issue with simulators is that these systems may not accurately capture the complexity and variability of real-world environments, and their results may not always be representative of real-world performance. Additionally, simulators are

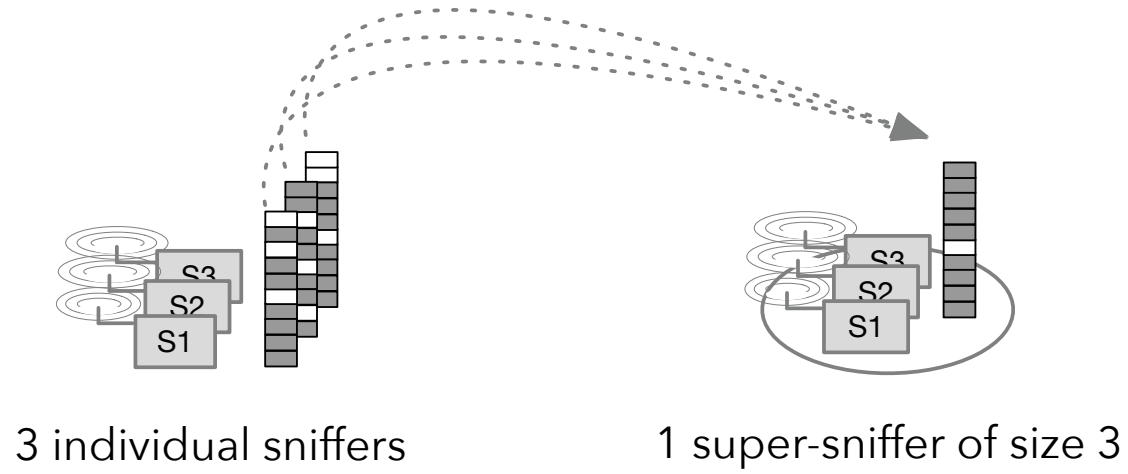


Figure 1.2. – Three sniffers forming a super-sniffer.

limited by the accuracy of the underlying models and assumptions used in the simulation, which may not always hold true in practice [20]. Real-world experiments find a preference among the researchers for the following reasons:

— **Validation of theoretical models:** Real-world experimentation helps to validate the theoretical models developed for wireless networks. Theoretical models are often based on assumptions that may not hold true in the real world. By conducting real-world experiments, researchers can verify if their assumptions are correct and adjust their models accordingly.
— **Performance evaluation:** Real-world experimentation allows researchers to evaluate the performance of wireless networks under real-world conditions. This helps to identify the strengths and weaknesses of different wireless network technologies and protocols and to determine which ones are best suited for different applications and scenarios.
— **Identification of problems:** Real-world experimentation can help to identify problems that may not be apparent in simulation or modeling

studies. For example, interference from other wireless devices or environmental factors such as buildings, trees, or terrain can affect the performance of wireless networks in ways that cannot be accurately modeled.

We, therefore, rely on real-world experimentation to bring to attention a wireless sniffing shortcoming[2]. It also allows us to highlight the extent of the impact. As we will see in the next chapters, we evaluate the quality of capture of individual sniffers with up to fourteen sniffers. We use sniffers composed of two models of Raspberry Pi (3B and 4B) to ensure diversity in the capturing devices. We take into consideration two different scenarios, office and residential, with varying traffic loads. The individual sniffers achieve relatively low completeness (between 30% and 54% depending on the scenario), which confirms the need for redundancy. Secondly, commercial off-the-shelf (COTS) devices such as Raspberry Pi are powerful enough to play the role of a sniffer in each scenario. The sniffers uniformly miss packets, indicating that the environment essentially dictates the quality of the sniffing process.

1.4 CONTRIBUTIONS

This thesis introduces the need for redundancy in the number of sniffers in wireless passive measurements based on real-world experiments, and therefore, has three contributions.

Contribution # :1 Trace completeness and the need for redundancy

Trace completeness is an important aspect of wireless measurements and the challenge that we face is the low completeness of individual sniffers. There is,

2. Although we capture only public headers, they still contain MAC addresses. We ensure privacy by using a hashing and truncation technique.

therefore, a need to have some reference for the evaluation of completeness. The contribution for trace completeness is as follows:

— **Metric for completeness.** We propose a formal definition of completeness that incorporates the notion of redundancy. We introduce the term *super-sniffer* and formulate the completeness of the super-sniffer.
— **Experimental evaluation.** We follow an experimental approach to assess the completeness achieved by individual sniffers, hence proving a single sniffer is not sufficient and there is a need for redundancy. We study the behavior of the temporal variation in completeness over time.
— **Traffic load.** We investigate the effect of varying traffic loads on the completeness of traces. The amount of traffic varies at different times of the day as the number of users/devices keeps varying.
— **Performance of individual sniffers.** We analyze the performance of individual sniffers to highlight the fact that all sniffers do not perform in the same manner and to help us identify the sniffers that perform consistently poorly despite the conditions in the wireless medium changing over time.
— **Pairwise completeness.** The sniffers complement each other in improving the quality of passive measurements when considering them in pairs. We call this pairwise completeness and evaluate it for eleven sniffers after cleaning the dataset.

Contribution # 2: PyPal, a tool for Wi-Fi trace synchronization and merging

We introduce the concept of trace completeness along with its mathematical representation. However, there lies a challenge for the evaluation of completeness: time-synchronization of the traces.

Since the individual sniffers have a local clock and are not connected to the Internet at the time of capture, there is a need for the traces to be time-synchronized before the merging and redundancy evaluation analysis can be

done. Therefore, the second contribution of this thesis is a Python tool, called PyPal, for the time-synchronization and merging of the traces [21]. Apart from that, the tool can concatenate the synchronized traces as well as produce per-MAC-address traces. We explain it in detail in Section 4.2.

Contribution # 3: Leveraging redundancy in distance estimation

Distance estimation and localization are a huge part of the research, particularly in sensor networks and the Internet of Things (IoT). The Received Signal Strength Indicator (RSSI) values are of interest to detect the presence of a node in a target area and evaluate its position relative to the sniffer [22, 23, 24, 25]. Furthermore, these values are available in most operating systems in a straightforward fashion. However, the downside of using the RSSI concerns its accuracy. Although there are some workarounds like machine learning to reduce the error in distance estimation [26, 27], the applicability of the RSSI is still questioned [28, 29, 30]. The challenge that the individual traces pose is the incoherence of RSSI values, bursts of consecutive packets missing, and errors in distance estimation.

As a part of the ANR MITIK project [16], we collect Wi-Fi traces through passive measurements to capture the mobility of wireless nodes in a given area. In the project, we focus on probe request messages, the only messages a user's device sends when it is not associated with any Wi-Fi access point. The problem is that their sending rate can be as low as 55 packets per hour or as high as 2,000 packets per hour, depending on the device [31]. Missing these probe requests may have severe consequences on the quality of the measurement campaign, as it may lead to nodes not being detected or biased mobility estimations.

We collect wireless traces using several sniffers and one source node generating Wi-Fi traffic. The source is the node we want to characterize. We co-locate the sniffers (i.e., introduce redundancy) to investigate the consistency of the RSSI values. We do the measurements outdoors and monitor the traffic generated

by a controlled source for which we know the ground truth of its distance from the sniffers. In our experiments, we place the source node at multiple distances from the sniffers and make the following observations:

— All sniffers do not necessarily miss the same packet. It means the packet is undoubtedly not lost because of the collisions at the sniffers.
— The number of consecutive packets missed by a sniffer can be huge at times. Moreover, the sniffers miss the capture for several seconds, which is problematic if one has to analyze mobility.
— At times, there is incoherence in RSSI values of the same packet captured by different sniffers, which leads to frequent errors in distance estimation.

The redundancy improves the capture quality by *reducing the number of packets missed* and *the gaps due to consecutive packet misses*. Furthermore, this leads us to the third *contribution of this thesis*, we could *evaluate the problem of the incoherence of RSSI values and reduce the error in distance estimation*.

1.5 THESIS OUTLINE

The rest of the thesis is organized as follows. We provide the background of the topic and the state of the art in Chapter 2. In Chapter 3, we provide the experimental evidence that a single sniffer is not enough for wireless passive measurements highlighting the need for redundancy. In Chapter 4, we provide the definition and mathematical representation of trace completeness for super-sniffers. Moreover, we differentiate between *relative* and *absolute completeness* and provide the experimental evaluation. This chapter also includes the detailed working of our tool PyPal. We explain the application of redundancy in wireless passive measurements in the shape of distance estimation in Chapter 5. Finally, we conclude the thesis and mention future work in Chapter 6.

2

Background and positioning

The increasingly ongoing need for passive measurements in wireless networks for purposes such as network management and diagnosis, new protocols, localization, location-based services, and trajectory reconstruction has driven the need for improving the quality of measurements and traces captured. In this chapter, we review the work that has been carried out in this regard.

2.1 Wireless measurements: active vs. passive

Researchers carry out measurement experiments for several reasons such as troubleshooting or improving the network, realizing new network protocols, assessment of wireless sniffing, trajectory reconstruction, etc. [5, 6, 7, 8, 9, 32, 33, 34, 35]. However, there is a choice to be made between active and passive measurements for the mode of experiments.

Active measurements play a major role in the speed testing of the networks, as well as in experiments involving cellular networks. Smart devices such as mobile phones and tablets make use of the cellular network and users can be requested to volunteer and help in measurements using an application [36]. The dependence on users to install and run an application on their mobile devices in active measurements can result in unwanted bias in the traces collected.

On the other hand, for Wi-Fi, there are comparatively more types of devices such as computers, smart home appliances, sensors, etc. apart from smartphones. Obtaining access to these devices or finding volunteers willing to install measurement applications is an exceedingly challenging task. Hence, this makes passive measurements more relevant [37]. Passive measurements give the researchers an opportunity to carry out measurements without depending on any infrastructure or people for crowd-sourcing the data. There is also no impact on actual traffic in the network as no redundant traffic is generated that can impact the actual traffic of the users circulating in the medium.

We summarize the advantages of passive measurements as follows:

— **Non-intrusive.** Non-intrusive in wireless measurements refers to the ability to collect information or data without interfering with the wireless system one intends to measure. It means doing the measurements without interrupting or modifying the infrastructure to capture the packets being transmitted or received by the devices in the coverage area. Moreover, there is no need for the users to install any application or software on their devices.

— **No disruption.** As there is no traffic generated for doing the measurements, there is no disruption of the actual user traffic. Unlike active measurements, there is no fear of causing congestion either, which can not only introduce a bias in the results but also impact the users' Quality of Experience (QoE).

— **Low complexity.** The complexity level is low since there is no need to create any specific software or tool to install on certain devices. There are already tools such as tcpdump [38], tshark [39], Wireshark [40], and scapy [41] available specifically for passive measurements.

— **Scalability.** The scalability of passive measurements arises from their ability to observe wireless network traffic without requiring any interaction with the network or its users, thereby avoiding any impact on the network's operation. The number of measurement devices can be

increased to make the measurements more scalable; it comes at a financial cost but there is a trade-off between the cost and the measurement. Whereas, scalability is becoming increasingly difficult in active measurements due to the need to either have access to more infrastructure devices in the wireless network or more users volunteering to install the application/software on their personal devices.

— **Comprehensive.** Passive measurements provide a detailed view of the network [42]. Those yield an unbiased view of the wireless network performance and are used to analyze various aspects of the network, including network utilization, traffic patterns, protocol usage, and application behavior.

2.2 BUILDING AN EFFICIENT SNIFFER

There are several small and low-cost computers available, including ESP32 [43], Arduino [44], BeagleBone [45], UDOO [46], and Raspberry Pi [47]. Each of these computers has different specifications and supports various operating systems.

Among these devices, ESP32 is the only one that can be used as a Wi-Fi sniffer without the use of any external Wi-Fi adapter. Its internal Wi-Fi card can be set to promiscuous mode, allowing it to capture packets using libraries such as ESP32-Paxcounter and the ESP32-Sniffer. However, ESP32 is unable to capture control and error frames [48].

Arduino is not typically used as a Wi-Fi sniffer due to the absence of built-in Wi-Fi connectivity and its unsuitability for high-performance networking applications. However, an external adapter can be used to provide Wi-Fi connectivity, and it has found the most usage in wireless sensor networks.

Similarly, BeagleBone and UDOO boards require an external Wi-Fi adapter to work as a sniffer since their internal cards can not be put in monitor mode. However, those are rarely used as sniffers except for some exceptions [49].

Raspberry Pi is widely used in the world of research, especially for wireless measurements, but it also needs an external Wi-Fi adapter that can be put into monitor mode for measurements. We discuss the widespread use of RPi in various fields.

Selection of a sniffer

The Raspberry Pi (RPi hereafter) is a series of low-cost, single-board computers developed by the Raspberry Pi Foundation in the United Kingdom [47]. These boards are designed to be user-friendly and can be utilized for a wide range of applications, for example, hobby projects, educational purposes, and commercial applications. Researchers have also found the RPi to be a versatile tool for conducting experiments across a variety of fields, including mobility [50, 51], Internet of Things (IoT)[52, 53], wireless sensor networks[54, 55, 56], privacy [57, 58], network monitoring and security [59, 60], communication systems [61], biotechnology [62], and biology [63]. The RPi has also proved to be a valuable resource in the field of education, particularly in subjects such as Physics [64], Chemistry [65, 66], and image processing [67].

Researchers commonly use RPi as a sniffer for wireless measurements. In their study, Turk et al. describe a technique that utilizes an RPi and a USB wireless adapter to capture and analyze wireless network traffic, enabling real-time wireless packet monitoring at a low cost [68]. Li et al. also employ an RPi3 for measurements to investigate the impact of channel settings and access point configurations on wireless sniffing performance [32]. Additionally, RPi has been utilized in calculating the waiting time of passengers at bus stops [69], determining public transport occupancy on a bus [70], and estimating the number of travelers by collecting Wi-Fi signals at a bus stop [71].

Silva et al. conducted a study on the impact of privacy preservation on wireless frames during the sniffing process. They developed an on-the-fly hashing module using the Scapy tool and evaluated the results on an RPi platform [57]. Jais et al. used an RPi-based access point to sniff RSSI values for localization purposes [72]. Friess designed a multichannel sniffing system using RPi3 [73]. Minambres et al. conducted a study to demonstrate the suitability of RPi devices for wireless sniffing [74].

Discussion

RPi is a widely adopted technology in various domains, including education, due to its affordability, versatility, and processing capabilities. As previously mentioned, it is also a popular choice among researchers for conducting wireless measurement experiments, primarily as a sniffer. Previous studies have demonstrated that RPi devices are capable of performing wireless sniffing tasks effectively. Therefore, these features make it a suitable choice for building a sniffer for our real-world experiments.

2.3 TRACE QUALITY IMPROVEMENT

The presence of discrepancies in the traces captured by passive sniffing mainly due to the behavior of the wireless medium is a reality. Therefore, the researchers take some steps of either pre- or post-processing to improve the quality of the trace. This allows them to do an analysis more representative of the actual network (conditions). We take a look at works that are focused on improving the quality of wireless network traces captured through passive sniffing.

One such process is to merge the individual traces into a single trace. Xu et al. merge the individual traces into a single and then run an inference procedure to reconstruct the missing packets [75]. It needs at least one packet

of a conversation in a trace to infer the missing packets and its accuracy also depends on the capture percentage. The evaluation is dependent on a simulation where the process removes packets from the trace randomly.

Another approach used by Sammarco et al. is to compute inter-flow similarity for the traces collected from multiple sniffers. They use graph theory to determine the sequence in which the traces should be merged to achieve maximum completeness. This method can decrease the number of merge operations as a subset of traces can achieve the same result [76].

Schulman et al. estimate the number of missed packets using sequence numbers and re-transmission bit [77] but they do not capture traffic of their own and rely on datasets available on CRAWDAD [17]. The dependence on the re-transmission bit would create some bias since it is hard to infer how many packets are actually re-transmitted since these packets have the same sequence number.

Mahanti et al. examine the beacon and acknowledgment frames, MAC-layer sequence numbers, and placement of sniffer to address the incomplete traces [78]. They use the results from one sniffer to create a layout of four sniffers on three floors. The amount of packets captured in 24 hours is nearly the same as that captured by our sniffers in 10 minutes.

Discussion

Overall, these methods are aimed at improving the accuracy of wireless network analysis by addressing the incomplete traces captured through passive sniffing. However, there is less focus on the use of redundancy in the number of sniffers to study its impact on the level of trace completeness. Our work stands distinctive as firstly we prove that a single sniffer is not enough to capture a quality trace and then we focus on redundancy for trace completeness based on real-world experiments in an uncontrolled environment and perform an exhaustive analysis for different scenarios. Moreover, our solution

is more oriented toward contact traces and it can apply to different wireless technologies.

2.4 ANALYSIS OF TRACES

In order to analyze the impact of redundancy, we need a tool or system. Redundant sniffers operate independently and without Internet connectivity, resulting in a lack of global time synchronization. To evaluate redundancy, we require a time-synchronization tool that also provides a unified view of the traces. This section explores various tools and systems available in the literature.

Wit is a software tool designed to merge multiple views of monitors into a single view, and reconstruct missing packets using an engine based on formal language techniques. This engine can infer whether missing packets were received by the destination, by analyzing frames such as Association Request and Response [14].

PMSW is a passive monitoring system that relies on sequence numbers to infer the missing packets in a wireless sensor network. However, it only captures data and acknowledgment packets, leading to a complex synchronization solution [79]. There are no conversation, data, or association frames as we rely on probe requests for contact traces.

Garcia et al. develop a passive monitoring system called EPMOSt which focuses on election to choose the nodes of the target area for their packets to be captured by the sniffers but more in terms of energy consumption. The system is validated and tested on the MicaZ platform, and the results show that the energy consumption of the sniffers decreases by up to 69.3% when the election mechanism is used. The energy consumption decreases by 78.5% when the election mechanism and aggregation of headers are used. However,

it reduces the number of packets captured by 0.62% when 11 sniffers are used [13].

LiveNet provides a platform for monitoring and processing passive traces but the transfer of packets to the serial port seems to result in packet loss and the validation is also based on the data measured in a controlled environment [80]. Jigsaw is a system by Cheng et al. that relies on multiple monitors to collect a large number of traces to provide a cross-layer unified view of the network. It can merge traces from a number of monitors and the authors merge traces captured by each vantage point across four floors of a university campus. [15].

Serrano et al. make use of merging for single and multiple sniffers to analyze the fidelity of COTS sniffers by studying differences in traces obtained from controlled experiments inside an anechoic chamber [81]. Claveirole et al. merge the independent but incomplete traces from different sniffers into one complete trace using their tool Wipal [82]. The main focus is, however, on the performance of the tool. WiPal is for offline post-processing.

Discussion

Time synchronization of the traces is an important aspect in the world of passive sniffing and it lays the basis for the analysis of the quality of the traces captured. However, there is a lack of tools available for the purpose of time synchronization and merging of the traces. Most of the tools mentioned are no longer available, whereas WiPal is still available to be downloaded [83]. However, WiPal is not usable in its current form with the latest operating systems and compilers. It can be used with Docker but it is not convenient for doing extended analysis on servers and clusters. This, therefore, defines our need to have a readily usable and easily available tool for trace synchronization and it leads us to a readily usable and easily available Python version of WiPal, called PyPal.

2.5 DISTANCE MEASUREMENT

Distance measurement and localization play crucial roles in opportunistic networks, as those facilitate the provision of personalized and localized services, enable the identification of nodes, and, more recently, support contact tracing efforts for COVID-19. In this context, we consider the use of RPi for such purposes [84, 85, 86] as an opportunity for the application of redundancy. By introducing redundancy, it is possible to reduce the potential errors in distance estimates, thereby enhancing the overall accuracy and reliability of these measurements.

Distance measurement is important for pedestrian trajectory reconstruction as it helps to accurately track the movement and position of objects over time. Trajectory construction involves creating a path or route for a person's movement based on the data captured over time. Small errors in distance estimation can lead to significant errors in trajectory construction, which can affect the accuracy of subsequent analyses and predictions. Researchers make use of the Received Signal Strength Indicator (RSSI) values for this purpose.

The RSSI values are of interest to detect the presence of a node in a target area and evaluate its position relative to the sniffer [22, 23, 24, 25]. Furthermore, those are available in most operating systems in a straightforward fashion. However, the downside of using the RSSI concerns its accuracy. Although there are some workarounds like machine learning to reduce the error in distance estimation [26, 27], the applicability of the RSSI is still questioned [28, 29, 30].

Adel et al. use ten maximum RSSI values for indoor localization using Bluetooth Low Energy (BLE). They use median, mean, mode, and single direction outlier removal to smooth the RSSI and improve the indoor distance estimate [87]. This method requires testing in a trilateration setup. Venkatesh et al. use the mean and median filters to stabilize the RSSI values to enhance the distance estimation accuracy for indoor localization in BLE [88].

Salomon et al. make a comparison for distance estimation by RSSI and Channel State Information (CSI) for Wi-Fi by doing experiments on RPi4 devices, one as an Access Point (sender) and one as receiver [89]. Forbes et al. use a single RPi4 as a capturing device to perform distance estimation in Wi-Fi using CSI [90]. They capture the traffic generated by an Access Point in response to the packets it receives from a computer. The use of a single receiver is in itself risky as it can miss several packets due to the characteristics and unpredictability of the wireless medium. Chuku et al. remove the RSSI outliers using clustering to improve the distance estimates [91].

Discussion

We make use of redundancy in the number of devices capturing the traffic, which means we have multiple co-located sniffers at the same place. The focus in the literature has always been on removing the RSSI outliers from the dataset without using redundancy [91, 92, 93]. We prove that even a single packet can have different RSSI values at different co-located sniffers at the time of capture. Secondly, we highlight the fact that a single sniffer can miss a burst of consecutive packets at a time which can result in flawed or biased analyses. The emphasis on redundancy aids in the elimination of RSSI outliers and bursts of missing packets, thereby decreasing the error in distance estimation.

2.6 CONCLUSION

Passive measurements provide an unbiased view of wireless network performance, making them a suitable method for analyzing various aspects of the network. Among the low-cost computers available for building an efficient sniffer, RPi is a popular choice due to its affordability, versatility, and processing capabilities. One study particularly proves that RPi works well as a sniffer. Hence, conforming to our choice to use RPi as a sniffer.

We explore the state-of-the-art techniques used to capture and improve the quality of wireless network traces through passive sniffing. While various methods, such as merging individual traces, inferring missing packets, and computing inter-flow similarity, have been used to improve the accuracy of wireless network analysis, there is no focus on the use of redundancy in the number of sniffers to study its impact on the level of trace completeness. Our work stands out by firstly proving that a single sniffer is not enough to capture a quality trace, and then focusing on redundancy for trace completeness based on real-world experiments in an uncontrolled environment, and doing an exhaustive analysis for different scenarios. Moreover, our solution is more oriented toward contact traces and can be applied to different wireless technologies.

To evaluate redundancy, we require a time-synchronization tool but the tools in the literature are either not available or out-of-date and not easy to use. We develop our tool for Wi-Fi trace synchronization and merging, which can additionally provide per-MAC address traces.

Finally, we highlight the distance measurement techniques using the RSSI values. Even though there is some work for the removal of outliers, it is based on a single sniffer. We apply the idea of redundancy in the number of sniffers and prove that it reduces the error in distance estimation. Furthermore, the use of an outlier removal technique with redundancy reduces the error further.

In the next chapter, we start by explaining our experimental methodology to do measurements in two different environments and hence prove that a single sniffer does not capture a trace representative enough of the wireless medium, irrespective of the environment.

3

Sniffing a wireless environment with COTS sniffers

A single sniffer might not be able to have a comprehensive view of the wireless network due to the inherent characteristics of the medium, such as:

- **Interference:** Passive wireless measurements are often performed in environments with multiple signal sources and types, which can lead to interference and signal degradation.
- **Multi-path propagation:** The signals in the wireless medium can follow different paths due to the surrounding environment and factors, for example reflection, refraction, diffraction, scattering, etc. It is entirely possible that the signal dissipates even before it reaches a (single) sniffer.
- **Power limitations:** Passive wireless measurements rely on the reception of signals from other devices, which means that the performance of the sniffer is limited by the strength and quality of the received signals. A single sniffer may not be able to receive all relevant signals, especially in environments with weak signals.

Therefore, we decided to conduct a thorough assessment of the performance of a single sniffer in real-world environments. To accomplish this, we construct

an experimental configuration that involves specific hardware and software components, which we explain in the next section.

3.1 EXPERIMENTAL SETUP

Individual sniffing nodes. We have several sniffers in our measurement set-up, composed of Raspberry Pi models 3B (RPi3 hereafter) and 4B (RPi4 hereafter) [94, 95]. We use an external Wi-Fi module, Alfa AWUS 051NH, one per sniffer [96]. The community of the tool aircrack-ng [97] recommends this adapter. The advantage of this specific external Wi-Fi module is that it can be easily set to monitor mode. The monitor mode is a radio mode that allows the Wi-Fi card to listen to all Wi-Fi traffic in the wireless medium passively. We select the 2.4 GHz band for doing the measurements and channel 1 for our measurements. We stick to one channel for our measurements since channel hopping or multi-channel measurements lead to less number of packets captured [98]. Figure 3.1 shows the components of our sniffer using an RPi4.

Trace capture. Sniffers run *tcpdump* to collect traces [38]. We configure some filters to gather only the header fields that are essential for the objectives of this work (for example, to avoid capturing personal data as discussed below). The outcome of the capture process is one *pcap* file per individual sniffer.

Privacy preserving. The privacy of the users is a top priority for us. We anonymize the traces by running several protection techniques on the packets. Firstly, we do not disclose the geographic locations of our measurements. Secondly, we configure the sniffers to capture only the headers of the packets. In our work, we need the header as it brings the necessary information to let us combine traces from different sniffers. But, since headers contain the MAC addresses of the devices, which are considered personal information, we need to provide some extra privacy guarantees. To this end, we hash and truncate

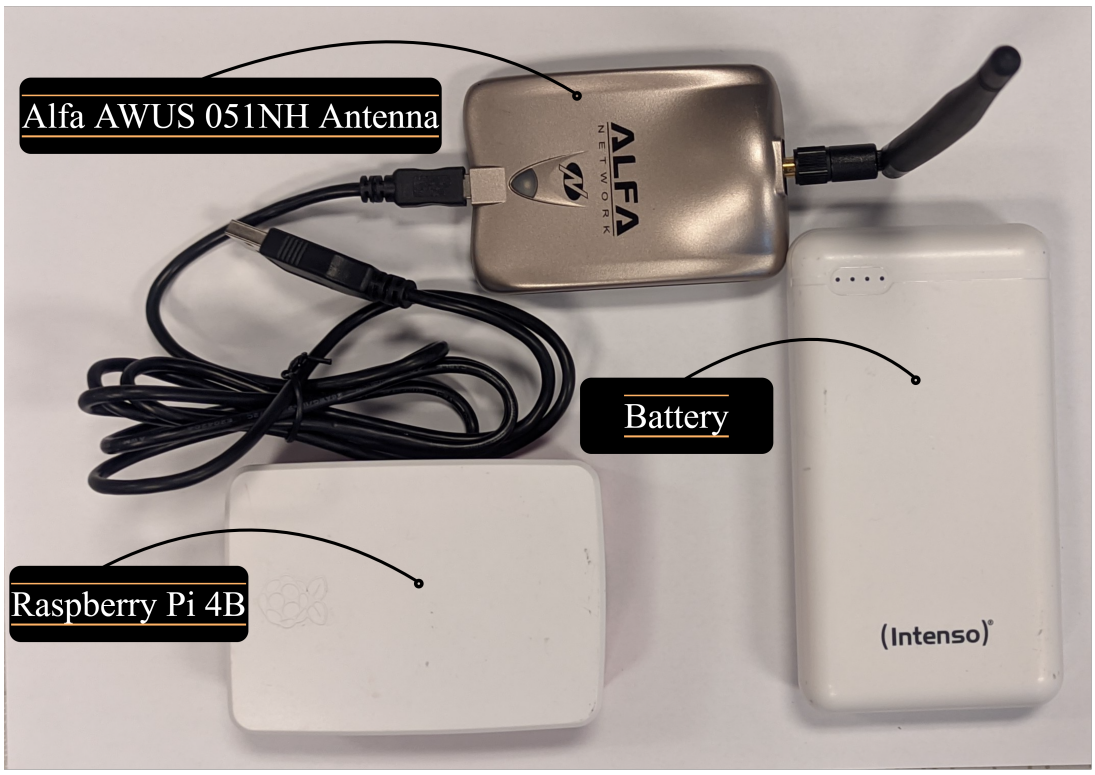


Figure 3.1. – A single sniffer. It is composed of a Raspberry Pi 4B node, an Alfa AWUS 051NH antenna, and an external power supply (20,000 mAH/74/0 Wh and 3.1 A output) to allow us to carry out the measurements for a longer period of time.

the MAC addresses to make the data secure while making sure there are no hashing collisions.

3.2 PERFORMANCE OF INDIVIDUAL SNIFFERS

To evaluate the impact of the environment on wireless passive capture, we co-locate ten sniffers to collect wireless traffic. The merge of all these traces provides a *full* capture of the environment, which is then used as the reference to compute the percentage of packets captured for each sniffer trace.

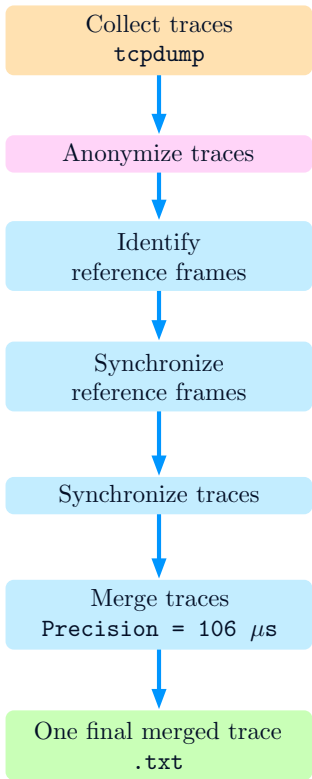


Figure 3.2. – Experimental methodology. The steps involved in the complete process of collection, privacy protection, processing, and analysis and outcome of the traces.

3.2.1 *Experimental Setup*

We use the following setup to measure the performance of the individual sniffers. We deploy the ten individual sniffers in the way depicted in Figure 3.3. We place the sniffers at a distance of ∼20 cm from each other[3]. Note that s_i refers to RPi3 nodes if i is odd and to RPi4 if i is even.

3. The minimum separation between 2 antennas is half the wavelength. The wavelength of the 2.4 GHz band is 12.5 cm, so the minimum separation is 6.25 cm. At the same time, the separation should not be in multiples of the wavelength. We choose the value of ∼20 cm to rule out any chance of interference among the sniffers. Moreover, we present the results for separations of 0, 5, 10, 15, and 20 cm between the sniffers in [99].

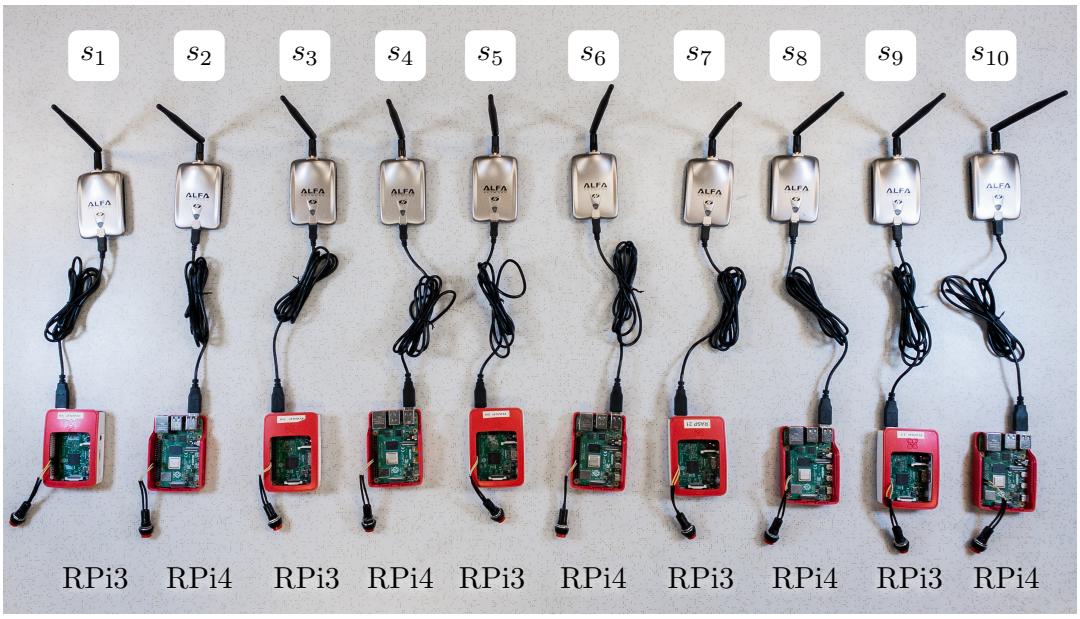
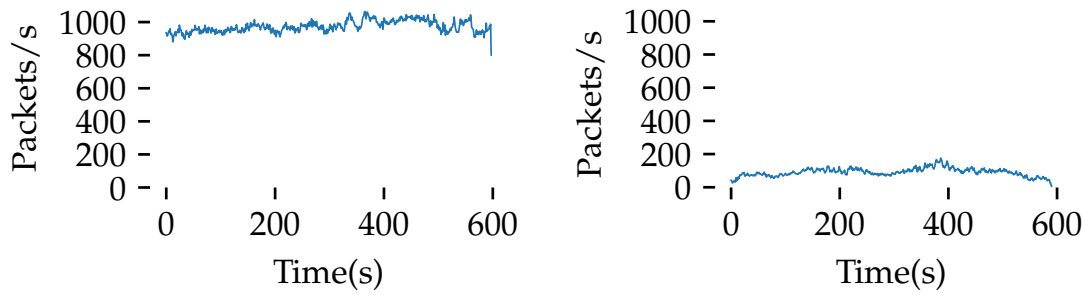


Figure 3.3. – Arrangement of the ten sniffers. We used it to carry out experiments for the evaluation of the performance of individual sniffers. We use different RPi models to see if the choice of hardware has any impact on the quality of the capture.

Scenarios. We conduct trace capture experiments in two distinct scenarios to evaluate the performance of our wireless sniffing system under varying levels of traffic intensity. The first scenario involves a *residential* area, while the second scenario consists of an *office* environment. Our aim is to assess the performance of our commercial-off-the-shelf (COTS) sniffing system, particularly in the presence of higher traffic volumes, as we expect to encounter in the second scenario.

Execution. We run each test 10 times at three different spots in the target scenarios to rule out anomalies. The duration of each data collection is 10 minutes, and the sniffers remain stationary for the whole capture period.

To evaluate the proportion of traffic a single sniffer can capture, we average the percentage of the number of packets captured for each sniffer using the



(a) Office area.

(b) Residential area.

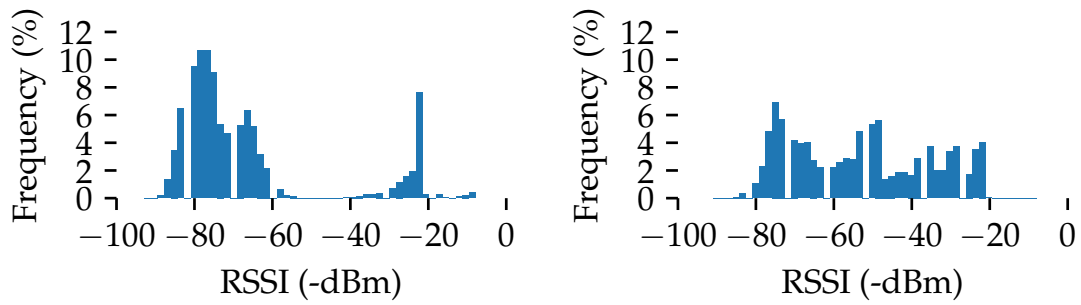
Figure 3.4. – Capture of traffic in two different environments. It is the average of all 10 sniffers.

30 different runs of each scenario. In the following section, we present an analysis of the percentage of packets captured per individual sniffer.

3.2.2 *Characterization of the environment*

In Figure 3.4, we show the difference in the traffic load present in the wireless medium at the time of capture for office and residential scenarios. The office area of our experimentation is dense, whereas the residential area is scarcely populated. As we see in Figure 3.4a, the traffic in the office area is dense with an average of roughly 1,000 packets per second. The measurement location in the residential area (Figure 3.4b) is isolated and has less Wi-Fi activity in the surroundings. Therefore, the traffic in the residential area is sparse – an average of around 100 packets per second.

We also investigate the reception quality of the packets by plotting the RSSI (received signal strength indicator) values of all the packets that all 10 sniffers observe. We show the results in Figure 3.5. In the residential environment (Figure 3.5b), we note that a large proportion of the packets have a low RSSI: 20% of the packets have an RSSI below -80 dBm and about 30% between



(a) RSSI in the office area.

(b) RSSI in the residential area.

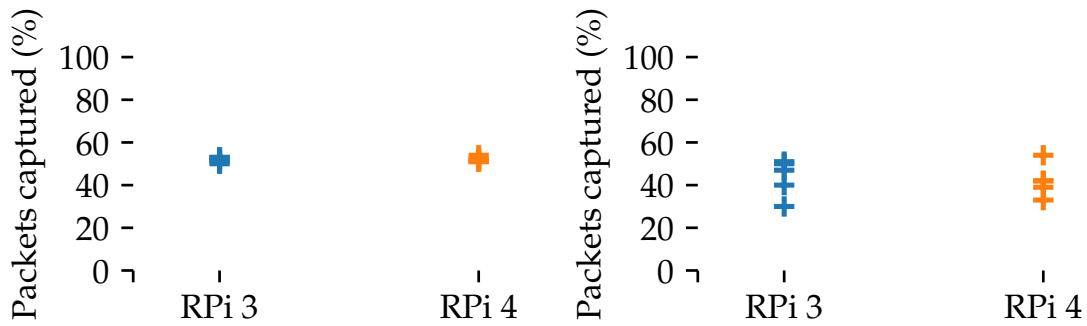
Figure 3.5. – Reception quality of traffic. The distribution of RSSI values of the packets captured by the sniffers in office and residential areas.

-70 dBm and -80 dBm. While the RSSI value is not an absolute measure of the quality of a link [9], degradation of reception is often considered altered under -70 dBm and poor under -80 dBm. Thus, half of the packets that we collected had a low-quality reception. In the second environment, we observe that while the number of packets is much higher than in the previous scenario, only 25% of the RSSI values are below -70 dBm. So, in the dense environment, 75% of the packets reach the sniffers with a good RSSI.

3.2.3 *Representativity of the measures*

We show in Figure 3.6 the percentage of the average number of packets that each sniffer captures for both the residential and office scenarios. To make a fair comparison between both types of devices, we plot, for each scenario, the values obtained with RPi3 and RPi4 sniffers.

We observe that with a low traffic load in the residential environment, a single sniffer captures between 30% and 50% of the total number of packets collected with RPi3 sniffers, while the same values for RPi4 sniffers are between 33%



(a) Office area.

(b) Residential area.

Figure 3.6. – Average percentage of packets captured by individual sniffers. Each dot represents an individual sniffer. This is the average capture ratio achieved by each sniffer individually over 30 tests.

and 54%. In the heavy-traffic environment of the office scenario, the capture ratio appears slightly better, ranging between 50% and 54%.

The most striking element is the low value of the packets captured by each sniffer for both scenarios. The best individual sniffers only 54% of the packets in both scenarios. By comparing both scenarios, we deduce that our sniffers are powerful enough to handle the low and high traffic loads as the percentage of packets captured remains similar when the traffic load is increased by a factor of 10 (residential and office scenarios). Thus, it seems that capture misses are due to the conditions of the wireless medium.

To explain the low values of the percentage of the packets captured, we focus on the RSSI variations of both environments presented in Section 3.2.1. While the signal strength is weak or poor for about 50% of the packets in the residential area, it appears much stronger in the office environment. This leads, as expected, to relatively low values of packets captured. The comparison of the variation of the packets captured for the individual sniffers between both environments confirms the observation. In the residential case, we observe a difference of 20% between the best and the worst capture values and a

difference of less than 5% in the office environment. Note that the most recent devices (RPi4) tend to capture slightly more packets than RPi3 devices in both scenarios.

To conclude this first analysis, we show that the wireless environment is challenging to capture, and signal strength plays a substantial role in the quality of the captured traces. A packet is more likely to be missing from a trace if the signal strength is lower, but other nearby devices can capture it. In other words, introducing redundancy in the number of sniffing devices is beneficial for global capture. Now the question is: "are individual sniffers complementary to each other?" To address this question, we propose to measure the similarity between all sniffer traces.

Jaccard similarity. We use the Jaccard index to measure the similarity of the traces in a pairwise way. The higher the percentage, the more similar the traces. The goal is to verify whether sniffers capture the same packets or not. If so, there would be no interest in deploying super-sniffers. If not, the conclusion is that we can improve the quality of the capture by adding redundancy.

Results for residential and office scenarios are given respectively in Tables 3.1 and 3.2. In Table 3.1 we observe that the similarity indexes remain relatively stable, varying from 63% to 73%. So, whichever is the combination of two sniffers, the two respective traces differ by 27% – 37%. Thus, we conclude that any of the sniffers can bring useful information to the global trace. This result is even more evident in the office scenario shown in Table 3.2, as the similarity indexes are bounded between 62% and 64%. This strong stability combined with a low similarity index value denotes each collected trace is complementary to any other one.

Table 3.1. – Jaccard similarity: residential area.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
Rel. compl.	47%	42%	50%	33%	40%	42%	30%	54%	51%	39%
s_1	–	0.66	0.66	0.71	0.67	0.66	0.70	0.66	0.68	0.67
s_2	0.66	–	0.66	0.68	0.65	0.65	0.69	0.65	0.67	0.67
s_3	0.66	0.66	–	0.72	0.67	0.65	0.70	0.63	0.66	0.68
s_4	0.71	0.68	0.72	–	0.67	0.69	0.69	0.72	0.70	0.73
s_5	0.67	0.65	0.67	0.6738	–	0.65	0.66	0.66	0.66	0.69
s_6	0.66	0.65	0.65	0.69	0.65	–	0.67	0.64	0.66	0.67
s_7	0.70	0.69	0.70	0.69	0.66	0.67	–	0.70	0.69	0.73
s_8	0.66	0.65	0.63	0.72	0.66	0.64	0.69	–	0.64	0.67
s_9	0.68	0.67	0.66	0.70	0.66	0.66	0.69	0.64	–	0.70
s_{10}	0.67	0.67	0.68	0.73	0.69	0.67	0.73	0.67	0.70	–

3.3 REDUNDANCY IS NECESSARY

Redundancy in the number of sniffers is important to ensure reliable and continuous monitoring of network traffic in case one or more sniffers fail. This helps to minimize disruptions and ensures that all network activity is captured and analyzed, providing a more complete picture of network behavior and potential issues. We show that the wireless environment is challenging to capture, and signal strength plays a substantial role in the quality of the captured traces. We can therefore conclude the need to combine sniffers to capture the traffic in a wireless environment adequately based on the results we present in the previous section.

3.4 CONCLUSION

We present the analysis for traces captured simultaneously by ten co-located sniffers of two different types in low and high-activity scenarios. We draw attention to the differences in the number of packets captured depending on

Table 3.2. – Jaccard similarity: office area.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
Rel. compl.	50%	52%	53%	51%	53%	52%	51%	54%	52%	51%
s_1	–	0.62	0.62	0.64	0.62	0.63	0.63	0.63	0.63	0.64
s_2	0.62	–	0.62	0.63	0.62	0.63	0.63	0.62	0.63	0.63
s_3	0.62	0.62	–	0.63	0.62	0.63	0.63	0.62	0.62	0.63
s_4	0.64	0.63	0.63	–	0.63	0.63	0.64	0.63	0.63	0.64
s_5	0.62	0.62	0.62	0.63	–	0.62	0.63	0.62	0.62	0.63
s_6	0.63	0.62	0.63	0.63	0.62	–	0.63	0.63	0.63	0.63
s_7	0.63	0.63	0.63	0.64	0.63	0.63	–	0.63	0.63	0.63
s_8	0.63	0.62	0.62	0.63	0.62	0.62	0.63	–	0.62	0.63
s_9	0.63	0.62	0.62	0.63	0.62	0.62	0.63	0.62	–	0.63
s_{10}	0.63	0.63	0.63	0.64	0.63	0.63	0.63	0.63	0.63	–

the type of environment, but at the same time, there is a consistency in the level of the number of packets captured that an individual sniffer can achieve in real-world experiments. We show that both types of Raspberry Pi devices achieve similar capture levels, despite having different hardware specifications. It indicates that the quality of the traces is poor due to the environment and the characteristics of the wireless medium, independent of the hardware.

We introduce the concept of a *super-sniffer* which is a solution to circumvent the problem of the low number of packets captured by single sniffers. It consists of introducing redundancy in the system by tying two or more sniffers together in order to increase the probability that at least one of the sniffers captures a packet. In Chapter 4, we address the question that *how the level of redundancy helps improve the quality of measure*. To provide an answer to this question, we propose the concept of trace *completeness*. We provide the definition of trace completeness, explain how to formulate and measure it, and then evaluate it through real-world experiments. Although we focus on Wi-Fi traces, our methodology is general and can be applied to other technologies.

4

Trace completeness through super-sniffers

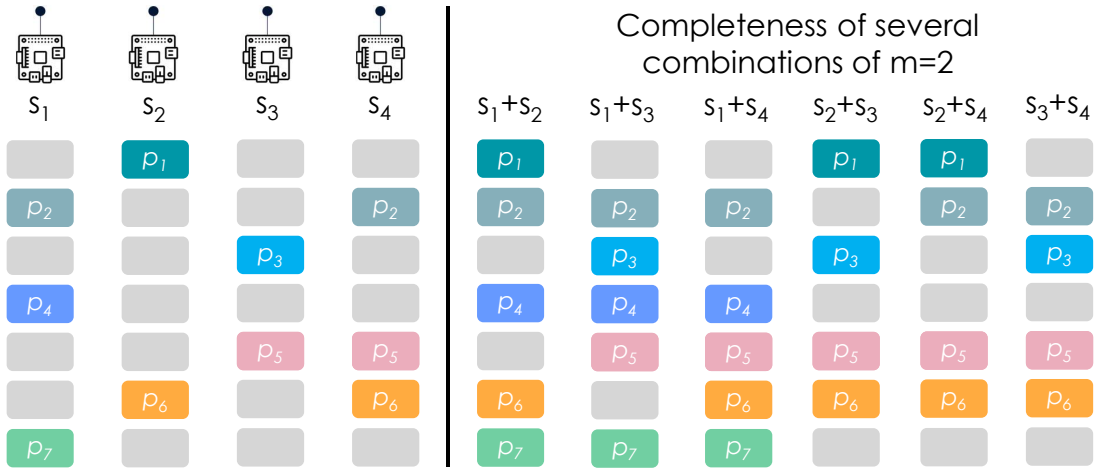
Trace completeness is important for a variety of purposes, such as network troubleshooting, performance optimization, and capacity planning. A trace that is not complete may result in inaccurate or incomplete analysis and lead to incorrect conclusions or decisions. High trace completeness ensures that the data analyzed is accurate and representative of the actual network performance.

It helps us understand the quality of traces of the individual sniffers as well as the combination of different sniffers when we introduce redundancy. We define trace completeness as:

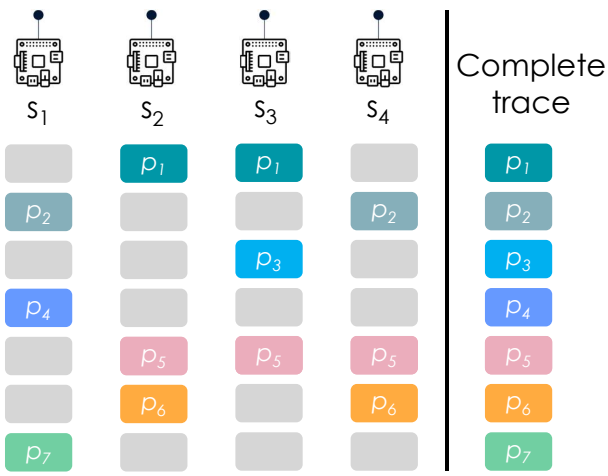
Definition 1 (Trace completeness) *Trace completeness in Wi-Fi refers to the percentage of Wi-Fi packets that are successfully captured and analyzed in a Wi-Fi network. It is a measure of the accuracy of Wi-Fi data analysis and is used to evaluate the quality of Wi-Fi network performance.*

We show how to express completeness mathematically and also differentiate between the *relative* and *absolute* completenesses in the following section. We evaluate it with real-world experiments in different setups that we explain later in this chapter.

4.1 MATHEMATICAL DESCRIPTION



(a) Completeness for all combinations of super-sniffers of size 2.



(b) Completeness for a super-sniffer of the maximum size.

Figure 4.1. – An illustration of the concept of trace completeness for super-sniffers of all sizes.

The quality of a passive measurement system improves as the redundancy level of a super-sniffer increases. In Figure 4.1a, for example, it is not possible to obtain the complete trace through any combination of the traces of 2 sniffers ($s_1 + s_2$, $s_1 + s_3$, ..., $s_3 + s_4$). However, we obtain a *relatively*[4] complete trace by the super-sniffer of maximum size in Figure 4.1b. We need to estimate how much the level of redundancy of the super-sniffer impacts the number of packets captured, as adding more and more sniffers to a super-sniffer comes at a financial as well as management cost.

As we introduce redundancy in the number of sniffers, we coin the term *super-sniffer*. A super-sniffer is a collection of sniffers that are co-located and operate side-by-side to increase the completeness of the trace. The number of co-located sniffers gives the redundancy level of the super-sniffer. A super-sniffer of redundancy m is composed of m individual sniffers. In Figure 4.2, we illustrate a super-sniffer of size three, where each individual sniffer is composed of a Raspberry Pi computing unit and an Alfa antenna. A super-sniffer merging the traces from each of the 3 sniffers would obtain the complete trace. But if the super-sniffer is only composed of 2 sniffers, it may or may not capture all the packets.

Super-sniffer. We formalize the completeness as follows. Let $S = \{s_1, s_2, \ldots, s_M\}$ be the set of M sniffers that we have at our disposal to compose a super-sniffer, T_{s_i} be the trace (i.e., set of packets) captured by sniffer $s_i \in S$, and $\mathcal{T} = \{T_{s_1}, T_{s_2}, \ldots, T_{s_M}\}$.

We define $\pi_{\cdot}^{m}$ as a subset of m elements of $\mathcal{T}$ and Π^m be the set of all instances of different combinations of $\pi_{\cdot}^{m}$:

4. The trace is relatively complete because the completeness in this case is relative to whatever is captured by the sniffers. We have no guarantee that the sniffers are able to capture every packet that is present in the medium at the time of measurement because we have no control and knowledge of the devices present in the vicinity.

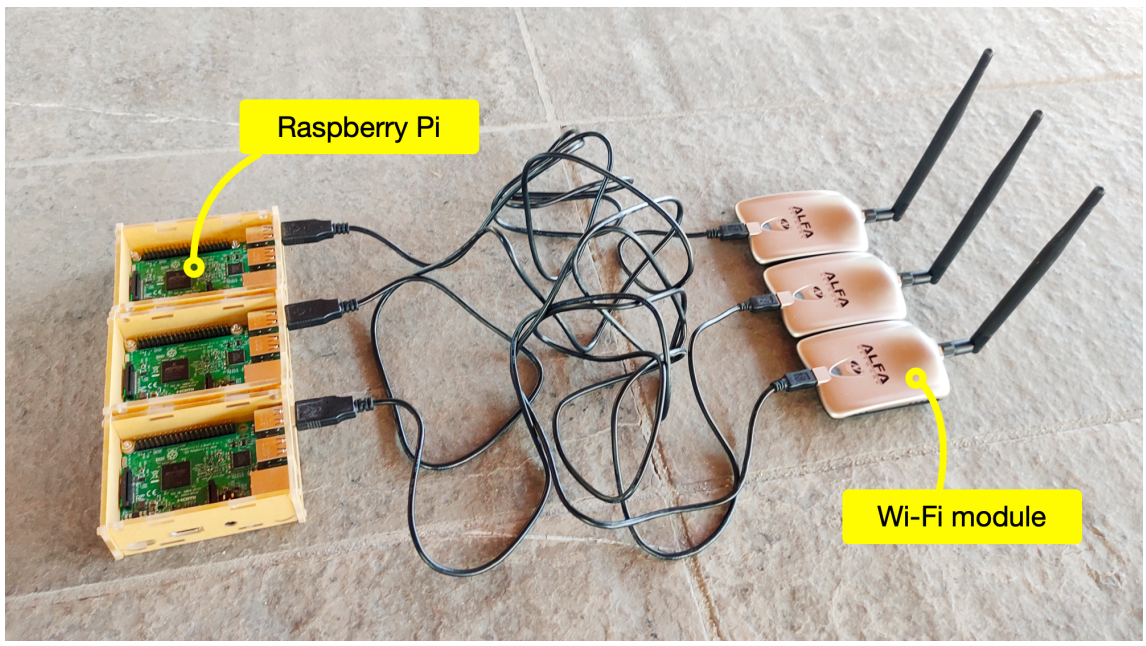


Figure 4.2. – A super-sniffer. 3 individual co-located sniffers grouped together form a super-sniffer of size 3.

$$\Pi^m = \{\pi_1^m, \pi_2^m, \ldots, \pi_{\binom{M}{m}}^m\} = \{X = \{x_1, x_2, \ldots, x_m\}, \\ x_1, x_2, \ldots, x_m \in \mathcal{T}, \quad x_1 \neq x_2 \neq \ldots \neq x_m\} \quad (4.1)$$

where $\binom{M}{m}$ is the number of combinations of super-sniffers of size m that can be built out of M sniffers.

The outcome trace of a super-sniffer is a single trace resulting from the combination of the individual traces of the sniffers composing the super-sniffer. We refer to such a trace as $A^{\pi_i^m}$, i.e., as the union of the traces $\pi_i^m \in \Pi^m$, $i = 1, 2, \ldots \binom{M}{m}$:

$$A^{\pi_i^m} = T_a \cup T_b \cup \ldots \cup T_m, \quad T_a, T_b, \ldots, T_m \in \pi_i^m, \quad (4.2)$$

and

$$T_a \neq T_b \neq \ldots \neq T_m. \tag{4.3}$$

In a real situation, it is likely that we do not possess the complete trace to estimate the exact completeness of a capture. Thus, in this definition, we compute the completeness of a trace with respect to the union of all traces from all the individual sniffers that participated in the capture (which explains why we use the term "relative").

Definition 2 (Relative completeness) *The relative completeness is the proportion of packets that an individual sniffer or a super-sniffer captures "relative" to the number of packets captured by the super-sniffer of maximum size.*

As underlined earlier, the *maximum reachable quality* is obtained when the super-sniffer is M-fold redundant (i.e., it is composed of all M individual sniffers):

$$A_{\max} = A^{\pi^M} = T_{s_1} \cup T_{s_2} \cup \ldots \cup T_{s_M}. \tag{4.4}$$

We need to make two observations now. Firstly, note from Equation 4.1 that Π^M has a single element, which is π^M. Therefore, the quality of a capture is denoted by $A^{\pi_i^m}$. The value of this measure quality is obtained by taking its ratio with the result of maximum value when all M sniffers are considered. Secondly, $A_{\max}$ is the best result that we can obtain. That is why we consider it as the reference number to define the "relative" completeness:

$$C(A^{\pi_i^m}) = \frac{|A^{\pi_i^m}|}{|A_{\max}|}. \tag{4.5}$$

There are multiple super-sniffers of size m, each one resulting from a different combination of m out of M sniffers. Each of the $\binom{M}{m}$ super-sniffers leads to a different value of completeness. We can then define two special cases, which come, respectively, from the super-sniffer that leads to the largest completeness and the super-sniffer that leads to the smallest completeness:

$$C^m_{\max} = \max_{i=1,2,\ldots\binom{M}{m}} C(A^{\pi^m_i}) \tag{4.6}$$

and

$$C^m_{\min} = \min_{i=1,2,\ldots\binom{M}{m}} C(A^{\pi^m_i}). \tag{4.7}$$

Definition 3 (Absolute completeness) *The absolute completeness gives the share of the packets that a sniffer captures for the actual traffic from a specific source.*

Unless the capture is performed in an anechoic chamber, it is not possible to control all the Wi-Fi devices in the target area. It is thus hard to estimate or measure absolute completeness in wireless networks without the knowledge of the traffic generated.

The fundamental difference between absolute and relative completeness is the set of packets that we consider our universe. In our experiments for absolute completeness, a source node generates controlled traffic. This serves as a reference to define the largest trace in a given environment and the base to define absolute completeness.

Let A_{abs} be the set of packets generated by the source node that circulated in the network at the time of the capture. Note that $A_{\max} \subseteq A_{\text{abs}}$. The absolute completeness is:

$$C(A^{\pi_i^m}) = \frac{|A^{\pi_i^m}|}{|A_{\text{abs}}|}. \tag{4.8}$$

Equations 4.6 and 4.7 still hold here.

Number of traces per size of super-sniffer. We need to build traces π_i^m for all combinations of sniffers of different sizes. If we consider $m = 4$, then Π^m in Equation 4.1 is equivalent to $\{\pi_1^4, \pi_2^4, \ldots, \pi_{\binom{M}{4}}^4\}$ which means that we need to build traces for all combinations of $m = 4$ sniffers out of the total M sniffers. It represents all combinations of sniffer s_1 with combinations of three sniffers other than s_1, similarly combinations of sniffer s_2 with three sniffers other than s_2 itself, and so on.

We define and formulate the relative and absolute completenesses, but before we study completeness experimentally, we need to define a methodology. We explain the tools and hardware we need to do the experiments in Section 3.2.1 but we are also in need of another tool that allows us to analyze the traces for super-sniffers. We explain the need and working of a tool that we created in the next section.

4.2 PYPAL: A PYTHON TOOL FOR WI-FI TRACE SYNCHRONIZATION AND MERGING

The principle behind a super-sniffer is its ability to merge traces collected by its individual sniffers. The sniffers' traces are not synchronized among them since the sniffers have their own local clocks. We need synchronization to be able to compare the traces that the sniffers collect at the same time. Whereas, the merging process requires that input traces be synchronized so that a packet that appears in multiple individual traces is identified in an

unambiguous manner. We developed a Python tool called *PyPal* that performs such a synchronization operation [21].

PyPal is an updated Python version of WiPal [100]. It is a stand-alone and offline tool independent of the sniffing or monitoring process. One can use any tool for capturing the data as it does not affect the working of PyPal. As of now, PyPal expects the input traces in CSV/TEXT format containing the fields required for the process of time synchronization. We mention the fields required for synchronization in Appendix B.1. However, functionality can be added to the tool to accommodate other formats specifically PCAP by including a command that can extract the required fields from the PCAP trace itself. A command to extract the required fields from a PCAP file to be used with the current version of PyPal is given in Appendix B.2.

The main idea of PyPal is to synchronize the traces captured by different sniffers at the same time and to be able to merge them by removing duplicate frames. The process is composed of five modes: (i) identifying reference frames, (ii) extraction of unique frames, (iii) intersection of unique reference frames, (iv) synchronization, and (v) merging. We explain each of these modules.

Identifying reference frames. A frame is said to be unique when it appears "in the air" once and only once for the whole duration of the measurement. A frame that is unique within each trace but that actually appeared twice on the wireless medium should not be considered as unique. The process of extracting unique frames finds candidates to become reference frames [100].

Extraction of unique frames. The beacons and probe response frames are the closest representatives of real-time clocks. Moreover, the IEEE 802.11 standard dictates that these frames have a fixed timestamp in the header, added by the access point, further improving synchronization precision. We use these frames as a basis for the synchronization process, as illustrated in Figure 4.3.

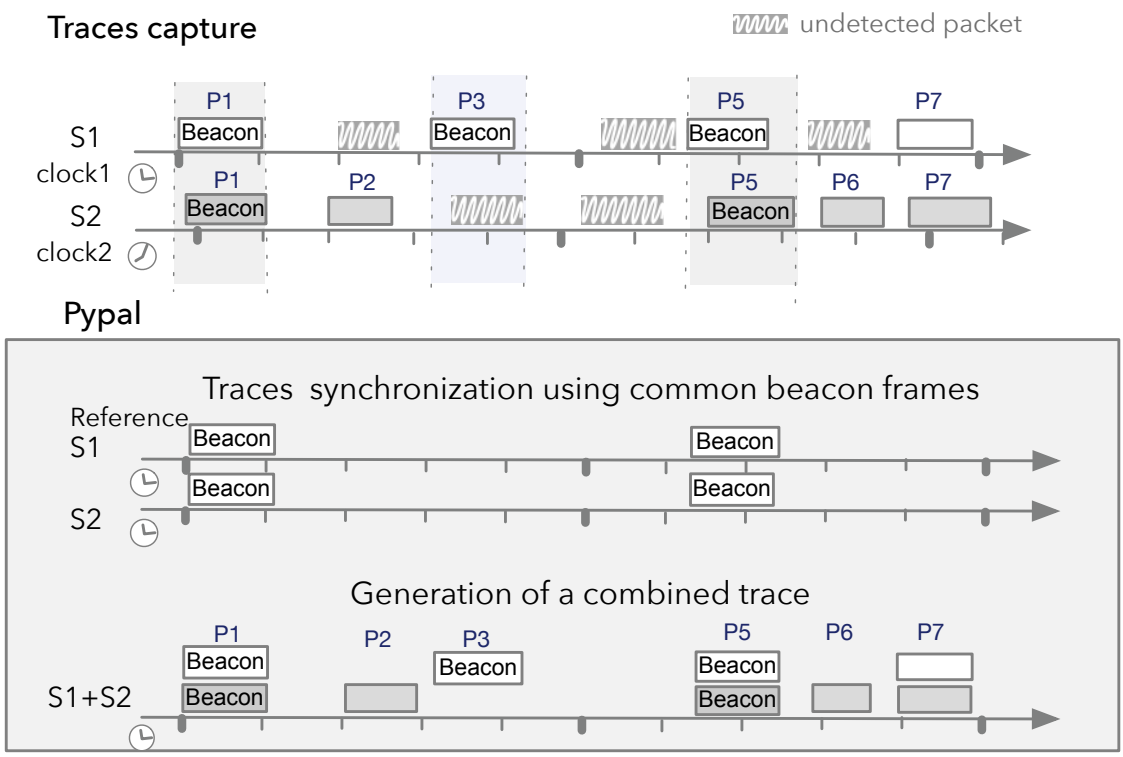


Figure 4.3. – PyPal's methodology for synchronizing traces. It takes two traces as inputs: one as a reference trace and the second as the one to be synchronized.

Intersection of unique frames. This step consists of extracting the unique frames that appear in both traces. Note that the coverage areas of the sniffers capturing these traces should overlap; otherwise, the traces will be disjoint. The frames, that we obtain by the intersection of unique frames, serve as reference frames.

Synchronization. Synchronizing two traces means mapping the first trace's timestamps to values compatible with the second trace's timestamps. We compute this mapping with an affine function $t2 = at1 + b$. It estimates a and b with the help of reference frames as the process runs.

The synchronization process operates on windows of w+1 reference frames. For each reference frame R_i, the process performs a linear regression using reference frames $R_{\lfloor i-w/2 \rfloor}, \ldots, R_{\lceil i+w/2 \rceil}$. At the beginning and at the end of the trace, we use $R_1, \ldots, R_w$ and $R_{N-w}, \ldots, R_N$ (N is the number of reference frames). The result gives a and b for all frames between R_i and R_{i+1}.

Merging. The role of merging is to copy frames from synchronized traces to the output trace. Of course, it must organize its output correctly while avoiding duplicate frames.

We make use of a combination of different header fields such as frame type and sub-type, sequence number, fragment number, frame check sequence (FCS), fixed timestamp, channel, and source MAC address to help us identify the duplicate frames. Although the FCS is for error check and control, it can make the synchronization more precise. This header field along with a couple of more fields should ideally be enough for identifying the duplicate frames but it is not always present in the captured PCAP file. The reason for the FCS field's absence is the driver of the antenna. For example, the Alfa AWUS 051NH antenna drops the FCS field before allowing the capture tool to write the frames to the PCAP/CSV/TEXT file.

Using the tool. We detail the prerequisites and instructions to run the tool in Appendix B.

At this stage, we have the definition and formulation of trace completeness as well as the tool for time-synchronization and merging of the traces. We experimentally evaluate the relative and absolute completenesses in the next two sections.

4.3 RELATIVE COMPLETENESS

4.3.1 *Experimental Setup*

We have fourteen sniffers in our measurement set-up, all RPi4. We deploy the sniffers in the form of super-sniffer as we depict it in Figure 4.4. As mentioned in Chapter 3, the distance between the sniffers is ∼20 cm. We capture the traces indoors in an office scenario where the traffic load is high as mentioned in Section 3.2.1, which allows us to study the variation in the amount of traffic over time. We co-locate the sniffers that remain stationary for the whole duration of the capture. We perform one test for 24 hours. We collected the traces from 17:30 on 12th July 2022 to 17:30 on 13th July 2022.

Trace organization. As we collect all the traffic over an extended period of time, the size traces that we collect are 84 GB. As a first step, we synchronize the traces. The number of merge operations we do for measuring redundancy makes the processing complicated and overloaded. We analyze all possible combinations of sniffers for a given redundancy, which gives a total of $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ combinations. We have 16,369 possible combinations with 14 sniffers. As the second step after time synchronization, we select a time granularity of 5 minutes for ease of processing and speeding up the calculation and analysis. We split each trace into 288 5-minute sub-traces to cover the whole 24-hour period. We merge the traces for each 5-minute slot independently for all combinations of super-sniffers of all sizes.

4.3.2 *Data characteristics*

In this section, we detail the characteristics of the data we collect. We discuss (i) the traffic load in the medium (ii) what kind of traffic we capture (iii)

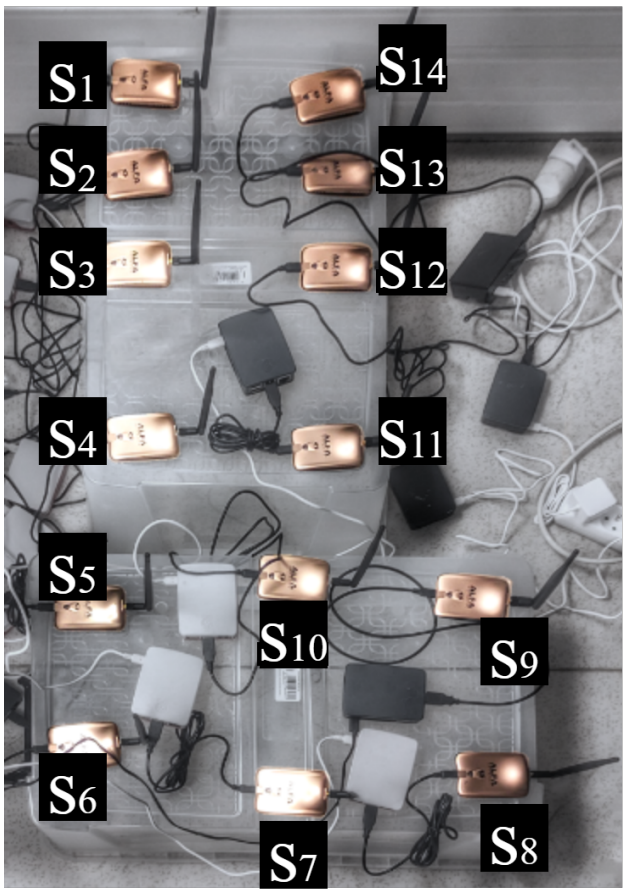


Figure 4.4. – Composition of the super-sniffer. The order in which we place our co-located sniffers to form a super-sniffer of maximum size 14.

why we capture all kinds of traffic (iv) the number of different source MAC addresses detected over 24 hours.

Traffic in the medium. Figure 4.5 shows the number of packets per 5 minutes for the duration of 24 hours. We only capture headers so these curves do not fully characterize the load in terms of the utilization of the medium i.e. the data packets are less in number but those may occupy a larger portion of the medium due to a bigger size. Note also that our capture is based

on the sniffers that only support the IEEE 802.11n, so we do not capture IEEE 802.11ac and 802.11ax traffic, and data packets sent with the highest Modulation and Coding Scheme (MCS) values [101]. Management frames (probe request/response, beacons, etc. [102]) are often sent at a lower data rate, which is why those make up a large portion of the load and at a consistent rate.

The yellow, green, and red curves in the figure depict the data, management, and control frames respectively. The blue line represents all the traffic. The x-axis and y-axis represent time and the packets captured per 5 minutes respectively. We see in the figure that the traffic load is higher in the late afternoon and morning since more people are present during office hours. At the same time, there is comparatively less traffic in the evening and at night. The traffic remains consistent in the non-peak hours; in fact, it is mostly the management frames that make up the traffic during that time. The traffic varies more during the peak (or office) hours when we also see control and data frames. The data frames cause small peaks but the control frames contribute the most to the load variation over time. There are a few sharp spikes when the control frames shoot that result in high fluctuation of the traffic load in the medium. This is the kind of variation that will help us understand the impact of traffic load on the level of completeness that the sniffers can achieve.

Reason for capturing all traffic. We see that management frames have negligible variation over the course of 24 hours and can, hence, not be used to study the impact of load variation on completeness over time. Likewise, the data frames exhibit little variation and are notably absent during nighttime periods when there is no activity in the office space. We see the most variation in the case of control frames but we cannot solely rely on them as those fall to a very low level during the night, indicating the lack of traffic for a meaningful analysis. Therefore, we consider all these frames in our experiments as those

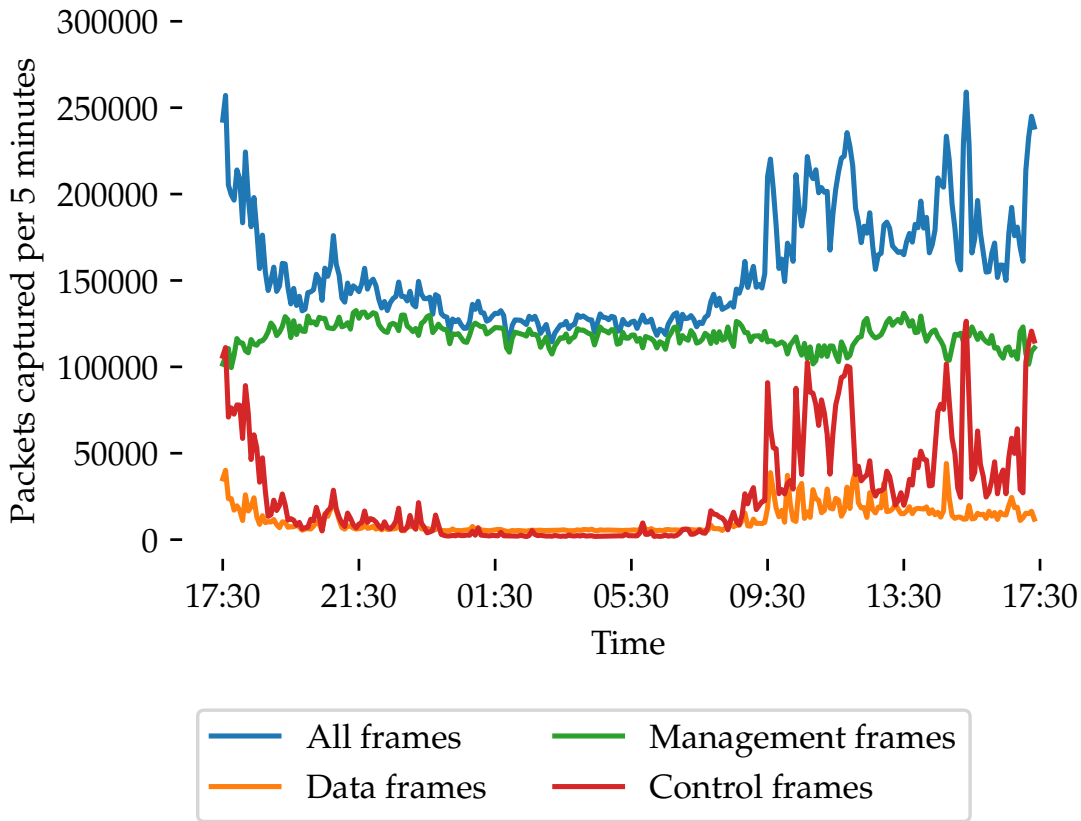
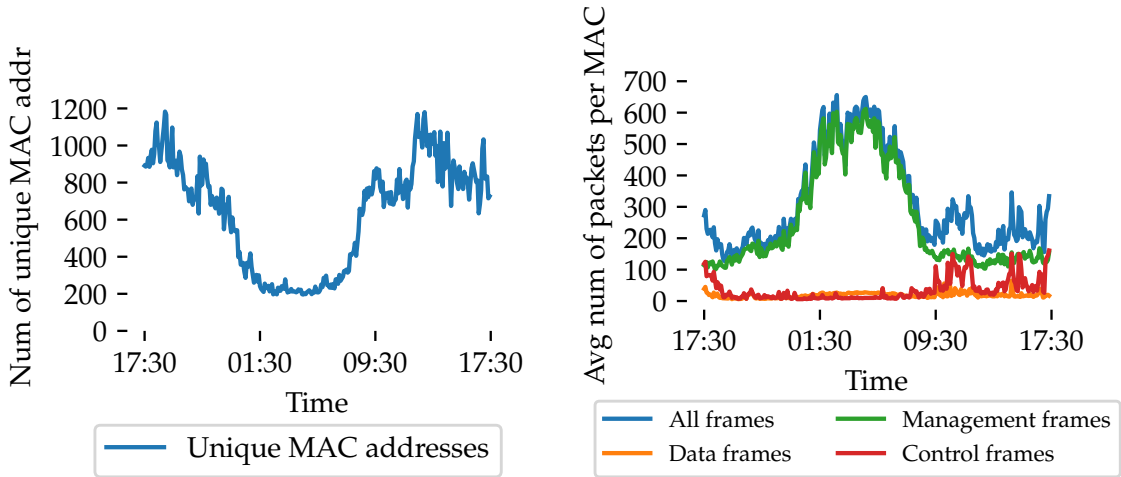


Figure 4.5. – Traffic load for the whole duration of 24 hours. The traffic is split into the data, management, and control frames, per 5 minutes each.

truly help us to understand the impact of the evolution of the traffic load in the wireless medium on the quality of trace capture.

Number of sources detected over 24 hours. Figure 4.6a presents the number of unique MAC addresses[5] per 5 minutes on the y-axis as a function of time on the x-axis. We see that the number of unique MAC addresses per

5. The number of MAC addresses does not equate to the number of devices due to the MAC address randomization implemented by the devices to change their MAC addresses randomly for privacy reasons [103], that is why we notice a higher number of unique MAC addresses per 5 minutes during office timings.



(a) Number of unique MAC addresses per 5 minutes.

(b) Average number of packets per MAC address per 5 minutes.

Figure 4.6. – The distribution of the number of unique MAC addresses and the average number of data, management, control, and all frames per MAC address. The results are per 5 minutes.

5 minutes corresponds to the traffic load in Figure 4.5. The traffic in the medium increases when the number of unique addresses increases i.e. there are more devices present in the office area. The value also decreases during the night since fewer people are present in the vicinity. We observe a surge in the number from 13:30 onwards, as there are more devices in the office area during that period.

Figure 4.6b displays the average number of packets per MAC address per 5 minutes (on the y-axis) over the course of 24 hours. This figure helps us understand the density of the traffic present in the medium over time. With more devices during the daytime, the average number of control frames per MAC address rises. The average number of management frames per source also drops during this time as a higher number of devices results in higher activity. Similarly, during nighttime, the average number of data and control frames per MAC address falls to a negligible level. It is coherent with the fact

that the number of unique sources drops during the night due to the presence of no people. As there are no people, the devices that generate traffic are the access points (APs). That is why we see no data and control frames during the night. As there are a few APs on the premises that send periodic beacons (management frames) consistently during the nighttime, that is the reason for seeing a spike in the average number of packets per MAC during the night despite having a low number of unique MAC addresses as no users connect to the Wi-Fi network.

Completeness as a function of load. Figure 4.7 shows the average relative completeness of all 14 sniffers per 5 minutes. The blue line represents the average completeness whereas the red bars are representative of the standard deviations of the completeness values of all 14 sniffers. We have the completeness values on the y-axis and time on the x-axis.

We observe that the completeness values decrease as the traffic load (Figure 4.5) in the medium increases. The completeness crosses 60% when the load is low during the night, but it falls as low as 40% when the load is higher. In either case, the average completeness appears low. Each single sniffer misses around 40% to 60% of the packets depending on the time of the day.

When we look at the red bars, we notice that the values of the standard deviation differ, and are comparatively larger at some points. This means that there is a significant difference in the completeness values of the individual sniffers. A couple of questions arise: 1) are some of the sniffers faulty? 2) is it the same sniffer(s) that consistently performs poorly and leads to bad results?

Not all sniffers are good. Figure 4.8 shows the completeness of all 14 individual sniffers over 24 hours. We see that there are a few sniffers that perform consistently poorly. When we look at the zoomed parts of the figure, we identify that 3 sniffers, namely s_5, s_{11}, and s_{14}, achieve low completeness

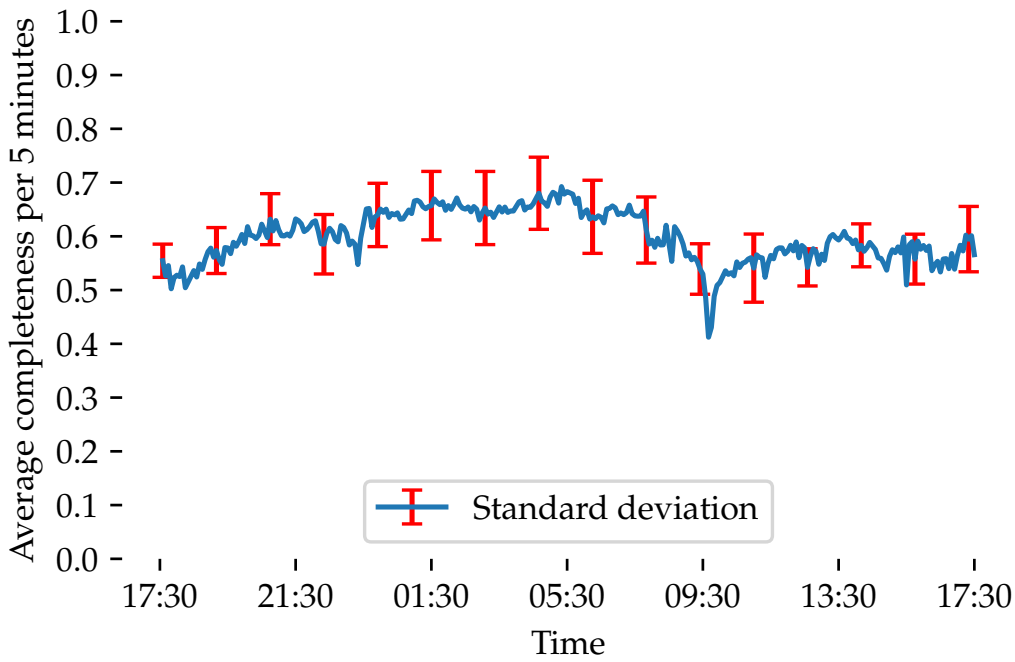


Figure 4.7. – Average relative completeness of all single sniffers.
The red lines represent the standard deviation of the completeness of all 14 sniffers individually. The higher standard deviation values show that all the sniffers do not perform in the same manner.

values. We recall from Figure 4.6a in Section 4.3.2 that the number of devices keeps changing over the period of 24 hours. It means that the medium and the conditions change over time. This implies that the problem comes from the device itself, not the environment (multi-path, collisions, etc.). [6]

RSSI. We explore the RSSI values of packets captured by each sniffer to further highlight the fact the aforementioned three sniffers are faulty. While the RSSI value is not an absolute measure of the quality of a link [9], quality of reception is often considered acceptable above -70 dBm and poor under

6. We show in Chapter 3 and in more detail in our work [104] that the location of the sniffers does not have an impact on the level of completeness achieved by a sniffer.

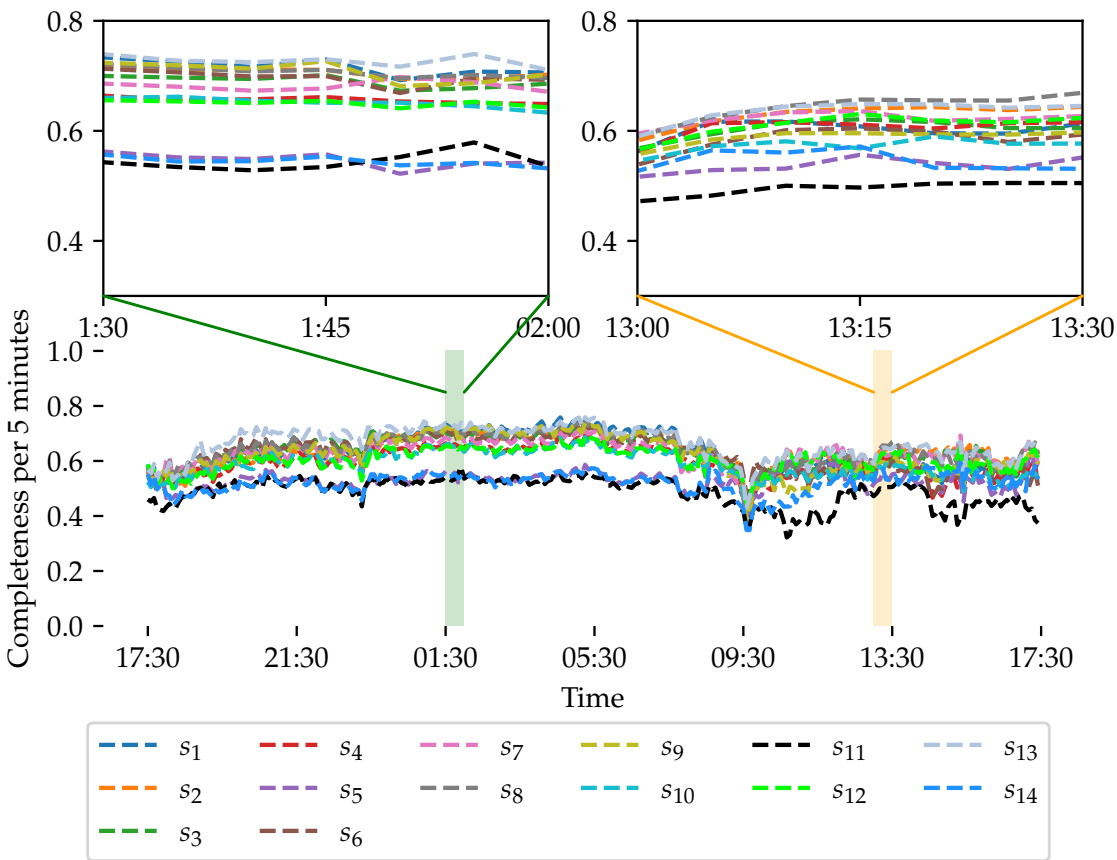


Figure 4.8. – Completeness of each sniffer. The completeness level of each individual sniffer. The zoomed area highlights that 3 sniffers perform consistently poorly.

-70 dBm [105]. Figure 4.9 presents the percentage of packets captured with acceptable and poor RSSI, and packets missed, for each of the 14 sniffers. We see that the sniffers s_5, s_{11}, and s_{14} capture the least percentage of packets with poor RSSI, proportionally. Sniffer s_{11} captures a negligible amount of packets with poor RSSI in comparison with other sniffers. It reiterates our finding that these sniffers are faulty[7] and can lead to a biased analysis.

7. The fault they present is that they only capture packets with good values of RSSI. The percentage of packets that they capture with poor RSSI is very low compared to the other 11 sniffers.

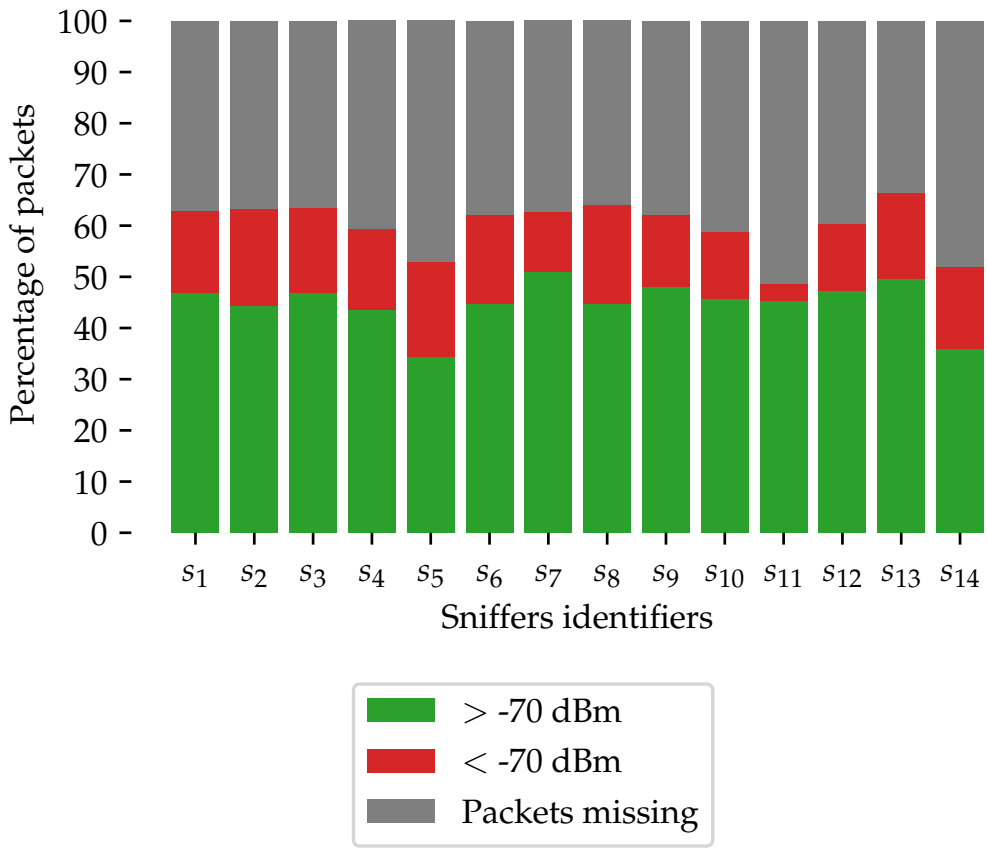


Figure 4.9. – RSSI. The percentage of packets captured with good (> -70 dBm) and bad (< -70 dBm) RSSI, as well as the percentage of packets missed by each individual sniffer.

Cleaning the dataset. From this point on, we use a subset of the dataset for our analysis. We remove the worst performing sniffers s_5, s_{11}, and s_{14} from the analysis part, we are, hence, left with 11 sniffers (i.e., a super-sniffer of maximum size 11).

We understand that the decision of pursuing with 11 sniffers can introduce some bias in our analysis but we want to continue with consistent sniffers. So, 21% of the sniffers lead to a poor dataset in our experiments. We need to investigate more in the future what could be the exact reason for this

malfunction and whether is there a possibility of fixing it. Along with that, we need to devise a strategy to select how to choose the best-performing sniffers for the experiments.

4.3.3 *Pairwise completeness: Combination of 2 single sniffers*

We define the *pairwise completeness* as a metric to rank the individual contribution of a sniffer towards completeness when paired in all combinations with other sniffers. In other words, when two traces are merged, how much information comes exclusively from the first trace, and how much comes from the second. Figure 4.10 represents the pairwise completeness. We compare the pairwise completeness of each sniffer with s_{13} as it has the highest individual completeness.

The values inside the circles represent the values of completeness of individual sniffers over 24 hours. The labels outside the circles identify the sniffer. The value on each edge of this star indicates the improvement in completeness the 2 sniffers bring as a pair.

The node s_3 brings an improvement of 11% when it is considered as a pair with node s_{13}, and s_{13} brings an improvement of 14% to s_3. We see a 13% improvement when the node s_1 is paired with s_{13}, and the improvement is 17% when we pair s_{13} with s_1. The minimum and maximum values of improvement are 9% and 18% respectively.

There is a trend that with increasing individual completeness of sniffers, their average pairwise gain also increases. We believe this metric can enhance the quality of measurements if only two sniffers are to be used. This graph can help us in the following ways, alternatively:

— It enables us to carry out an experiment of a small duration with all m sniffers co-located and then find out the best two sniffers for further experiments.

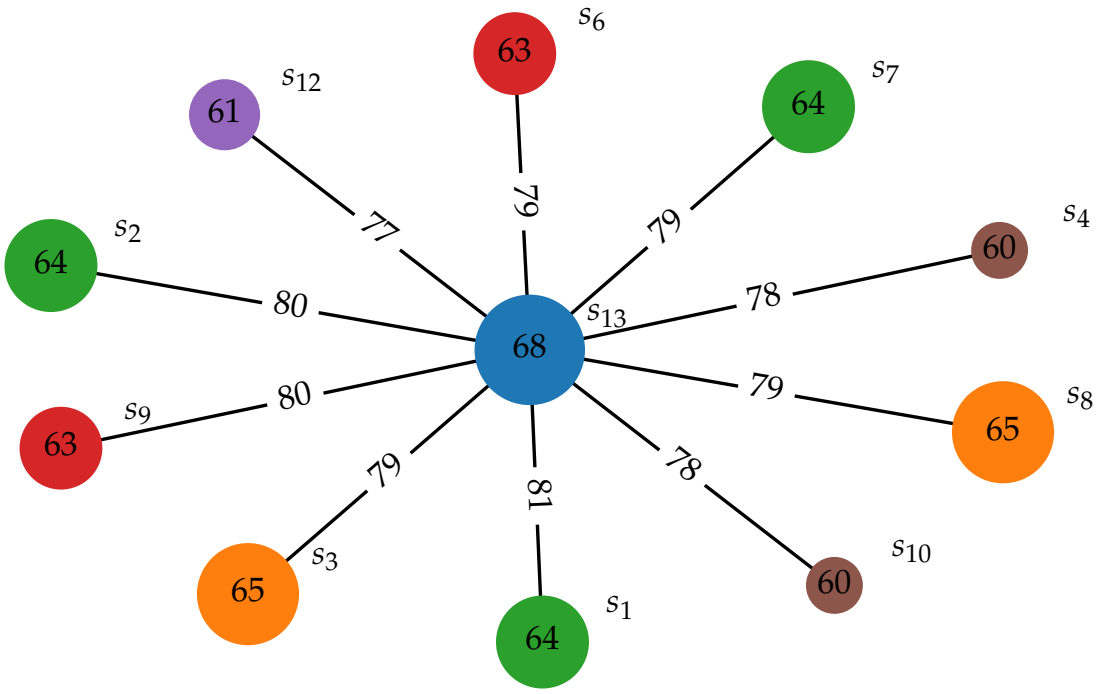


Figure 4.10. – Pairwise completeness. The improvement in completeness that 2 sniffers bring to each other when considered as a pair. The values inside the circles depict the completeness level of each sniffer and the edges' weights highlight the improvement 2 sniffers bring as a pair.

— We do the experimentation as planned and then we create this star as an initial analysis to select the two best nodes.

In this way, we are sure of getting the best traces for different analyses.

4.3.4 *Completeness gain*

We define the completeness gain as the improvement a super-sniffer of each size introduces. The completeness that we consider here is the average of all combinations of the super-sniffer of a specific size. Hake's method of finding

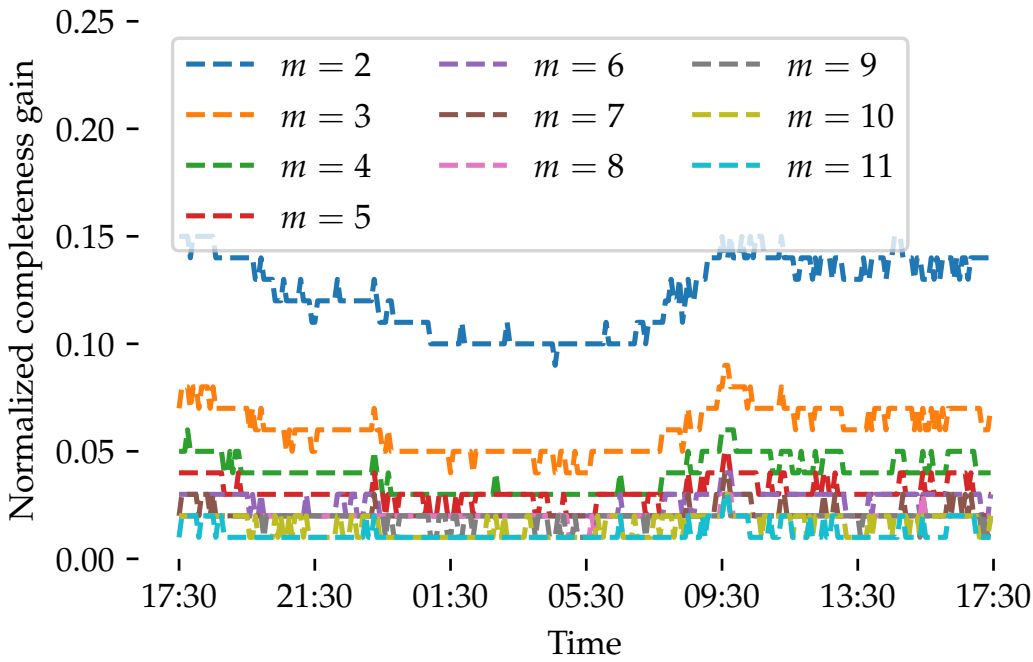


Figure 4.11. – Normalized gain of average completeness over time for super-sniffers of all sizes.

the normalized gain is widely used for determining the quality a new method brings in comparison with the existing methods/results [106]. The formula is as follows:

$$<g> = \frac{<post> - <pre>}{100 - <pre>}, \tag{4.9}$$

where <*pre*> and <*post*> refer to the results obtained before and after the improvements, respectively. The normalized gain is also known as the g-factor.

Figure 4.11 represents the normalized gain of average completeness for super-sniffers of all sizes. We see that adding a single sniffer to compose a super-sniffer of size 2 results in a significant gain in completeness. There is still

significant gain when we add another sniffer to the super-sniffer, i.e., $m = 3$. We get some gain as we keep adding sniffers until we get the super-sniffer of maximum size $m = 11$; however, the value of gain keeps reducing.

4.3.5 *Evaluation*

In this section, we present the results of completeness for super-sniffers of all sizes as well as our reference super-sniffer. We build our super-sniffer as follows:

- Single sniffer: $m = 1$. The reference in this case is the single sniffer that gives the best completeness.
- Reference super-sniffer of size $m = 2$; which is the combination giving the best completeness among all super-sniffers of size 2 containing the reference single sniffer (i.e., $m = 1$).
- Reference super-sniffer of size $m = 3$ is the one giving the best completeness among all super-sniffers of size 3 containing the reference super-sniffer of size $m = 2$.
- We proceed the same way for the remaining super-sniffers up to $m = 11$.

Figure 4.12 shows the minimum, maximum, and average completeness for all combinations of m sniffers, and reference completeness of our reference super-sniffer, represented by blue, yellow, green, and red lines, respectively. The numbers above the lines represent the improvement that our reference super-sniffer of each size brings to the table. The x-axis represents the time, while the y-axis gives the completeness for a combination of up to 11 sniffers. These are the results over 24 hours.

We observe that the completeness improves by 13% by adding only one sniffer to make a reference super-sniffer of size $m = 2$. Adding one more sniffer brings a further improvement of 6% in the value of completeness. We keep seeing some improvement as we keep increasing the size of our reference super-sniffer. The rate of improvement keeps decreasing with every addition

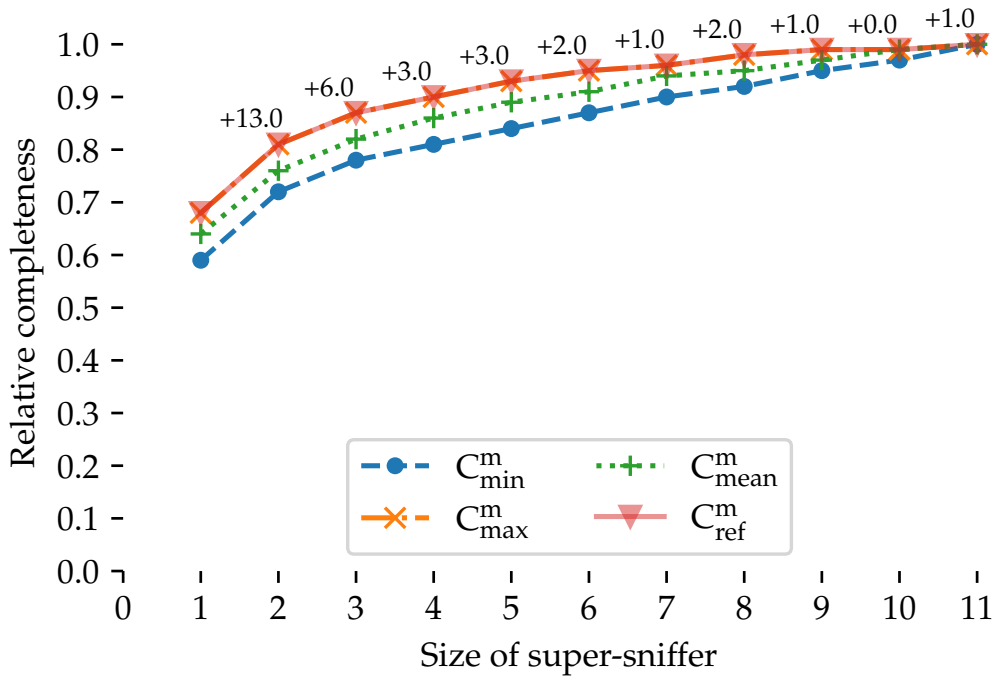


Figure 4.12. – Super-sniffer completeness. Minimum, maximum, average, and reference completeness as a function of the size of super-sniffer for super-sniffers of all sizes.

of a sniffer to the super-sniffer, but each sniffer brings new information to the super-sniffer. We observe that when we go from the super-sniffer of size $m = 9$ to $m = 10$ there is no improvement in reference/maximum completeness in our case; there is still an improvement of 2% in the case of minimum and average completeness though.

We also notice that the maximum and reference completeness are identical for super-sniffers of all sizes. It means that the super-sniffer of a certain size that achieves maximum completeness is part of the best-performing super-sniffer of the succeeding size. The minimum and maximum completenesses are also not too far apart.

Figure 4.13 shows the completenesses of our reference super-sniffer as a function of time. We see that the completeness improves significantly by adding

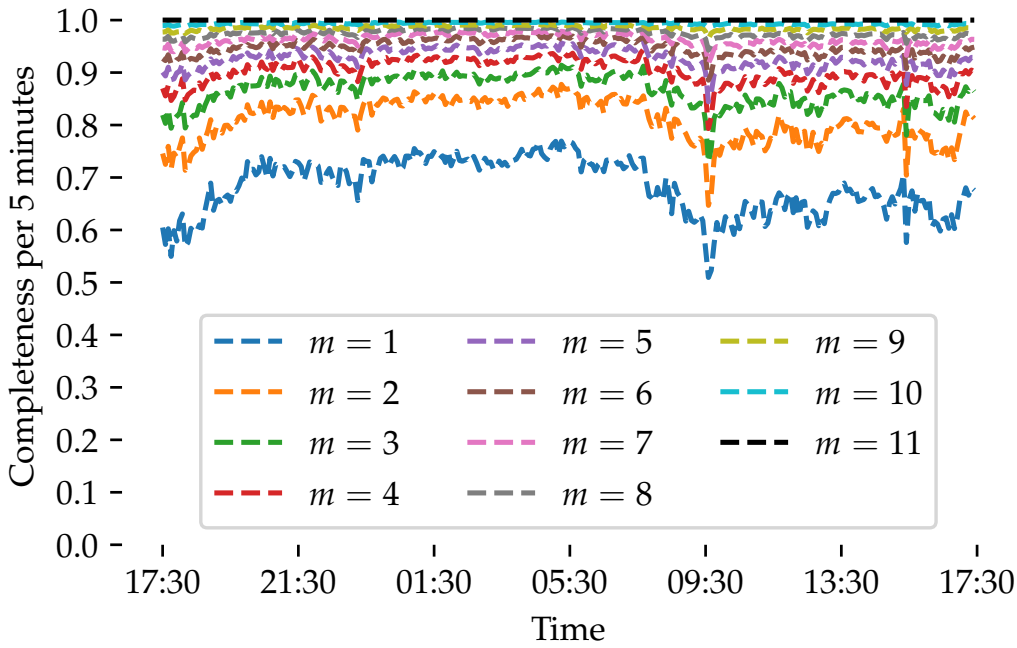


Figure 4.13. – Completeness of reference super-sniffer. The completeness of our reference super-sniffers of all sizes over the course of 24 hours. This shows the variation in the level of completeness over time.

just one sniffer to make the super-sniffer of size $m = 2$. The improvement is around 10% for the whole duration of 24 hours and it also varies concerning the traffic load. The rate of improvement keeps decreasing with every addition of a sniffer to the super-sniffer, but each sniffer brings new information to the super-sniffer. We note improvement in the quality of the trace capture with redundancy irrespective of the traffic load and time of the day.

The value of completeness decreases when there is more traffic in the medium during office hours, most notably around 09:30 in the morning. The use of a super-sniffer, however, helps improve the quality of capture even during the high load. It indicates that, comparatively, a higher number of sniffers

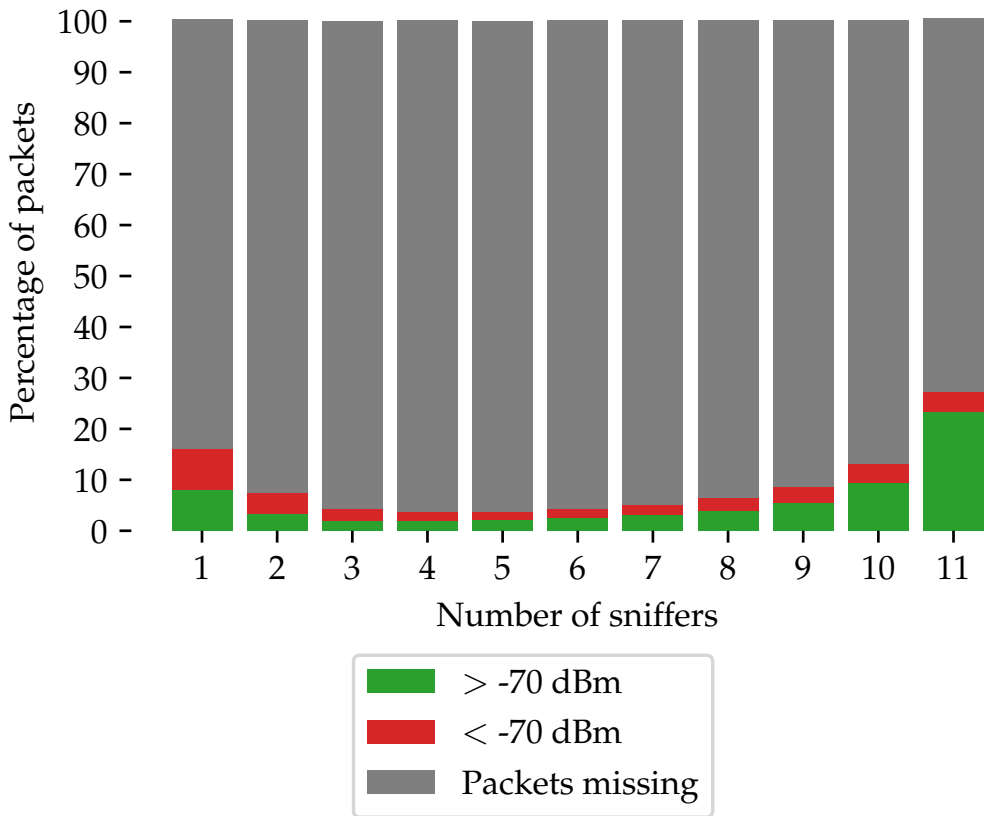


Figure 4.14. – Super-sniffer wise RSSI distribution. The percentage of packets captured with good (> -70 dBm) and bad (< -70 dBm) RSSI, as well as the percentage of packets missed by the combination of each number of sniffers.

are needed when the traffic in the medium is really high. In either case, our concept of super-sniffer increases the value of completeness.

RSSI. Figure 4.14 depicts the percentage of packets captured by combinations of each number of sniffers with good or bad RSSI values, as well as the percentage of the packets missed. We see that around 80% of the packets are missed if we use single sniffers since 20% of the packets are captured by only individual sniffers. 10% packets are captured by strictly 2 sniffers.

Similarly, around 15% and 30% of packets are captured by 10 and all 11 sniffers respectively. There are redundant packets that are removed in the process of merging, but there is a percentage of packets missed by a fewer number of sniffers. This finding further strengthens the use of a super-sniffer to improve the quality of the capture.

4.4 ABSOLUTE COMPLETENESS

4.4.1 *Experimental Setup*

As discussed in Section 4.3, there is no guarantee that a sniffer is able to capture all the traffic that was present in the medium at the time of the capture. Since we have no control over the devices generating Wi-Fi traffic in the vicinity, we do not know how much traffic is actually present in the medium. We, therefore, use our own source node to generate Wi-Fi traffic so that we exactly know what amount of traffic is generated and what proportion is captured by the sniffers [8].

We have eleven RPi4 nodes in our measurement set-up, ten as sniffers and one as the source node for generating Wi-Fi traffic. The source node sends Wi-Fi traffic at a rate of 10 packets per second. We run experiments in the outdoor scenario to examine the traces collected by the individual sniffers. We place the source node at distances of 1, 10, 20, 40, and 50 meters from the sniffers for these experiments. We run the test 5 times at each distance. The duration of each test is one minute. We present the results for all combinations of $m = \{1, 2, \ldots, 10\}$ sniffers for each distance. Table 4.1 shows the number of

8. We realize that our experimental setup has a limitation as there is no benchmark set in a controlled environment. There could still be interference from the other nodes present in the medium at the time of measurements. However, we plan to perform experiments in an anechoic chamber in the near future where we will not have any external interference. It will help us measure and define *absolute completeness* in the true sense.

super-sniffers of size m that we can obtain from our 10 sniffers. The sniffers remain stationary for the whole duration of the experiments.

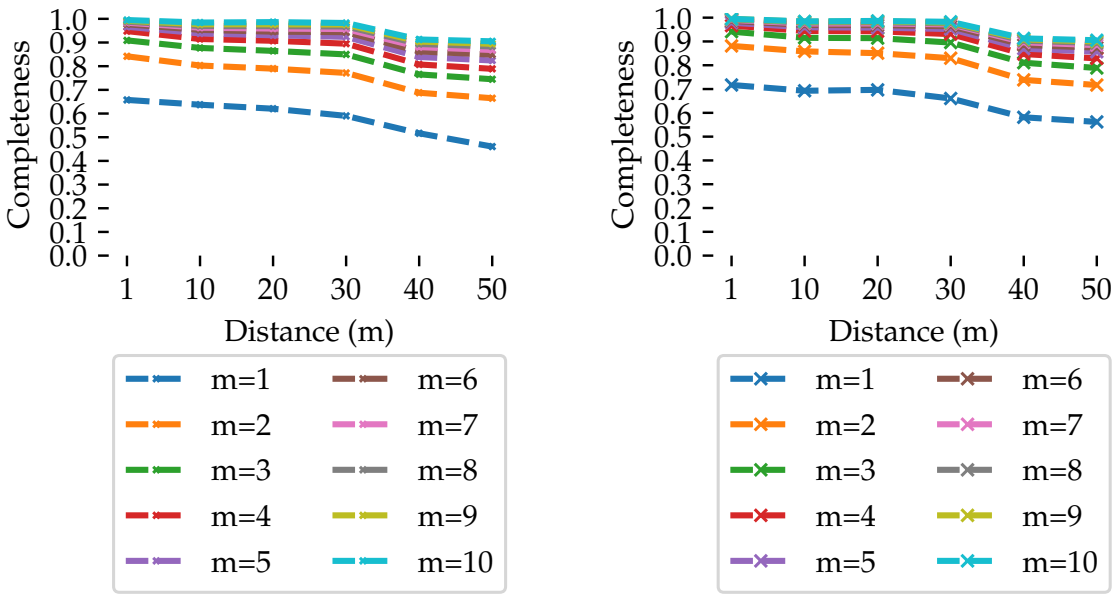
Table 4.1. – Number of combinations of super-sniffers of size m for our experimental scenario of 10 sniffers.

	Combinations of super-sniffers								
Size	2	3	4	5	6	7	8	9	10
Number	45	120	210	252	210	120	45	10	1

4.4.2 *Evaluation*

We know that the farther the source, the fewer the chances of its traffic being captured by the sniffers. It means there are more chances of a sniffer missing packets if the source of traffic is far. Figure 4.15 shows the impact of *distance* of source from the sniffers on the *minimum* and *maximum* completeness of all super-sniffers of sizes $m = \{1, 2, \ldots, 10\}$. The dark blue line at the bottom represents the results for m=1 for each distance of the source and the sky blue line at the top for m=10.

Figures 4.15a and 4.15b present respectively the minimum and maximum completeness of super-sniffers up to size ten for each distance. The improvement in completeness stands out for super-sniffers of sizes up to 5. The results are closely spaced for super-sniffers of size 5 and more. These observations hold for both cases. The minimum completeness for a 50m distance is 46% when $m = 1$ but it reaches 90% for a super-sniffer of size $m = 10$. We see that the completeness improves for each distance as the size of the super-sniffer increases. It is also evident from the bottom two lines that completeness improves massively just by increasing the number of sniffers from $m = 1$ to $m = 2$. This result holds for all distances.



(a) Minimum completeness.

(b) Maximum completeness.

Figure 4.15. – Minimum and maximum completeness of super-sniffers of the same sizes at each distance. The completeness for each m decreases as the distance increases.

Table 4.2 shows the maximum completeness (C^m_{max}) gain for the super-sniffers of each size at distances of 1m and 50m (note the trend is similar for the remaining distances). We get a noteworthy improvement of 16.5% by adding just one sniffer ($m = 2$) for 1m. The rate of improvement keeps falling as we keep adding a sniffer to the super-sniffer. It eventually becomes zero for $m = 10$. We see a similar trend for 50 m where there is an improvement of 15.5% for $m = 2$. The value of improvement again keeps decreasing for increasing m. The improvement in completeness is stagnant for super-sniffers of size greater than 5 and 7 for 1m and 50m respectively. It implies that more sniffers are needed if the source is far from the sniffers. The redundancy in the number of sniffers, therefore, improves the quality of the traces captured as it increases completeness.

Table 4.2. – Maximum gain of completeness for combinations of different numbers of sniffers compared to a single sniffer at 1m, 10m, 20m, 30m, 40m, and 50m.

Number of sniffers	1m	10m	20m	30m	40m	50m
2	16.4	16.5	15.4	16.9	15.7	15.5
3	22.4	22.2	21.7	23.5	22.9	22.7
4	24.9	25.1	24.6	27.0	26.6	26.7
5	26.1	26.6	26.1	29.1	28.9	29.5
6	26.8	27.6	27.1	30.3	30.4	31.3
7	27.2	28.2	27.8	31.1	31.5	32.3
8	27.4	28.6	28.3	31.5	32.2	33.2
9	27.6	28.9	28.7	31.8	32.7	33.8
10	27.6	29.1	28.9	32.1	33.1	34.3

Distribution of Packets Captured

Figure 4.16 shows the average percentage of packets lost by super-sniffers of all sizes ($m = 1, 2, \ldots, 10$) at each distance i.e., not a single sniffer captured these packets. The dark blue line at the top and teal line at the bottom represent the results for the minimum and maximum size of the super-sniffer respectively. As expected, hardly any packets are lost at 1m for $m = 10$ but the values are higher for a lower number of m, especially 1 and 2. For $m = 10$, around 1.75% of packets are lost up to the distance of 30m, and then there is a sharp rise at 40m where the percentage of packets lost reaches 9.5%. The maximum amount of packets is lost at a 50m distance which is the expected behavior. If we consider the individual sniffer, around 31% of packets are lost at a distance of 1m. The percentage increases as the distance of the source increases. Nearly 48% of packets are lost by the single sniffer for a distance of 50m. Whereas, the percentage falls to 9.5% for the same distance when $m = 10$. We notice that the completeness of individual sniffers ($m = 1$) is not identical for all tests which means the lowest and the highest packet loss is not always from the same sniffer. This observation coupled with the fact that the packets are

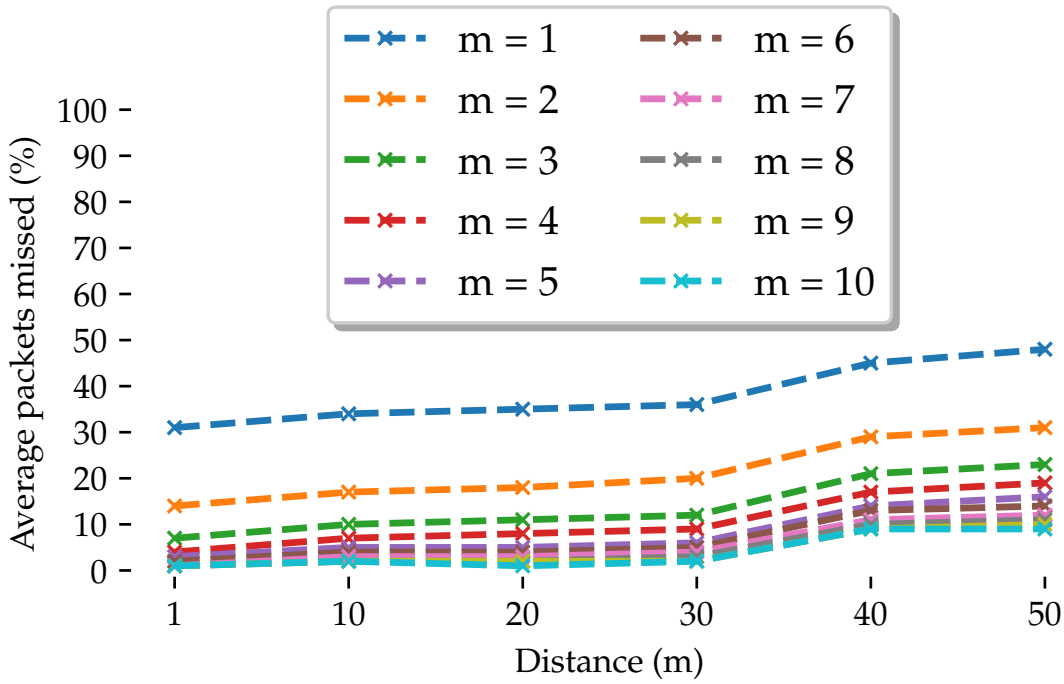


Figure 4.16. – Missed packets. Average packets missed by super-sniffers of all sizes ($m = 1, 2, \ldots, 10$) for different distances between the source and super-sniffer.

not missed by all sniffers simultaneously indicates that it is likely to be due to poor reception and not necessarily collisions at the receiver(s).

It is hard to differentiate between the average percentage of packets missed for super-sniffers of size 5 and more in Figure 4.16. To address the matter in question, Table 4.3 presents 95% Confidence Interval (CI) in the form of range for super-sniffers of size up to 9 for distances 1, 30, and 50 meters (the results are similar for 10, 20, and 40 meters). We see that the difference between the ranges is quite small for super-sniffers of size greater than 5 for all distances and that is why those are very closely spaced in the graph. The CI ranges reduce with increasing m which means we are more confident about our results with the introduction of the super-sniffer. However, there is no overlap in the intervals. This means that each new sniffer still brings new information

Table 4.3. – Confidence interval of the percentage of missed packets.

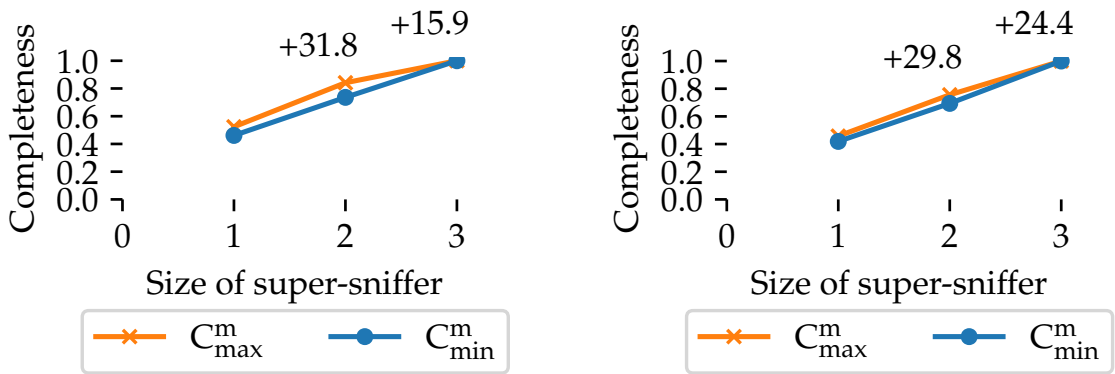
Size of the super-sniffer	**Distance** 10m	30m	50m
1	32.46 - 34.97	34.50 - 37.59	45.24 - 49.82
2	16.49 - 17.23	19.10 - 19.92	30.60 - 31.50
3	10.07 - 10.36	12.21 - 12.56	23.04 - 23.41
4	6.85 - 7.01	8.44 - 8.64	18.60 - 18.81
5	4.99 - 5.10	6.13 - 6.26	15.70 - 15.85
6	3.79 - 3.89	4.60 - 4.71	13.68 - 13.81
7	2.96 - 3.07	3.54 - 3.65	12.19 - 12.32
8	2.34 - 2.48	2.75 - 2.88	11.04 - 11.20
9	1.82 - 2.07	2.13 - 2.32	10.10 - 10.35

to the super-sniffer, albeit minimal. Therefore, it is up to the designer which level of completeness is desirable. It could be interesting to run a preliminary test of 5 minutes to have an idea of how to visualize the measuring system.

We carried out detailed analyses of the relative and absolute completeness in the previous and this section respectively. We assess the possibility of using multiple Wi-Fi antennas with a single RPi node to achieve redundancy at a lower cost in the following section.

4.5 ALTERNATE REDUNDANCY

We introduce redundancy in the number of sniffers but it comes at a financial cost. Since an RPi node boasts multiple USB ports, multiple wireless adapters can be plugged in at the same time. This can allow us to achieve redundancy at a lower cost.



(a) Multiple antennas.

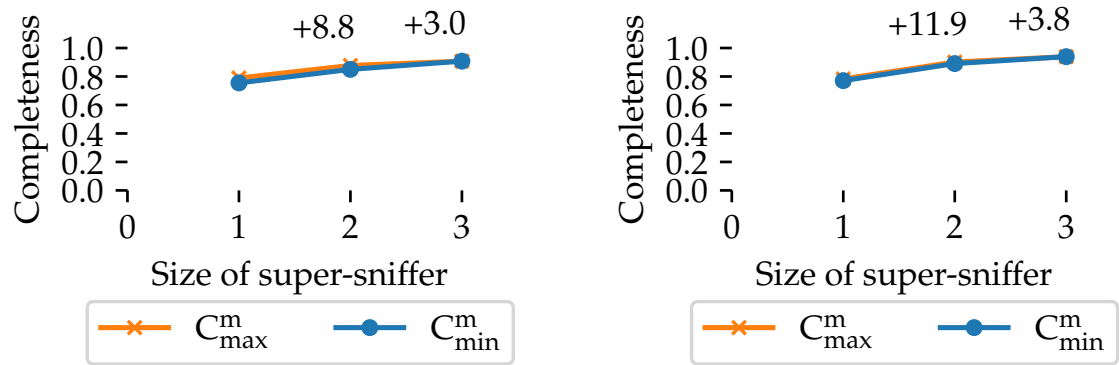
(b) Single antennas.

Figure 4.17. – Relative completeness. Evaluation of relative completeness for alternate redundancy, i.e. a) 3 (multiple) antennas attached to a single RPi4 node. b) 3 RPi4 nodes with one antenna each.

4.5.1 *Experimental setup*

We perform experiments with RPi4 nodes. We have 3 RPi4 nodes with 1 antenna each and 1 RPi4 node with 3 antennas simultaneously [9]. We do the testing for both relative and absolute completenesses in the office environment. We perform 10 tests and each test lasts 5 minutes. The sniffers simultaneously take a 2-minute gap between tests for relative completeness. On the other hand, the sniffers keep sniffing for the whole duration of the experiment for absolute completeness and it is the source node that takes 2-minute gaps between each 5-minute stream of generating traffic. In this case, we configure the sniffers to capture data only from our own source node, which is also an RPi4.

9. An RPi4 node has 4 USB ports but 3 is the maximum number of Wi-Fi antennas that it can handle at the same time. We tried to do the testing with more than 3 antennas connected to it but the antennas kept fluctuating.



(a) Multiple antennas.

(b) Single antennas.

Figure 4.18. – Absolute completeness. Evaluation of absolute completeness for alternate redundancy, i.e. a) 3 (multiple) antennas attached to a single RPi4 node. b) 3 RPi4 nodes with one antenna each.

4.5.2 *Evaluation*

We plot the completeness for RPi4 nodes with multiple and single antennas to make a comparison of performance. Figure 4.17 shows one such completeness plot for relative completeness. We see the plot completeness achieved with three antennas attached to a single RPi4 and with 3 RPi4 nodes each having one antenna in Figures 4.17a and 4.17b respectively.

The orange and blue lines represent the maximum and minimum completenesses respectively. The values are averaged over ten tests. We observe that the RPi4 with multiple antennas achieves slightly higher maximum completeness for one sniffer/antenna with 52%. The maximum value in the case of single antennas is 46%. The completeness increases to 84% in the maximum and 76% in the minimum case with the addition of an antenna. Whereas, the same values are 76% and 69% respectively when one sniffer is added to make a super-sniffer of size $m = 2$ with 2 single antennas. The values similarly increase for both cases for a super-sniffer of size $m = 3$. We see that

the alternate redundancy works well for relative completeness and we can, therefore, achieve the desired results with redundancy at a lower financial cost.

Figure 4.18 presents the results of alternate redundancy for absolute completeness. We observe that there is not much difference between multiple and single antenna cases. The maximum completeness achieved by one sniffer/antenna is 79% and 78% in multiple and single scenarios respectively. There is a similar kind of increase in completeness for both cases as the size of the super-sniffer increases. It further strengthens our ascertainment that a single sniffer is not enough for capturing a major chunk of the traffic and the use of a super-sniffer improves the quality of the traces. Furthermore, alternate redundancy works equally well for absolute completeness, thus reducing the cost of redundancy.

4.6 CONCLUSION

We know that a single sniffer performs poorly irrespective of the conditions of the wireless medium. We experimentally proved in Chapter 3 that there is a need to introduce redundancy in the number of sniffers to improve the quality of the traces. As a pre-requisite to the evaluation of redundancy, we i) define and formulate trace completeness and relative and absolute completeness, ii) create a tool called PyPal for time-synchronization of Wi-Fi traces, iii) do experimental evaluations of redundancy in different scenarios to prove that it significantly improves the quality of capture that helps to do a better analysis.

The redundancy in the number of sniffers, however, comes at a financial as well as management cost. There is a trade-off between the cost of the sniffers and the level of performance in the quality of trace capture. We advise choosing a higher number of sniffers when the traffic load in the medium is

high. At the same time, one needs to be careful about the choice of sniffers. We notice that all the sniffers do not behave the same way despite the conditions of the medium changing over the course of 24 hours. We show that the faulty sniffers can be removed from the detailed analysis after some initial diagnosis but it is not easy to have a consistent platform from the very beginning.

There is a possibility of using the concept of super-sniffers for various purposes. One such application of redundancy in passive measurements is distance measurement using the RSSI values. We evaluate this application in the next chapter, while highlighting the use of redundancy in the number of sniffers can improve the error in distance estimation.

5

Application of redundant passive measurement: Distance estimation

Distance estimation and localization are important factors in sensor networks, as well as for trajectory reconstruction and location-centric services. As mentioned in Chapter 1, RSSI is a popular choice for distance estimation and localization. In this chapter, we highlight the incoherence in the RSSI value of a single packet as seen by different sniffers and we also provide experimental proof that the use of a super-sniffer improves the error in distance estimation.

5.1 Experimental setup and dataset

We use six RPi4 nodes in our measurement set-up, five as co-located sniffers and one as the source of Wi-Fi traffic. We capture traces outdoors while maintaining a line of sight between the source node and the sniffers. We place sniffers side by side with a distance of 20 cm between each of them, and those remain stationary for the duration of the experiments. We place the source node at distances of 1, 10, 20, 30, 40, and 50 meters from the sniffers. We perform four tests for each distance, each lasting five minutes.

Dataset

Each sniffer generates one trace per test for each distance. As discussed earlier, we run the test 4 times. So we obtain 120 traces in total. As the first step after time synchronization, we organize all traces within a single test – we perform this step separately for each distance. Once we have a single trace per test, we combine the traces from each test to get one single trace for each distance. At this stage, we have all the traffic captured in the four tests at a given distance in one final trace. We perform the analysis to identify the packets that are captured by multiple sniffers and group them by their RSSI values. So, we have a single concatenated file per distance representing the RSSI values of each packet as seen by different sniffers. For the sake of illustration, we provide in Table 5.1 a snapshot of such a trace for the distance of 50 m. Note that, for example, sniffer s_4 misses the first packet, while sniffers s_1, s_3, and s_5 miss the second packet. We use this dataset for all the analyses we present in this chapter.

5.2 MEASURING DISTANCE WITH RSSI

We use the Log-Distance Path Loss (LDPL) model to estimate the distance and put values into context. The formula for LDPL is as follows [107, 108]:

$$RSSI(d) = RSSI(d_0) - 10 \times \eta \times \log\left(\frac{d}{d_0}\right), \tag{5.1}$$

where $RSSI(d)$ is the RSSI value seen by the sniffers when those are at a distance d from the source, $RSSI(d_0)$ is the RSSI value at a close-in reference distance d_0, and η is the path loss index which is dependent on the propagation environment. We use the following values for our experimental setup:

— d_0 = 1 meter,
— average RSSI at d_0 = -19.7 dBm, and

Table 5.1. – Per-packet per-sniffer RSSI (dBm) at 50 m for a subset of the collected trace. "–" means that a particular sniffer does not capture the packet.

	RSSI captured per sniffer				
Packet	s_1	s_2	s_3	s_4	s_5
1	-48	-54	-48	–	-46
2	–	-78	–	-48	–
3	–	-72	–	–	–
4	–	–	-64	–	–
5	-48	-56	-48	-48	-44
6	-48	-54	-50	-48	-46
7	-46	-54	-50	-48	-44
8	-46	–	-48	-48	-44
9	-48	–	–	–	–
10	–	–	–	–	-44
11	–	–	–	-50	–
12	-46	-54	-50	-48	-44

— η = 1.75.

In our experiments, the average RSSI value at a reference distance of 1 m is -19.7 dBm. Although there are a few building walls on the side, we maintain Line of Sight (LoS) in our experiments, so we choose 1.75 as the value of η. The specifications of the hardware are as follows:

— transmission power: 27 dBm
— antenna gain: 5 dBi
— receiver sensitivity: -93 dBm

5.3 IMPACT OF BURST SIZE: CONSECUTIVE PACKETS MISSING

We now look into the *burst size* of consecutive packets missing. It is an important parameter as it translates into a period during which the sniffers

fail to detect the presence of devices – depending on their mobility, the devices/users may traverse the monitored zone without being even detected. We show in Table 5.2 the maximum burst size (M.B.S.) of consecutive packet misses for each one of the five sniffers at each distance, along with the ratio between the number of maximum size bursts and the total number of bursts (Freq.). The numbers are far from negligible in certain cases. We see a gap of 259 consecutive packet misses by sniffer s_3 for the distance of 20 m. Recall that our source sends packets at a rate of 10 packets/s. Thus, s_3 misses packets for 25.9 seconds, which is massive! For contextualization, the average walking speed of an adult is between 1.2 m/s and 1.4 m/s, which means that a mobile user would have moved between 31.08 m and 36.26 m in 25.9 seconds [109, 110].

The probe request frames are often used for trajectory reconstruction [111, 112]. The phenomenon is even worse if we consider an average transmission rate of 1,028 probe request frames per hour [31]. If a sniffer misses 259 consecutive packets, then it will end up missing traffic for 15.24 minutes. The user would have covered a distance between 1,097 and 1,280 meters. It means by the time this sniffer captures the next packet, the user is most likely to be out of the coverage area of several following sniffers.

We do the same analysis using the notion of super-sniffers that we introduced in Section 4.1. We analyze all possible combinations of sniffers for a given size of a super-sniffer, which gives a total of $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ combinations. We show in Table 5.3 the results for the average maximum burst size of consecutive packets missing. We see that even the average value of maximum burst size for single sniffers is quite high for all distances. However, the burst size decreases significantly for a super-sniffer of size $m = 2$ for each distance. The results improve further for super-sniffers of larger sizes. The results are zero for the captured traffic when we consider the combination of all sniffers since each packet is captured by at least one sniffer.

Table 5.2. – Maximum burst size (M.B.S.) for individual sniffers and the ratio between the number of maximum size bursts and the total number of bursts (Freq.).

	1 m		**10 m**		**20 m**		**30 m**		**40 m**		**50 m**	
Sniffer	**M.B.S.**	**Freq.**	**M.B.S.**	**Freq.**	**M.B.S.**	**Freq.**	**M.B.S.**	**Freq.**	**M.B.S.**	**Freq.**	**M.B.S.**	**Freq.**
s_1	7	1/1997	7	2/1926	10	1/2065	9	1/2120	9	1/2090	12	1/2098
s_2	6	1/2063	8	1/1952	8	2/2030	7	5/2077	8	2/2042	9	3/2067
s_3	9	1/2142	9	1/2078	259	1/2040	7	3/2147	11	1/2058	9	2/2110
s_4	6	4/2077	8	1/1973	8	1/2035	11	1/2187	9	1/2081	11	1/2082
s_5	54	1/2078	7	2/1931	7	3/2101	8	1/2079	13	1/2115	13	1/2073

Table 5.3. – Average burst size of consecutive packets missing for multiple sniffers.

Number of combined sniffers	**Average burst size per distance**					
	1 m	10 m	20 m	30 m	40 m	50 m
1	16	8	58	8	10	11
2	4	5	5	5	5	6
3	3	3	3	4	3	4
4	2	2	2	2	2	3
5	0	0	0	0	0	0

5.4 USING SUPER-SNIFFERS TO ESTIMATE DISTANCE

We see in Table 5.1 that -46 dBm and -48 dBm are the most common RSSI values for all sniffers, but we see some values as -78 dBm, -72 dBm, -56 dBm, and -54 dBm. We classify these values as outliers by using the Majority Rule Scheme (MRS) strategy [91]. These values are not coherent with other measures and result in large errors in distance estimation. We have such RSSI outliers for all distances in our experiments. We calculate the average RSSI value for each packet for all combinations of sniffers for redundancy of 1 to 5.

Table 5.4 shows the average per-packet error in distance estimate in meters, using two sniffers with the least error, for combinations of all numbers of sniffers at 50 m. We see in row 1 that the average error of estimated distance for a combination of two sniffers is 6.24 m which is 2.35 m less than the average of individual sniffers. Similarly, if we consider only the average of single sniffers in the second row, the average error is huge, around 1050 m. However, the error goes down to 8.59 m when we take the average of the combination of two sniffers with the least error. Although the redundancy helps improve the error, the presence of these outliers still results in massive errors, as we see for combinations of four and five sniffers in the second row. We clean the dataset to remove the outliers. The RSSI values below

Table 5.4. – Raw per-packet average distance error for the distance of 50 m.

	Number of sniffers				
Packet	**1**	**2**	**3**	**4**	**5**
1	8.59	6.24	0.86	0.86	2.76
2	1051.8	8.59	8.59	128.32	248.05
3	924.03	924.03	924.03	924.03	924.03
4	289.96	289.96	289.96	289.96	289.96
5	8.59	6.24	0.86	2.76	3.99
6	6.24	2.76	0.86	0.45	1.5
7	6.24	2.76	0.86	1.6	6.35
8	8.59	8.59	10.41	13.66	16
9	8.59	8.59	8.59	8.59	8.59
10	25.53	25.53	25.53	25.53	25.53
11	3.88	3.88	3.88	3.88	3.88
12	6.24	2.76	0.86	1.6	6.35

-56 dBm appear in only 2.86% of the cases; thus, we remove them from the 50 m dataset. Similarly, values greater than -46 dBm account for only 1.66% of the values. We recalculate the average values in Table 5.5 and we already see a huge improvement. We see in row seven that the average error with five sniffers goes from 6.24 m to only 0.5 m. The rows with no values mean that the sniffers presented outlier values.

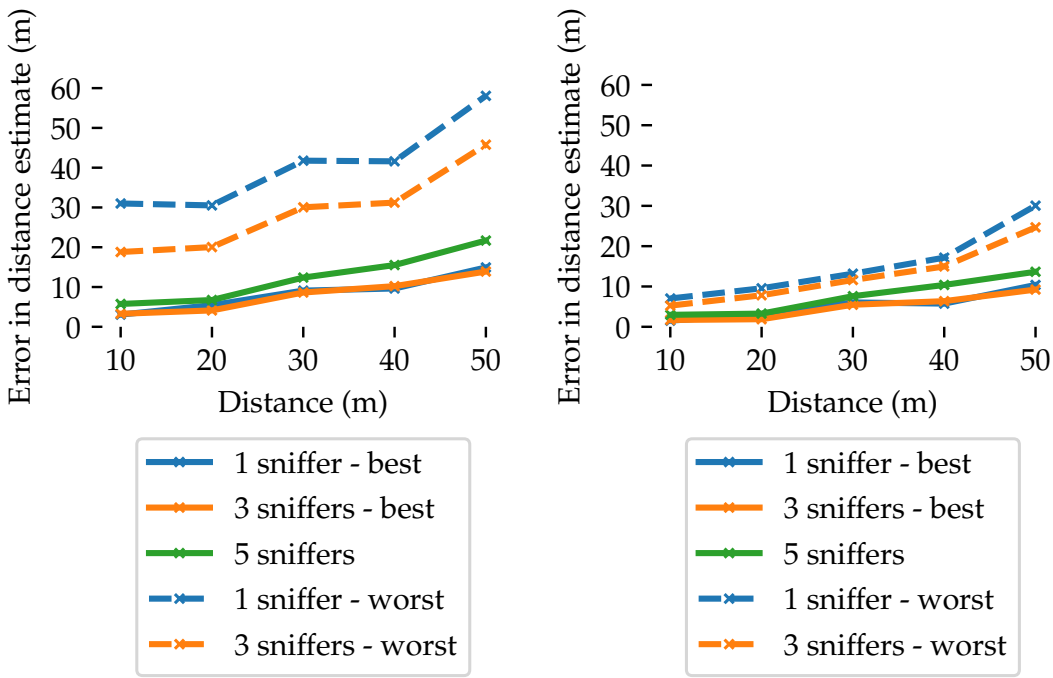
In Figure 5.1, we represent the best and worst-case results of average error in distance estimation for one, three, and all five sniffers, with and without outliers. We see that the worst-case error is massive for a single sniffer for all distances. Three sniffers yield an improvement, but the worst-case error is still significant. We also note in Figure 5.1b that removing RSSI outliers significantly improves the average error, including the worst case. Even a single sniffer gives low error for short distances but increases strikingly for longer distances, particularly for the worst-case scenario.

Table 5.5. – Per-packet average distance error after removing the outliers for the distance of 50 m.

	Number of sniffers				
Packet	**1**	**2**	**3**	**4**	**5**
1	8.59	6.24	0.86	0.86	2.76
2	8.59	8.59	8.59	8.59	8.59
3	–	–	–	–	–
4	–	–	–	–	–
5	8.59	8.59	8.59	6.24	3.88
6	6.24	2.76	0.86	0.45	1.5
7	6.24	3.32	1.81	0.66	0.45
8	8.59	8.59	8.59	10.41	12.23
9	8.59	8.59	8.59	8.59	8.59
10	–	–	–	–	–
11	3.88	3.88	3.88	3.88	3.88
12	6.24	3.32	1.81	0.66	0.45

Kalman filter: smoothing out RSSI outliers

Kalman filter is a very powerful filtering solution for minimizing the mean of the squared error [113, 114]. It is often used to filter the RSSI values to be then used for trajectory and positioning [92, 93, 115, 116, 117]. We use the Kalman filter to smooth out the RSSI outlier values in our dataset before computing the distance. Figure 5.2a shows the error in distance estimate in meters for the resulting dataset. When we compare it to the distance error that we obtain with MRS strategy outlier removal (in Figure 5.1b, we see some visible difference in the error level, especially for higher distances. We plot the difference between the MRS strategy and Kalman filter distance errors in meters in Figure 5.2b. The positive values mean the Kalman filter performs better, whereas negative values highlight better performance of the MRS strategy. We notice that the Kalman filter smoothness improves the error in the distance also for the worst sniffers. The error is smaller for even higher distances. However, the MRS strategy is very slightly better in the case of

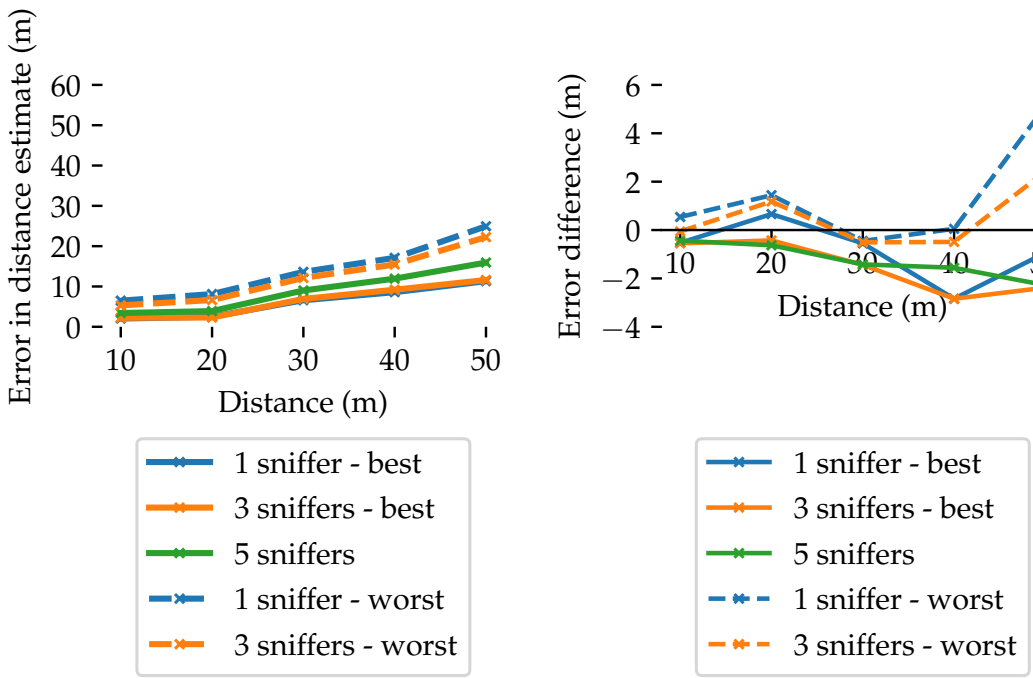


(a) With outliers.

(b) Without outliers.

Figure 5.1. – Error in distance estimate for 1, 3, and 5 sniffers with and without outliers.

best sniffers. The error with MRS is less than 1 m less than that of Kalman except for 40 m where the difference is around 3 m. However, when Kalman performs better, its error can be up to 5 m less than that of MRS. We believe there is not much to separate the two techniques, however, smoothing the RSSI values using the Kalman filter is better instead of completely removing the outlier RSSI values using the MRS strategy.



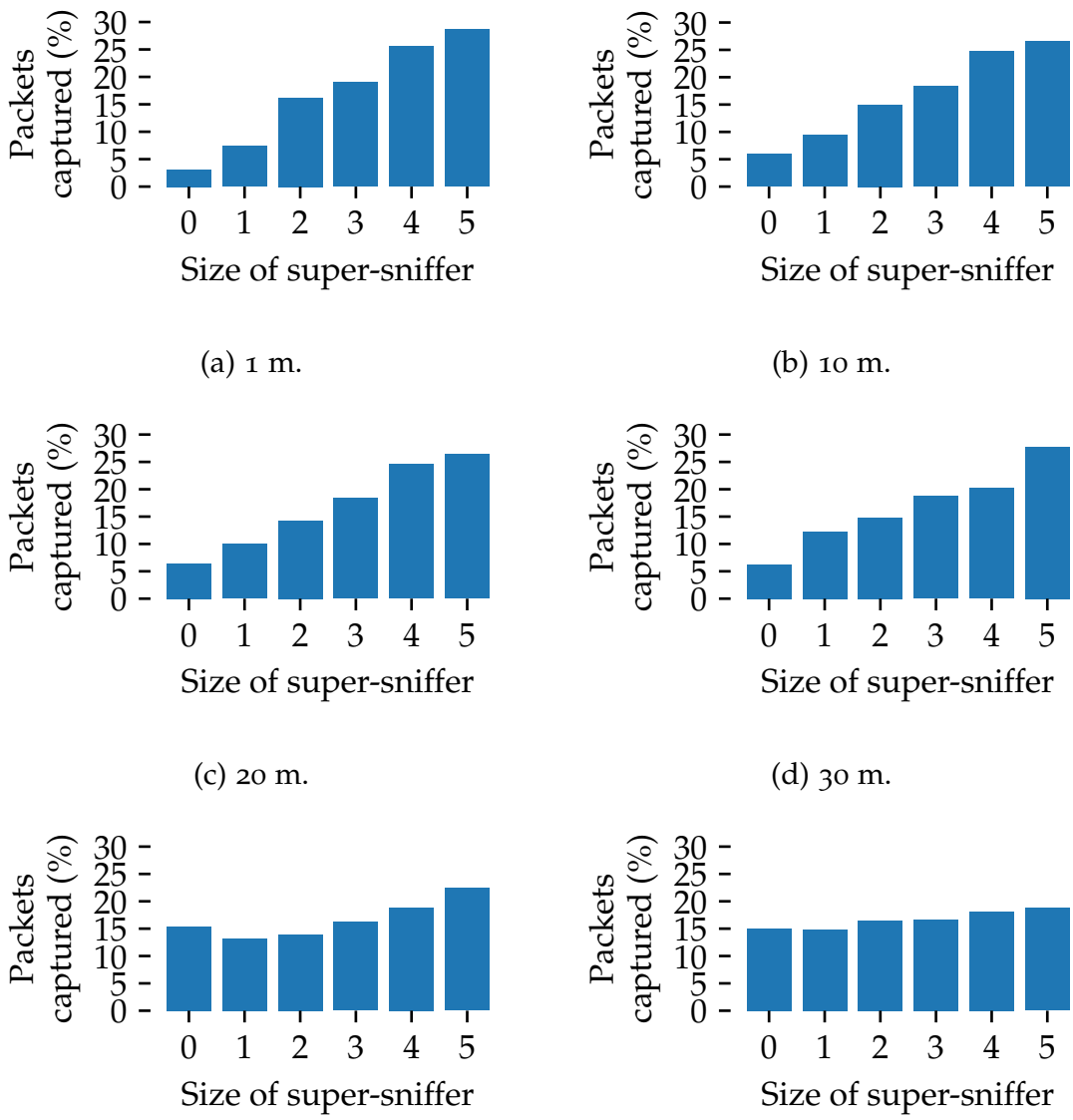
(a) Without outliers.

(b) Comparison.

Figure 5.2. – Comparison between MRS strategy and Kalman filter.
a) Error in distance estimate for 1, 3, and 5 sniffers using Kalman filter to remove outliers.
b) Distance error between MRS strategy and Kalman filter. The positive values mean the Kalman filter performs better, whereas negative values highlight better performance of the MRS strategy.

5.5 HISTOGRAM COMPARISON

It is quite challenging to make a correct estimate because of the inherent characteristics of the wireless medium. We discussed in the previous section that using redundancy in the number of sniffers improves the accuracy of distance estimation but the error still exists. In this section, we introduce our concept of distance estimation depending on the number of packets captured by the super-sniffers of all sizes.



(a) 1 m.

(b) 10 m.

(c) 20 m.

(d) 30 m.

(e) 40 m.

(f) 50 m.

Figure 5.3. – Histograms of packets captured and missed at 1 m, 10 m, 20 m, 30 m, 40 m, and 50 m distances.

5.5.1 *Comparison methodology*

Figure 5.3 represents histograms of the percentage of packets captured by super-sniffers of all sizes as well as the packets missed. We observe that there is a trend in the ratio of packets missed and captured as the distance increases. We use this information to compare the histograms to be able to estimate the distance. We compare the histograms using the Chi-squared test [118] and compare it with the ground truth to determine the coherence among the sniffers.

As described at the beginning of this chapter, we have 5 sniffers and we perform the test 4 times. So we have traces of 20 minutes in total per sniffer for each distance. We divide the 20-minute traces into 10 equal parts based on the number of packets for all distances so that we have one training and 9 testing tests. Figure 5.4 shows that we have a histogram for each of the 10 parts where τ_1 is the training set and π_1, π_2, ... π_9 are the testing sets. We follow the k-fold cross-validation mechanism [119] so we take all the tests as a training set once. We then do the histogram comparison in the following fashion:

— Compare 1 m histogram of each testing set with histograms of the training set at all distances.
— Repeat the same step for all other distance histograms of each testing set. We end up with 90 results for each distance (640 in total as we have 6 distances).
— Calculate the percentage of distance estimates for each distance i.e. what percentage of 1 m testing sets were correctly classified as 1 m or incorrectly identified as 10 m, 20 m, 30 m, 40 m, and 50 m. This step helps us evaluate our method.

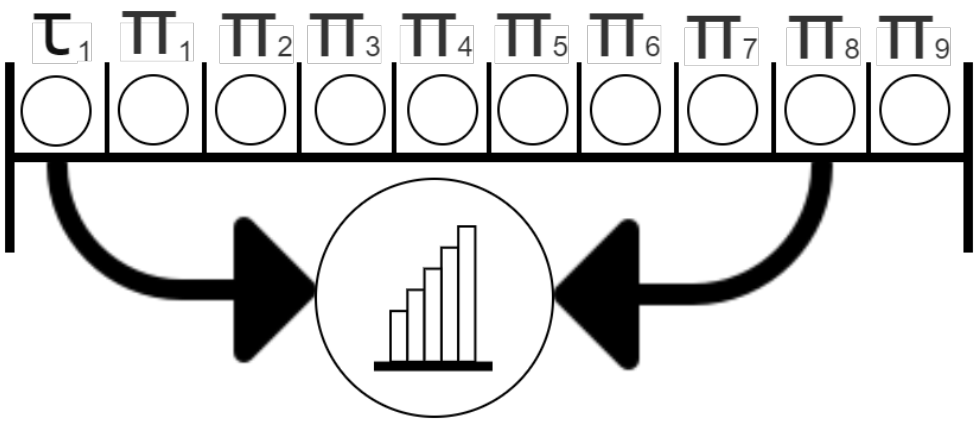


Figure 5.4. – Histogram comparison methodology using k-fold cross-validation mechanism.

5.5.2 *Evaluation*

We present the results in Table 5.6. Each row represents the results of the comparison of the testing set at a certain distance with all other distances of the training set. The idea behind this comparison is that the histograms should be similar at the same distance, hence the percentage values in bold (in diagonal) should be the highest.

We see in Table 5.6 that 63.33% of training sets of 1m are correctly identified as 1 m whereas 27.78%, 6.67%, and 2.22% are classified as 10 m, 20 m, and 30 m respectively. None of the tests are determined to be of distances 40 m or 50 m.

The results are less satisfactory for 10m and 33.33% and 23.33% testing sets are determined to belong to 1m and 20m respectively, while 23.33% are correctly identified as 10m. 5.56% are even estimated to be at a distance of 40m.

A little more than $1/3^{rd}$ are identified to be at a distance of 20 m and 40 for 20 m and 40 m tests respectively, whereas a majority (56.37%) of 30 m tests

Table 5.6. – Percentage of correct distance estimations among all tests.

Distances	1 m	10 m	20 m	30 m	40 m	50 m
1 m	**63.33**	27.78	6.67	2.22	0.00	0.00
10 m	33.33	**23.33**	23.33	14.44	5.56	0.00
20 m	5.56	12.22	**34.44**	24.44	23.33	0.00
30 m	3.33	7.78	25.56	**23.33**	31.11	8.89
40 m	0.00	0.00	23.33	21.11	**36.67**	18.89
50 m	0.00	0.00	0.00	6.67	11.11	**82.22**

are estimated to be either a distance of 20 m or 40 m. For 30 m, we notice that a certain percentage of tests has been identified for all the distances which means more error.

We get the best results for 50 m where 82.22% of the testing sets are correctly identified as 50 m and none of them are estimated to be at distances of 1 m, 10 m, or 20 m.

The results are encouraging and we plan to explore it further and combine it with RSSI values to improve the level of precision for distance estimation. We also intend to find and define the minimum number of packets and testing time required for this analysis.

5.6 CONCLUSION

The information of bursts of consecutive packets missing, and outliers in the RSSI values lead us to the decision that using a single sniffer is not enough, as it can have a massive impact if one has to measure mobility. The results show that combining the sniffers improves all three issues that we highlight. On the one hand, the results of our experiments in Table 5.5 show that using five sniffers primarily results in a minimum error in distance estimate. On the other hand, the results in Table 5.3 show that even three sniffers (row 3) reduce

the average burst size to 3 or 4, meaning the sniffers miss the traffic for 0.3∼0.4 seconds. The user could move between 0.36 and 0.56 meters in this period. Therefore, we propose that the redundancy of size 3 is good enough for an outdoor capture to rule out the anomalies in the trace capture and lead to more complete results and analysis. The best combination of 3 sniffers in Figure 5.1 slightly outperforms the best case single sniffer which leads us to conclude that a super-sniffer of size 3 is advisable for improving the quality of results while maintaining a comparatively low financial cost. Furthermore, we introduce a new technique of histogram comparison for distance estimation. The initial results are encouraging and we plan to combine histogram comparison with RSSI values to improve the accuracy.

6

Conclusion and future work

6.1 Conclusion

Passive sniffing is a widely used method in wireless networks, particularly for trajectory reconstruction and localization. However, it is not without its challenges. Due to the inherent characteristics of the wireless medium, a single sniffer is unable to capture traffic that is representative enough, which can result in incomplete and inaccurate analyses. Thus, it is crucial to measure trace completeness, but there is a lack of mathematical representation and empirical analysis of completeness in real-world experiments. Additionally, there is no readily available and easy-to-use tool for synchronizing and merging Wi-Fi (IEEE 802.11) traces to investigate the impact of redundancy on trace completeness. This thesis aims to address these issues.

Firstly, we developed a tool called PyPal, a Python version of WiPal, for the synchronization and merging of the Wi-Fi traces. This tool requires two traces as input in the text format, one as a reference and the other that needs to be synchronized. It utilizes specific header fields to aid in the synchronization process (refer to Appendix B for a command to extract the required fields from a pcap file). Once the traces are in the required format, PyPal provides the functionality of time-synchronizing the traces. It also provides the option of merging the traces while removing duplicate frames

in the process. Additionally, it provides the option to simply concatenate the two synchronized traces and create per-MAC-address traces.

In the second place, we present experimental evidence that a single sniffer is inadequate for capturing wireless medium traces that are sufficiently representative. We conduct an analysis of traces captured simultaneously by ten co-located sniffers of two different types in low and high-activity scenarios. Our findings indicate that, even in less active situations, the completeness of both types of sniffers does not exceed 54%. To further compare the two types of sniffers, we employ Jaccard similarity and show that the low completeness levels are not attributable to the sniffers themselves, but to the characteristics of the medium.

Thirdly, we introduce redundancy through an increased number of sniffers. However, to analyze the effectiveness of redundancy, we require a metric. We thus introduce the concepts of trace completeness and super-sniffer and provide a mathematical representation for both relative and absolute completeness. Relative completeness refers to the ability to capture all frames in the medium, while absolute completeness pertains to frames captured from a controlled source with known traffic volume. A super-sniffer comprises a collection of redundant co-located sniffers, such as a combination of three co-located sniffers forming a super-sniffer of size three. We conduct a 24-hour measurement experiment in an office environment using fourteen sniffers and apply our definition of trace completeness to redundancy. Our findings indicate that the addition of a single sniffer (to form a super-sniffer of size two) improves the capture quality by 13% in the best-case scenario. Furthermore, the completeness level increases further with an increase in the super-sniffer size. Our analysis of the RSSI values of the captured packets by each sniffer highlights the importance of analyzing individual sniffer performance before evaluating completeness due to the possibility of sniffer malfunction. We identified three out of fourteen sniffers as faulty and removed them from the analysis.

As previously mentioned, we introduce the concept of absolute completeness to measure the ratio of captured traffic compared to the actual amount of traffic generated. We generate Wi-Fi traffic using our own source node and place it at various distances during the experiment. Our findings indicate that, even in this scenario, a single sniffer fails to capture a substantial number of packets, and the ratio of missing packets increases with distance. However, the introduction of redundancy via the super-sniffer concept significantly improves the completeness of traces at all distances.

We acknowledge that redundancy incurs a financial cost. However, by utilizing the multiple USB ports available in a Raspberry Pi node, we can implement redundancy at a lower cost. Specifically, we can connect multiple antennas to a single RPi node, rather than employing multiple RPi nodes, each with a single antenna. We refer to this as alternate redundancy and evaluate it for both relative and absolute completeness. Our experiments demonstrate that an RPi node can support up to three external Wi-Fi antennas. We, therefore, use three antennas with a single RPi for the experiments, creating a super-sniffer of size three. We also construct another super-sniffer of size three using three RPi nodes, each with one antenna. Our results indicate that the RPi with three antennas performs equally well as the super-sniffer formed of three single sniffers. This approach provides a cost-effective solution for researchers seeking to implement redundancy in their systems.

Finally, we present an application of redundancy in the form of distance estimation. We utilize our own source node to obtain the ground truth of the actual distance between the source node and the sniffers. Despite the widespread use of RSSI in the literature for distance estimation and localization, concerns have been raised about the reliability of RSSI due to the characteristics of the wireless medium. Our experiments indicate that there is an inconsistency in RSSI values, i.e., the RSSI values of the same packet captured by different co-located sniffers can vary significantly at each sniffer, leading to flawed analysis. Moreover, a single sniffer may miss a burst of

packets at a given time, which can have a significant impact on trajectory reconstruction. However, we demonstrate that these issues can be resolved through redundancy, and the error in distance estimation can be reduced by using the RSSI values with the Log-Distance Path Loss (LDPL) model along with redundancy. We also introduce a new technique for distance estimation based on the number of packets missed and captured by super-sniffers of all sizes.

6.2 PERSPECTIVES

We have presented our contributions in this manuscript to improve the quality of Wi-Fi traces. We plan to extend the work to look at the concept and uses of redundancy from different angles. We list some open questions that can be investigated in future work:

- **Channel and Spectrum.** We did all the experiments on a single channel i.e. channel 1 of the 2.4 GHz spectrum. We intend to do experiments on different channels to analyze the traffic trend as well as trace completeness. We also plan to study the aspect of completeness for the 5 GHz spectrum.
  The possible impact of channel hopping on trace completeness is another possible future work. However, it would need some kind of synchronization among the co-located sniffers to make them hop channels simultaneously and also in the same order.
- **Antenna.** We use Alfa Network AWUS 051NH antennas for all our experiments. Our aim is to extend the experimentation to different antennas with different specifications that can be easily set to monitor mode. One such antenna that is easily available in the market is TP-Link WN722N [120].
- **Distance estimation.** We plan to investigate the inter-burst size distribution and its impact on long-term mobility analysis. We also intend

to determine the value of the path-loss index from the RSSI value that appears the most for each distance and use that to find the error distribution in the distance estimation.
We did the distance estimation analysis for outdoor experiments. We intend to extend it to indoor localization with redundancy. We believe it will similarly help with RSSI incoherence as there are a lot of phenomena happening more intensely in the wireless medium indoors for example reflection, refraction, diffraction, scattering, etc. The redundancy will also help in minimizing estimation errors.
We place the source node at different distances in our experiments. We get a histogram of packets captured for all sizes of super-sniffers for each distance. We plan to use this data as a training set and explore the possibility of localization with histogram comparison techniques.
Finally, we have the idea to extend our work to accommodate mobility with the introduction of moving source nodes. We believe it is a fascinating challenge to experiment with redundancy.

— **Histogram comparison.** The results of our histogram comparison technique for distance estimation are encouraging but there are still errors in the estimation. We plan to work on it further and combine this technique with RSSI values to improve the level of accuracy in distance estimation.

— **Mobility.** The sniffers remain stationary in all our experiments. We intend to study the impact of the mobility of the sniffers on trace completeness. We believe it is an interesting aspect of sniffing to study as there is a study for the impact of mobility on the accuracy of sensing [121].

— **BLE - BLEPal.** Similarly to PyPal, we have a tool for Bluetooth trace synchronization and merging, called BLEPal [122]. It was created by a group of students in a networking project that we supervised. A possible future work is the use of BLEPal to study the impact of redun-

dancy on trace completeness in Bluetooth, as well as to do analyses for distance estimation and localization.

A

Résumé détaillé en Français

Le nombre d'utilisateurs de smartphones est un chiffre en constante augmentation et devrait atteindre 7,516 milliards d'ici 2026 [1]. De plus, alors que de plus en plus d'appareils se connectent à Internet [2, 3], les prévisions suggèrent qu'à la fin de l'année 2022, il y aura environ 362 millions de hotspots Wi-Fi publics disponibles dans le monde entier [123]. Ces chiffres conduisent à une amplification de la dynamique de la topologie et à des problèmes de gestion de réseau de plus en plus complexes. Par conséquent, notre dépendance vis-à-vis des techniques de mesure efficaces pour caractériser précisément le réseau et comprendre la mobilité des utilisateurs augmente également.

A.1 CONTEXTE

L'air est le médium préféré et gagnant en raison de sa portabilité, de son faible coût et de l'augmentation constante des débits de données. Les réseaux sans fil sont partout, et comprendre leur comportement pour améliorer leurs performances est d'une importance capitale [5, 6, 7, 8, 9]. Néanmoins, mesurer le trafic sans fil (sans fil) est difficile en raison de la nature intrinsèquement volatile des liens sans fil [10]. Bien que les opérateurs cellulaires produisent beaucoup de données de localisation, elles ne sont pas accessibles au public. Par conséquent, la communauté de recherche continue de s'appuyer sur un

ensemble limité de traces, ce qui restreint l'univers des observations possibles. Il est donc nécessaire de réaliser des mesures sans fil pour construire des traces que les chercheurs peuvent utiliser pour évaluer et améliorer les approches de mise en réseau.

A.2 MOTIVATION ET ÉNONCÉ DU PROBLÈME

La collect active du trafic est fastidieuse car elle nécessite le déploiement de sondes sur différents nœuds, qui sont susceptibles d'être sous le contrôle de diverses entités administratives. Par exemple, pour collecter activement du trafic à partir de smartphones, on peut soit déployer des sondes à tous les points d'accès auxquels l'utilisateur est associé, soit créer une application de mesure et demander aux utilisateurs de l'installer sur leurs appareils. Les utilisateurs qui se portent volontaires peuvent constituer un échantillon insuffisant de la population, ce qui entraîne des résultats inexactes.

Une alternative efficace est de réaliser des mesures passives en déployant plusieurs *sniffeurs* (appareils qui collectent des paquets sans fil en mode monitor) dans la zone cible [13, 14, 15] [10]. C'est une stratégie de mesure peu coûteuse et « scalable » qui ne nécessite pas de déranger les utilisateurs avec des services intrusifs. Le concept du projet ANR-MITIK, auquel nous participons, consiste à déduire des traces de contacts grâce à des méthodes non intrusives telles que le sniffing passif [16]. Néanmoins, en raison des contraintes de transmission sans fil telles que les effets multi-trajets, les effets de fading ou les collisions, il n'y a aucune garantie qu'un seul sniffeur peut capturer tous les paquets, ce qui conduit à des traces incomplètes. Dans la Figure A.1, nous illustrons un scénario typique où quatre sniffeurs ($s_1, \ldots, s_4$) n'ont pas la même "vue" du trafic sans fil en raison de captures manquées.

10. Il est cependant essentiel de savoir quelles données on peut récupérer en fonction de l'emplacement de la campagne de mesure tout en préservant la vie privée des utilisateurs.

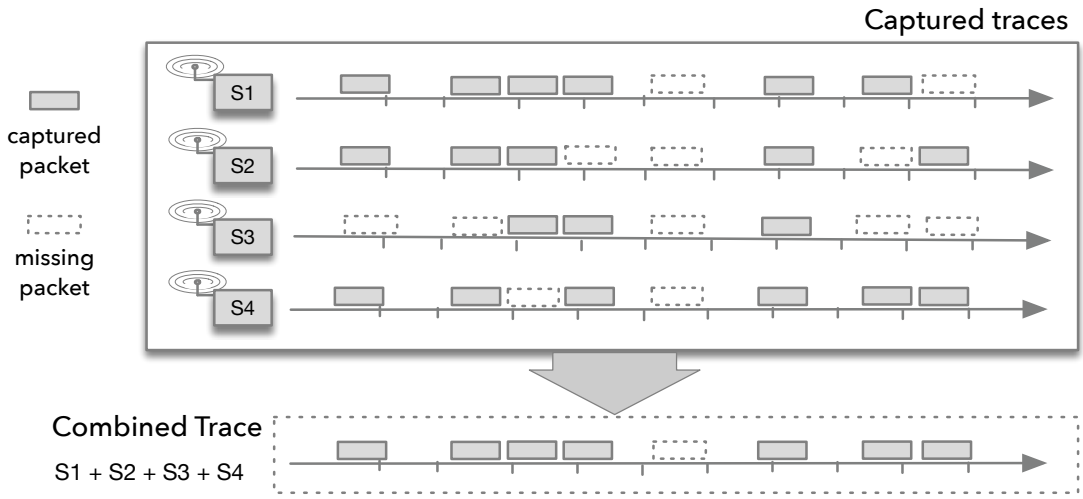


Figure A.1. – Complétude de trace. En raison de la nature du support sans fil, les sniffeurs peuvent manquer plusieurs paquets. Nous devons combiner les traces individuelles pour obtenir une trace aussi complète que possible.

Cela entraîne des divergences dans les mesures, et les analyses ultérieures reposant sur de telles traces incomplètes sont susceptibles d'être faussées.

La solution pour contourner le problème repose sur l'utilisation de *super-sniffeurs*. Elle consiste à introduire de la redondance dans le système en reliant deux ou plusieurs sniffeurs pour augmenter la probabilité qu'au moins l'un d'entre eux capture un paquet. Dans la Figure A.2, nous montrons un super-sniffeur triple, où chaque sniffeur individuel est composé d'une unité de calcul Raspberry Pi et d'une antenne Alfa. La principale question à laquelle nous répondons dans cet article est *comment le niveau de redondance contribue-t-il à améliorer la qualité de la mesure*. Pour répondre à cette question, nous proposons une définition de la *complétude relative* d'une trace et l'évaluons à travers des expériences réelles. Bien que nous nous concentrions sur les traces Wi-Fi, notre méthodologie est générale et peut s'appliquer à d'autres technologies.

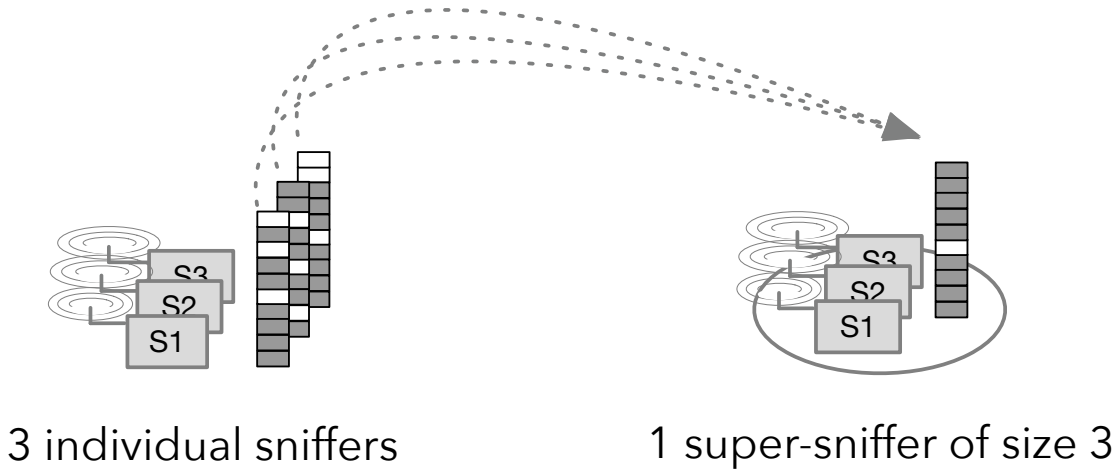


Figure A.2. – Trois sniffeurs formant un super-sniffeur.

A.3 MÉTHODOLOGIE

Bien qu'il existe des ensembles de données disponibles sur des plateformes telles que Crawdad [17], on ne sait pas quelles pré- ou post-traitements ont été effectués avant que l'ensemble de données ne soit rendu public, ce qui peut conduire à des résultats biaisés de manière involontaire.Par ailleurs, il existe aussi plusieurs simulateurs de réseau [19]. Le problème avec les simulateurs est qu'ils peuvent ne pas capturer avec précision la complexité et la variabilité des environnements réels, et leurs résultats ne sont pas toujours représentatifs des performances réelles. De plus, les simulateurs sont limités par l'exactitude des modèles et des hypothèses sous-jacents utilisés dans la simulation, qui peuvent ne pas toujours être vrais dans la pratique. Les expériences réelles sont préférées par les chercheurs pour les raisons suivantes :

— Validation des modèles théoriques : Les expériences dans le monde réel aident à valider les modèles théoriques développés pour les réseaux sans fil. Les modèles théoriques reposent souvent sur des hypothèses qui peuvent ne pas être vraies dans le monde réel. En menant des

expériences dans le monde réel, les chercheurs peuvent vérifier si leurs hypothèses sont correctes et ajuster leurs modèles en conséquence.
— L'évaluation des performances : L'expérimentation dans le monde réel permet aux chercheurs d'évaluer les performances des réseaux sans fil dans des conditions réelles. Cela permet d'identifier les forces et les faiblesses des différentes technologies et protocoles de réseau sans fil et de déterminer ceux qui conviennent le mieux à différentes applications et scénarios.
— Identification des problèmes: Les expériences en monde réel peuvent aider à identifier des problèmes qui ne seraient pas évidents dans des études de simulation ou de modélisation. Par exemple, l'interférence d'autres dispositifs sans fil ou des facteurs environnementaux tels que des bâtiments, des arbres ou un terrain peuvent affecter les performances des réseaux sans fil de manière qui ne peut pas être modélisée avec précision.

A.4 CONTRIBUTIONS

Cette thèse introduit la nécessité d'introduire une redondance dans le nombre de sniffeurs pour les mesures passives sans fil basées sur des expériences réelles. Nous proposons comporte trois contributions.

Contribution # :1 Complétude de trace

La complétude de la trace est un aspect important des mesures sans fil. Il est nécessaire d'avoir une référence pour l'évaluation de la complétude. La contribution pour la complétude de la trace est la suivante :

— **Métrique pour la complétude.** Nous proposons une définition formelle de la complétude qui intègre la notion de redondance. Nous

introduisons le terme *super-sniffeur* et formulons la complétude du super-sniffeur.

— **Évaluation expérimentale.** Nous suivons une approche expérimentale pour évaluer la complétude atteinte par les sniffeurs individuels, prouvant ainsi qu'un seul sniffeur n'est pas suffisant et qu'il est nécessaire d'avoir de la redondance. Nous étudions le comportement de la variation temporelle de la complétude au fil du temps.

— **Charge de trafic.** Nous étudions l'effet de la charge de trafic variable sur la complétude des traces. La quantité de trafic varie à différents moments de la journée car le nombre d'utilisateurs / d'appareils varie.

— **Rôle des sniffeurs.** Nous analysons les performances des sniffeurs individuels pour mettre en évidence le fait que tous les sniffeurs ne se comportent pas de la même manière et nous aider à identifier les sniffeurs qui ont des performances constamment médiocres malgré les conditions dans le milieu sans fil qui changent au fil du temps.

— **« Pairwise completeness ».** Les sniffeurs se complètent mutuellement pour améliorer la qualité des mesures passives lorsqu'on les considère par paire. Nous appelons cela la complétude par paire et l'évaluons pour onze sniffeurs après nettoyage du jeu de données.

Contribution # 2: PyPal, un outil de synchronisation et de fusion de traces Wi-Fi

Nous introduisons le concept de complétude de la trace ainsi que sa représentation mathématique. Cependant, il existe un défi pour l'évaluation de la complétude :

Comme les sniffeurs individuels ont une horloge locale et ne sont pas connectés à Internet au moment de la capture, il est nécessaire que les traces soient synchronisées dans le temps avant que l'analyse de fusion et d'évaluation de la redondance puisse être effectuée. Par conséquent, la deuxième contribution de cette thèse est un outil Python, appelé PyPal, pour la synchronisation dans

le temps et la fusion des traces [21]. En plus de cela, l'outil peut concaténer les traces synchronisées ainsi que produire des traces par adresse MAC.

Contribution # 3: Exploiter la redondance dans l'estimation de distance

Les valeurs d'indicateur de force du signal reçu (RSSI) sont utiles pour détecter la présence d'un nœud dans une zone cible et évaluer sa position par rapport au sniffeur [22, 23, 24, 25]. De plus, elles sont disponibles dans la plupart des systèmes d'exploitation de manière simple. Cependant, l'inconvénient de l'utilisation du RSSI concerne sa précision. Bien qu'il existe des moyens de contourner ce problème, tels que l'apprentissage automatique pour réduire l'erreur dans l'estimation de distance [26, 27], l'applicabilité du RSSI est toujours remise en question [28, 29, 30].

Dans le cadre du projet ANR MITIK [16], nous collectons des traces Wi-Fi à travers des mesures passives pour capturer la mobilité des nœuds sans fil dans une zone donnée. Dans le projet, nous nous concentrons sur les messages de demande de sonde, les seuls messages qu'un appareil d'utilisateur envoie lorsqu'il n'est associé à aucun point d'accès Wi-Fi. Le problème est que leur taux d'envoi peut être aussi bas que 55 paquets par heure ou aussi élevé que 2 000 paquets par heure, selon l'appareil [31]. Manquer ces demandes de sonde peut avoir de graves conséquences sur la qualité de la campagne de mesure, car cela peut entraîner la non-détection de nœuds ou des estimations de mobilité biaisées.

Nous collectons des traces sans fil à l'aide de plusieurs sniffeurs et d'un nœud source générant du trafic Wi-Fi. La source est le nœud que nous voulons caractériser. Nous co-localisons les sniffeurs (c'est-à-dire introduisons de la redondance) pour étudier la cohérence des valeurs RSSI. Nous réalisons les mesures à l'extérieur et surveillons le trafic généré par une source contrôlée pour laquelle nous connaissons la vérité terrain de sa distance par rapport aux sniffeurs. Dans nos expériences, nous plaçons le nœud source à plusieurs distances des sniffeurs et faisons les observations suivantes :

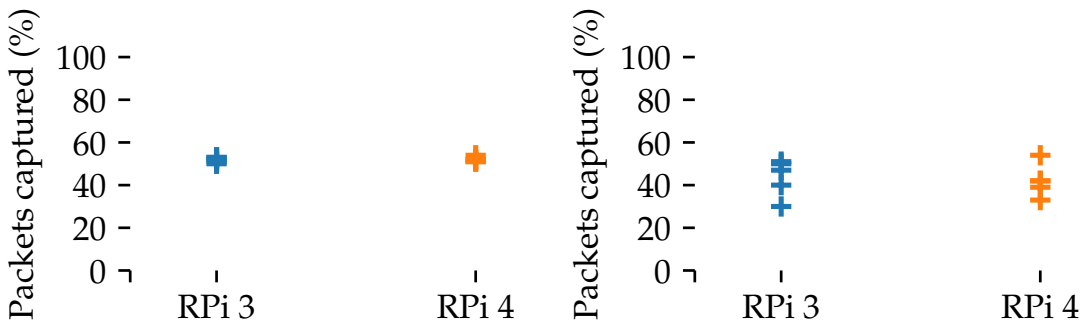
— Tous les sniffeurs ne ratent pas nécessairement le même paquet. Cela signifie que le paquet n'est sans aucun doute pas perdu en raison des collisions sur les sniffeurs.
— Le nombre de paquets consécutifs manqués par un sniffeur peut être énorme parfois. De plus, les sniffeurs peuvent manquer la capture pendant plusieurs secondes, ce qui pose un problème si l'on doit analyser la mobilité.
— Parfois, il y a une incohérence dans les valeurs RSSI du même paquet capturé par différents sniffeurs, ce qui entraîne des erreurs fréquentes dans l'estimation de la distance.

A.5 LES SNIFFEURS INDIVIDUELS POURRAIENT NE PAS SUFFIRE

Un seul sniffeur pourrait ne pas être en mesure d'avoir une vue complète du réseau sans fil en raison des caractéristiques inhérentes du milieu sans fil.

Pour évaluer l'impact de l'environnement sur la capture passive sans fil, nous avons co-localisé dix sniffeurs pour collecter le trafic sans fil. La fusion de toutes ces traces fournit une capture *complète* de l'environnement, qui est ensuite utilisée comme référence pour calculer la complétude individuelle de chaque trace de sniffeur.

Nous capturons des traces dans deux scénarios différents. Le but est de tester le comportement du système de capture pour différentes intensités de trafic sans fil. Le premier scénario correspond à une zone *résidentielle*, tandis que le deuxième scénario implique des *bureaux*. Comme nous le verrons plus tard dans l'article, le deuxième scénario est beaucoup plus dense et sollicite davantage les sniffeurs. Nous montrons que le trafic dans la zone de bureau est dense avec une moyenne d'environ 1,000 paquets par seconde. L'emplacement de mesure dans la zone résidentielle est isolé et a moins d'activité Wi-Fi dans



(a) Zone bureaux.

(b) Zone résidentielle.

Figure A.3. – Complétude moyenne des sniffeurs individuels. Chaque point représente un sniffeur individuel. Il s'agit de la complétude moyenne obtenue par chaque sniffeur individuellement sur 30 tests..

les environs. Par conséquent, le trafic dans la zone résidentielle est plus clairsemé – une moyenne d'environ 100 paquets par seconde.

Nous montrons dans la Figure A.3 la complétude moyenne de chaque trace de sniffeur pour les scénarios résidentiels et de bureaux. Pour faire une comparaison équitable entre les deux types d'appareils, nous traçons, pour chaque scénario, les valeurs obtenues avec les sniffeurs RPi3 et RPi4.

Nous observons qu'avec une faible charge de trafic dans l'environnement résidentiel, la complétude moyenne varie de 50% dans le meilleur des cas à 30% dans le pire des cas avec les sniffeurs RPi3, tandis que les mêmes valeurs pour les sniffeurs RPi4 sont de 54% et 33% respectivement. Dans l'environnement de bureau à forte charge de trafic, la complétude semble légèrement meilleure, variant entre 50% et 54%.

La redondance dans le nombre de sniffeurs est importante pour assurer une surveillance fiable et continue du trafic réseau au cas où un ou plusieurs sniffeurs échouent. Cela contribue à minimiser les interruptions et à garantir que toute l'activité du réseau est capturée et analysée, fournissant ainsi une

image plus complète du comportement du réseau et des problèmes potentiels. Nous montrons que l'environnement sans fil est difficile à capturer et que la force du signal joue un rôle important dans la qualité des traces capturées. Nous pouvons donc conclure la nécessité de combiner les sniffeurs pour capturer le trafic dans un environnement sans fil de manière adéquate en se basant sur les résultats présentés dans la section précédente.

Nous introduisons le concept de *super-sniffeur*, qui est une solution pour contourner le problème de la faible complétude de trace. Il consiste à introduire de la redondance dans le système en reliant deux ou plusieurs sniffeurs pour augmenter la probabilité qu'au moins l'un des sniffeurs capture un paquet. Nous proposons une définition de la *complétude* de la trace et l'évaluons grâce à des expériences réelles. Bien que nous nous concentrions sur les traces Wi-Fi, notre méthodologie est générale et peut s'appliquer à d'autres technologies.

A.6 PYPAL

Avant d'étudier expérimentalement la complétude, nous avons besoin de définir une méthodologie. Nous avons besoin d'un nouvel outil qui nous permet de analyser les traces pour les super-sniffeurs. Le principe derrière un super-sniffeur est sa capacité à fusionner les traces collectées par ses sniffeurs individuels. Les traces des sniffeurs ne sont pas synchronisées entre elles car les sniffeurs ont leurs propres horloges locales. Nous avons besoin de synchronisation pour pouvoir comparer les traces que les sniffeurs collectent en même temps. En revanche, le processus de fusion nécessite que les traces d'entrée soient synchronisées afin qu'un paquet qui apparaît dans plusieurs traces individuelles soit identifié de manière non ambiguë. Nous avons développé un outil Python appelé *PyPal* qui effectue une telle opération de synchronisation [21].

PyPal est une version mise à jour en Python de WiPal [100]. L'idée principale est de synchroniser les traces capturées par différents sniffeurs au même moment et de pouvoir les fusionner en supprimant les trames en double. Le processus est composé de cinq modes: (i) identification des trames de référence, (ii) extraction des trames uniques, (iii) intersection des trames de référence uniques, (iv) synchronisation et (v) fusion. Nous expliquons chacun de ces modules.

Identification des trames de référence. La première étape de la synchronisation des traces consiste à identifier les trames de référence, qui sont des trames qui n'apparaissent qu'une seule fois dans l'air pendant toute la période de mesure. Les trames qui apparaissent plusieurs fois sur le support sans fil, même si elles sont uniques dans chaque trace, ne doivent pas être considérées comme des trames de référence. Le processus d'extraction de trames uniques trouve des candidats pour devenir des trames de référence [100].

Extraction de trames uniques. Les trames de *beacons* et de *probe response* sont les représentants les plus proches des horloges en temps réel. De plus, la norme IEEE 802.11 dicte que ces trames ont une horodatage fixe dans l'en-tête, ajouté par le point d'accès, améliorant ainsi la précision de la synchronisation. Nous utilisons ces trames comme base pour le processus de synchronisation tel qu'illustré par la Figure A.4.

Intersection de trames uniques. La troisième étape consiste à identifier les trames uniques qui apparaissent dans les deux traces. Il est important de noter que les zones de couverture des sniffers capturant ces traces doivent se chevaucher ; sinon, les traces seront disjointes. Les trames obtenues par l'intersection des trames uniques servent de trames de référence pour la synchronisation.

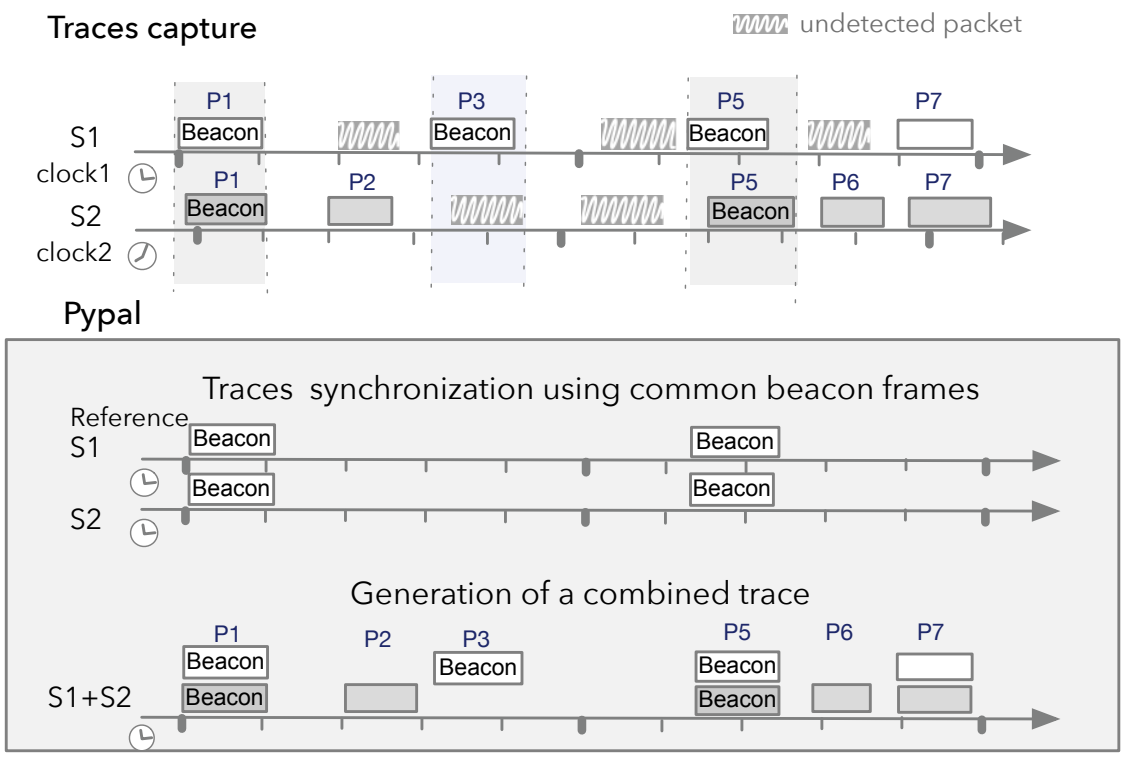


Figure A.4. – La méthodologie de PyPal pour synchroniser les traces. Elle consiste à prendre deux traces en entrée : l'une en tant que trace de référence et la deuxième en tant que trace à synchroniser.

Synchronisation. Synchroniser deux traces signifie mapper les horodatages de la première trace à des valeurs compatibles avec les horodatages de la deuxième trace. Nous calculons cette correspondance avec une fonction affine $t2 = at1 + b$. Elle estime a et b à l'aide de cadres de référence pendant que le processus s'exécute.

Le processus de synchronisation fonctionne sur des fenêtres de $w+1$ trames de référence. Pour chaque trame de référence R_i, le processus effectue une régression linéaire en utilisant les trames de référence $R_{\lfloor i-w/2 \rfloor}, \ldots, R_{\lceil i+w/2 \rceil}$. Au début et à la fin de la trace, nous utilisons $R_1, \ldots, R_w$ et $R_{N-w}, \ldots, R_N$

(où N est le nombre de trames de référence). Le résultat donne les valeurs de a et b pour toutes les trames entre R_i et R_{i+1}.

Fusion. Le rôle de la fusion est de copier les trames des traces synchronisées vers la trace de sortie. Bien sûr, il doit organiser correctement sa sortie en évitant les trames en double. Nous utilisons l'algorithme de fusion WiPal dans PyPal.

A.7 COMPLÉTUDE DE TRACE À TRAVERS LES SUPER-SNIFFEURS

Nous présentons les résultats de complétude pour les super-sniffeurs de toutes tailles ainsi que notre super-sniffeur de référence. Nous construisons notre super-sniffeur de la manière suivante :

- Simple sniffeur : $m = 1$. La référence dans ce cas est le simple sniffeur qui donne la meilleure complétude.
- Super-sniffeur de référence de taille $m = 2$; qui est la combinaison offrant la meilleure complétude parmi tous les super-sniffeurs de taille 2 contenant le sniffeur unique de référence (c'est-à-dire, $m = 1$).
- La référence du super-sniffeur de taille $m = 3$ est celui qui donne la meilleure complétude parmi tous les super-sniffeurs de taille 3 contenant la référence du super-sniffeur de taille $m = 2$.
- Nous procédons de la même manière pour les super-sniffeurs restants jusqu'à $m = 11$.

La Figure A.5 montre la complétude minimale, maximale et moyenne pour toutes les combinaisons de m sniffuers, ainsi que la complétude de référence de notre super-sniffuer de référence, représentée par des lignes bleues, jaunes, vertes et rouges, respectivement. Les nombres au-dessus des lignes représentent l'amélioration que notre super-sniffuer de référence apporte pour chaque taille. L'axe des abscisses représente le temps, tandis que l'axe des ordonnées

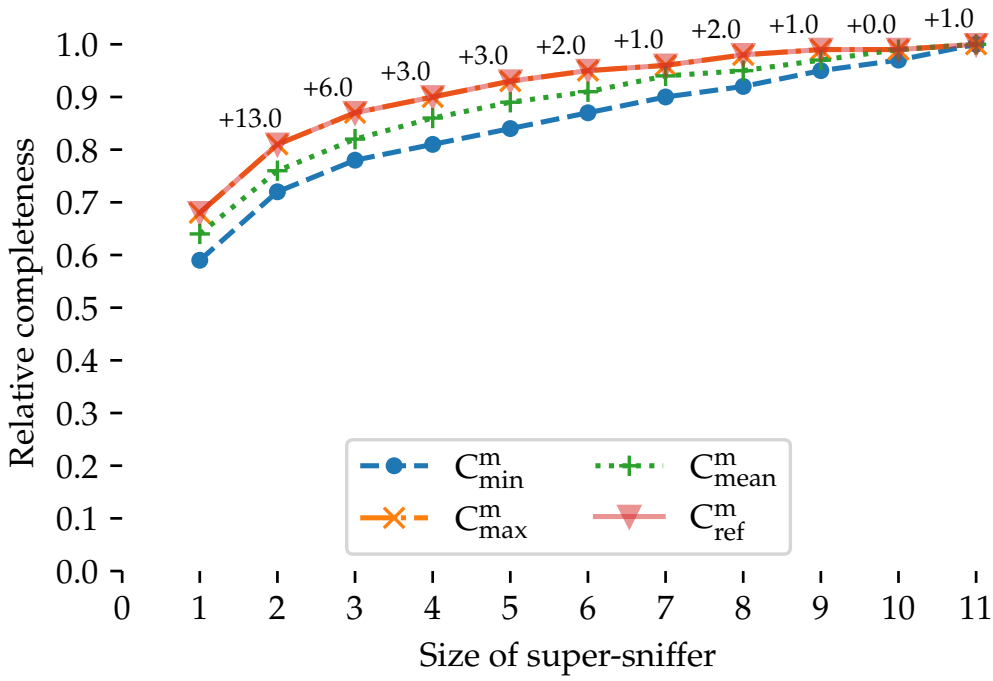


Figure A.5. – Complétude des super-sniffeurs. Complétude minimale, maximale, moyenne et de référence en fonction de la taille des super-sniffeurs pour des super-sniffeurs de toutes tailles.

donne la complétude pour une combinaison allant jusqu'à 11 sniffuers. Ce sont les résultats sur 24 heures.

Nous observons que la complétude s'améliore de 13% en ajoutant seulement un sniffuer pour créer un super-sniffuer de référence de taille $m = 2$. L'ajout d'un autre sniffuer apporte une amélioration supplémentaire de 6% de la valeur de la complétude. Nous continuons à observer une amélioration en augmentant la taille de notre super-sniffuer de référence. Le taux d'amélioration diminue avec chaque ajout de sniffuer au super-sniffuer, mais chaque sniffuer apporte de nouvelles informations au super-sniffuer. Nous observons que lorsque nous passons du super-sniffuer de taille $m = 9$ à $m = 10$, il n'y a pas d'amélioration de la complétude de référence/maximum dans notre cas ;

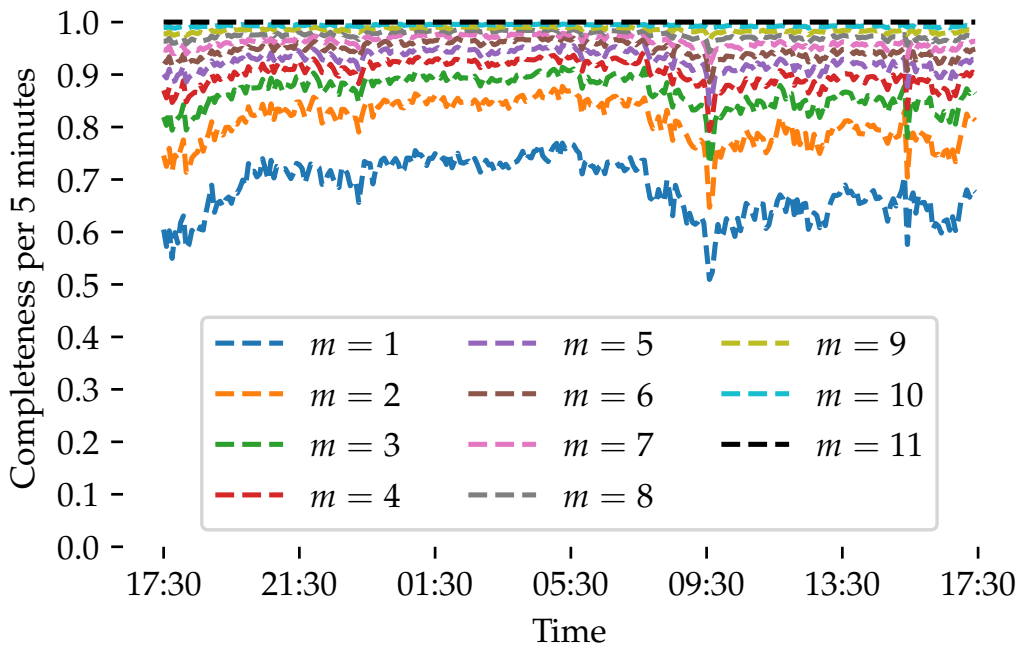


Figure A.6. – Complétude du super-sniffeur de référence. La complétude de nos super-sniffeurs de référence de toutes tailles au cours de 24 heures. Cela montre la variation du niveau de complétude au fil du temps.

cependant, il y a toujours une amélioration de 2% dans le cas de la complétude minimum et moyenne.

Nous remarquons également que la complétude maximale et de référence sont identiques pour les super-sniffeurs de toutes tailles. Cela signifie que le super-sniffeur d'une certaine taille qui atteint une complétude maximale fait partie du super-sniffeur de meilleure performance de la taille suivante. La complétude minimale et maximale ne sont pas non plus trop éloignées.

La figure A.6 montre les complétudes de notre super-sniffeur de référence en fonction du temps. Nous constatons que la complétude s'améliore considérablement en ajoutant simplement un sniffeur pour faire le super-sniffeur de taille $m = 2$. L'amélioration est d'environ 10% pendant toute la durée

de 24 heures et elle varie également en fonction de la charge du trafic. Le taux d'amélioration diminue à chaque ajout d'un sniffeur au super-sniffeur, mais chaque sniffeur apporte de nouvelles informations au super-sniffeur. Nous constatons une amélioration de la qualité de la capture de la trace avec la redondance indépendamment de la charge de trafic et de l'heure de la journée.

A.8 UTILISATION DE SUPER-SNIFFEURS POUR ESTIMER LA DISTANCE

Le Tableau A.1 montre l'erreur moyenne par paquet dans l'estimation de la distance en mètres, en utilisant les deux sniffeurs avec la moindre erreur, pour des combinaisons de tous les nombres de sniffeurs à 50m. Nous constatons dans la ligne 1 que l'erreur moyenne d'estimation de la distance pour une combinaison de deux sniffeurs est de 6,24m, soit 2,35m de moins que la moyenne des sniffeurs individuels. De même, si nous considérons uniquement la moyenne des sniffeurs individuels dans la deuxième ligne, l'erreur moyenne est énorme, environ 1050m. Cependant, l'erreur descend à 8,59m lorsque nous prenons la moyenne de la combinaison de deux sniffeurs avec la moindre erreur. Bien que la redondance aide à réduire l'erreur, la présence de ces valeurs aberrantes entraîne encore des erreurs massives, comme nous le voyons pour les combinaisons de quatre et cinq sniffeurs dans la deuxième ligne. Nous nettoyons l'ensemble de données pour supprimer les valeurs aberrantes [91]. Les valeurs RSSI inférieures à -56dBm n'apparaissent que dans 2,86% des cas ; nous les supprimons donc de l'ensemble de données à 50m. De même, les valeurs supérieures à -46dBm ne représentent que 1,66% des valeurs. Nous recalculons les valeurs moyennes dans le Tableau A.2 et nous constatons déjà une énorme amélioration. Nous voyons dans la ligne sept que l'erreur moyenne avec cinq sniffeurs passe de 6,24m à seulement

Table A.1. – Erreur moyenne brute par paquet pour une distance de 50 m.

	Nombre de sniffeurs				
Pacquet	**1**	**2**	**3**	**4**	**5**
1	8.59	6.24	0.86	0.86	2.76
2	1051.8	8.59	8.59	128.32	248.05
3	924.03	924.03	924.03	924.03	924.03
4	289.96	289.96	289.96	289.96	289.96
5	8.59	6.24	0.86	2.76	3.99
6	6.24	2.76	0.86	0.45	1.5
7	6.24	2.76	0.86	1.6	6.35
8	8.59	8.59	10.41	13.66	16
9	8.59	8.59	8.59	8.59	8.59
10	25.53	25.53	25.53	25.53	25.53
11	3.88	3.88	3.88	3.88	3.88
12	6.24	2.76	0.86	1.6	6.35

0,5m. Les lignes sans valeurs signifient que les sniffeurs présentaient des valeurs aberrantes.

Dans la Figure A.7, nous représentons les résultats des meilleurs et pires cas d'erreur moyenne dans l'estimation de la distance pour un, trois et tous les cinq sniffeurs, avec et sans les valeurs aberrantes. Nous constatons que l'erreur dans le pire des cas est énorme pour un seul sniffeur pour toutes les distances. Trois sniffeurs donnent une amélioration, mais l'erreur dans le pire des cas est encore significative. Nous notons également dans la Figure A.7b que la suppression des valeurs aberrantes de RSSI réduit considérablement l'erreur moyenne, y compris dans le pire des cas. Même un seul sniffeur donne une faible erreur pour les courtes distances, mais elle augmente considérablement pour les longues distances, en particulier dans le pire des cas.

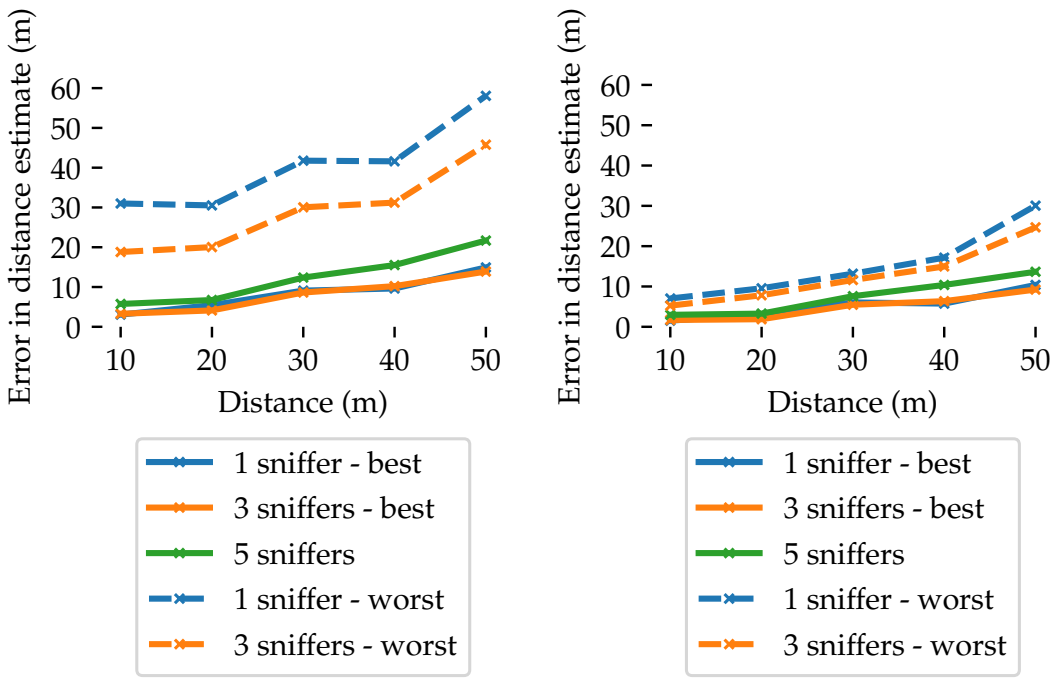
Table A.2. – Erreur de distance moyenne par paquet après suppression des valeurs aberrantes pour une distance de 50 m.

	Nombre de sniffeurs				
Pacquet	**1**	**2**	**3**	**4**	**5**
1	8.59	6.24	0.86	0.86	2.76
2	8.59	8.59	8.59	8.59	8.59
3	–	–	–	–	–
4	–	–	–	–	–
5	8.59	8.59	8.59	6.24	3.88
6	6.24	2.76	0.86	0.45	1.5
7	6.24	3.32	1.81	0.66	0.45
8	8.59	8.59	8.59	10.41	12.23
9	8.59	8.59	8.59	8.59	8.59
10	–	–	–	–	–
11	3.88	3.88	3.88	3.88	3.88
12	6.24	3.32	1.81	0.66	0.45

A.9 CONCLUSION

La surveillance passive est une méthode largement utilisée dans les réseaux sans fil, en particulier pour la reconstruction de trajectoires et la localisation. Cependant, elle n'est pas sans ses défis. En raison des caractéristiques inhérentes du support sans fil, un seul sniffeur est incapable de capturer un trafic suffisamment représentatif, ce qui peut entraîner des analyses incomplètes et inexactes. Il est donc crucial de mesurer la complétude de la trace, mais il existe un manque de représentation mathématique et d'analyse empirique de la complétude dans les expériences réelles. De plus, il n'existe pas d'outil facilement disponible et facile à utiliser pour synchroniser et fusionner les traces Wi-Fi (IEEE 802.11) afin d'étudier l'impact de la redondance sur la complétude de la trace. Cette thèse vise à résoudre ces problèmes.

Tout d'abord, nous avons développé un outil appelé PyPal, une version Python de WiPal, pour la synchronisation et la fusion des traces Wi-Fi. Cet outil néces-



(a) Avec les valeurs aberrantes.

(b) Sans les valeurs aberrantes.

Figure A.7. – Erreur d'estimation de distance pour 1, 3 et 5 sniffeurs avec et sans valeurs aberrantes.

site deux traces en entrée au format texte, l'une servant de référence et l'autre devant être synchronisée. Il utilise des champs d'en-tête spécifiques pour faciliter le processus de synchronisation (voir l'Annexe B pour une commande permettant d'extraire les champs requis à partir d'un fichier pcap). Une fois que les traces sont dans le format requis, PyPal permet de synchroniser les traces dans le temps. Il offre également la possibilité de fusionner les traces tout en supprimant les trames en double dans le processus. De plus, il offre la possibilité de simplement concaténer les deux traces synchronisées et de créer des traces par adresse MAC.

En second lieu, nous présentons des preuves expérimentales montrant qu'un seul sniffeur est insuffisant pour capturer des traces de support sans fil

suffisamment représentatives. Nous avons effectué une analyse de traces capturées simultanément par dix sniffeurs co-localisés de deux types différents dans des scénarios à faible et forte activité. Nos résultats indiquent que, même dans des situations moins actives, la complétude des deux types de sniffeurs ne dépasse pas 54%. Pour comparer davantage les deux types de sniffeurs, nous utilisons la similarité de Jaccard et montrons que les faibles niveaux de complétude ne sont pas imputables aux sniffeurs eux-mêmes, mais aux caractéristiques du support.

Troisièmement, nous introduisons de la redondance en augmentant le nombre de sniffeurs. Cependant, pour analyser l'efficacité de la redondance, nous avons besoin d'une métrique. Nous introduisons donc les concepts de complétude de trace et de super-sniffeur, et fournissons une représentation mathématique de la complétude relative et absolue. La complétude relative se réfère à la capacité de capturer toutes les trames dans le support, tandis que la complétude absolue se rapporte aux trames capturées à partir d'une source contrôlée avec un volume de trafic connu. Un super-sniffeur comprend une collection de sniffeurs redondants co-localisés, tels qu'une combinaison de trois sniffeurs co-localisés formant un super-sniffeur de taille trois. Nous avons mené une expérience de mesure de 24 heures dans un environnement de bureau en utilisant quatorze sniffeurs et appliqué notre définition de complétude de trace à la redondance. Nos résultats indiquent que l'ajout d'un seul sniffeur (pour former un super-sniffeur de taille deux) améliore la qualité de capture de 13% dans le meilleur des cas. De plus, le niveau de complétude augmente davantage avec une augmentation de la taille du super-sniffeur. Notre analyse des valeurs de RSSI des paquets capturés par chaque sniffeur met en évidence l'importance d'analyser les performances individuelles des sniffeurs avant d'évaluer la complétude en raison de la possibilité de dysfonctionnement des sniffeurs. Nous avons identifié trois des quatorze sniffeurs comme défectueux et les avons exclus de l'analyse.

Comme mentionné précédemment, nous introduisons le concept de complétude absolue pour mesurer le ratio de trafic capturé par rapport à la quantité réelle de trafic généré. Nous générons du trafic Wi-Fi en utilisant notre propre nœud source et le plaçons à différentes distances pendant l'expérience. Nos résultats indiquent que, même dans ce scénario, un seul sniffeur échoue à capturer un nombre substantiel de paquets, et le ratio de paquets manquants augmente avec la distance. Cependant, l'introduction de la redondance via le concept de super-sniffeur améliore considérablement la complétude des traces à toutes les distances.

Nous reconnaissons que la redondance engendre un coût financier. Cependant, en utilisant les ports USB multiples disponibles dans un nœud Raspberry Pi, nous pouvons mettre en place la redondance à moindre coût. Plus précisément, nous pouvons connecter plusieurs antennes à un seul nœud RPi, plutôt que d'utiliser plusieurs nœuds RPi, chacun avec une seule antenne. Nous appelons cela la redondance alternative et l'évaluons pour la complétude relative et absolue. Nos expériences démontrent qu'un nœud RPi peut prendre en charge jusqu'à trois antennes Wi-Fi externes. Nous utilisons donc trois antennes avec un seul RPi pour les expériences, créant ainsi un super-sniffeur de taille trois. Nous construisons également un autre super-sniffeur de taille trois en utilisant trois nœuds RPi, chacun avec une antenne. Nos résultats indiquent que le RPi avec trois antennes fonctionne aussi bien que le super-sniffeur formé de trois sniffeurs simples. Cette approche offre une solution rentable pour les chercheurs souhaitant mettre en place la redondance dans leurs systèmes.

Enfin, nous présentons une application de la redondance sous forme d'estimation de distance. Nous utilisons notre propre nœud source pour obtenir la vérité terrain de la distance réelle entre le nœud source et les sniffeurs. Malgré l'utilisation généralisée du RSSI dans la littérature pour l'estimation de distance et la localisation, des préoccupations ont été soulevées quant à la fiabilité du RSSI en raison des caractéristiques du milieu sans fil. Nos expériences indiquent qu'il existe une incohérence dans les valeurs

de RSSI, c'est-à-dire que les valeurs de RSSI du même paquet capturé par différents sniffeurs co-localisés peuvent varier considérablement à chaque sniffeur, ce qui entraîne une analyse erronée. De plus, un seul sniffeur peut manquer une rafale de paquets à un moment donné, ce qui peut avoir un impact significatif sur la reconstruction de la trajectoire. Cependant, nous démontrons que ces problèmes peuvent être résolus grâce à la redondance, et que l'erreur d'estimation de distance peut être réduite en utilisant les valeurs de RSSI avec le modèle Log-Distance Path Loss (LDPL) ainsi que la redondance.

A.10 PERSPECTIVE FUTURE

Nous avons présenté nos contributions dans ce manuscrit pour améliorer la qualité des traces Wi-Fi. Nous prévoyons d'étendre ce travail pour explorer le concept et les utilisations de la redondance sous différents angles. Nous listons ci-dessous quelques questions ouvertes qui pourront être étudiées dans des travaux futurs :

— **Canal et spectre.** Nous avons effectué toutes les expériences sur un seul canal, à savoir le canal 1 du spectre de 2,4 GHz. Nous prévoyons de réaliser des expériences sur différents canaux pour analyser la tendance du trafic ainsi que la complétude des traces. Nous prévoyons également d'étudier l'aspect de la complétude pour le spectre de 5 GHz.
L'impact potentiel du balayage de canaux sur la complétude des traces est une autre possibilité de travaux futurs. Cependant, il faudrait une sorte de synchronisation entre les sniffeurs co-localisés pour qu'ils puissent balayer les canaux simultanément et dans le même ordre.
— **Antenne.** Nous avons utilisé des antennes Alfa Network AWUS 051NH pour toutes nos expériences. Notre objectif est d'étendre l'expérimentation à différentes antennes avec des spécifications différentes qui peuvent être facilement configurées en mode monitor. Une

telle antenne qui est facilement disponible sur le marché est TP-Link WN722N [120].

— **Estimation de distance.** Nous prévoyons d'étudier la distribution de la taille inter-rafale et son impact sur l'analyse de mobilité à long terme. Nous avons également l'intention de déterminer la valeur de l'indice de perte de trajet à partir de la valeur RSSI qui apparaît le plus pour chaque distance et l'utiliser pour trouver la distribution d'erreur dans l'estimation de distance.

Nous avons effectué l'analyse de l'estimation de distance pour des expériences en extérieur. Nous avons l'intention de l'étendre à la localisation en intérieur avec la redondance. Nous pensons que cela aidera également à résoudre les problèmes d'incohérence RSSI, car il y a beaucoup de phénomènes qui se produisent plus intensément dans le milieu sans fil à l'intérieur, comme la réflexion, la réfraction, la diffraction, la diffusion, etc. La redondance aidera également à minimiser les erreurs d'estimation.

Nous avons placé le nœud source à différentes distances dans nos expériences. Nous obtenons un histogramme des paquets capturés pour toutes les tailles de super-sniffeurs pour chaque distance. Nous avons l'intention d'utiliser ces données comme ensemble d'entraînement et d'explorer la possibilité de la localisation avec des techniques de comparaison d'histogrammes.

Enfin, nous avons l'idée d'étendre notre travail pour prendre en compte la mobilité avec l'introduction de nœuds source mobiles. Nous pensons que c'est un défi vraiment intéressant à expérimenter avec la redondance.

— **Mobilité.** Les détecteurs restent immobiles dans toutes nos expériences. Nous avons l'intention d'étudier l'impact de la mobilité des détecteurs sur la complétude de la trace. Nous croyons que c'est un aspect intéressant de la détection à étudier, car il existe une étude sur l'impact de la mobilité sur la précision de la détection [121].

— **BLE - BLEPal.** De même que PyPal, nous disposons d'un outil pour la synchronisation et la fusion des traces Bluetooth, appelé BLEPal [122]. Il a été créé par un groupe d'étudiants dans le cadre d'un projet de réseau que nous avons supervisé. Un possible travail futur est l'utilisation de BLEPal pour étudier l'impact de la redondance sur la complétude des traces en Bluetooth, ainsi que pour effectuer des analyses pour l'estimation de distance et la localisation.

B

PyPal Manual

B.1 FIELDS REQUIRED IN THE TRACE

The tool takes two traces (in csv or txt format) as input and then performs the option you select. You would need to have the following fields in the traces [11]:

- frame.number: Frame_number
- frame.time_epoch: Frame_time_epoch
- wlan.fixed.timestamp: Fixed_timestamp
- wlan_radio.signal_dbm: RSSI_dBm
- wlan_radio.channel: Channel
- wlan.fc.type: Frame_type
- wlan.fc.type_subtype: Frame_subtype
- wlan.fc.retry: Retransmission
- wlan.fcs: Checksum
- wlan.sa: Source_MAC_address
- wlan.seq: Sequence_number
- wlan.frag: Fragment_number

11. It is, however, essential to clearly define which data one can sniff depending on the location of the measurement campaign to preserve the privacy of the users. It is also necessary to carry out hashing of MAC addresses to preserve privacy.

B.2 TSHARK COMMAND TO EXTRACT THE REQUIRED FIELDS FROM A PCAP TRACE

You can use the following tshark command to extract the above-mentioned fields from a pcap file.

tshark -r pcap_input_file -Y '!_ws.malformed and wlan_radio.channel==1'

-T fields -E header=y -E separator=/t -e frame.number -e frame.time_epoch

-e wlan.fixed.timestamp -e wlan_radio.signal_dbm -e wlan_radio.channel

-e wlan.fc.type -e wlan.fc.type_subtype -e wlan.fc.retry -e wlan.fcs -e wlan.sa

-e wlan.seq -e wlan.frag > csv_or_txt_output_file

B.3 HOW TO USE THE TOOL

python3 pypal.py -h will also show you the information on how to operate the tool [12].

B.4 LIBRARIES REQUIRED

You need to have the following libraries installed:

- numpy
- pandas
- scikit-learn

12. It is preferable to use Python3.

B.5 INPUT ARGUMENTS OF THE TOOL

The tool has to positional arguments and those are the two traces:

- trace1: trace to be synchronized
- trace2: reference trace

There are several optional arguments, but you have to tell the tool which one you want to use. You can use only one optional argument at a time. The arguments are given below:

- -U : extract unique frames
- -R : extract unique reference frames
- -SR : synchronize unique reference frames
- -S : synchronize traces
- -C : concatenate traces (and keep the duplicate frames)
- -M : merge the traces and remove the duplicate frames within a time difference of 106µs.

B.6 TIME SYNCHRONIZATION ERROR

The time synchronization error (the difference between two timestamps of different sniffers for the same frame) has to be less than half the minimum gap between two valid IEEE 802.11 frames. In the IEEE 802.11b protocol, the minimum gap can be calculated as the 192-microsecond preamble delay plus 10-microsecond SIFS (Short Inter-Frame Space) plus 10-microsecond minimum transmission time for a MAC frame, to be a total of 212 microseconds. So the precision is 212/2 = 106µs [124].

C

Hardware configuration

We encountered a few challenges while configuring the Raspberry Pi (RPi) nodes. We highlight them to make you aware so that you can adopt a suitable solution.

— The boot time of the Raspberry Pi 3 (RPi3) and Raspberry Pi 4 (RPi4) differs due to variations in their hardware. The RPi3 can take several minutes to boot on some occasions, while the RPi4 typically boots successfully within a few seconds. However, there may be minor variations in boot time among different RPi4 nodes, even when powered up simultaneously.
Removing the memory card from an RPi node, particularly for the purpose of altering any files on it, can lead to an extended boot time during the subsequent startup. This is an important consideration to bear in mind when conducting experiments to study completeness, as the capture must commence simultaneously on all sniffers.

— When executing *python* code or a utility such as *tcpdump* in the terminal, it is not necessary to enter the complete path for the tool. However, when configuring a command or script to run on boot using crontab, it is advisable to include the complete path for the tool, script, or Python itself. Similarly, when running a tool such as tcpdump within a script, it is recommended to include the path within the script as well.

— As previously mentioned, it is essential to utilize full paths. Additionally, it is imperative to verify the path, as it may change following an update, especially when utilizing the Raspberry Pi operating system (OS). We encountered an issue when a script was unable to locate tcpdump after an OS update. We discovered that the path of tcpdump had changed from */usr/sbin/tcpdump* to */usr/bin/tcpdump*. The update from Debian Buster to Debian Bullseye for the RPi OS equally created difficulties.

— The RPi utilizes a traditional interface naming convention, in which interfaces are labeled as eth0, eth1, wlan0, wlan1, and so on. The interface for an RPi's internal Wi-Fi card is consistently labeled as wlan0. However, an issue can arise when using an external Wi-Fi adapter, as it should logically be named wlan1. However, during the RPi's boot process, it labels the interfaces in the order in which it detects them. Due to this reason, the interface names wlan0 and wlan1 may occasionally be reversed. If a script is configured for traffic capture using *wlan1*, it may fail if the interface names are reversed, as the internal Wi-Fi card of an RPi cannot be set to monitor mode.
One potential solution is to allow the RPi to boot completely before plugging in the Wi-Fi adapter. Alternatively, one could enable the predictable naming option on the RPi. In this case, the interface name will initiate with *wlx* and end with the adapter's *MAC address* [125, 126]. However, predictable naming necessitates the consistent usage of the same Wi-Fi adapter, and some sort of identification will be required in the case of multiple co-located RPi sniffers.

— It is essential to verify that the Wi-Fi adapter is functioning correctly before conducting experiments. Not all Wi-Fi adapters are plug-and-play, and some may require additional or external drivers to operate. The Alfa AWUS 051NH adapter is plug-and-play and does not require any additional drivers to function properly. However, version 3 of the TP-Link WN722N Wi-Fi dongle is not plug-and-play. Installing separate

drivers is necessary, and even then, capturing traffic in monitor mode can cause difficulties.

— Like any other system that runs on an operating system, the RPi must be shut down correctly. In experiments where the RPi does not have a screen attached, it is crucial to include the shutdown command at the end of the script. Directly unplugging the RPi while the operating system is still running can damage the memory card, which could lead to it becoming read-only permanently.

— If you intend to use an external battery to power your RPi, it is essential to ensure that it has sufficient power to make the antenna work. For instance, a battery with a 12,000 mAh capacity and an output of 5 Volts and 1 Ampere is adequate to power the RPi node itself, but not enough to provide power to the USB ports to run the Wi-Fi adapter. However, a battery with a 20,000 mAh capacity and an output of 5 Volts and 3.1 Amperes can handle the RPi4 sniffer very well.

  Additionally, it is important to check if a single battery can simultaneously power multiple RPis or not. For example, a battery with a 26,800 mAh capacity and an output of 5 Volts, and a total of 5.5 Amperes across three USB ports (or 2.4 Amperes individual) can power up to three RPi4 sniffers simultaneously.

D

List of my publications

JOURNAL ARTICLES

M. I. Syed, A. Fladenmuller, and M. Dias de Amorim. "Unity is strength: Improving Wi-Fi passive measurements through sniffer redundancy". In: *Ad Hoc Networks (2023), p. 103287. issn: 1570-8705*. doi: https://doi.org/10.1016/j.adhoc.2023.103287

CONFERENCE PROCEEDINGS

M. I. Syed, A. Fladenmuller et M. Dias de Amorim, "Jusqu'où la redondance peut aider dans la capture passive de trafic Wi-Fi," in *CORES 2022 – 7ème Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*, Saint-Rémy-Lès-Chevreuse, France, May 2022.

M. I. Syed, A. Fladenmuller, and M. Dias de Amorim, "Assessing the completeness of passive Wi-Fi traffic capture," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, 2022.

M. I. Syed, A. Fladenmuller, and M. Dias de Amorim, "How much can sniffer redundancy improve Wi-Fi traffic?," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022.

M. I. Syed, A. Fladenmuller, and M. Dias de Amorim, "RSSI: Lost and alone, a case for redundancy," in *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2022.

TECHNICAL REPORTS

M. I. Syed, A. Flandenmuller, and M. Dias de Amorim, "PyPal: Wi-Fi Trace Synchronization and Merging Python Tool," LIP6 UMR 7606, UPMC Sorbonne Université, France, Technical Report, Mar. 2022.

M. F. Akli, N. N. E. A. Boukerras, N. Derradji, D. Laga, and **M. I. Syed**, "BLEPal," Aug. 2022.

N. Bencherif, M. Chabane, L. Mehidi, L. Paredes, and **M. I. Syed**, "Impact of TP-Link WN 722N and Sniffer placement on trace completeness," Aug. 2022.

Bibliography

[1] S. O'Dea, "Number of smartphone users from 2016 to 2021," 2023, https://statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[2] Statista Research Department, "Public wireless Internet usage in selected locations according to Internet users worldwide as of june 2015, by device," 2015, https://tinyurl.com/5y768zz6.

[3] Statista Research Department, "Wireless Internet access according to Internet users worldwide as of june 2015, by device," 2015, https://statista.com/statistics/463301/wireless-internet-access-by-device-worldwide/.

[4] Cisco, "Cisco annual Internet report (2018–2023) white paper," 2020, https://tinyurl.com/4pf6em46.

[5] P. A. K. Acharya, A. Sharma, E. M. Belding, K. C. Almeroth, and K. Papagiannaki, "Congestion-aware rate adaptation in wireless networks: A measurement-driven approach," in *2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2008.

[6] A. Galanopoulos, V. Valls, G. Iosifidis, and D. J. Leith, "Measurement-driven analysis of an edge-assisted object recognition system," in *IEEE ICC*, 2020.

[7] W. Zhou, Z. Wang, and W. Zhu, "Mining urban WiFi QoS factors: A data driven approach," in *IEEE BigMM*, 2017.

[8] P. De Vaere, T. Bühler, M. Kühlewind, and B. Trammell, "Three bits suffice: Explicit support for passive measurement of Internet latency in QUIC and TCP," in *Proceedings of the Internet Measurement Conference 2018*, 2018.

[9] J. Wang, Y. Zheng, Y. Ni, C. Xu, F. Qian, W. Li, W. Jiang, Y. Cheng, Z. Cheng, Y. Li, X. Xie, Y. Sun, and Z. Wang, "An active-passive measurement study of TCP performance over LTE on high-speed rails," in *ACM Mobicom*, New York, NY, USA, 2019.

[10] M. D. Corner, B. N. Levine, O. Ismail, and A. Upreti, "Advertising-based measurement: A platform of 7 billion mobile devices," in *ACM Mobicom*, Snowbird, UT, USA, oct 2010.

[11] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan, "WiFi, LTE, or Both? Measuring Multi-Homed Wireless Internet Performance," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. New York, NY, USA: Association for Computing Machinery, 2014.

[12] F. Guillemin, W. Robitza, S. Wunderer, and T. Hoßfeld, "Definitions of crowdsourced network and QoE measurements," *SIGMultimedia Rec.*, vol. 12, jul 2022.

[13] F. Garcia, R. Andrade, C. De Oliveira, and J. Souza, "EPMOSt: An energy-efficient passive monitoring system for wireless sensor networks," *Sensors (Basel, Switzerland)*, vol. 14, pp. 10 804–10 828, 06 2014.

[14] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Analyzing the MAC-level behavior of wireless networks in the wild," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New York, NY, USA: Association for Computing Machinery, 2006.

[15] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New York, NY, USA: Association for Computing Machinery, 2006.

[16] "ANR MITIK," 2020, https://project.inria.fr/mitik/.

[17] C. team, "Crawdad," https://ieee-dataport.org/collections/crawdad.

[18] S. M. King, F. Nawab, and K. Obraczka, "A survey of open source user activity traces with applications to user mobility characterization and modeling," *arXiv e-prints*, p. arXiv:2110.06382, Oct. 2021.

[19] G. F. Riley and T. R. Henderson, "The NS-3 network simulator." in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Günes, and J. Gross, Eds. Springer, 2010, pp. 15–34.

[20] A. Abuarqoub, F. Alfayez, M. Hammoudeh, and T. Alsbou'i, "Simulation issues in wireless sensor networks: A survey," in *Proceedings of the 2012 International Conference on Sensor Technologies and Applications (SENSORCOM)*, 01 2012.

[21] M. I. Syed, A. Flandenmuller, and M. Dias de Amorim, "PyPal: Wi-Fi Trace Synchronization and Merging Python Tool," LIP6 UMR 7606, UPMC Sorbonne Université, France, Technical Report, Mar. 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03618014

[22] C. Bertier, "Quantification in Device-to-Device Networks : from Link Estimation to Graph Utility," Theses, Sorbonne Université, Jan. 2020. [Online]. Available: https://tel.archives-ouvertes.fr/tel-03347660

[23] T. I. Chowdhury, M. M. Rahman, S.-A. Parvez, A. K. M. M. Alam, A. Basher, A. Alam, and S. Rizwan, "A multi-step approach for RSSI-based distance estimation using smartphones," in *2015 International Conference on Networking Systems and Security (NSysS)*, 2015.

[24] S. Sadowski and P. Spachos, "RSSI-based indoor localization with the Internet of Things," *IEEE Access*, pp. 30 149–30 161, 2018.

[25] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, "Wireless RSSI fingerprinting localization," *Signal Processing*, pp. 235–244, 2017.

[26] G. G. Anagnostopoulos and A. Kalousis, "A reproducible comparison of RSSI fingerprinting localization methods using LoRaWAN," in *2019 16th Workshop on Positioning, Navigation and Communications (WPNC)*, 2019.

[27] M. Anjum, M. A. Khan, S. A. Hassan, A. Mahmood, H. K. Qureshi, and M. Gidlund, "RSSI fingerprinting-based localization using machine learning in LoRa networks," *IEEE Internet of Things Magazine*, no. 4, pp. 53–59, 2020.

[28] K. Heurtefeux and F. Valois, "Is RSSI a good choice for localization in wireless sensor network?" in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, 2012.

[29] Q. Dong and W. Dargie, "Evaluation of the reliability of RSSI for indoor localization," in *2012 International Conference on Wireless Communications in Underground and Confined Areas*, 2012.

[30] A. Parameswaran, M. I. Husain, and S. Upadhyaya, "Is RSSI a reliable parameter in sensor localization algorithms: an experimental study," *Field Failure Data Analysis Workshop (F2DA'09)*, Jan 2009.

[31] J. Freudiger, "How talkative is your mobile device? An experimental study of Wi-Fi probe requests," in *Proceedings of the 8th ACM Conference*

on Security and Privacy in Wireless and Mobile Networks. New York, NY, USA: Association for Computing Machinery, 2015.

[32] Y. Li, J. Barthelemy, S. Sun, P. Perez, and B. Moran, "A case study of WiFi sniffing performance evaluation," *IEEE Access*, vol. 8, pp. 129 224–129 235, 2020.

[33] C. Caol, W. Gong, W. Dong, J. Yu, C. Chen, and J. Liu, "Network measurement in multihop wireless networks with lossy and correlated links," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 1340–1348.

[34] A. Kumar Mishra, A. Carneiro Viana, and N. Achir, "Do WiFi probe-requests reveal your trajectory?" in *WCNC 2023 - IEEE Wireless Communications and Networking Conference*, Glasgow, United Kingdom, Mar. 2023.

[35] Y. Cai, "Data monitoring and analysis in wireless networks," Ph.D. dissertation, Graduate Center, City University of New York, Sep 2015. [Online]. Available: https://academicworks.cuny.edu/gc_etds/878/

[36] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic forecasts achieve high throughput and low delay over cellular networks," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*. USA: USENIX Association, 2013.

[37] S. Sundaresan, N. Feamster, and R. Teixeira, "Measuring the performance of user traffic in home wireless networks," in *Passive and Active Measurement*, J. Mirkovic and Y. Liu, Eds., 2015.

[38] The Tcpdump Group, "Tcpdump and libpcap," https://tcpdump.org.

[39] G. Combs and Wireshark Foundation, "Tshark," https://tshark.dev/.

[40] G. Combs and Wireshark Foundation, "Wireshark," https://www.wireshark.org/.

[41] P. Biondi, "Scapy," https://scapy.net/.

[42] M. Freire and M. Sousa, *Encyclopedia of Internet Technologies and Applications*. Idea-Group Inc., 2008.

[43] ESP32, "Esp32," https://www.espressif.com/en/products/socs/esp32.

[44] Arduino, "Arduino," https://www.arduino.cc/.

[45] BeagleBone, "Beaglebone," https://beagleboard.org/.

[46] Udoo, "Udoo," https://www.udoo.org/.

[47] RPi, "Raspberry Pi," https://www.raspberrypi.com/.

[48] E. Litvinov, "Building sniffer on the basis of ESP32. Listening on Wi-Fi, aiming at Bluetooth!" https://hackmag.com/security/esp32-sniffer/.

[49] A. Haka, V. Aleksieva, H. Valchanov, and D. Dinev, "Analysis of ZigBee network using simulations and experiments," in *2020 International Conference Automatics and Informatics (ICAI)*, 2020.

[50] G. Solmaz and F.-J. Wu, "Together or alone: Detecting group mobility with wireless fingerprints," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7.

[51] M. Uras, R. Cossu, E. Ferrara, A. Liotta, and L. Atzori, "PmA: A real-world system for people mobility monitoring and analysis based on Wi-Fi probes," *Journal of Cleaner Production*, vol. 270, 2020.

[52] A. N. Ansari, M. Sedky, N. Sharma, and A. Tyagi, "An internet of things approach for motion detection using Raspberry Pi," in *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, 2015, pp. 131–134.

[53] M. Ibrahim, A. Elgamri, S. Babiker, and A. Mohamed, "Internet of things based smart environmental monitoring using the Raspberry-Pi computer," in *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, 2015, pp. 159–164.

[54] K. O. Flores, I. M. Butaslac, J. E. M. Gonzales, S. M. G. Dumlao, and R. S. Reyes, "Precision agriculture monitoring system using wireless sensor network and Raspberry Pi local server," in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 3018–3021.

[55] S. G. Nikhade, "Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications," in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, 2015, pp. 376–381.

[56] C. N. Cabaccan, F. R. G. Cruz, and I. C. Agulto, "Wireless sensor network for agricultural environment using Raspberry Pi based sensor nodes," in *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2017, pp. 1–5.

[57] F. D. de Mello Silva, A. K. Mishra, A. C. Viana, N. Achir, A. Fladenmuller, and L. H. M. K. Costa, "Performance analysis of a privacy-preserving frame sniffer on a Raspberry Pi," in *2022 6th Cyber Security in Networking Conference (CSNet)*, 2022, pp. 1–7.

[58] O. Westerlund and R. Asif, "Drone hacking with Raspberry-Pi 3 and WiFi Pineapple: Security and privacy threats for the Internet-of-Things," in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, 2019, pp. 1–10.

[59] N. Patil, S. Ambatkar, and S. Kakde, "IoT based smart surveillance security system using Raspberry Pi," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, 2017, pp. 0344–0348.

[60] S. Tripathi and R. Kumar, "Raspberry Pi as an intrusion detection system, a honeypot and a packet analyzer," in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 2018, pp. 80–85.

[61] Wardi, A. Achmad, Z. B. Hasanuddin, D. Asrun, and M. S. Lutfi, "Portable IP-based communication system using Raspberry Pi as exchange," in *2017 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2017, pp. 198–204.

[62] S. Vappangi, N. K. Penjarla, S. E. Mathe, and H. K. Kondaveeti, "Applications of Raspberry Pi in Bio-Technology: A review," in *2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP)*, 2022, pp. 1–6.

[63] J. W. Jolles, "Broad-scale applications of the Raspberry Pi: A review and guide for biologists," *Methods in Ecology and Evolution*, vol. 12, no. 9, pp. 1562–1579, Jun. 2021. [Online]. Available: https://doi.org/10.1111/2041-210x.13652

[64] A. Martínez, C. Nieves, and A. Rúa, "Implementing Raspberry Pi 3 and Python in the Physics laboratory," *The Physics Teacher*, vol. 59, no. 2, pp. 134–135, Feb. 2021. [Online]. Available: https://doi.org/10.1119/10.0003472

[65] J. J. K. Chng and M. Y. Patuwo, "Building a Raspberry Pi Spectrophotometer for undergraduate Chemistry classes," *Journal of Chemical Education*, vol. 98, no. 2, pp. 682–688, Dec. 2020. [Online]. Available: https://doi.org/10.1021/acs.jchemed.0c00987

[66] M. S. Cubberley and W. A. Hess, "An inexpensive programmable dual-syringe pump for the Chemistry laboratory," *Journal of Chemical Education*, vol. 94, no. 1, pp. 72–74, Nov. 2016. [Online]. Available: https://doi.org/10.1021/acs.jchemed.6b00598

[67] J. Marot and S. Bourennane, "Raspberry Pi for image processing education," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 2364–2366.

[68] Y. Turk, O. Demir, and S. Gören, "Real time wireless packet monitoring with Raspberry Pi sniffer," in *Information Sciences and Systems 2014*, T. Czachórski, E. Gelenbe, and R. Lent, Eds. Cham: Springer International Publishing, 2014.

[69] S. El-Tawab, R. Oram, M. Garcia, C. Johns, and B. B. Park, "Data analysis of transit systems using low-cost IoT technology," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 497–502.

[70] L. Mikkelsen, R. Buchakchiev, T. Madsen, and H. P. Schwefel, "Public transport occupancy estimation using WLAN probing," in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2016, pp. 302–308.

[71] T. Oransirikul, R. Nishide, I. Piumarta, and H. Takada, "Feasibility of analyzing Wi-Fi activity to estimate transit passenger population," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016, pp. 362–369.

[72] M. I. Jais, T. Sabapathy, M. Jusoh, P. Ehkan, L. Murukesan, I. Ismail, and R. B. Ahmad, "Received signal strength indication (RSSI) code assessment for wireless sensors network (WSN) deployed Raspberry-Pi," in *2016 International Conference on Robotics, Automation and Sciences (ICORAS)*, 2016, pp. 1–4.

[73] K. Friess, "Multichannel-sniffing-system for real-world analysing of Wi-Fi-packets," in *Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 358–364.

[74] M. D. Miñambres, V. Rojo, and D. R. Llanos, "Distance estimation to BLE beacons using Raspberry Pi 3 boards for indoor positioning and social tracking applications," in *SARTECO 20-21*, 2021.

[75] X. Xu, C. Tong, and J. Wan, "Improve the completeness of passive monitoring trace in wireless sensor network," in *2010 Asia-Pacific Services Computing Conference (APSCC 2010)*. Los Alamitos, CA, USA: IEEE Computer Society, dec 2010.

[76] M. Sammarco, M. E. M. Campista, and M. D. de Amorim, "Trace selection for improved WLAN monitoring," in *Proceedings of the 5th ACM Workshop on HotPlanet*. New York, NY, USA: Association for Computing Machinery, 2013.

[77] A. Schulman, D. Levin, and N. Spring, "On the fidelity of 802.11 packet traces," in *9th International Conference on Passive and Active Network Measurement*. Berlin, Heidelberg: Springer-Verlag, 2008.

[78] A. Mahanti, M. Arlitt, and C. Williamson, "Assessing the completeness of wireless-side tracing mechanisms," in *IEEE WoWMoM*, 2007.

[79] X. Xu, J. Wan, W. Zhang, C. Tong, and C. Wu, "PMSW: A passive monitoring system in wireless sensor networks," *International Journal of Network Management*, vol. 21, no. 4, pp. 300–325, 2011.

[80] B.-r. Chen, G. Peterson, G. Mainland, and M. Welsh, "LiveNet: Using passive monitoring to reconstruct sensor network dynamics," in *Distributed Computing in Sensor Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[81] P. Serrano, M. Zink, and J. Kurose, "Assessing the fidelity of COTS 802.11 sniffers," in *IEEE INFOCOM 2009*, 2009.

[82] T. Claveirole and M. Dias de Amorim, "WiPal: Efficient offline merging of IEEE 802.11 traces," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, no. 4, p. 39–46, mar 2010.

[83] T. Claveirole, "WiPal webpage," http://wipal.lip6.fr/download.html.

[84] G. Rego, N. Bazhenov, and D. Korzun, "Trajectory construction for autonomous robot movement based on sensed physical parameters and video data," in *2021 30th Conference of Open Innovations Association FRUCT*, 2021, pp. 200–206.

[85] M. Wenger, J. CARRERA, and Z. ZHAO, "Indoor positioning using Raspberry Pi with UWB," Bachelor's Thesis, University of Bern, 2019.

[86] C. Gentner, D. Günther, and P. H. Kindt, "Identifying the BLE advertising channel for reliable distance estimation on smartphones," *IEEE Access*, vol. 10, pp. 9563–9575, 2022.

[87] A. Mussina and S. Aubakirov, "RSSI based Bluetooth Low Energy indoor positioning," in *2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*, 2018.

[88] V. R, V. Mittal, and H. Tammana, "Indoor localization in BLE using mean and median filtered RSSI values," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2021.

[89] E. Salomon, L. H. Botler, K. Diwold, C. A. Boano, and K. Römer, "Poster: Comparison of Channel State Information driven and RSSI-based WiFi distance estimation," in *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 2021.

[90] G. Forbes, S. Massie, and S. Craw, "WiFi-based human activity recognition using Raspberry Pi," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020.

[91] N. Chuku and A. Nasipuri, "RSSI-based localization schemes for wireless sensor networks using outlier detection," *Journal of Sensor and Actuator Networks*, vol. 10, 2021.

[92] A. A. Jose, P. H. Rishikesh, and S. Shaju, "Mitigation of RSSI variations using frequency analysis and Kalman filtering," in *Proceedings of the International Conference on Cognitive and Intelligent Computing*, A. Kumar, G. Ghinea, S. Merugu, and T. Hashimoto, Eds. Singapore: Springer Nature Singapore, 2023.

[93] L. Alsmadi, X. Kong, K. Sandrasegaran, and G. Fang, "An improved indoor positioning accuracy using filtered RSSI and beacon weight," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 18 205–18 213, 2021.

[94] RPi, "Raspberry Pi 3 model B," https://tinyurl.com/2p88aa94.

[95] RPi, "Raspberry Pi 4 model B," https://tinyurl.com/2p89uund.

[96] AlfaNetwork, "Alfa Network AWUS051NH Wi-Fi adapter," https://tinyurl.com/yk8vk3vz.

[97] Aircrack, "Aircrack-ng," https://aircrack-ng.org/.

[98] L. Oliveira, D. Schneider, J. De Souza, and W. Shen, "Mobile device detection through WiFi probe request analysis," *IEEE Access*, vol. 7, pp. 98 579–98 588, 2019.

[99] N. Bencherif, M. Chabane, L. Mehidi, L. Paredes, and M. I. Syed, "Impact of TP-Link WN 722N and Sniffer placement on trace completeness," May 2022. [Online]. Available: https://hal.science/hal-03752126

[100] T. Claveirole, "Activités Wi-Fi en environnement ouvert : outils, mesures et analyses," Ph.D. dissertation, Université Pierre-et-Marie-Curie, 2010. [Online]. Available: http://www.theses.fr/2010PA066020

[101] IEEE, "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 5: Enhancements for higher throughput," *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, pp. 1–565, 2009.

[102] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O'Reilly Media, Inc., 2005.

[103] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A study of MAC address randomization in mobile devices and when it fails," 2017.

[104] M. I. Syed, A. Fladenmuller, and M. Dias de Amorim, "Assessing the completeness of passive Wi-Fi traffic capture," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, 2022.

[105] Juniper, "RSSI values for good/bad signal strength," https://www.mist.com/documentation/rssi-values-good-bad-signal-strength/.

[106] R. R. Hake, "Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory Physics courses," *American Journal of Physics*, vol. 66, no. 1, pp. 64–74, 1998.

[107] A. Thaljaoui, T. Val, N. Nasri, and D. Brulin, "BLE localization using RSSI measurements and iRingLA," in *IEEE ICIT*, 2015.

[108] T. S. Rappaport, *Wireless communications - principles and practice.* Prentice Hall, 1996.

[109] A. Mansfield, E. L. Inness, and W. E. Mcilroy, "Chapter 13 - stroke," in *Balance, Gait, and Falls*, ser. Handbook of Clinical Neurology, B. L. Day and S. R. Lord, Eds. Elsevier, 2018, vol. 159, pp. 205–228.

[110] O. Mohamed and H. Appling, "5 - clinical assessment of gait," in *Orthotics and Prosthetics in Rehabilitation (Fourth Edition)*. St. Louis (MO): Elsevier, 2020, pp. 102–143.

[111] M. W. Traunmueller, N. Johnson, A. Malik, and C. E. Kontokosta, "Digital footprints: Using wifi probe and locational data to analyze human mobility trajectories in cities," *Computers, Environment and Urban Systems*, vol. 72, pp. 4–12, 2018.

[112] T. Trasberg, B. Soundararaj, and J. Cheshire, "Using wi-fi probe requests from mobile phones to quantify the impact of pedestrian flows on retail turnover," *Computers, Environment and Urban Systems*, vol. 87, p. 101601, 2021.

[113] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960.

[114] G. F. Welch, *Kalman Filter*. Cham: Springer International Publishing, 2020, pp. 1–3.

[115] Z. Huang, X. Zhu, Y. Lin, L. Xu, and Y. Mao, "A novel WIFI-oriented RSSI signal processing method for tracking low-speed pedestrians," in *2019 5th International Conference on Transportation Information and Safety (ICTIS)*, 2019, pp. 1018–1023.

[116] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, "Indoor positioning algorithm based on the improved RSSI distance model," *Sensors*, vol. 18, no. 9, p. 2820, Aug. 2018.

[117] C. Zhou, J. Yuan, H. Liu, and J. Qiu, "Bluetooth indoor positioning based on RSSI and Kalman filter," *Wireless Personal Communications*, vol. 96, no. 3, pp. 4115–4130, Jul. 2017.

[118] M. Jeng, "Error in statistical tests of error in statistical tests," *BMC medical research methodology*, vol. 6, p. 45, 02 2006.

[119] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*. Boston, MA: Springer US, 2009, pp. 532–538. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_565

[120] TPLink, "TP-Link WN722N Wi-Fi adapter," https://www.tp-link.com/fr/home-networking/adapter/tl-wn722n/.

[121] P. Cruz, J. B. P. Neto, M. E. M. Campista, and L. H. M. K. Costa, "On the accuracy of data sensing in the presence of mobility," in *2016 7th International Conference on the Network of the Future (NOF)*, 2016, pp. 1–5.

[122] M. F. Akli, N. N. E. A. Boukerras, N. Derradji, D. Laga, and M. I. Syed, "BLEPal," Aug. 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03765103

[123] L. S. Vailshery, "Number of public Wi-Fi hotspots worldwide from 2016 to 2022," 2022, https://tinyurl.com/4wfebd7j.

[124] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless LAN monitoring and its applications," in *Proceedings of the 3rd ACM Workshop on Wireless Security*. New York, NY, USA: Association for Computing Machinery, 2004.

[125] A. Pratama, "Assign fixed name to the network interface on Raspbian," https://tinyurl.com/396ka8nd.

[126] D. Lawson, "Enable predictable names," https://tinyurl.com/332nbkv3.