Enhanced Location Information and its Application for Improving Wireless Networks

vorgelegt von

Filip Lemić
(M.Sc. in Electrical Engineering and Information Technology)
geb. in Pula, Kroatien

von der Fakultät IV
Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Stephan Völker, Technische Universität Berlin
Gutachter: Prof. Dr.-Ing. Adam Wolisz, Technische Universität Berlin
Gutachterin: Prof. Dr. Niki Trigoni, The University of Oxford
Gutachter: Prof. Dr. Gunnar Karlsson, KTH Royal Institute of Technology
Gutachter: Prof. Dr. Mahesh K. Marina, The University of Edinburgh

Tag der wissenschaftlichen Aussprache: 30. November 2017

Berlin 2018

...za Zecu...

Acknowledgments

Five years later, ten years older, and I am done with this journey. And what a ride it has been! Sometimes challenging, borderline annoying, nevertheless largely exciting, adventurous, and fun. Everything but boring. And I owe it primarily to my supervisor, Professor Adam Wolisz. Dear Professor Wolisz, I am very honored to have had you as my mentor. You made it challenging, you made it adventurous, you made it fun. And I have learned a lot in the process. Thank you for that, thank you sincerely.

I am immensely thankful to my co-supervisor and friend, Vlado Handziski. Vlado, the advices, encouragement, and support you gave me are much more than what I dared to hope for. I definitely wouldn't get to this point without you! I genuinely enjoyed learning from you and I hope many other PhD students will get the same chance. Throughout my PhD I also received an exquisite support from Arash Behboodi. Arash, without your kind words and support, coming to the end of this journey would have been much much harder for me. Thank you truly, my friend.

Jasper Büsch and Mikolaj Chwałisz were hugely helpful in the initial stages of my PhD. Petra Hutt helped me to swim through (or not to drown in) the German bureaucracy. Petra, thank you a lot and sorry for sometimes annoying you in the process. Irina Piens provided very valuable support in project work and organizational issues. Tolja Zubow provided excellent feedback in the last stages of my PhD. All of them, together with my other colleagues at TKN, made the last five years a great experience for me, and for that I am immensely grateful.

My sincere thanks also goes to my supervisors at the UC Berkeley, Professors John Wawrzynek and Jan Rabaey. I enjoyed learning from you and thank you for giving me that opportunity. A big thanks also goes to my colleagues from UC Berkeley. Thank you for making my stay so enjoyable! I highly appreciate the support received from my colleagues in the EVARILOS project. It was my great pleasure to work with all of you. I would also like to thank my PhD committee members for accepting to review my thesis and for their valuable feedback.

Last but not least, a huge thanks to my family and friends for being there with me and for me. To my mom: "Evo mama, gotov sam!"[1]. I couldn't ask for a cooler and more supportive family. Mama, tata, buraz, seka, I love you all. Zvone, Miha, Daniel, Artem, and Amir, it is probably because of you that I feel ten years older now. Nevertheless, I really enjoyed those fantastic years at Vater bar and would do it all over again. Siljan, Đole, Mojsa, Lido, Sočec, Mislav, Necko, Kešan and all the others I missed to mention here (you know who you are), I'm happy to call you my friends. Seka, I don't owe you anything. I don't know why I'm even mentioning you here. Ivana, the memories we share, thank you for them. I will always carry the dragons and devils in my heart. Cimi Ana, you have been with me in my more recent ups and downs. Thank you for that. I am truly happy to have you around.

Throughout this thesis "We" is used intentionally to express my gratitude to the people who made my PhD journey an enjoyable one. After all, their contributions were the crucial ones.

[1] "Here mom, I'm done!"

Abstract

A highly desired feature of next-generation wireless networks is their contextual awareness, in which location information of mobile and nomadic devices arguably plays the most critical role. On one side, location information is used as underpinning to user-facing location-based services and, on the other, it shows significant potential as a driver for optimization of core aspects of the underlying infrastructure like performance enhancement of next-generation wireless networks. To fully achieve its potential, location information has to be accurate, robust, and seamlessly available. Unfortunately, these goals still face significant unresolved challenges, especially in indoor environments, where a seamless source of location information is of immense importance, but currently lacking. RF-based localization solutions show high promise for localization indoors. This work, therefore, aims at addressing some of the aforementioned challenges for primarily RF-based indoor localization.

For addressing these challenges, one first needs solid understanding of the performance of RF-based solutions for indoor localization. This is, unfortunately, hindered by the lack of a unified methodology and efficient evaluation process. As a contribution toward better objectiveness, efficiency, and productivity in performance evaluation, we first provide an infrastructure for experimental evaluation of RF-based indoor localization solutions. The infrastructure is publicly available, remotely accessible, and supports flexible experimentation. The objectiveness of evaluation is supported through unified evaluation methodology, controlling and monitoring external conditions, and automated positioning of a mobile device to be localized at different evaluation locations without involvement of a test person. We demonstrate the usability and capabilities of the infrastructure by evaluating a large number of indoor localization solutions in the form of a remote competition. To facilitate more efficient remote evaluation of RF-based indoor localization solutions in different environments under various conditions, we also contribute to the public domain a large set of raw measurements that can be used as an input to an RF-based indoor localization solution. We further provide an online platform for enabling streamlined remote evaluation of RF-based indoor localization solutions using these pre-collected raw datasets.

WiFi-based fingerprinting is a promising candidate for an indoor localization service because of its well-known advantages like the ubiquitous availability of WiFi infrastructures and possibility of using off-the-shelf devices. We therefore apply the developed methodology and infrastructure toward better understanding and formalization of WiFi-based fingerprinting. We additionally aim at increasing the robustness of fingerprinting to harsh environments and conditions, reducing its deployment overheads, and trying to predict its performance in new environments.

Despite the ongoing improvement of WiFi fingerprinting, fingerprinting is still only a single source of location information. In contrast to outdoors where GNSS covers most use-case requirements, in indoors there is no generic solution that provides full satisfaction. This is mainly because different solutions are deployed in diverse indoor environments and seemingly also maintained by various entities. Additionally, the requirements for accuracy, latency, and power consumption of provisioning of location information are very diverse across applications. To achieve seamless provisioning of location information to the applications, there is a need for a middleware platform that will integrate these individual sources of location information. While the platform should isolate the applications and the individual sources, it also needs to remain aware

of their requirements to be able to optimally guide the orchestration of these individual sources. To achieve this vision, we propose a localization service whose core entity is a middleware platform for integration, handover, and fusion of individual sources of location information. We further contribute with a set of interfaces for standardized interaction between the components of the proposed service. The invocation of different sources of location information administered by the middleware platform is in a given space at a certain time based on the requirements from the applications and the availability and provisioning features of different provisioning sources. This raises the question of which sources to invoke for provisioning of location information at a given time and space. We contribute with two algorithms for such an invocation, and perform an in-depth evaluation of the contributions of these algorithms to the overall performance of the proposed middleware platform. Under the control of the middleware platform, we instantiate a set of sources of location information based on previously developed WiFi fingerprinting-based solutions. We follow by evaluating the developed middleware platform using the proposed performance evaluation infrastructure, and show its benefits in terms of accuracy, latency, and availability enhancements of such a unified approach compared to individual localization solutions.

Finally, on two examples, we demonstrate how this refined location-awareness can be a valuable source of information for improving the performance of wireless networks. First, we propose a location-based mechanism for deciding if a device-to-device communication link between two mobile devices should be established. Second, we contribute a location-based mechanism for positioning of a mobile relay in a way that optimizes the link quality between the end-devices that want to establish a communication link. By leveraging the previously developed flexible experimentation infrastructure we show that, for practically achievable location information that is burdened with a certain level of inaccuracies, the proposed mechanisms indeed lead to very reliable device-to-device link establishment and close-to-optimal positioning of a mobile relay.

Zusammenfassung

Ein höchst erwünschtes Merkmal drahtloser Netzwerke der nächsten Generation ist ein kontextuelles Bewusstsein, bei dem die Standortinformationen von mobilen und nomadischen Geräten die wichtigste Rolle spielt. Einerseits werden Standortinformationen als Untermauerung für Nutzer-orientierte ortsbezogene Dienste verwendet und zum anderen zeigen sie ein erhebliches Potenzial für die Optimierung von Kernaspekten der zugrunde liegenden Infrastruktur, wie die Leistungssteigerung der drahtlosen Netzwerke der nächsten Generation. Um dieses Potenzial voll auszuschöpfen, muss die Standortinformation genau, robust und nahtlos verfügbar sein. Leider stehen diesen Zielen noch immer erhebliche ungelöste Herausforderungen gegenüber, vor allem in Innenbereichen, wo eine nahtlose Quelle von Standortinformationen von immenser Bedeutung ist, aber derzeit fehlt. RF-basierte Lösungen sind sehr vielversprechend für die Lokalisierung im Innenbereich. Diese Arbeit zielt daher darauf ab, einige der genannten Herausforderungen insbesondere für die RF-basierte Innenbereichslokalisierung zu bewältigen.

Um den Herausforderungen gerecht zu werden, braucht man zunächst ein solides Verständnis für die Leistungsfähigkeit der verfügbaren RF-basierten Lösungen für die Innenbereichslokalisierung. Dies wird leider durch das Fehlen einer einheitlichen Methodik und eines effizienten Evaluierungsprozesses behindert. Als Beitrag zur besseren Objektivität, Effizienz und Produktivität bei der Leistungsbewertung stellen wir zunächst eine Infrastruktur für die experimentelle Auswertung von RF-basierten Innenbereichslokalisierungslösungen bereit. Die Infrastruktur ist öffentlich verfügbar, fern zugänglich und unterstützt flexibles Experimentieren. Die Objektivität der Evaluierung wird durch eine einheitliche Bewertungsmethodologie unterstützt, die externe Bedingungen steuert und überwacht sowie die automatisierte Positionierung eines mobilen Geräts, das an verschiedenen Auswertungsorten ohne Beteiligung einer Testperson lokalisiert werden soll, ermöglicht. Wir zeigen die Benutzbarkeit und die Leistungsfähigkeit der Infrastruktur durch die Auswertung einer Vielzahl von Lokalisierungslösungen für den Innenbereich im Rahmen eines Fernwettkampfs. Um eine effizientere Fernbewertung von RF-basierten Innenbereichslokalisierungslösungen unter verschiedenen Bedingungen und Umgebungen zu ermöglichen, tragen wir ebenfalls mit einer großen Anzahl von Rohdatensätzen zur öffentlichen Nutzung bei, die als Eingabe für eine RF-basierte Innenbereichslokalisierungslösung genutzt werden kann. Wir bieten darüber hinaus eine Online-Plattform für die effiziente Remote-Evaluierung zahlreicher RF-basierter Innenbereichslokalisierungslösungen mit diesen vorab gesammelten Rohdatensätzen.

Der WiFi-basierte Fingerabdruck ist aufgrund seiner bekannte Vorteile, wie die allgegenwärtige Verfügbarkeit von WiFi-Infrastrukturen und der Möglichkeit handelsübliche Geräte zu benutzen, ein vielversprechender Kandidat für Innenbereichslokalisierungsdienste. Wir verwenden daher die entwickelte Methodologie und Infrastruktur für ein besseres Verständnis und eine Formalisierung des WiFi-Fingerabdrucks. Zusätzliche Ziele waren eine Steigerung der Robustheit in rauen Umgebungen und Bedingungen, eine Verringerung des Bereitstellungsaufwands und die Vorhersage der Leistung in neuen Umgebungen.

Trotz der laufenden Verbesserung des WiFi-Fingerabdrucks ist dieser dennoch nur eine einzelne Quelle von Standortinformationen. Im Gegensatz zu Anwendungsfällen im Freien, wo GNSS die meisten Use-Case-Anforderungen abdeckt, gibt es im Innenbereich keine zufriedenstellende generische Lösung. Das liegt daran, daß in diversen Innenbereichen unterschiedliche Lösungen eingesetzt und anscheinend auch von verschiedenen Entitäten gepflegt werden. Darüber hinaus sind die Anforderungen an Genauigkeit, Latenz und Energieverbrauch zur Bereitstellung von Standortinformationen sehr vielfältig in den jeweiligen Anwendungsgebieten. Um eine nahtlose Bereitstellung von Standortinformationen für die Anwendungen zu erreichen, besteht ein Bedarf an Middleware-Plattformen, die diese einzelnen Quellen von Standortinformationen integrieren. Während eine solche Middleware-Plattform Applikationen und die einzelnen Standortquellen isolieren soll, müssen die jeweiligen speziellen Anforderungen bekannt sein, um eine Orchestrierung der einzelnen Quellen optimal begleiten zu können. Um diese Vision zu erreichen, schlagen wir einen Lokalisierungsdienst vor, dessen Kern eine Middleware-Plattform für die Integration, Übergabe und Verschmelzung von einheitlichen Quellen von Standortinformationen ist. Wir vereinheitlichen einen Satz von Schnittstellen für die standardisierte Interaktion zwischen den Komponenten des vorgeschlagenen Dienstes. Die Aufrufe verschiedener Quellen von Standortinformationen, die von der Middleware-Plattform verwaltet werden, beziehen sich auf einen bestimmten Ort zu einem bestimmten Zeitpunkt, basierend auf den Anforderungen der Anwendungen und der Verfügbarkeit und Merkmalen verschiedener Bereitstellungsquellen. Dies wirft die Frage auf, welche Quellen für die Bereitstellung von Standortinformationen zu einem bestimmten Zeitpunkt und Ort herangezogen werden sollen. Wir tragen mit zwei Algorithmen dazu bei, eine Entscheidung über einen solchen Aufruf zu treffen und eine eingehende Bewertung ihrer Beiträge zur Gesamtleistung des vorgeschlagenen Middleware-Plattform durchzuführen. Unter der Kontrolle der Middleware-Plattform instanziieren wir eine Reihe von Quellen von Standortinformationen basierend auf zuvor entwickelten WiFi Fingerabdrucks-basierten Lösungen. Anschließend bewerten wir die entwickelte Middleware-Plattform mit der vorgeschlagenen Performance-Evaluierungs-Infrastruktur und zeigen ihre Vorteile in Bezug auf Genauigkeit, Latenz und Verfügbarkeitserweiterung eines solchen einheitlichen Ansatzes im Vergleich zu einzelnen Lokalisierungslösungen.

Abschließend zeigen wir in zwei Beispielen, wie dieses verfeinerte Standortbewusstsein eine wertvolle Informationsquelle für die Verbesserung der Leistungsfähigkeit von drahtlosen Netzwerken sein kann. Zuerst schlagen wir einen ortsbasierten Mechanismus vor, um zu entscheiden, ob eine Gerät-zu-Gerät-Kommunikationsverbindung zwischen zwei mobilen Geräten aufgebaut werden soll. Dann setzen wir einen ortsbasierten Mechanismus zur Positionierung eines mobilen Relais derart ein, dass die Verbindungsqualität zwischen den Endgeräten, die eine Kommunikationsverbindung aufbauen wollen, optimiert wird. Durch die Nutzung der zuvor entwickelten flexiblen Experimentierinfrastruktur zeigen wir, dass die vorgeschlagenen Mechanismen für tatsächlich gesammelte Standortinformationen, die mit einem gewissen Maß an Ungenauigkeiten belastet sind, in der Tat zu einer sehr zuverlässigen Gerät-zu-Gerät-Link-Einrichtung und einer nahezu optimalen Positionierung eines mobilen Relais führen.

Contents

List of Acronyms

2D 2-Dimensional

3D 3-Dimensional

AP Access Point

API Application Programming Interface

ARB Arbitrary Waveform Generator

AWGN Additive White Gaussian Noise

AoA Angle of Arrival

AoD Angle of Departure

BSON Binary JSON

CDF Cumulative Distribution Function

COTS Commercial Off-The-Shelf

CS Carrier Sensing

CSI Channel State Information

CSMA Carrier Sense Multiple Access

D2D Device-to-Device

DECT Digital Enhanced Cordless Telecommunications

ED Euclidean Distance

ETS Enriched Training Service

GDP Global Data Plane

GNSS Global Navigation Satellite System

GSM Global System for Mobile Communications

HD High-Definition

HTTP Hypertext Transfer Protocol

IDWI Inverse Distance Weighted Interpolation

IEEE Institute of Electrical and Electronics Engineers

IMU Inertial Measurement Unit

IR Infra-Red

ISM Industrial, Scientific and Medical

ISO/IEC International Organization for Standardization / International Electrotechnical Commission

IoT	Internet of Things
JSON	JavaScript Object Notation
kNN	k-Nearest Neighbors
KL	Kullback-Leibler
LIDAR	Light Detection and Ranging
LQI	Link Quality Indicator
LTE	Long Term Evolution
LoS	Line-of-Sight
MAC	Media Access Control
MW	Multi-Wall
MvG	Multivariate Gaussian
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NoSQL	Not only SQL
OMF	cOntrol and Management Framework
OS	Operating System
PC	Personal Computer
PDF	Probability Density Function
PH	Pompeiu-Hausdorff
PLCP	Physical Layer Convergence Protocol
PRR	Packet Reception Rate
PRSA	Per-Request Satisfaction Algorithm
PSI	Provisioning Service Interface
PTSA	Per-Time-bucket Satisfaction Algorithm
QoS	Quality of Service
REM	Radio Environmental Map
REST	Representational State Transfer
RF	Radio Frequency
RFID	Radio Frequency Identification
RII	Resource Interaction Interface
RPC	Remote Procedure Call
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator

SDK Software Development Kit

SINR Signal to Interference plus Noise Ratio

SLI Standardized Localization Interface

SLSR Standardized Localization Service

SNR Signal to Noise Ratio

SoA State-of-the-Art

SSH Secure Shell

SUT System Under Test

TCP Transmission Control Protocol

TKN Telecommunication Networks Group

TUB Technische Universität Berlin

TWIST TKN Wireless Indoor Sensor Network Testbed

ToF Time of Flight

UAV Unmanned Aerial Vehicle

UDP User Datagram Protocol

URI Uniform Resource Identifier

UWB Ultra-Wideband

WiFi Wireless Fidelity

XMPP Extensible Messaging and Presence Protocol

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The knowledge of location information of mobile and nomadic devices in wireless communicational networks is an important information for a large variety of use-cases. First and foremost, location information is an enabler of location-based services [1, 2, 3]. Location-based services range from navigation services, location-grounded targeted advertising [4], public emergency operations [5], personalized weather services, etc. Location-based services are also an integral part of social networks [6], used among others by Facebook, LinkedIn, Google, and Amazon. Such services can further be found in specialized domains, for example in health-care, where they are used for tracking of the patients and assets [7], and in smart households, where they allow intelligent control of household appliances, e.g. heating and light systems [8, 9]. Location information is a decisive aspect of the Internet of Things (IoT) vision of smart environments, in which locations of devices and services usually play a role in their selection and usage [10].

Location information also plays an important, even critical role as a source of context information in wireless networks. Location information as a part of context-awareness has a potential of improving and enhancing different aspects of wireless networks across the whole protocol stack [11]. At the physical layer, a Radio Environmental Map (REM) could be exploited for finding white spaces for enabling secondary radio transmissions in the cognitive radio domain [12]. The REM could also be leveraged for interference coordination and avoidance [13], reducing signaling overhead [14], handover [15], data-offloading [16], etc. Building the REM of a given environment is time consuming and has to be performed periodically [17]. The usual approach for building it is through crowd-sourcing, i.e. basing its buildup on the opportunistic measurements taken by mobile users' devices [18, 19]. Obviously, this approach requires location information of the mobile devices for building the REM. Location information could also be harnessed at the link layer, for example for avoiding hidden terminal problems [20], for assignment of channels to specific geographical areas [21], and for reducing the signaling overhead of link establishment [22]. On the network and transport layers, location information could be used for geographical routing [23], improving video streaming [24, 25], balancing of traffic in a network [26], etc. In general terms, knowing location information of mobile devices in wireless networks provides a plethora of opportunities for novel users focused use-cases and businesses, as well as for improving the performance of wireless networks.

1.2 Challenges

The Global Navigation Satellite System (GNSS) is a service that can reliably, accurately, and timely provide location information. However, the GNSS usually fails to provide such location information in indoor and urban environments, i.e. where people essentially spend most of their time. Hence, alternative approaches for providing location information indoors are needed. This has been recognized in the research community, which resulted in a plethora of approaches for indoor localization. These approaches leverage a large variety of technologies, signal features, and algorithms for location estimation. Multiple categorizations of indoor localization approaches are given in the literature, e.g. [27, 28, 29]. One of the most promising indoor localization approaches is Radio Frequency (RF)-based localization, using some characteristics of wireless signals for localizing a mobile device. RF-based indoor localization approaches show promise because of the ubiquitous availability of RF infrastructures and the possibility of using Commercial Off-The-Shelf (COTS) devices [28]. Particularly interesting are the 2.4 GHz and 5 GHz Industrial, Scientific and Medical (ISM) frequency bands because of their public availability and the possibility of unlicensed usage. For that reason, various technologies operate in these frequency bands, e.g. Institute of Electrical and Electronics Engineers (IEEE) 802.11 (WiFi), IEEE 802.15.4 (ZigBee) and IEEE 802.15 (Bluetooth), all of them having the potential of being used for indoor localization purposes. Following this realization, various localization approaches of such kind have been proposed in the research community, some examples being [30, 31, 32, 33, 34, 35, 36, 37, 38, 39].

The usability of these prototypical localization solutions was, in their early stages, usually demonstrated by experimentally evaluating their localization accuracy in a given proprietary environment. After the initial rollover and demonstration of their feasibility, it became clear that a comparative experimental evaluation of different RF-based indoor localization solutions is needed. This is necessary both for finding the best solution and for finding the best instantiation of a certain localization solution for certain conditions. This realization opens a set of challenges pertaining to the objectiveness of the evaluation of RF-based indoor localization solutions. The first clear challenge is that, for an objective comparative evaluation of different solutions or different parameterizations of the same solution, the evaluation environment has to be the same for all instances of experiments. Additionally, the evaluation methodology and the followed procedures have to be the same for guaranteeing an objective comparative evaluation. This pertains to the same number and locations of evaluation points, same metrics for characterizing the performance of different solutions, and same conditions in which the evaluation is performed. In addition, experimental evaluations usually require large efforts and induce substantial time overheads, making such experimentations in many cases impractical or even unfeasible. Hence, apart for guarantying the objectiveness of experimental evaluation of RF-based localization solutions, reduction in the overhead of performing such experimentations is also an open research challenge.

Among different approaches for RF-based localization, fingerprinting-based localization using commodity Wireless Fidelity (WiFi) infrastructures offers significant advantages. To name some, WiFi infrastructures are ubiquitously available, fingerprinting does not depend on a usually unreliable power-to-distance relation, it can be implemented on COTS mobile devices to be localized, and it does not require adjustments of an existing infrastructure, hence it can be piggybacked on the already ubiquitously available ones. However, WiFi-based fingerprinting is usually viewed as a black-box localization solution [40, 41, 42, 43], while treating it as a white-box could provide additional insights about its performance, which could be a base for its further improvements. Moreover, fingerprinting requires a training set, i.e. a survey of an environment at predefined locations, and its generation is laborious and time consuming. That is to say, reducing its deployment overheads is still an unresolved problem. Since it is costly to deploy a WiFi fingerprinting-based localization solution, it would be beneficial to understand, before deploying the solution, if its localization accuracy in a targeted environment will be acceptable for the envisioned use-cases. However, the prediction of the accuracy of WiFi-based fingerprinting in new environments is currently missing. Finally, it is well known that RF interference can degrade the performance of WiFi in terms of its communicational capabilities [44, 45]. It is therefore intuitive that the same degradation of localization accuracy can be expected if RF interference exists in an environment serviced by a WiFi fingerprinting-based indoor localization solution. However, a definite answer and consequently the characterization of this performance degradation are currently lacking. Obviously, in the likely case that interference really harms the performance of WiFi-based fingerprinting, there is a need for mitigating this negative influence for enhancing the robustness of the performance of fingerprinting.

Despite the ongoing improvements in WiFi fingerprinting-based localization, WiFi fingerprinting is still capable of providing location information only for limited geographical areas. Similar observations can be made for other sources of location information, e.g. the GNSS can reliably and accurately provide location information only outdoors. In general, there is no single prominent solution that can seamlessly provide accurate and timely location information. In the future, we can expect some sources being capable of providing location information in certain environments, while some others will provide better performance in other environments. The deployments of such sources of location information are also going to overlap significantly. From the perspective of entities consuming location information, they require such information to be seamlessly available, regardless of a source that is providing it. To provide the perception of seamless provisioning of location information to the end-users and applications, there is a need for an entity for integration, fusion, and handover of different location estimation methods. Such an entity would provide a perception of seamless provisioning of location information regardless of the provisioning source, where this information has a potential of being more robust and accurate, in comparison to any single source of location information. This has been recognized already in localization services available on today's mobile devices. However, such services are proprietary and use a limited set of sources of location information. Integration of new location estimation methods is therefore cumbersome, requiring adaptation to the specific interfaces of the proprietary localization service. In addition, location-based applications are tightly interwoven with the localization service that is typically provided by the operating system, hence these applications require significant restructuring to be able to run with another localization service.

Location information of mobile and nomadic devices currently mostly being used as a driver for location-based services and applications. Nonetheless, location information is can also be used for enhancing wireless networks and improving their performance. As mentioned previously, there have been suggestions about using location information for improving various aspects of wireless networks. However, the vast majority of such suggestions rely on perfect location information, which in practice is very unlikely to be the case. The unavoidable inaccuracies of location information should be taken into account when developing location-based approaches that advance certain aspects of wireless communicational networks and thus improve their performance, otherwise the feasibility of such approaches for practical scenarios will be limited.

1.3 Contributions

This section summarizes the contributions made in this thesis toward addressing above listed challenges, pertaining to:

- Enabling an objective, fast, user-friendly, and remotely accessible experimental evaluation of RF-based localization solutions by providing to the public a testbed and cloud infrastructures, as well as a clear evaluation methodology.
- Addressing multiple challenges of WiFi fingerprinting-based localization solutions, i.e. their decoupling and white-box testing, robustness improvements, reduction of complexity and deployment overhead, and predictability of their performance in new environments.
- Design, implementation, and evaluation of a modular localization service architecture that allows seamless and standardized provisioning of location information to applications by supporting handover, fusion, and integration of underlying sources of location information.
- Two examples of leveraging imperfect location information for improving selected aspects of wireless networks, more specifically a location-based decision-making mechanism for Device-to-Device (D2D) link establishment and a location-based mechanism for positioning of a mobile relay.

In the following we discuss in more details the contributions made in this thesis.

1.3.1 Infrastructure for Objective Evaluation of RF-based Indoor Localization Solutions

To address the need for testing systems that enable objective evaluation of their properties, we develop a testbed infrastructure for supporting automatized evaluation of RF-based indoor localization solutions under controlled interference. For evaluating the impact of RF interference on the performance of evaluated solutions, the testbed infrastructure leverages various interference generation and monitoring devices. Furthermore, the testbed infrastructure uses a robotic mobility platform serving as a reference localization system and providing the capability of transporting a device to be localized to different evaluation locations in an autonomous and repeatable manner. We evaluate the accuracy of the autonomous mobility platform in two setups, showing that, due to the high accuracy, the location estimation provided by the platform can be considered as the reference localization system for evaluation of RF-based indoor localization solutions. The infrastructure requests location estimates from an indoor localization System Under Test (SUT) using a well defined interface and the estimates are subsequently processed in a dedicated cloud-based metrics computation engine. The results of the experiments can be stored into dedicated publicly available repository, therefore creating a portfolio of evaluated solutions.

We further contribute by reporting on the design, execution, and results from a remote localization competition in which several different RF-based indoor localization algorithms have been evaluated are evaluated along a set of standardized metrics under unified and representative condition with the help of the proposed remotely accessible and automated testbed and cloud infrastructures, which reduces the overheads of such evaluation. For the performance evaluation of the competing algorithms we propose a combination of precision, latency and sensitivity metrics, under four evaluation scenarios, overall resulting in 28 different evaluation experiments. The execution of this competition provided us with the possibility of evaluating the usability and complexity of usage of the developed testbed infrastructure by the users that did not have any previous knowledge of its capabilities. Furthermore, we concluded that many of the evaluated algorithms use the same signal feature for localization, i.e. the Received Signal Strength (RSS). Hence, providing this signal feature to can be used as an input to a RSS-based localization algorithm could reduce the complexity and costs of performance evaluation. In addition, the obtained results confirmed that specific types of RF-interference generally and noticeably degrade the localization performance of different RF-based indoor localization solutions. As a side note, the obtained results also confirmed the previously well-known fact that WiFi-based fingerprinting is a promising candidate for an indoor localization service.

Binding the previous conclusion that many algorithms use the same signal feature for localization, we collect and offer to the public an extensive set of raw measurements that can be used as input data for evaluating primarily RSS-based indoor localization algorithms. We further provide a publicly accessible cloud platform for streamlined experimental evaluation of RF-based indoor localization algorithms. By contrasting its fidelity and usability with respect to remote experiments on the previously proposed physical testbed infrastructure, we show that the cloud platform produces comparative performance results, while removing the need for physical experimentation, hence offering a significant reduction in the experimentation complexity, and time and labor overheads.

1.3.2 Advancing WiFi Fingerprinting-based Localization Solutions

We advance WiFi-based fingerprinting with the following contributions: decoupling and white-box testing of WiFi-based fingerprinting, reduction of complexity and deployment overhead, interference robustness improvements, and predictability of WiFi fingerprinting performance in new environments.

1. The usual work-flow of fingerprinting algorithms can be decomposed in common phases (collection of raw measurements, creation of fingerprints, pattern matching, and post-processing). However, the performance of WiFi fingerprinting is usually focused on black-box comparison of complete algorithms. Moreover, the current practice of comparative experimental evaluation of WiFi fingerprinting-based localization algorithms is lacking rigor, with studies typically following ad-hoc evaluation procedures. To address these issues, we present a systematic evaluation methodology that is focused on gaining fine-grained insight about the relative contributions of the individual phases of fingerprinting-based localization algorithms to their overall performance. We illustrate the application of the proposed methodology using a comprehensive experimental case-study of 3 fingerprinting algorithms with 4 raw RSSI collection procedures, 3 fingerprint creation and pattern matching procedures, 4 different post-processing procedures in 3 testbeds and 4 evaluation scenarios. The results demonstrate that in the evaluated scenarios, a lower number

of WiFi APs and rather simple fingerprint creation and pattern matching procedures can achieve better performance in terms of localization accuracy than more sophisticated alternatives. The results also show that post-processing steps like k-Nearest Neighbors (kNN) procedure are indeed effective in reducing the localization error variability and extremes, thus increasing the stability of location estimates. Finally, the results demonstrate that WiFi-based fingerprinting provides reasonably accurate location information for office-like environments, while a substantial degradation of performance is visible in industrial, open-space like environments.

2. In fingerprinting, there is a trade-off between the accuracy of location estimation and the density of a laborious and time consuming survey for collecting training fingerprints. A generally accepted concept of increasing the density of a training dataset, without an increase in the amount of physical labor and time needed for surveying an environment for additional fingerprints, is to leverage a propagation model for the generation of virtual training fingerprints. This process, however, incur an overhead in terms of implementing a propagation model, defining locations of virtual training fingerprints, generating virtual fingerprints, and storing the generated fingerprints in a training database. To address this issue, we propose the Enriched Training Service (ETS), a web-service that enables storage and management of training fingerprints, with an additional "enriching" functionality. The user can leverage the enriching functionality to automatically generate virtual training fingerprints based on propagation modeling in the virtual training points. We further propose a method for defining the locations of virtual training fingerprints based on modified Voronoi diagrams, which removes the burden of defining virtual training points manually and which automatically "covers" the regions without sufficient density of training fingerprints. The evaluation results show that the usage of automated generation of virtual training fingerprints in the ETS results in more than 25% increase in point accuracy and 15% in room-level accuracy of fingerprinting.
3. As we have demonstrated by the competition that RF interference can adversely influence the accuracy of WiFi-based fingerprinting algorithms, we follow by typifying this impact and proposing a procedure for reducing the influence of RF interference on WiFi-based fingerprinting. We first study the effect of RF interference on Received Signal Strength Indicator (RSSI) signal feature used for localization, particularly the effect of interference on packet-based RSSI reported by IEEE 802.11 and IEEE 802.15.4 technologies. Using an information theoretic formulation, we distinguish three operational regimes and show that the RSSI values, in dBm, remain unchanged in the noise-limited regime, increase roughly linearly with interference power in dBm in the interference-limited regime, and cannot be obtained due to packet-loss in the collision regime. Second, we propose a procedure for adjusting the WiFi beacon packets' RSSI measurements based on estimates of their distortion caused by RF interference. For estimating the distortion of RSSI measurements, the procedure uses information about the spectrum power levels in the frequency band where a fingerprinting algorithm performs. The proposed procedure can be inserted in the previously proposed formalized work-flow of WiFi fingerprinting. Our experimental evaluation in different interference scenarios shows that the proposed procedure for mitigating the influence of RF interference significantly improves the accuracy, without notably increasing

the latency of location information provisioning.

4. As mentioned, WiFi-based fingerprinting solutions are typically evaluated only in a single or small number of discrete environments. When the end-user's environment is not part of the evaluated set, it remains unclear if and to what extent the reported performance results can be extrapolated to this new environment. We contribute by establishing a relationship between the similarities among a set of different deployment environments and parameterizations of fingerprinting algorithms on one side, and the performance of these algorithms on the other. We hypothesize about the factors that can be used to capture the degree of similarity among environments and parameterizations of the algorithms, and proceed to systematically analyze the performance of two fingerprinting algorithms across four environments with different levels of similarity. The results show that the localization error distributions have small statistical difference across environments and parameterizations that are considered similar according to our hypothesis. As the level of similarity is decreased, we demonstrate that the relative performance of the algorithms can still be preserved across environments. For dissimilar environments, the localization errors demonstrate larger statistical differences.

1.3.3 Proposal for Standardized Localization Service

To address the aforementioned interoperability and integration problems, a high-level localization service architecture that consists of location-based applications, a middleware entity for integration, handover, and fusion of different sources of location information (i.e. location information provisioning services), and resources for generating location information. We further propose a set of interfaces for unified interaction among these components.

The middleware entity in envisioned to select and invoke the available provisioning services and, in the later step, aggregate the received information and provide it to location-based applications. The selection and aggregation of the provisioning services is envisioned to take into account the accuracy and latency requirements from the applications and accuracy, latency, and power consumption characteristics of the provisioning services. We further propose two algorithms for the selection of provisioning services aiming at meeting latency and subsequently accuracy requirements from the applications, one subject to minimizing per-request power consumption, while the other subject to a per-time bucket power minimization. In the considered examples, we show that the per-time bucket optimization achieves better performance in terms of power consumption, while trading-off accuracy satisfaction.

We further contribute with the Standardized Localization Service (SLSR), a prototypical implementation of the proposed localization service architecture for supporting handover, fusion, and integration of different location information provisioning services. We instantiate the SLSR in an office environment and performed exhaustive performance benchmarking in the previously proposed testbed specifically designed for supporting such experimentation. Our results characterize the effects of different functional components envisioned in the SLSR on its final performance. In addition, our results quantify the accuracy benefits of fusion of representative sources of location information.

1.3.4 Using Location Information for Improving Wireless Networks Operation

We demonstrate the potential of using realistic location information for improving wireless networks in two scenarios, i.e. D2D communication and mobile relaying.

D2D communications commonly refer to the technologies that enable devices to communicate directly (without base stations or Access Points (APs) [46]. D2D communication has a high potential in reducing the amount of network traffic and improving the latency and energy efficiency of communication. Currently, D2D link establishment decisions are based on active probing between devices that wish to establish a D2D link. The main drawback of such approaches is a large overhead as during active probing no data communication can take place. We leverage physical locations of the devices that wish to establish a D2D link for estimating the probability of success in establishing the link before making an attempt to communicate. The probability of success is derived in a closed form that takes into account the imperfections of location information of the devices and intrinsic randomness of wireless environments. We experimentally evaluate the proposed location-based decision-making mechanism for D2D link establishment in a complex office-like indoor environment. We show that setting the Signal to Noise Ratio (SNR) threshold of the proposed mechanism to a value that is 5 dB higher than the nominal SNR required for communication results in reliable link establishment with false positive rate of less than 2%. Furthermore, we show a relatively small loss of link establishment potential due to an increase in the inaccuracies of location information.

Leveraging a relay for communication is a promising approach for improving throughput, coverage, and energy efficiency in wireless networks. If the destination device is nomadic, transmitting through a relay that is always at the same location is usually suboptimal in terms of maximizing the benefits of relaying. A mobile relay that is capable of positioning itself at different locations opens the possibility for dynamic optimization of the path quality between the source and the nomadic destination. How to optimally position the mobile relay in order to maximize the path quality, however, remains a challenging task. Under the assumption that the physical location information of the devices are either known or can be estimated, we propose a mechanism for positioning of the mobile relay with the aim of maximizing the SNR between the source and the destination. The proposed mechanism takes into account the practically unavoidable inaccuracies of estimated locations, as well as the propagation characteristics of the served environment. Using WiFi as an example technology, we experimentally evaluate the proposed mechanism in a complex indoor environment with the support of a specifically designed testbed infrastructure. For relatively small localization errors, our results show less than 4 dB average difference between the measured SNR at optimal locations of the mobile relay vs. the SNR at locations yielded by our positioning mechanism. Our results also illustrate how the quality of the paths created by the proposed positioning mechanism degrades in the face of increasing localization errors.

1.4 Thesis Structure

This thesis is structured as follows. In the next chapter, we provide an overview of basic concepts and definitions relevant for this work, followed by an overview of the related contributions for the literature. In Chapter 3 we present the infrastructure for evaluation of RF-based indoor localization solutions under artificially generated RF interference context. The same chapter reports on the execution and results of a remote competition in which multiple RF-based indoor localization solutions have been evaluated with the help of the proposed infrastructure. Chapter 3 also overviews a cloud platform for streamlined evaluation of RF-based indoor localization solutions. In Chapter 4 we present a set of contributions pertaining to enhancing and improving WiFi fingerprinting-based indoor localization solutions. More specifically, we contribute by formalizing WiFi-based fingerprinting, reducing its deployment overheads, increasing its interference robustness, and evaluating the possibility of predicting its performance in new environments. In Chapter 5 we present the Standardized Localization Service, a middleware architecture supporting handover, fusion, and integration of different sources of location information. In Chapter 6, we demonstrate how location information can be leveraged for improving the decision-making mechanism for D2D link establishment. In the same chapter, we show how a mobile relay can be positioned in a way that optimizes the expected SNR between a source and a destination by leveraging location information of the devices participating in the communication. Finally, in Chapter 7 we conclude the thesis and underline suggestions for future improvements and potential research directions.

2 Fundamentals and Related Work

2.1 Basic Concepts and Definitions

This section provides a a set of basic concepts and definitions of terms used throughout the rest of the thesis.

Localization Solution and Location-based Applications

A localization solution (or a location information provisioning service) is an entity with the capability of determining physical locations of a mobile device in a certain environment. An indoor localization solution enables determining location information in indoor environments, in which the GNSS generally fails to do so accurately [47]. In addition to determining the physical location of a mobile device, a localization solution makes the physical location of the mobile device available for location-based applications for enabling services such as navigating, tracking, monitoring. Alternatively, location-based applications can use location information for optimizing the performance of wireless networks. In general terms, location-based applications are entities that make use of the physical location of a mobile device for a certain purpose [48].

Classification of RF-based Indoor Localization Solutions

Based on the signal types used for estimating location, wireless indoor localization solutions can be roughly classified as infrared, ultrasound, Ultra-Wideband (UWB), and RF-based ones [49]. RF-based indoor localization solutions are predominantly discussed in this thesis. While RF frequencies range from a few kilohertz to a few hundreds of gigahertz, in this work we focus on the 2.4 GHz ISM frequency band. Hence, when stating "RF-based indoor localization solution", we assume an indoor localization solution operating in the 2.4 GHz ISM frequency band, if not explicitly stated otherwise. Technologies that are usually used for RF-based indoor localization in the 2.4 GHz ISM band are WiFi (IEEE 802.11), ZigBee (IEEE 802.15.4), and Bluetooth (IEEE 802.15.1). Different types of features derived from the received signals can be used for localization purposes in RF frequencies. Based on the signal features used for generating location, indoor localization solutions can roughly be classified as Angle of Arrival (AoA), Time of Flight (ToF), and RSSI-based ones. AoA is a signal feature that determines the direction of propagation of a signal incident on a receiving device. Contrasting AoA, Angle of Departure (AoD) determines the direction of propagation of a signal in reference to a transmitting device. ToF is a signal feature that specifies the time that it takes for a signal to travel between a transmitter and a receiver through a medium. Signal features processing procedures used for indoor localization can also vary. Based on the signal features processing method RF-based indoor localization solutions can be classified as proximity, multilateration / multiangulation, and fingerprinting-based [50].

Proximity-based localization solutions use different signal features for reporting the proximity of user's device and an AP [51]. Multilateration / multiangulation-based indoor localization solutions use geometrical dependencies between static (anchors) and a mobile device for estimating mobile device's location. Finally, fingerprinting-based localization solutions, also known as scene analysis, compute the similarities between stored fingerprints and the user's fingerprints for estimating location.

Multilateration and Multiangulation

Multilateration and multiangulation are localization procedures that use geometrical dependencies between a mobile device and each anchor node for estimating location of the mobile device. In particular, multilateration-based localization procedures use distances between the mobile device and all localization anchors. For obtaining the distances between devices different signal features can be used, with mostly adopted ones being the received signal strength and different variations of the ToF [52]. On the other side, for estimating location multiangulation-based localization procedures use the AoD feature of the signal issued by each localization anchor and/or AoA signal feature received by the mobile device, or vice-versa.

Received Signal Strength Indicator in WiFi

In the IEEE 802.11 standard (i.e. WiFi), RSSI is the relative received signal strength in a wireless environment, in arbitrary units [53]. The RSSI indicates the power level of the signal received by the IEEE 802.11 physical layer receive radio after the antenna and possible cable losses. The RSSI is measured by the receive radio during the reception of the Physical Layer Convergence Protocol (PLCP) preamble transmitted by the transmit radio. The RSSI decreases with the distance between the node that is receiving and the WiFi device that is transmitting the signal. Furthermore, RSSI is represented by 8 bits (256 levels). There is no standardized relationship of any particular physical parameter to the RSSI reading in the IEEE 802.11 standard. In other words, the IEEE 802.11 standard does not define any relationship between the RSSI value and the power level in mW or dBm. The IEEE 802.11 standard specifies that the RSSI can be on a scale of 0 to up to 255 and that each chipset manufacturer can define their own "RSSI_Max" value. Cisco, for example, uses a 0-100, while Atheros uses 0-60 scale. In this work, we focus on WiFi RSSI-based indoor localization solutions and in particular WiFi RSSI-based fingerprinting.

Fingerprinting-based Indoor Localization

Fingerprinting-based localization solutions (or shortly fingerprinting) can be divided into a training and a runtime steps (often also referred to as phases). The training step is performed *off-line* for obtaining a set of fingerprints of a wireless environment. In other words, the localization area is surveyed at a number of segments by collecting certain signal features at each segment. Furthermore, a training fingerprint of a segment is created following some methodology for processing the collected signal features. Using the obtained training fingerprints of a surveyed environment, the training database is created and stored on a fingerprinting server. The runtime step consists of three individual phases. Two of them are mirroring the phases of the first step: a number of scans are first created by the user at an unknown location, and the runtime fingerprint for the location is created following the same procedure as in the training step. On the server's side, the runtime fingerprint is compared with the training dataset following some predefined matching procedure and the most similar training fingerprint is reported as the estimated location. Alternatively, the output of the matching phase can also be a set of training fingerprints, which is post-processed in the post-processing phase for obtaining the estimated location.

Used Fingerprinting Algorithms

Throughout this thesis we use two WiFi RSSI-based fingerprinting algorithms. These algorithms are in details discussed in Chapter 4.1.1, while here we provide their high-level summaries.

Euclidean distance of averaged RSSI vectors: This fingerprinting algorithm uses the computed average value of RSSI measurements obtained from each AP used for localization. The fingerprint is a vector of average values of the RSSI measurements obtained from each AP used for localization in both the training and the runtime step, where K is the length of the vector. Let $\boldsymbol{\mu}_{t,m} = [\overline{RSSI}_{t,1}, ..., \overline{RSSI}_{t,k}, ..., \overline{RSSI}_{t,K}]$ be the vector of averaged RSSI values $\overline{RSSI}_{t,i}$ from each AP i obtained in training step at point $m \in 1, ..., M_t$, i.e. training fingerprint. In the same manner, let $\boldsymbol{\mu}_r = [\overline{RSSI}_{r,1}, ..., \overline{RSSI}_{r,k}, ..., \overline{RSSI}_{r,K}]$ be the vector of averaged RSSI values $\overline{RSSI}_{r,i}$ from each AP i obtained in the runtime step. The pattern matching procedure uses the Euclidean Distance (ED) between a training fingerprint at the cell m and the runtime fingerprint and it is given as:

$$D_E(\overline{\boldsymbol{X}}_{t,m}, \overline{\boldsymbol{X}}_r) = |\overline{\boldsymbol{X}}_{t,m} - \overline{\boldsymbol{X}}_r|. \tag{2.1}$$

$\overline{\boldsymbol{X}}_{t,m}$ and $\overline{\boldsymbol{X}}_r$ are the fingerprint vectors in the training and the runtime steps, respectively. The training fingerprints with the smallest distance (also called smallest weight) are then used in the post-processing procedure. In the post-processing procedure we used the weighed kNN method with the parameter *k* set to 4, since this method achieves the best performance results in comparison to a large set of other post-processing methods evaluated in Chapter 3 of the thesis.

Pompeiu-Hausdorff distance of RSSI quantiles: This fingerprinting procedure uses q quantiles of the RSSI values from each AP as fingerprints, which are calculated in two steps. First the Cumulative Distribution Function (CDF) of the RSSI measurements from each AP is computed. Second, the quantiles, i.e. RSSI values with probabilities $k/(q-1)$, where $k = 0, 1, ..., q-1$, are calculated. The result of the quantile calculation in both training and runtime steps is a quantile matrix $Q_{K,q}$, where K is the number of APs visible at the given location and q is a number of quantiles. The pattern matching procedure of this algorithm uses the Pompeiu-Hausdorff (PH) metric for capturing similarities between training fingerprints and an runtime one [54], as follows:

$$D_{PH}(\boldsymbol{X}_{t,m}, \boldsymbol{X}_r) = \max_{x_{t,k} \in \boldsymbol{X}_{t,m}} \min_{x_{r,k} \in \boldsymbol{X}_r} d(x_{t,k}, x_{r,k}) \tag{2.2}$$

Here $d(x_{t,k}, x_{r,k})$ is the Euclidean Distance measurement between elements of the runtime fingerprint $\boldsymbol{X}_r$ and training fingerprint $\boldsymbol{X}_{t,m}$ at point m. Pompeiu-Hausdorff (PH) distance measures how far two subsets of a metric space are from each other. The training point with the smallest PH distance with the runtime fingerprint is reported as an estimated location. Same as for the previous algorithm, 4 training fingerprints with the smallest distance D_{PH} are then used in a weighted kNN procedure.

Radio Frequency Interference

RF interference are the unwanted, interfering RF signals that interfere with desired RF signals and disrupt normal operations of devices operating in these frequencies. As mentioned, unless explicitly stated otherwise, in this work IEEE 802.11 is the desired technology and the 2.4 GHz ISM frequency band is the operating frequency band. Because of the IEEE 802.11 Media Access Control (MAC) protocols, a sufficiently strong interfering RF signal in the same frequency can appear as a bogus 802.11 station transmitting a packet. For this reason, IEEE 802.11 stations could potentially wait for long periods of time before attempting to access the transmission medium. Furthermore, RF interference is not limited to the IEEE 802.11 protocols, so the interfering RF signal might start while a legitimate IEEE 802.11 station is in the process of transmitting a packet. If this occurs, the destination station will receive the packet with errors and will not reply to the source station with an acknowledgment packet. In return, the source station will attempt to retransmit the initial packet, which will add overhead on the network. The RF interference types can be categorized into interference from neighboring WiFi devices, IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (ZigBee), and synthetic interferences such as the ones originating from microwaves, Digital Enhanced Cordless Telecommunicationss (DECTs), and amateur audio and video devices.

Fusion of Diverse Sources of Location Information

Fusion of diverse sources of location information results in improved accuracy, seamlessness, and robustness of location information provisioning. Literature discusses two types of such fusion, i.e. on the level of raw data (i.e. resources) for generating location information and on the level of location estimates provided by different sources of such information. In this dissertation, we focus on the latter.

2.2 Related Work

In this section, we provide an overview of previous works related to the four major contributions made in this thesis. In particular, the overview of related work pertains to infrastructures for experimental evaluation of RF-based indoor localization solutions, advancing WiFi RSSI-based indoor fingerprinting, localization service architectures that integrate diverse sources of location information, and location-based D2D link establishment and relaying in wireless networks.

2.2.1 Infrastructure for Evaluation of RF-based Indoor Localization Solutions

Indoor positioning has been a hot research topic in the research community over the past years. As discussed previously, an objective performance evaluation of different indoor localization solutions is needed, which has been recognized in the community. Hence, infrastructures for supporting such evaluation have been discussed in the literature. The VirTIL testbed is a customized testbed infrastructure for evaluation of indoor localization algorithms [55] in an office-like facility. The testbed features a low-cost TurtleBot robot that serves as a reference localization system. The robot also carries the SUT while directed remotely through hallways of an office building. The w-iLab.t II testbed [56] also uses an autonomous mobility platform as a source of ground-truth and for remote positioning of the SUT at different locations in the testbed. However, this testbed is deployed in an open-space indoor environment. A very similar localization testbed deployed in an industrial automation environment is presented in [57]. The Mobile Emulab is a remotely-accessible mobile wireless and sensor testbed that also supports experimentation related to evaluation and benchmarking of indoor localization solutions [58]. Similar to the other testbeds, the Mobile Emulab makes use of an autonomous mobility platform capable of transporting a device to be localized to different evaluation locations in the environment. This autonomous mobility platform also serves as a source of ground-truth locations for the evaluation of the performance of an evaluated localization solution. In that regard, all the discussed testbeds are similar to the testbed infrastructure discussed in this thesis. The discussed testbeds are also remotely accessible, same as the proposed testbed infrastructure. Some testbeds are also office-like environments similar to the environment in which the proposed testbed in deployed. In addition to the capabilities provided by other testbed infrastructures, the testbed we propose offers the possibilities of generating and monitoring wireless interference context. In contrast to the other testbeds, the proposed testbed infrastructure supports calculation and storage of metrics for characterizing the performance of the evaluated SUT. Hence, the proposed testbed infrastructure supports a remote and fully automated evaluation of RF-based indoor localization solutions.

Public raw datasets represent an important bedrock of scientific research in disciplines ranging from astronomy to studies of climate change and the humane genome. Dedicated sites, like the Amazon Public Dataset[1], act as central repositories of popular and large datasets, streamlining their use in cloud-based frameworks. These datasets can be leveraged as common benchmarks for assessing the relative performance of different systems. Popular examples are the use of National Institute of Standards and Technology (NIST) Language Recognition datasets in the domain of language recognition [59] or ImageNet datasets in computer vision [60]. In the domain of RF-based indoor localization, there has been only one previous effort to virtualize some aspects of the experimental evaluation of localization algorithms [55]. In [55], the authors focus on evaluation of range-based indoor localization algorithms by providing to the public the range measurements collected at various locations in their testbed environment. In our work, we promote and offer unprocessed low-level data such as RSSI and ToF measurements that are used in a wide range of indoor localization approaches.

Indoor localization competitions provide the possibility for evaluation of a large number of

[1] Amazon Public Datasets: https://aws.amazon.com/datasets

indoor localization solutions in the same setting, which increases the comparability of their performance results. However, such competitions are rare due to their cost, labor, and time intensity. In [61] the authors report on the experiences and lessons learned during the IPSN / Microsoft indoor localization competition. During the competition a large number of indoor localization solutions, including magnetic, light and ultrasound-based ones, were evaluated manually by carrying devices to be localized to different evaluation points. The set of EvAAL competitions is another popular line of competitions focused on indoor localization for the assisted living scenarios [62, 63, 64]. In these competitions, the evaluation was performed in multiple assisted living-related scenarios. The evaluation in EvAAL competitions was performed by a test person carrying a device to be localized to different locations in the environment. In both related competitions the evaluation was performed on-site, which required competitors to be physically present at the competition site. Our competition offered a possibility of remote participation, thus the cost for the competitors was significantly reduced. Furthermore, in our competition the focus was on the effects of interference on the performance of RF-based solutions.

The influence of RF interference on the performance of RF-based indoor localization solutions has been a vaguely investigated topic in the literature. However, some indication exist about the performance degradation of RF-based indoor localization solutions due to interference. The authors in [65] experimentally show that background IEEE 802.11 traffic as interference source can double the 80^{th} percentile localization error of an IEEE 802.15.4-based indoor localization solution in the worst case scenario. In [66], the authors give an analytical model and show by simulations the effect of several types of interference on the performance of an Radio Frequency Identification (RFID)-based indoor localization solution. We extend the current findings by showing that various RF interference patterns generally degrade the performance of a large number of RF-based indoor localization solutions in the 2.4 GHz ISM frequency band. This effect is mostly visible through the increased localization errors in the evaluation scenarios with RF interference, in comparison to the evaluation scenarios with minimized interference. The effect of channel interference on WiFi-fingerprint for different channel assignment schemes has been studied in [67]. The results show that the selection of channel the assignment scheme influences the accuracy WiFi fingerprinting. In contrast to these contributions, we experimental demonstrate the effect of various types of interference on multiple RF-based indoor localization solutions under unified conditions. Similar to the listed contributions, we demonstrate a generally negative impact of interference on the localization accuracy of different solutions.

2.2.2 WiFi Fingerprinting-based Localization Solutions

A number of research activities aimed at developing WiFi fingerprinting-based indoor localization solutions or improving its accuracy. An overview of WiFi-based fingerprinting, together with its advantages and limitations, is given multiple works, e.g. [68, 69, 70]. Numerous fingerprinting-based indoor localization algorithms support room level localization. One of the most popular ones of that type is RedPin [71]. RedPin is a hybrid localization solution that combines different technologies (WiFi, GSM, Bluetooth) for creating a fingerprint. Another well-known room level indoor WiFi-based fingerprinting approach is ARIEL [72]. ARIEL uses the room clustering in order to estimate the room where user stands, together with noise minimization and motion detections from the users' mobile phones. Besides the room level fingerprinting, various solutions are aiming to the more precise indoor localization. In [73], the authors propose an indoor localization algorithm that uses the difference of the signals from different WiFi and Bluetooth APs for creating a fingerprint. The authors in [40] use the WiFi beacon packets RSSI values for creating a fingerprint. They describe several algorithms and propose two new ones. As mentioned, the majority of current contributions focus on improving the accuracy of WiFi RSSI-based indoor fingerprinting, with some examples being [74, 75, 76, 77, 78, 79]. Contributions made in this thesis that are related to WiFi-based fingerprinting are not predominantly focused on improving its accuracy. Instead, we focus on less investigated topics, i.e. formalizing the fingerprinting procedure, reducing the deployment overheads, increasing its interference robustness, and predicting the performance of fingerprinting in new environments.

1. The majority of contributions study WiFi-based fingerprinting as a black-box solution, examples being [40, 41, 42, 43, 80]. We believe that the decoupling of WiFi fingerprinting into standard phases could provide additional insights into its performance, which could lead to further advancements of such solutions. We therefore distinguish different formal phases of WiFi fingerprinting and proceed to evaluate the contributions of methods in different phases to the overall performance of fingerprinting algorithms. Similar decoupling is proposed in [75], where the authors distinguish data collection, fingerprint definition, and location estimation algorithms as different phases of fingerprinting. In comparison [75], we evaluate the contributions of a set of different methods in each of the defined phases on the overall performance of fingerprinting algorithms. We perform the evaluation in a set of office environments, which is similar to the evaluation performed in [75]. Contrasting [75], we additionally carry out our evaluation in an open-space environment. In [81], the authors specify the following phases of WiFi-based indoor positioning systems: data collection, preprocessing, location calculation, post-processing, and data distribution. This formalization is similar to the one we propose specifically for WiFi-based fingerprinting. However, the evaluation in [81] aims at black-box testing of different WiFi-based indoor positioning systems and hence fails to evaluate the contributions of different methods in the specified phases on the overall performance of different positioning systems.
2. Collecting training fingerprints that are needed for enabling fingerprinting-based localization is a lengthy, costly, and cumbersome process. Training fingerprints have traditionally been collected by the administrator of a fingerprinting-based localization system before putting the system to use. More recent approaches enhance the traditional collection of training fingerprints with crowd-sourcing, i.e. an opportunistic collection of additional training fingerprints by the users of the system. This procedure reduces the efforts needed for collection of training fingerprints and improves the performance of a fingerprinting system [82]. Examples of crowd-sourcing-based collection of training fingerprints include [83, 84, 85, 86, 87, 88].

   Independent of the procedure for collection training fingerprints, the physically collected fingerprints can be leveraged for generating virtual training fingerprints. The generation of virtual training fingerprints is beneficial in terms of enhancing the accuracy and improving the deployment overhead of fingerprinting [89]. Various approaches exist for generating virtual training fingerprints based on propagation modeling at locations not covered with the physically collected training fingerprints. The authors in [89] investigate the influence of virtual training fingerprints generated using various propagation models on the accuracy of fingerprinting. The approach in [90] aims on generating virtual training fingerprints by modeling WiFi signal power levels based on kriging, while the one in [91] bases the creation of virtual training fingerprints on the support vector regression. The authors in [92] propose a learning algorithm that reduces the calibration efforts of fingerprinting by creating virtual training fingerprints based on linear interpolation. The authors in [93] base the creation of training fingerprints on Gaussian process regression, which in addition to modeling of received power at each fingerprint location allows modeling of the variance of the received power. The approach in [94] aims on generating virtual training fingerprints, i.e. increasing the density of a training database, based on discontinuity preserving smoothing. Finally, in [95] the authors aim on generating virtual training fingerprints based on the higher-order Voronoi tessellation. Contrary to the previously mentioned works, in which the focus is mainly on proposing novel methods for generating virtual training fingerprints, in this work we focus on the design of a training database that automates the application of different methods for generation of virtual fingerprints by providing a common platform. Similar tools have been proposed for different research domains, for example the ArcGIS Spatial Analyst [96] for automating the interpolation and modeling in spatial analysis. By using the proposed training database, without additional implementation burdens, the user is able to generate and store additional virtual training fingerprints.

3. In relation to the mitigation of the interference effect on WiFi fingerprinting, in [97] the authors propose the usage of a pair of properly positioned APs as a differential node. This is done in order to eliminate the undesired effects of environmental factors such as interference and noise. In addition, an approach for mitigating the influence of environmental factors with a very similar intuition has been presented in [98]. The proposed approaches result in a more stable differential strength of received signals and hence can be used for mitigating the effects of environmental dynamic factors. However, if a strong source of interference such as a jammer is present in a served environment, the distortion of received signals due to interference highly depends on the distance between the signal source and the measurement location, as well as the distance between the interference source and the measurement location [99, 100]. Hence, measurements from different signal sources will usually experience different distortions. The difference of such measurements will for that reason not be stable, hence the localization accuracy will be negatively affected. In contrast to the aforementioned contributions, we model the distortion of received signal strengths by leveraging spectrum information, which accounts for the distance between the signal source and the measurement location, as well as for the distance between the interference source and the measurement location.
4. Finally, the prediction of performance of WiFi-based fingerprinting is a less investigated topic in the literature. In [101], the authors propose a method for deriving the expected uncertainty of location estimation in WiFi-based fingerprinting. The proposed method uses the measurements collected in a training phase of fingerprinting for predicting the localization inaccuracy at any location in a served environment. In addition, in [102, 103] the same issue is addressed by proposing adverse methods for predicting the accuracy of fingerprinting from measurements collected in a training phase. Obviously, all these

contributions assume that the collection of training measurements is performed. Collecting training measurements in a new environment poses a large overhead that should be avoided if the expected accuracy of fingerprinting in this new environment is not acceptable for envisioned use-cases. In contrast to the aforementioned contributions, we hypothesize about the accuracy of WiFi-based fingerprinting in an entirely new environment. In other words, our approach does not require the collection of training measurements in a new environment.

2.2.3 Integration of Diverse Sources of Location Information

As discussed in the previous chapter, location information provided to location-based applications should be seamlessly available, accurate, and robust to harsh environment and external hindrances [11]. For achieving the desired accuracy in a seamless and robust way, there is a need for an entity for handover, fusion, and integration of different sources of location information. As mentioned, two types of such integration are discussed in the literature, i.e. fusion of raw data used for generating location information and fusion of location estimates provided by different sources of such information. In the following, we overview the contributions made in the literature pertaining to both types of fusion of diverse sources of location information.

A plethora of different fusion methods using various sensor readings has been proposed. For example, today's state-of-the-art localization services on smart-phones, i.e. Android's Google Play services location Application Programming Interfaces (APIs) and iOS's Core Location framework enhance GNSS readings with opportunistic WiFi and cellular measurements for estimating location. Moreover, various approaches combine WiFi with Inertial Measurement Unit (IMU) measurements for estimating location, for example [104, 105, 106, 18, 107]. Similarly, some contributions aim at combining WiFi with Bluetooth and opportunistic GNSS measurements, e.g. [71, 108, 109]. In [110], the authors propose an algorithm that combines WiFi, Bluetooth, Long Term Evolution (LTE), magnetic, and GNSS sensor measurements. Furthermore, in [111] the authors use a smart-phone camera for visual localization, which is then enhanced with a variety of other modalities such as IMU, GNSS, Bluetooth, and WiFi. Finally, various works demonstrate the feasibility of a localization approach based on readings from UWB enhanced with other sensors such as IMU [112], Bluetooth [113], WiFi [114], vision sensors [115], etc. There are drawbacks of fusion on the level of raw data, pertaining to organizational issues. For example, many sources of raw data that are envisioned to be used for localization are part of other systems to which a localization service does not have full access. Furthermore, this approach raises privacy issues in terms of leakage of resources for generating location information.

Integration on the level of fusion of location estimates provided by different sources of such information addresses the aforementioned issues, however it is a less investigated topic in the literature. Nevertheless, some proposals for such integration exists. Example-wise, a calibration-free localization solution based on triangulation, triangular interpolation, and extrapolation is proposed in [116]. Furthermore, a hybrid indoor localization solution that uses WiFi, where the RF propagation loss model and fingerprinting method have been combined, is proposed in [117]. Similar contribution has been made in [118]. The selective fusion location estimation (SELFLOC) algorithm that estimates location by fusing location information from multiple sources of location information that are not necessarily based on the same technology is given in [119].

While the authors in [119] mostly focus on the algorithmic aspect of such a system, we aim at a system-level specification of the localization service architecture that integrated various sources of location information. There have been existing proposals in this direction [120, 121], however they are generally under-defined. This pertains to the lack of a well-defined system architecture and interaction primitives. A step in this direction has been made in [122] where the authors propose the Location Stack, an abstraction that, among others, handles fusion of diverse sources of location information. Similar approach has been taken in [123], where the authors propose LocON, a platform capable of delivering location information obtained through fusion of several coexisting sources of location information. Although both contributions discuss the need for standardized interfaces as means of integrating diverse sources of location information, those interfaces remained unspecified. Moreover, the existing approaches to not focus on provisioning of location information based on the requirements from the applications leveraging such information, but generally aim at provisioning of the most accurate location information that can be obtained. Hence, to the best of our knowledge, we present a first full proposal for a modular standardized localization service architecture that aims at a comprehensive solution for integration on the level of fusion of location estimates provided by different sources of such information, while taking into account the requirements from location-based applications.

2.2.4 Using Location Information for Improving Wireless Networks Operation

Location information has a potential for improving the performance of wireless networks, as discussed e.g. in [11, 124]. In this thesis, we demonstrate how practically obtainable (i.e. erroneous) location information can be used for D2D link establishment and for positioning of a mobile relay.

Currently, mechanisms for establishing D2D links rely of manual pairing between devices that are in communication range [125]. Such manual pairings are widely available in WiFi and Bluetooth networks. More recent D2D communication link establishment use RSS for pairing [125, 126], i.e. links between devices are established if RSS values from probe packets transmitted by one and observed by the other device are above a given threshold [125, 126]. Such approaches require active probing, which induces latencies and reduces the communication potential. Location information of the potentially communicating devices can be used for addressing those issues. This has been recognized in the LTE supported D2D link establishment (e.g. [127, 128, 129]). In this context, LTE infrastructure informs the devices that are in proximity (e.g. connected to the same base station) and wish to establish a D2D link about their roles (transmitter and listener) and configuration for establishing a link (e.g. an operating channel). However, the problem of deciding if the devices that are in proximity and are hence good candidates for D2D operation can really establish a D2D link with a desired quality has not been addressed. Hence, LTE supported D2D link establishment still requires active probing (i.e. a discovery signal) to evaluate the quality of a D2D link between two devices [130]. Our location-based decision-making mechanism for D2D link establishment addresses this issue.

Location information can also used as an input to a decision-making mechanism in relaying, i.e. in the decision if and consequently which opportunistic relay should be utilized for transmitting information between end-devices, as discussed in [131, 132]. The aforementioned contributions assume that wireless propagation can be characterized with path-loss only, as well as that a perfectly accurate and instantaneous estimation of location information of the devices can be performed. Both assumptions are unrealistic in practice, which has already been recognized in the community. Hence, the authors in [133] consider the influence of information delay on location-based relaying. The influence of path-loss inaccuracies on location-based relaying is considered in [134]. In [135], where the authors consider the joint influence of erroneous and delayed location information for optimizing location-based relaying. In contrast to these contributions that are focused on deciding if a relay should be used and consequently on selecting the optimal one, we focus on finding an optimal location of the mobile relaying device. Furthermore, we assume that the destination is nomadic and hence we do not consider delayed location information to be significant. Finally, we assume that location information of both the destination and of the relay are burdened with errors, while in [135] the authors assumed only one of them is imperfect.

3 Infrastructure for Objective Evaluation of RF-based Indoor Localization Solutions

A flow of scientific method generally starts with an observation of a phenomena. Based on the observation, a hypothesis for its characterization is formed. The hypothesis is then evaluated to identify its correctness and suitability for realistic conditions. Based on the results of the evaluation, the hypothesis is accepted or revised. Obviously, the evaluation of the hypothesis has to be scientifically rigorous and objective, otherwise false statements about its applicability and usefulness can be deducted.

Location information of users and devices in indoor environments is an asset for various purposes, which can be characterized as an observed phenomena. Researchers therefore formed a large number of hypotheses, for example leveraging RF, Infra-Red (IR), magnetic, and various other types of environmental features for deriving locations of users and devices in indoor environments. These pioneering hypotheses have been initially experimentally evaluated to show their applicability, i.e. it has been shown that such approaches can indeed be used for indoor localization. Particularly interesting result for this thesis is the one demonstrating that RF signals can be used for indoor localization purposes.

After the initial roll-out, it became clear that the results of evaluation of different RF-based indoor localization solutions have to be objective and reproducible, i.e. the experimental evaluation of RF-based indoor localization approaches needs to be performed with more rigor. This is needed for finding the best localization solution, as well as for finding the optimal parameterization of a given localization solution for certain environments and conditions. In addition, initial experimental evaluations of RF-based indoor localization were slow and cumbersome processes, which posed an additional limitation for the researchers trying to advance indoor localization.

To be able to evaluate scientific hypotheses and to be able to do meaningful research toward advancing RF-based indoor localization, we first had to develop a methodology for objective and reproducible experimental evaluation of indoor localization solutions. We further had to guarantee that the experimental evaluation of indoor localization solutions is a fast and straightforward process. This chapter of the thesis overviews our contributions in those directions.

Parts of this chapter have been published previously. The proposed testbed infrastructure for experimental evaluation of indoor localization solutions has been discussed in [136]. A methodology for objective evaluation of indoor localization solutions, as well as the capabilities of the proposed testbed infrastructure and its usability have been discussed in [35]. The overview of a cloud platform for streamlined and fast evaluation of RF-based indoor localization solutions is given in [137].

3.1 Infrastructure for Evaluation of RF-based Indoor Localization Solutions

The performance of RF-based indoor localization solutions is usually being evaluated using local experimental setups in environments of variable sizes and different types. The evaluated localization solution is typically being carried by a test-person to each evaluation point, introducing inaccuracies due to errors in positioning and orientation of a person and influence of the person's body on the evaluated solution. The results are usually obtained for different scenarios and characterized by non-standardized metrics, with different number, density, and locations of evaluation points. Furthermore, the influence of RF interference is usually being neglected or only marginally considered in the current performance benchmarks, while it is intuitively expected that RF interference can indeed influence the performance of RF-based indoor localization.

Due to above specified reasons, the results of evaluation of RF-based indoor localization solutions are currently hardly repeatable. Furthermore, the comparative experimental evaluation of different solutions is typically biased. This is because a solution to be evaluated is usually optimally parameterized for given conditions, while the baseline solution is taken off-the-shelf, which results in its suboptimal performance.

We aim on addressing these issues by providing a testbed infrastructure for supporting objective evaluation of RF-based indoor localization solutions without the need of a test-person and with the capabilities of generating artificial RF interference and monitoring interference context in the testbed environment. We further present the cloud infrastructure for unified calculation of the performance metrics of indoor localization evaluation experiments.

3.1.1 Overview of the Flow of Evaluation

An overview of the flow of experimentation aiming at the evaluation of RF-based indoor localization solutions is given in Figure 3.1. In the first phase of evaluation experiments, i.e. the deployment phase, the user is expected to deploy the infrastructural nodes of the SUT in the testbed, deploy the mobile node of the SUT on a mobility platform, with the mobility platform being a means of transport of the mobile node in the later steps of evaluation procedure, and accommodate the SUT to provide location estimates meaningful for the testbed API. During the experimentation phase the user should firstly be able to generate different RF interference patterns and monitor the wireless spectrum. Secondly, the user should be able to start an evaluation experiment,i.e. start the procedure of positioning the mobile node of the SUT to different locations in the testbed environment. Upon coming to each location, the SUT should be requested to provide a location estimate, which should be, together with the ground truth location, sent for further processing. Additionally, other metrics should be captured, e.g. the response time of the SUT and room accuracy. In the third phase of the experimentation, i.e. the results management and storage phase, metrics characterizing the performance of the evaluated SUT are envisioned to be calculated, stored, and reported back to the user.

The above described flow of experimentation is to a certain level different from the usually followed one. First, we envision a mobility platform for transport of the mobile part of the SUT to different evaluation locations, while traditionally this operation is usually performed by a test-person. In contrast to the usual experimentation flow, we aim at enabling interference generation and monitoring capabilities. Furthermore, we aim at capturing multiple performance metrics, while usually the focus is mostly on the accuracy of an evaluated solutions.

For achieving the above described flow of evaluation experiments, we propose the infrastructure for objective evaluation of RF-based indoor localization in the environment with controlled and monitored interference context. The proposed infrastructure has two main functional components, i.e. a testbed infrastructure for performing evaluation experiments and a cloud service for the calculation of performance metrics. The testbed and cloud infrastructures can be used jointly, creating a full chain of experimentation, as specified by the flow of evaluation. Our remotely accessible and controllable testbed infrastructure leverages an autonomous mobility platform which enables accurate and repeatable positioning of indoor localization SUT and removes the necessity of manual rearrangement of a device to be localized. Furthermore, the testbed infrastructure integrates devices for generating controlled RF interference context, which is envisioned to be used for evaluating the influence of RF interference on the performance of the SUTs. For validation of the generated RF interference context, the infrastructure features monitoring devices that monitor the RF spectrum at different locations in the testbed. The same devices can be used for monitoring of the outside interferences in order to guarantee equal conditions for all evaluated SUTs. We also propose a cloud service for calculating and storing a set of metrics for characterizing the performance of different indoor localization solutions. This data is envisioned to be used as a portfolio of evaluated solutions, which could then be used for example for comparing the performance of a newly evaluated solution with the already evaluated ones or for finding a suitable solution for a given environment.

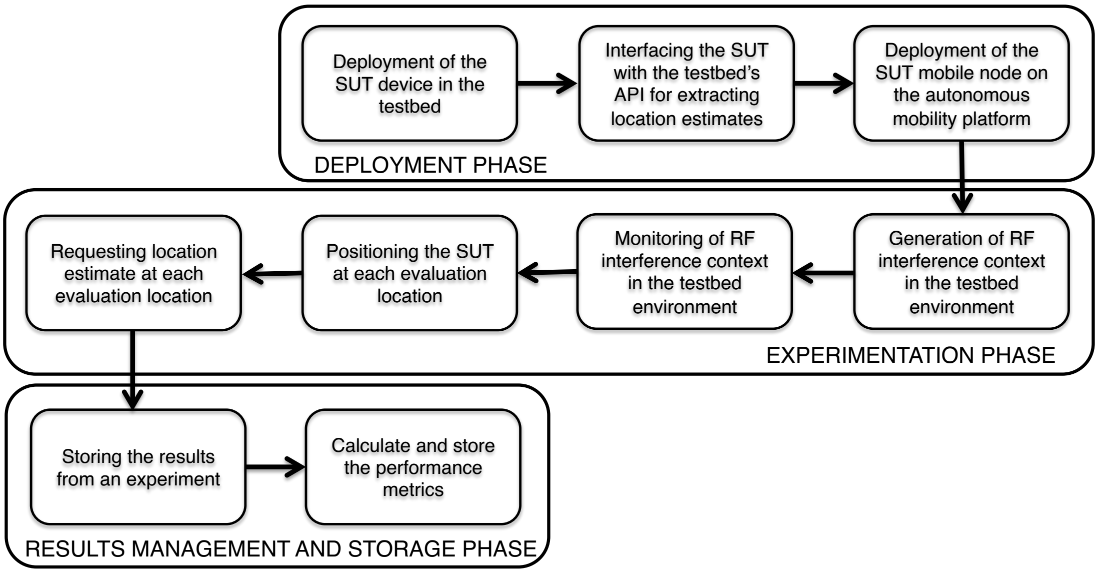

Deployment of the SUT device in the testbed
Interfacing the SUT with the testbed's API for extracting location estimates
Deployment of the SUT mobile node on the autonomous mobility platform
DEPLOYMENT PHASE
Requesting location estimate at each evaluation location
Positioning the SUT at each evaluation location
Monitoring of RF interference context in the testbed environment
Generation of RF interference context in the testbed environment
EXPERIMENTATION PHASE
Storing the results from an experiment
Calculate and store the performance metrics
RESULTS MANAGEMENT AND STORAGE PHASE

Figure 3.1: Flow diagram of the envisioned evaluation procedure

3.1.2 Testbed Environment and Hardware Components

This section gives a short and generic overview of the testbed environment and of different hardware components that are part of the infrastructure and currently available for experimentation. These are COTS components minimally modified to support the envisioned testbed functionalities. Their overview is given here for completeness purposes only.

Testbed Environment

The Telecommunication Networks Group (TKN) testbed is located on the 2^{nd} floor of a campus office building. The environment can be characterized as office space with brick walls, i.e. more than 400 m^2 area with more than 10 rooms. The testbed is comprised of three types of rooms, namely small offices ($14m^2$), big offices ($28m^2$) and laboratories ($42m^2$), as shown in Figure 3.4. It offers realistic indoor conditions, with a number of people moving around the premises, resulting in small environmental changes like opening of doors or slight movements of infrastructure (chairs, tables, etc.). A certain level of uncontrolled RF interference from external sources (such as WiFi APs from neighboring buildings) is expected in this environment.

Hardware Components of the Testbed

TKN Wireless Indoor Sensor Network Testbed: The TKN Wireless Indoor Sensor Network Testbed (TWIST) [138] is a multiplatform, hierarchical testbed architecture developed at the TKN. The selfconfiguration capability, the use of hardware with standardized interfaces and open source software make the TWIST architecture scalable, affordable, and easily replicable. The TWIST instance at the TKN office building is one of the largest remotely accessible indoor sensor network testbeds with 204 sockets, currently populated with 102 eyesIFX and 102 Tmote Sky nodes (Figure 3.2). The nodes are deployed in a 3-Dimensional (3D) grid spanning over 3 floors of an office building at the Technische Universität Berlin (TUB) campus, resulting in more than 1500 m^2 of instrumented office space. In small rooms, two nodes of each platform (i.e. eyesIFX and Tmote Sky) are deployed, while the larger ones have four nodes. This setup results in a fairly regular grid deployment pattern with intra node distance of 3 m, as shown in Figure 3.4. Within the rooms the sensor nodes are attached to the ceiling.

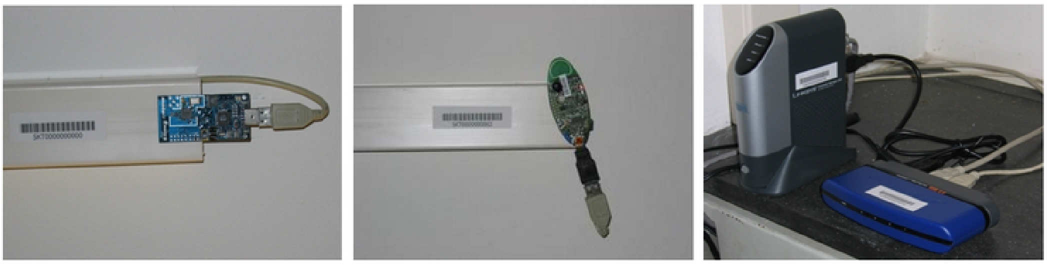
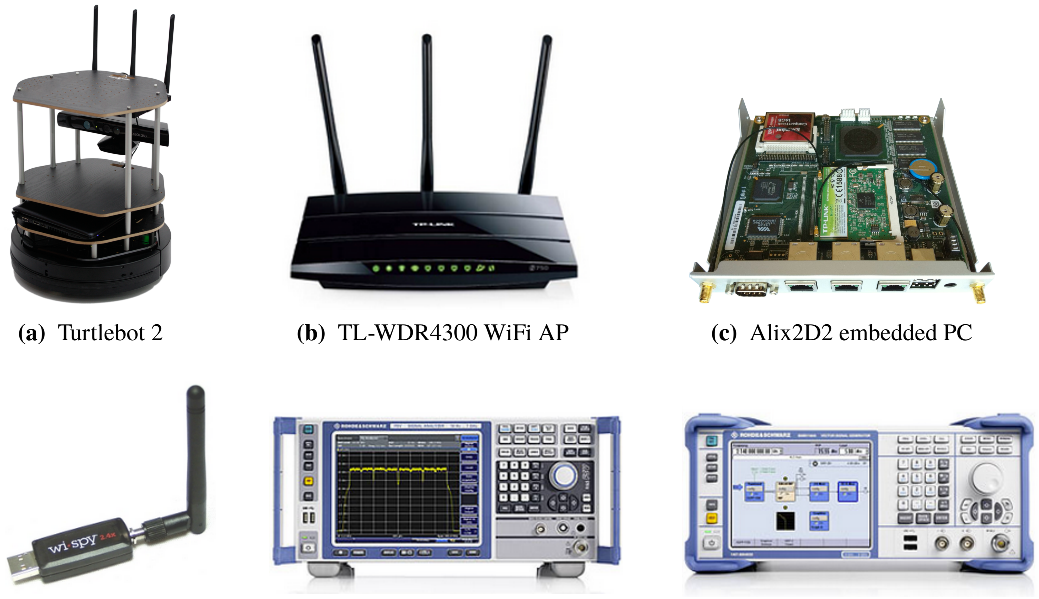


Figure 3.2: TWIST components: Tmote Sky, eyes IFXv2, NLSU2 supernode/USB Hub

TurtleBot II: The mobile robotic platform is based on an open source hardware and software development called Turtlebot-II and have been further enhanced to fit the testbed environment seamlessly. They consist of a mobile base (Kobuki), a laptop and a Microsoft Kinect 3D camera, as well as optional Light Detection and Ranging (LIDAR) sensor for accurate positioning of the mobility platform in the testbed environment (Figure 3.3a).

WiFi APs: TKN testbed is equipped with 18 dualband TP-link N750 APs (TL-WDR4300) (Figure 3.3b). They run OpenWRT as an Operating System (OS). Additionally to that we also have three ALIX2D2 embedded Personal Computers (PCs) (Figure 3.3c) equipped with Broadcom WL5011S 802.11b/g cards. They are running Ubuntu as an OS. All of them can be configured as APs as well as WiFi clients, allowing the flexible and easy configuration of experiments. Positions of the WiFi APs and embedded PCs in the 2^{nd} floor of TKN testbed are given in Figure 3.4.

R&S SMBV100A Signal Generator: Rhode&Schwarz SMBV100A is a very flexible signal generator (Figure 3.3f). It provides transmissions of baseband signals in the range of 9 kHz to 6 GHz. It is possible send any generated or stored signal with up to 120 MHz bandwidth. By applying toolboxes, the SMBV100A signal generator allows generating different standards conform signals like e.g. WiMAX, WiFi or LTE. Together with the R&S FSV7 Spectrum Analyzer complete transmission chains can be set.



(a) Turtlebot 2 **(b)** TL-WDR4300 WiFi AP **(c)** Alix2D2 embedded PC

(d) WiSpy dongle **(e)** R&S spectrum analyzer **(f)** R&S signal generator

Figure 3.3: Hardware components of the testbed

Low-Cost Spectrum Analyzers: The TKN infrastructure also comprises several WiSpy sensing devices (Figure 3.3d). These are low-cost spectrum scanners that monitor activity in either 868 MHz, 2.4 GHz or 5 GHz spectrum, and output the measured RF energy of the received signals. They can be attached, via USB port, to any PC in the testbed.

R&S FSV7 Spectrum Analyzer: Rhode&Schwarz FSV7 signal and spectrum analyzer (Figure 3.3e) is a very flexible and fast signal and spectrum analyzer covering the frequency range between 9kHz and 7 GHz. It is simple extensible by several measurement applications and toolboxes. Furthermore, it is possible, by buying appropriate licenses, to add complete receiver chains like Bluetooth, LTE, WiMAX or WiFi.

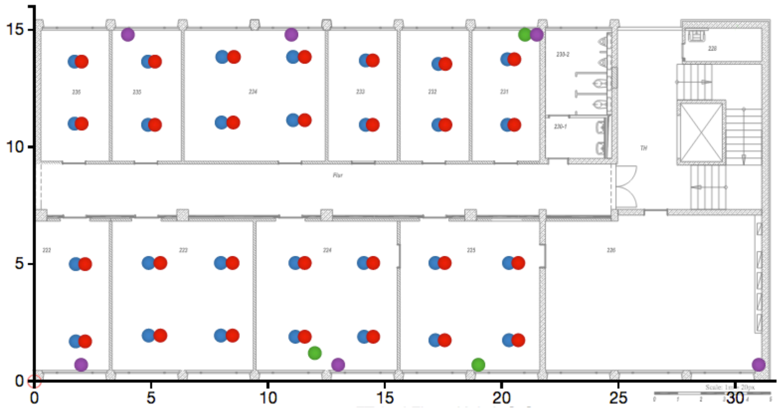


Figure 3.4: Locations of devices on the 2nd floor of the testbed (red: eyesIFX, blue: Tmote sky, purple: WiFi router, green: AlixD2D PC)

3.1.3 Functional Components of the Testbed

This section presents the functional components of our testbed infrastructure for supporting indoor localization benchmarking. A graphical presentation of the testbed components is given in Figure 3.5.

Central Testbed Control Engine: Central testbed control engine is used for instrumenting other functional components of the testbed: interference generation and monitoring, autonomous positioning of the SUT using the autonomous mobility platform and interfacing with the SUT. In other words, central testbed control engine is nothing more than a set of scripts that can be used for starting RF interference on the devices for generating interference or start monitoring the wireless spectrum on the devices envisioned for spectrum monitoring. Further, central control engine is used for autonomous navigation of the autonomous mobility platform and SUT to the evaluation points, requesting estimates from the SUT and sending the estimates and the raw data for further processing and storage in the cloud infrastructure.

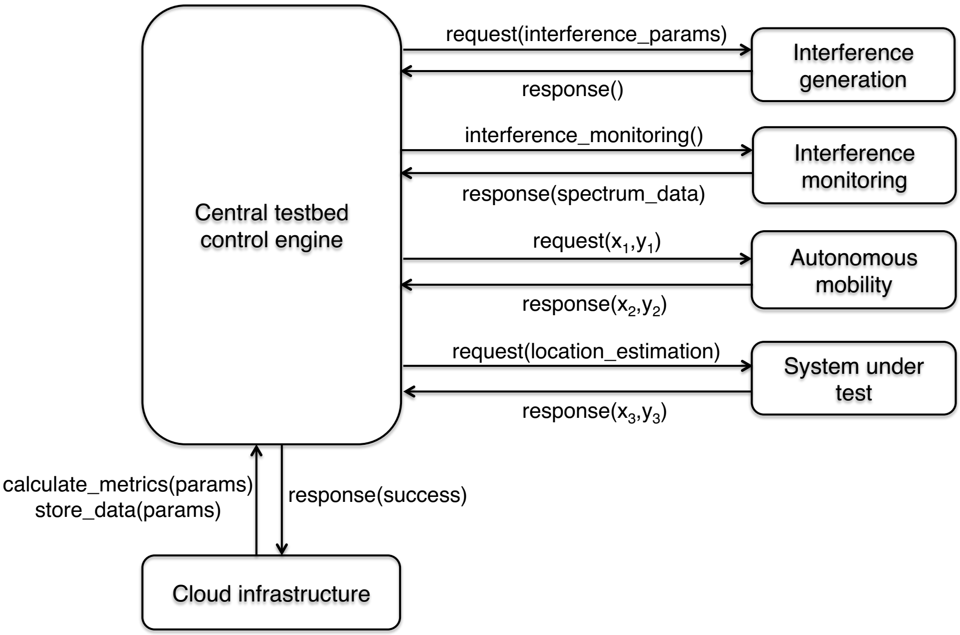


Figure 3.5: Overview of functional components of the testbed

Location Estimation Requests: The indoor localization SUT is considered a black-box solution and the only requirement for the SUT is to be able to report the estimated location on request. The only necessary interaction with the SUT, performed over a well defined API, is used to obtain location estimates. Due to their widespread usage, we have selected standard Hypertext Transfer Protocol (HTTP) requests and JavaScript Object Notation (JSON) notation for the API specification. Namely, the API is the HTTP Uniform Resource Identifier (URI) on which the SUT listens for requests for location estimation. Upon request, the SUT has to provide the location estimate as a JSON response in the format as specified with Listing 3.1.

JSON parameters *coordinate_x* and *coordinate_y* are x and y coordinates of the estimated location. These coordinates are expressed in meters and are relative to the zero-point in the testbed, with coordinates $(x, y, z) = (0, 0, 0)$, as presented in Figure 3.4, where the same zero-point is used by the autonomous mobility platform. They are required parameters and as such they must be reported upon request. Parameter *coordinate_z*, also expressed in meters, is an optional parameter, due to the 2-Dimensional (2D) evaluation environment. Finally, parameter *room_label* defines the room label in which the SUT estimates it is.

Listing 3.1: Format of reporting of location information

```
{
 "coordinate_x": 'Estimated location: coordinate x',
 "coordinate_y": 'Estimated location: coordinate y',
 "coordinate_z": 'Estimated location: coordinate z',
 "room_label":   'Estimated location: room'
}
```

Combined with reference location data from the autonomous mobility platform, the obtained location estimates are subsequently processed by a dedicated cloud engine that calculates relevant evaluation metrics.

Interference Generation: RF interference can have an impact on the performance of the RF-based indoor localization solutions. To evaluate this impact we have developed means to generate various types of interference, varying in the technology, interference power, and usage patterns. We distinguish three main types of interference that can be generated: WiFi traffic, low-power sensor nodes traffic and arbitrary signals.

As shown on Figure 3.6, we are using cabled backbone to control all devices participating in the interference generation. A common type of wireless activity in the 2.4 GHz ISM band is WiFi traffic. With the help of the TP-LINK wireless routers and ALIX2D2 embedded PCs, we can create the interference context of typical home or office environments. In addition to our interference generation system, we can, depending on the need of an experiment, introduce additional WiFi enabled devices, such as regular notebooks, to serve as entities for interference generation. We are also able to instrument a robotic mobility platform to carry such a device in our testbed environment, in order to introduce mobility in the interference generation. Furthermore, using distributed low-power sensor nodes provided by TWIST we can generate various types of IEEE 802.15.4 traffic patterns or constant carrier (jamming) as interference sources. Finally, using the signal generator we are able to generate power envelopes of any periodic signal with Arbitrary Waveform Generator (ARB) provided by the device. This includes IEEE 802.11 (WiFi), IEEE 802.15 (Bluetooth), but also synthetic interference such as microwave or DECT.

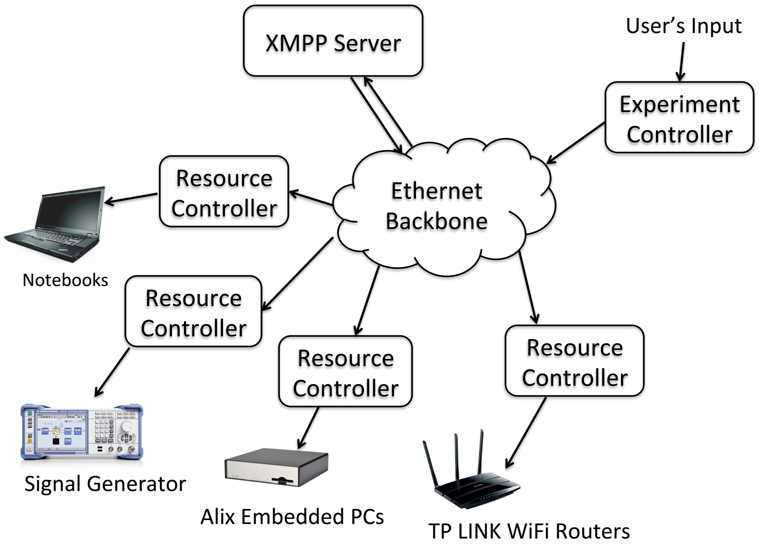


Figure 3.6: Architecture of the interference generation system

Interference Monitoring: The majority of our controlled interference devices is envisioned for interference generation in the 2.4 GHz ISM spectrum band, which is free for usage. Hence, the WiFi connectivity is ubiquitous in our office-like testbed environment, thus the uncontrolled RF interference should be expected. Depending on an experiment, in order to control if the intended interference is generated in a desired way or if the environment is relatively free of RF interference, it is necessary to monitor the wireless spectrum. We therefore leverage WiSpy devices for spectrum sensing at various locations in the testbed. In addition, one WiSpy device is attached to the autonomous mobility platform making sure that the measured interference is not exceeding the planned one at each evaluation point. Finally, for a fine-resolution monitoring of the spectrum we are able to use the high-end spectrum analyzer device, as shown in Figure 3.7. With the monitoring system and the interference generation mechanisms it is possible to detect and quantify the influence of interference on RF-based localization solutions. Both the interference generation and the interference monitoring systems are controlled using the cOntrol and Management Framework (OMF). OMF is a software tool for controlling the flow of experimentation. A specification of the experiment, in this context defined as the behavior of all hardware components, is stored on an Extensible Messaging and Presence Protocol (XMPP) server. The OMF control does not include the TWIST sensor network, which has its own publicly available control and management framework, as discussed in [138]. In the OMF, the user specifies the desired behavior of all components, which is then through the OMF experiment controller translated to the XMPP specification in the XMPP server. In the later step, the XMPP server assigns the desired operation to each of the specified hardware components. This specification is then sent to a resources controller of each hardware component, where it is translated to a set of natively understandable operations. The experiment can then be initiated, and through OMF the execution of the experiment can be monitored. This allows for simple specification and execution of experiments, which in addition can be remotely controlled.

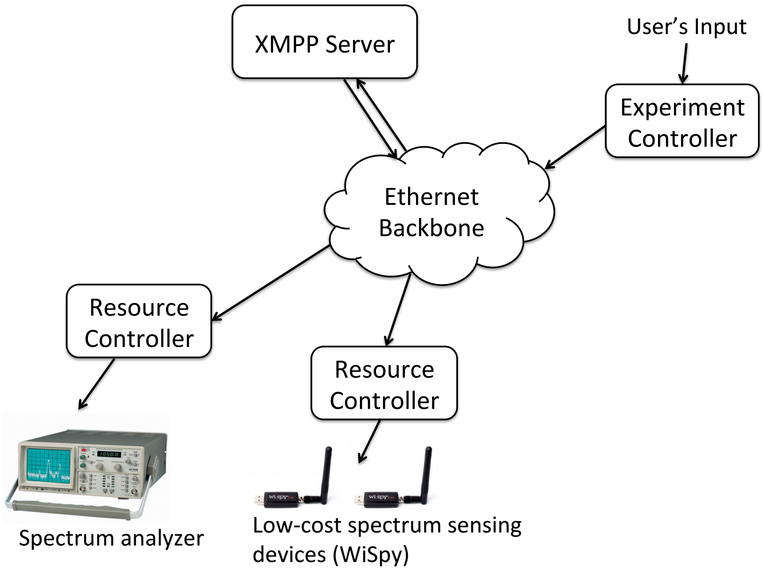


Figure 3.7: Architecture of the interference monitoring system

Mobility Support: The autonomous mobility platform can be used for transportation of the SUT to different evaluation points without the presence of a human test-person and in a repeatable way. The autonomous mobility platform provides a simple interface, named *Waypoint Controller*, for requesting autonomous movement to a given coordinate, as shown in Figure 3.8. Furthermore, as the specification of an experiment, one can define a set of evaluation locations. The autonomous mobility platform in that case iterates over each one of them, computing a shortest path to each goal by using the internal navigation system and by taking dynamic obstacles into account. Upon reaching an evaluation point, the autonomous mobility platform reports the reached location to the testbed's central control engine. The testbed's control engine then requests the estimated location from the SUT. After the SUT reports the estimated location, the autonomous mobility platform is requested to proceed with the next evaluation point.

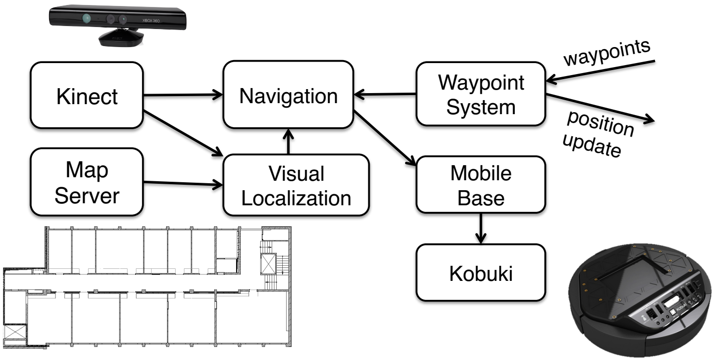


Figure 3.8: Architecture of the autonomous mobility platform

Autonomous localization and navigation of the autonomous mobility platform is supported by the Kobuki mobile base with dynamic obstacle avoidance inside buildings. This is done by leveraging a provided floor plan and combining it with depth readings from the visual sensor (Kinect), as shown in Figure 3.8. Obstacle avoidance is being a crucial property when the autonomous mobility platforms operate in a non-controlled office environment. Our enhancements, besides multiple minor adaptations to enable a fully autonomous operation within the testbed, include equipping the autonomous mobility platforms with high-performance WiFi routers that automatically monitor, select and connect to the best available WiFi AP. This is shown in Figure 3.9, depicting seamless connectivity of the autonomous mobility platform through by leveraging different APs based on the location of the platform in the testbed environment. The APs supporting connectivity of the autonomous mobility platform operate in 5 GHz ISM frequency band, in order to be out-of-band to the evaluated indoor localization SUTs to avoid interfering with them.

Accuracy of the Autonomous Mobility Platform

As stated previously, we consider the location estimates provided by the autonomous mobility platform as reference locations for the evaluation experiments. To be able to use this information in that manner, we needed to make sure that the accuracy of the autonomous mobility platform is at least one order of magnitude higher than the usual performance of the RF-based indoor localization solutions, which as a rule of thumb is currently around 1.5 m in average for the

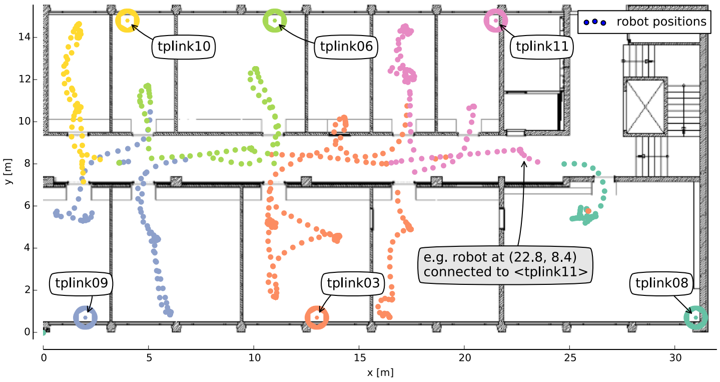


Figure 3.9: Automatic switching of APs for seamless connectivity

State-of-the-Art (SoA) solutions [61]. Due to that, we have evaluated the accuracy of our autonomous mobility platform in two different settings. The first evaluation was performed during the Microsoft Indoor Localization Competition in conjunction with International Conference on Information Processing in Sensor Networks 2014 (IPSN'14). The competition was performed in the the area with the size of around 200 m^2, consisting of two big conference rooms and a hallway [61]. Overall 20 evaluation point were defined, labeled on the floor and their coordinates were manually measured. Following that, we used our autonomous mobility platform to position itself on each evaluation point, and measured the errors using the laser distance measuring device. The average localization error of the autonomous mobility platform in this environment is less than 25 cm. Moreover, we performed the similar experiment in our testbed. Involving a sophisticated device Tachymeter Typ TS 06 Plus (Leica), with the accuracy of 2 mm at 100 m, we defined a set of 28 evaluation points in our testbed environment. We navigated the autonomous mobility platform to each of them and measured the positioning errors with the laser distance measuring device. The accuracy of our autonomous mobility platform in this environment is even better, achieving the overall average accuracy of less than 15 cm. The reason for better accuracy in this environment is better accuracy of the internal map of the robot, optimization of the internal parameters of the robot (e.g. improved navigation and dead-reckoning) and a very challenging set-up of the IPSN'14 environment. The CDF of the localization errors are given in Figure 3.10, while the summarized performance results are presented in Table 3.1.

Table 3.1: Summarized localization errors of the autonomous mobility platform

Metrics	IPSN'14	TKN environment
Mean localization error [m]	0.234	0.142
Localization error variance [m^2]	0.018	0.015
Localization error median [m]	0.221	0.136
Min/max localization error [m]	0.000 / 0.560	0.010 / 0.310

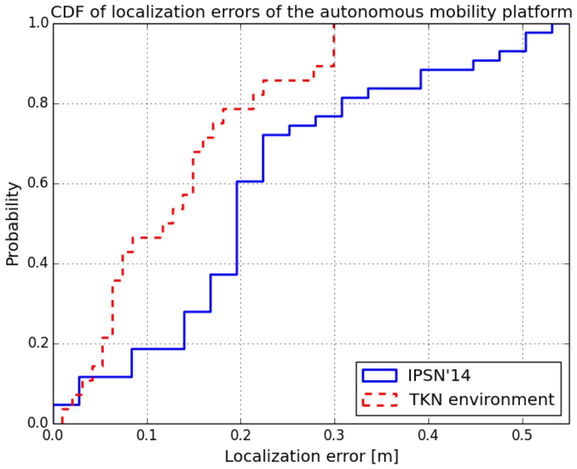


Figure 3.10: CDFs of localization errors of the autonomous mobility platform

3.1.4 Cloud Service for Metrics Calculation and Storage

The functions of the cloud infrastructure for evaluation of RF-based indoor localization solutions are storage of results of evaluation and calculation of metrics characterizing the performance of evaluated solutions. Given the information about a reference location, the central testbed control engine can request a location estimate from the SUT, using the previously described API. Upon receiving the estimate, the central testbed control engine can request the calculation of metrics and the storage of results from the cloud service. This is done by the central engine defining a message consisting of the ground truth location where the measurement was taken, estimated location, the latency needed to provide the estimate, and the name of an experiment. The central engine can then send the message to the cloud service for calculation and storage of the evaluation results. The cloud service is implemented as an Remote Procedure Call (RPC) that calculates metrics and, based on the experiment name, stores the results in a corresponding database. In case of multiple messages are sequentially received by the cloud service, i.e. for multiple evaluation locations, the service will automatically update the database with the results for each message, i.e. each evaluation point. The cloud service will also issue a response to the central testbed control engine, with the response specifying the calculated metrics for characterizing the performance of the evaluated SUT.

3.2 Remote Evaluation of Indoor Localization Solutions Under RF Interference

Having developed the testbed and cloud infrastructures, we are able to support the evaluation of RF-based indoor localization solutions. This supported evaluation can be performed in an non-ad-hoc fashion, in the same environment, and using the same evaluation procedure, which enhances the objectiveness of the comparison of performance of different approaches.

Localization competitions are one model for performing a comparative evaluation of different RF-based indoor localization solutions. The main benefit of the model is that the competitors can deploy custom hardware and are in full control of the localization systems, while the organizers are responsible only for the evaluation environment and procedure. In such competitions, the competitors are invited to physically deploy their localization solutions in a shared evaluation environment and are evaluated along a set of metrics reflecting different application and user requirements. Unfortunately, large localization competitions, like the one from Microsoft [61] with 21 teams from all around the world, are infrequent, mainly due to the high deployment and traveling costs. An important observation from this competition is that a high percentage of competing solutions utilize common hardware like WiFi APs and smartphones. For this class of solutions a shared hardware base deployed in a selected location is very attractive option for their comparative evaluation.

Hence, such a remote competition firstly provides a possibility of evaluating the feasibility of our testbed and cloud infrastructures for a large set of indoor localization solutions. Furthermore, the testbed infrastructure enables us to evaluate the impact of RF interference, which is unavoidable in real life, on the operation of a specific solution. We therefore in this section report the execution and results from a remote competition for RF-based indoor localization algorithms, in which participating teams competed in precise localization under the influence of different types of RF-interference in our testbed premises. The competing localization algorithms were remotely deployed on top of hardware resources available in the testbed. They have been subsequently evaluated following a combination of accuracy, latency and sensitivity metrics, under four different RF-interference scenarios. A short generic overview of the competing algorithms is given in the Appendix A of the thesis.

3.2.1 Performance Metrics and Evaluation Procedure

This section presents the procedure followed in the evaluation of RF-based indoor localization algorithms during the presented remote competition. While the focus in this work has been on typical and widely deployed RF technologies, the leveraged evaluation methodology can also be applied to any other type of indoor localization solution.

Evaluation Metrics

For characterizing the performance of evaluated indoor localization SUTs three primary and one secondary metrics have been used. The primary performance metrics are extracted directly from the evaluation experiments, like point and room level accuracy, and latency (response time). Subsequently, based on the primary performance metrics under different interference scenarios, the interference sensitivity of an SUT has been calculated as a secondary metric. The metrics have been defined during the course of EU Project EVARILOS [139] and are therefore not the author's sole contribution. For completeness purposes, in the following we provide the definitions of these metrics.

Point Level Accuracy: Point level accuracy at one evaluation point is defined as the Euclidean distance between the ground truth location (x_{GT}, y_{GT}) and the location estimated by an indoor localization algorithm (x_{EST}, y_{EST}). The point accuracy of location estimation for one point is defined by the following equation:

$$PointAccuracy = \sqrt{(x_{GT} - x_{EST})^2 + (y_{GT} - y_{EST})^2}\ [m] \tag{3.1}$$

Room Level Accuracy: Room level accuracy of location estimation is a binary metric stating the correctness of the estimated room, defined by the following equation:

$$RoomAccuracy = \begin{cases} 1 & \text{if the estimated room is correct;} \\ 0 & \text{if the estimated room is not correct;} \end{cases} \tag{3.2}$$

Latency of Location Estimation: Latency relates to the time that an SUT needs to report the location estimate when requested. The time measured in the evaluation is the difference between the moment when the request for location estimate has been sent to an SUT ($t_{request}$) and the moment when the response arrived ($t_{response}$), as given by the following equation:

$$Latency = t_{response} - t_{request}\ [s] \tag{3.3}$$

Interference Sensitivity: Interference sensitivity reflects the influence of different interference types on the performance of an indoor localization algorithm. It is the percentage of change in primary metrics in the scenarios with interference, in comparison to the performance in the scenario without interference (reference scenario). For the case of a generalized metric (M), the interference sensitivity is given according to the following equation:

$$InterferenceSensitivity = \frac{|M_{reference} - M_{interference}|}{M_{reference}} \cdot 100\ [\%] \tag{3.4}$$

where $M_{reference}$ is the value of a primary metric M in the reference scenario and $M_{interference}$ is the value of a metric M in a scenario with interference. Note that if the performance of an algorithm for the performance metric M is better in a scenario with interference in comparison to the reference scenario, then the interference sensitivity metric is set to 0 %.

Obtaining Evaluation Metrics

The evaluation procedure was organized in four evaluation scenarios. In each scenario, for each of the 20 evaluation points, the set of metrics (point accuracy, room accuracy, latency) was obtained. For each set, the 75^{th} percentiles of point level accuracy and latency were calculated, together with the percentage of correctly estimated rooms, as shown in Figure 3.11.

Interference sensitivity was calculated as the difference in each primary metric in each interference scenario, in comparison to the reference scenario, using Equation 3.4. The overall interference sensitivity was the averaged interference sensitivity over all interference scenarios and all performance metrics, given by the following equation:

$$\overline{M} = \frac{1}{9}\sum_{i=1}^{3}(M_1(i) + M_2(i) + M_3(i)) \tag{3.5}$$

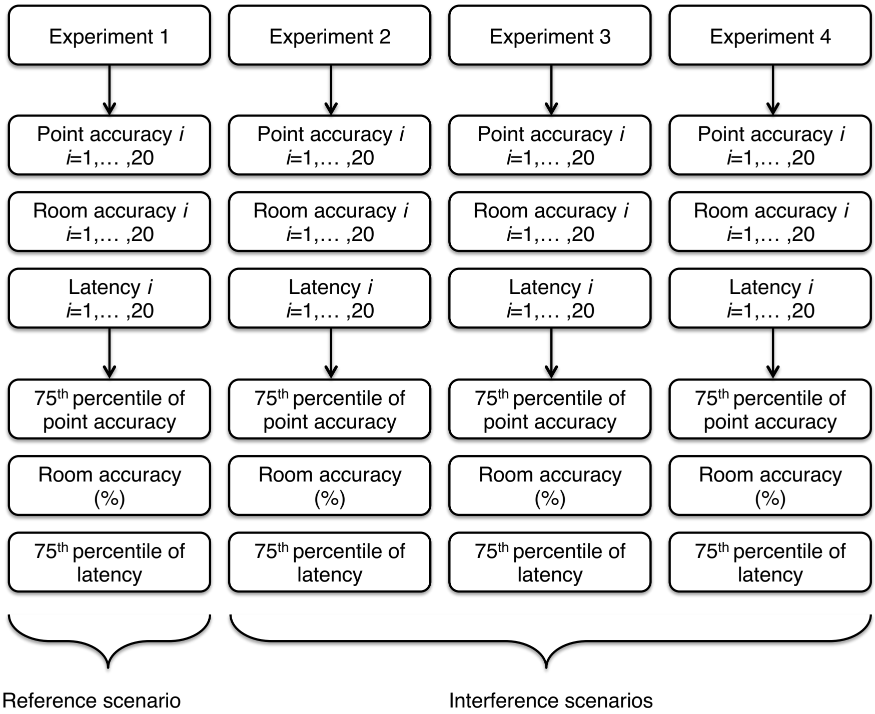


Figure 3.11: Procedure of capturing performance metrics

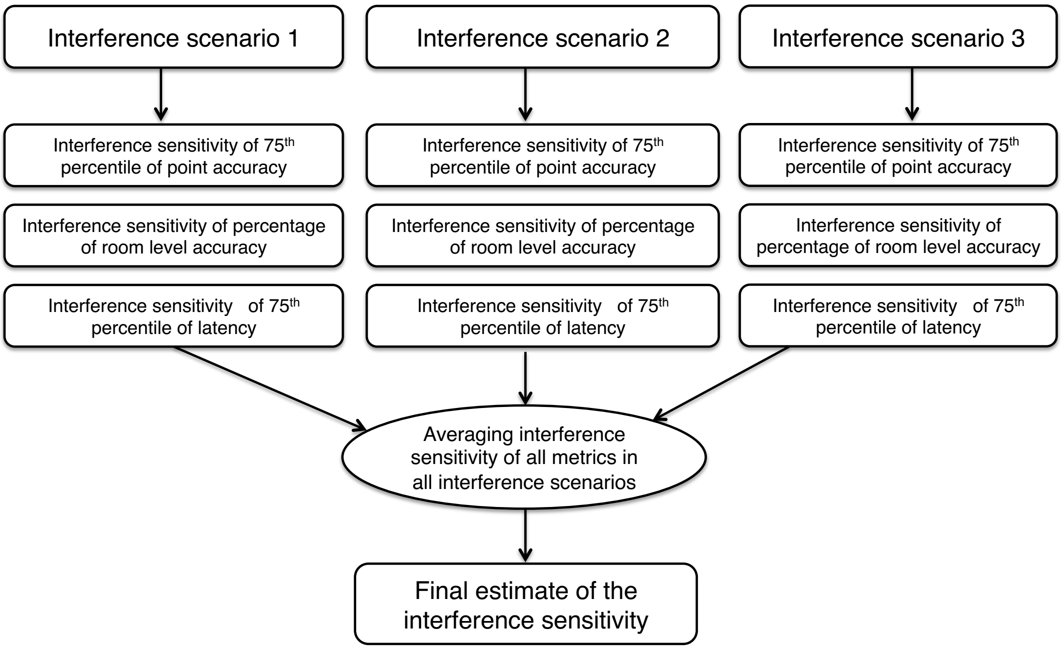

Interference scenario 1
Interference scenario 2
Interference scenario 3
Interference sensitivity of 75th percentile of point accuracy
Interference sensitivity of percentage of room level accuracy
Interference sensitivity of 75th percentile of latency
Averaging interference sensitivity of all metrics in all interference scenarios
Final estimate of the interference sensitivity

Figure 3.12: Procedure of capturing interference sensitivity

In the equation the sum goes over all three interference scenarios ($i = 1, 2, 3$), and $M_1(i)$, $M_2(i)$ and $M_3(i)$ are interference sensitivity of 75th percentile of point accuracy, interference sensitivity of room level accuracy and interference sensitivity of 75th percentile of latency for an interference scenario i, respectively, as depicted in Figure 3.12. The aim of averaging over all interference scenarios is to obtain one value that characterizes the influence of interference on the overall performance of a given solution.

Calculation of Final Scores

The score for each metric was calculated according to a linear function that is defined by specifying minimal and maximal acceptable values for each metric, as depicted in Figure 3.13. Furthermore, weighting factors have been utilized for defining the importance of each metric for a given category. In general, the linear translation function for calculating scores for each particular metric is given in Equation 3.6, with scores ranging from 0 to 10.

$$Score = \max\left(0, \min\left(10, 10\frac{m - M_{acceptable}}{M_{desired} - M_{acceptable}}\right)\right) \tag{3.6}$$

Acceptable and desired values are defined with $M_{acceptable}$ and $M_{desired}$, respectively. Note that $M_{acceptable}$ can be bigger than $M_{desired}$, e.g. in defining the acceptable point accuracy values one can discuss about acceptable localization error margins. Here $M_{acceptable}$ is the biggest acceptable, while $M_{desired}$ is the desired 75th percentile localization error.

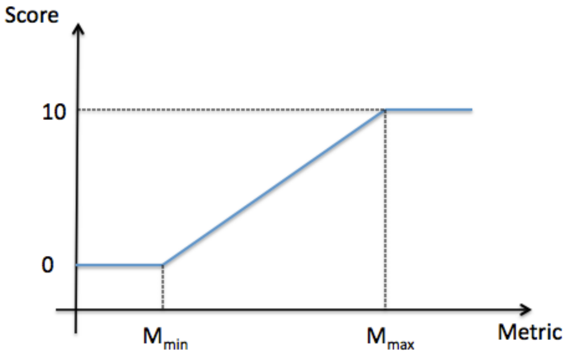


Figure 3.13: Linear translation function for each metric

The scores for each metric are weighted using predefined weighting factors and are summed together to produce the final score for a particular category, as shown in Figure 3.14. The winners of the competition were declared based on the final scores for the three different sets of marginal values and weights, as presented in Table 3.2. The desired values for different metrics have been selected with the intuition of satisfying the majority of use-cases for usage of location information, as discussed e.g. in [139, 11]. The acceptable values have been selected very generously with a practical purpose of not discouraging the competitors (i.e. getting a score of zero is something we wanted to avoid if possible). The weighting factors for different categories have been selected to emphasize the performance of the evaluated algorithms along a certain metric, however hot to disregard entirely other performance metrics. For example, in Category 2 the focus is on the latency of different algorithms, but some smaller weights are also given to other metrics to discourage competitors in reporting of random locations very fast.

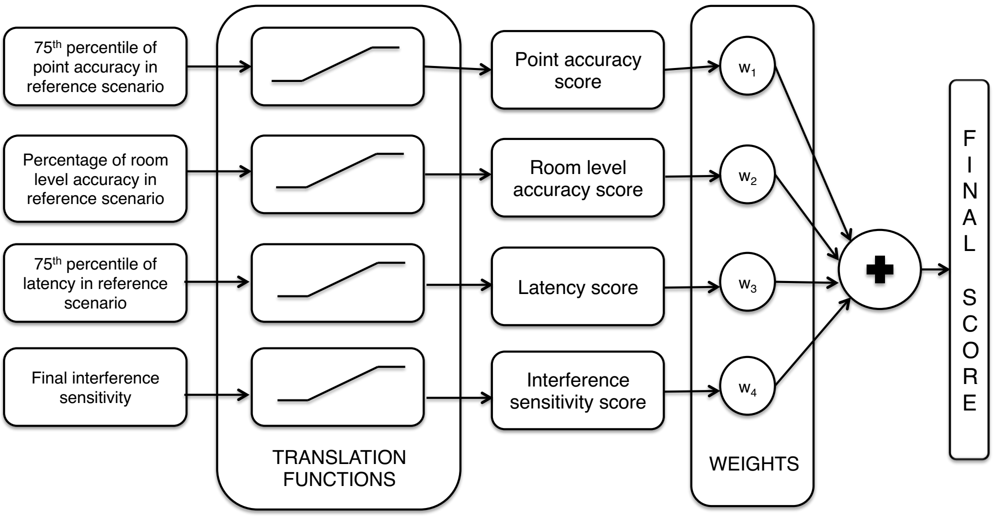

75th percentile of point accuracy in reference scenario
Percentage of room level accuracy in reference scenario
75th percentile of latency in reference scenario
Final interference sensitivity
TRANSLATION FUNCTIONS
Point accuracy score
Room level accuracy score
Latency score
Interference sensitivity score
w1
w2
w3
w4
WEIGHTS
FINAL SCORE

Figure 3.14: Calculation of final scores

Table 3.2: Marginal values and weights in different categories

Metric	$M_{acceptable}$	$M_{desired}$	Category 1 weight	Category 2 weight	Category 3 weight
Point accuracy	10 m	1 m	0.40	0.20	0.20
Room level accuracy	50 %	90 %	0.40	0.20	0.20
Latency	20 sec	1 sec	0.10	0.50	0.10
Interference sensitivity	50 %	10 %	0.10	0.10	0.50

3.2.2 Evaluation Scenarios

This section shortly presents the evaluation scenarios that were generated using the RF interference generation capabilities of the testbed. These interference patterns have been designed to capture the effects of different technologies (i.e. IEEE 802.11 and IEEE 802.15.4) on the performance of the evaluated localization algorithms. The interference scenarios have also been designed to reflect different common coexistence scenarios: one scenario in which there is a MAC protocol level mechanism supporting coexistence and 2 scenarios without coexistence support. Other interference patterns can be generated as well, as discussed previously in the thesis. Figure 3.15 gives an example spectrum dump for different evaluation scenarios taken by a WiSpy device at the location indicated in Figure 3.16.

Reference Scenario: In this scenario no artificial interference was generated and the presence of uncontrolled interference was minimized. Thus, the results achieved in this scenario are used as the reference for evaluating the impact of interference in the other scenarios.

Interference Scenario 1: In the first interference scenario the interference was created using IEEE 802.15.4 Tmote Sky nodes. The interference type was jamming on one IEEE 802.15.4 channel with a constant transmit power of 0 dBm. Five of these jamming nodes were present in the testbed, as shown in Figure 3.16. The channel on which the jamming was performed was selected such as to overlap with the channel used by a particular SUT. The summary of this interference scenario is given in Table 3.3.

Table 3.3: Summary of the interference scenario 1

Number of sources	5
Power	0 dBm
Waveform	Carrier jamming
Start & stop time	Beginning & end of experiment
Traffic model	IEEE 802.15.4 radio

Interference Scenario 2: The second interference scenario was comprised of several interference sources that are typical for office or home environments. Interference was emulated using 4 WiFi embedded PCs (AlixD2D) having the roles of a server, access point, data client, and video client. The interference transmission streams are schematically depicted in Figure 3.16. During this scenario, the server acted as a gateway for the emulated services. The data client was represented by a TCP client continuously sending data over the AP to the server. Similarly, the video client was emulated as a continuous UDP stream source of 500 kbps over the available bandwidth of 50 Mbps. The AP was working on a WiFi channel overlapping with the SUT's operating channel and with the transmission power set to 20 dBm (100 mW). The summary of the described interference scenario is given in Table 3.4.

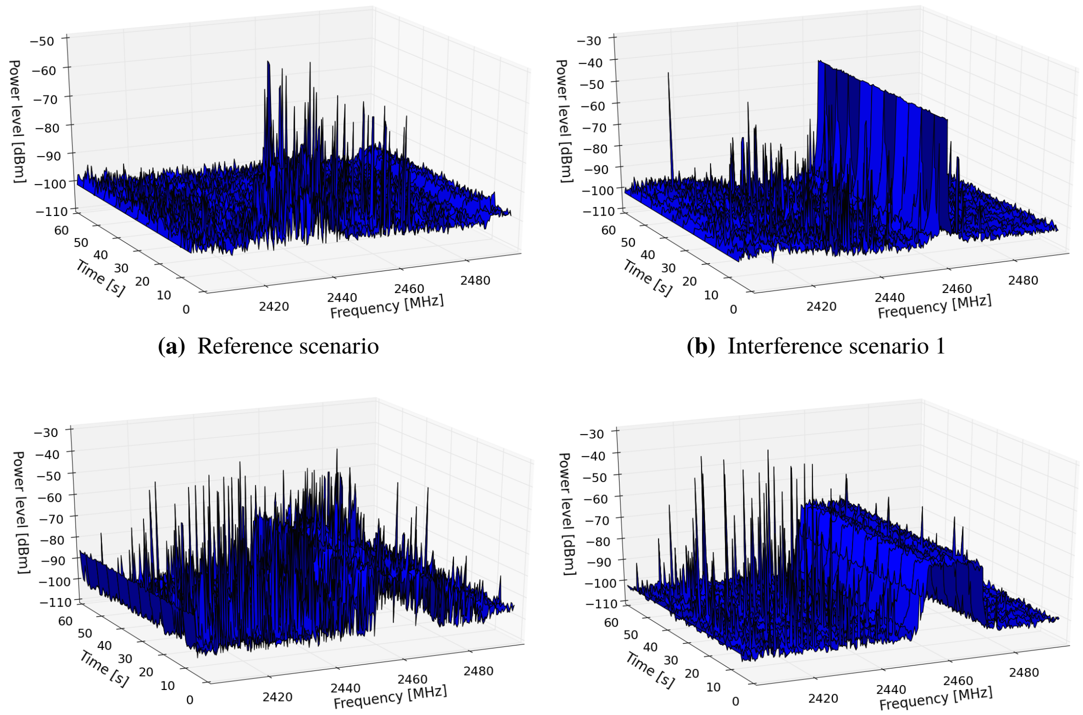
Table 3.4: Summary of the interference scenario 2

Number of sources	3
Power	20 dBm
Waveform	WiFi signals
Start & stop time	Beginning & end of experiment
Traffic model	IEEE 802.11 traffic

Interference Scenario 3: For the third interference scenario, a signal generator, with location given in Figure 3.16, was used to generate synthetic interference with an envelope that reflects WiFi modulated signals, but without Carrier Sensing (CS). The transmission power was set to 20 dBm, while the wireless channel on which the interference was performed depended on a particular evaluated SUT. The summary of the interference scenario 3 is given in Table 3.5.

Table 3.5: Summary of the interference scenario 3

Number of sources	1
Power	20 dBm
Waveform	Power envelope
Start & stop time	Beginning & end of experiment
Traffic model	Synthetic IEEE 802.11 traffic



(a) Reference scenario

(b) Interference scenario 1

(c) Interference scenario 2

(d) Interference scenario 3

Figure 3.15: Spectrum information for all evaluation scenarios

3.2.3 Execution of Evaluation Experiments

This section describes how the previously described testbed infrastructure was used in the evaluation experiments in the scope of the competition. All evaluation scenarios were instantiated on the 2^{nd} floor of the testbed, and for all of them the same 20 evaluation points were defined, with their locations depicted in Figure 3.16. At each evaluation point, the indoor localization SUT was requested to estimate the location. The SUT device was positioned at each evaluation point using the robotic mobility platform, at the height of 45 cm above the floor. The navigation stack of the robotic platform gives an order of magnitude more accurate location estimation than the usual SUTs, i.e. less than 15 cm in average in the evaluation environment, and due to that the location obtained from the robotic platform was considered as ground truth for the evaluation. The communication with the robotic platform is done in the 5 GHz ISM band, in order to avoid interfering with the evaluated systems, which all performed in the 2.4 GHz ISM band. The experiments were performed during the weekends afternoons, minimizing the influence of uncontrolled interference. Furthermore, the wireless spectrum was monitored using a WiSpy device attached to the robotic platform and another one at a control point in the testbed, used for assessing the level of uncontrolled interference and validating the correctness of generated controlled interference.

SUT Mobile Nodes: Different devices were used as mobile parts of SUTs, i.e. TelosB sensor node, Apple MacBook Pro laptop, Nexus S Android smartphone and Nexus 7 Android tablet. Users were able to use Secure Shell (SSH) tunneling to the desired nodes to deploy their algorithms on a desired device.

SUT Infrastructure Nodes: As infrastructural parts of SUTs, nodes from the wireless sensor network testbed or WiFi APs were used, depending on the requirements of a particular algorithm. The locations of the available infrastructure nodes were communicated to the competitors in advance.

Autonomous Mobility Platform: The mobility platform was accessible over a web interface where competitors were able to click on the location on which they wanted to position the robotic platform and their SUT. The competitors were able to send the platform to a location by setting the coordinates of a desired location. Also, it was possible to provide a set of way-points to the platform for a full automation of even the training phase of different algorithms. The platform was able to provide its current location or adequate messages if the desired location was not reachable.

Interference Generation: For training and parametrization purposes of their algorithms, competitors were also able to generate the interference scenarios using the previously discussed interference generation devices. The code for generating three previously described interference scenarios was provided and the users were able to select the nodes on which the code should run.

Interference Monitoring: Competitors were also able to use different devices for monitoring interference levels. Moreover, they were able to obtain the dumps of wireless spectrum using the WiSpy device on the robot or at the fixed location given in Figure 3.16.

Interfacing with the SUT: All competitors had to deploy their algorithms on one of the devices intended for deploying SUTs. Furthermore, competitors had to provide an HTTP URI on which their algorithm listens for location estimation requests. Upon a request, the algorithms needed to provide the location estimate as a JSON response in the previously specified format. JSON parameters *coordinate_x* and *coordinate_y*, expressed in meters, are required parameters and as such they had to be reported. Parameter *coordinate_z* is an optional parameter, due to the 2D evaluation environment. Finally, parameter *room_label* is an optional parameter that is either explicitly provided by the SUT, or automatically mapped from the estimated coordinates x and y. Coordinates (x, y) or (x, y, z) of the location estimates had to be calculated according to a predefined *zero-point* in the environment.

3.2.4 Evaluation Results

This section presents results of the evaluation performed in the scope of the competition. We evaluated 7 different indoor localization algorithms in four evaluation scenarios, resulting in 28 experiments in total. The short descriptions of the evaluated algorithms are given in the Appendix A of the thesis. We firstly present localization errors cumulatively over all competitors at each evaluation point, to show the spatial variability of localization errors due to a position in the environment and due to the RF interference generated in each interference scenario. The points are labeled from 1 to 20, as shown in Figure 3.16.

Secondly, we present the ranking of evaluated algorithms for three categories, each emphasizing different indoor localization performance metric, according to Table 3.2. In the table we present, in our view, only the most interesting results. A more extensive set, containing results for each competitor and detailed statistics for each metric, is given in [140].

Point Accuracies in Different Evaluation Scenarios

Localization errors per evaluation points for all scenarios are given in the regular box-plot fashion in Figure 3.17. In each group of box-plots, i.e. for each evaluation point, the first box-plot in a group depicts the localization errors obtained in the reference scenario, followed by the localization errors obtained in the interference scenarios 1, 2, and 3, respectively. Higher localization errors are generally achieved close to the borders of the environment, i.e. outside walls, for example in evaluation points 6, 10, 17 and 20. Consequently, in the center of the environment smaller localization errors are achieved, e.g. 2, 3, 9 and 14.

In the scenarios where controlled interference was generated in order to evaluate the influence of different RF interference patterns on the performance of indoor localization algorithms, higher localization errors are generally obtained. In the interference scenario 1, distributed jamming on one IEEE 802.15.4 channel, the achieved localization errors averaged over all competitors at each evaluation point are given in Figure 3.17. As it can be seen in figure, this interference scenario significantly degrades the averaged point accuracy of evaluated SUTs. Particularly interesting is to observe the tendency of achieving higher localization errors in the evaluation points close to the sources of interference, i.e. points 1, 16, 17, 18, 19 and 20.

Significantly smaller influence on the accuracy of localization is achieved in the interference scenario 2. However, the influence at some evaluation points is still visible, as shown in Figure 3.17. One reason for lower impact in this scenario on the averaged accuracy of localization is the fact that in this scenario WiFi traffic was generated, i.e. CS was enabled. In other words, the results achieved in the interference scenario 2 confirm the value of MAC coexistence as important mechanism for reducing the impact of interference, even for localization applications.

Finally, in the interference scenario 3, i.e. jamming on one IEEE 802.11 channel, the averaged accuracy of all evaluated SUTs is again significantly degraded compared to the reference scenario, as shown in Figure 3.17. Similarly to the interference scenario 1, the averaged localization errors are higher in the evaluation points close to the interference source, e.g. 19 and 20.

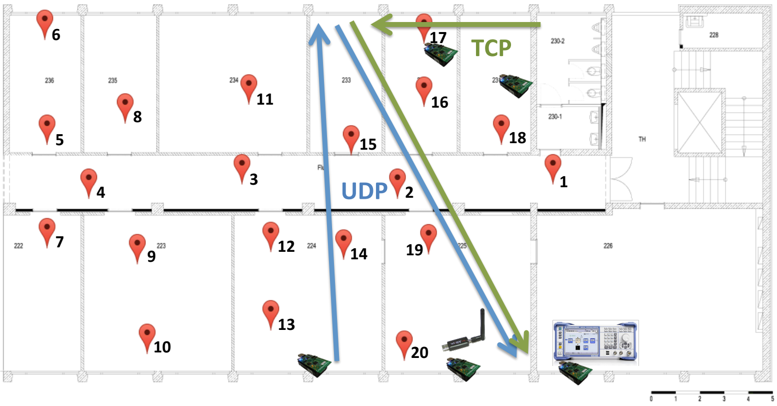


Figure 3.16: Locations of interferers and evaluation points

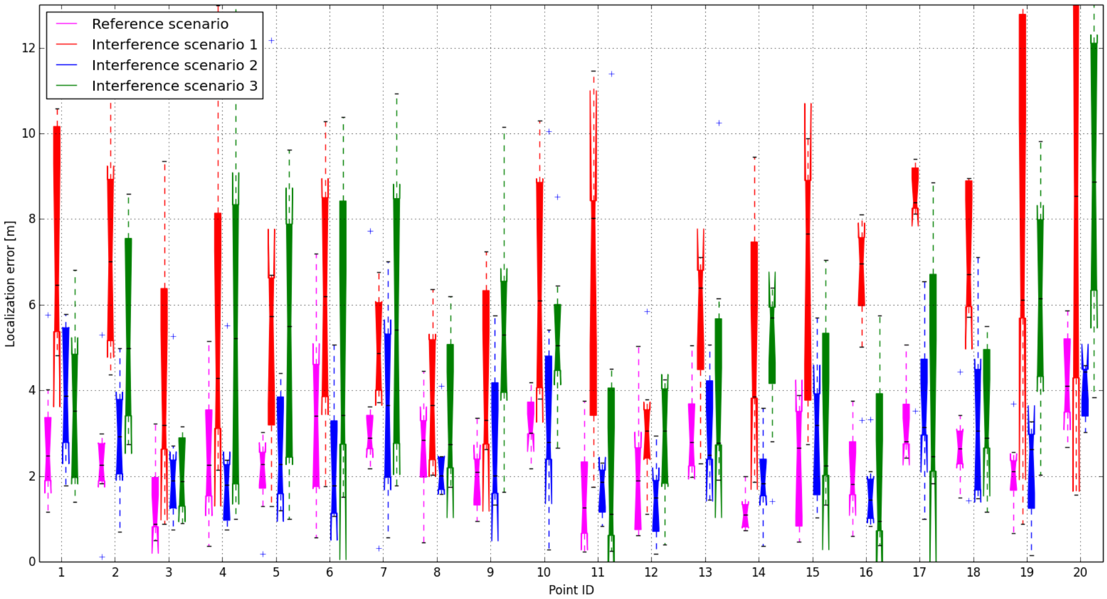


Figure 3.17: Localization errors per evaluation point for all evaluation scenarios

Ranking of the Competing Algorithms

The ranking of the competing algorithms with respect to each other for the three categories is given in Table 3.6. First category focuses on the point and room level accuracy of indoor localization by giving the highest scores to these metrics. The best performance is achieved by the fingerprinting-based algorithm “Indoor Geolocation for Android Smartphones with Airplace 1”, achieving average localization error of only 1.77 m and room accuracy of 80% in the reference scenario, and final score of 7.13/10. Note that the point accuracy in the table is the 75 percentile localization error, which was the metric used for calculating the scores.

In the second category, emphasizing the response time of evaluated algorithms, the best performance is again achieved by the fingerprinting-based algorithm “Indoor Geolocation for Android Smartphones with Airplace 1”, as shown in Table 3.6. Although this algorithm achieves the 75 percentile latency of 3.07 sec in the reference scenario, which is not the best latency result among the evaluated algorithms, due to the performance in other metrics this algorithm is a winning one with final score of 7.58/10. The best performance in terms of 75 percentile latency, namely only 0.01 sec, is achieved by the algorithm “Geo-n Localization”, based on low-power sensor nodes.

Finally, the ranking of competitors in the third category, emphasizing the interference sensitivity in different interference scenarios, is given in Table 3.6. The winning algorithm is the fingerprinting-based algorithm “Indoor Geolocation for Android Tablets with Airplace 3”. Although the other performance metrics achieved by this algorithm in the reference scenario are average, compared to other evaluated algorithms, the change in the metrics due to different controlled RF interference patterns is really small, resulting in the best score (6.71/10) in this category. In summary, the final scores of all competitors indicate that specific types of RF-interference noticeably degrade the performance of evaluated algorithms.

3.2.5 Lessons Learned

This section shortly summarizes the lessons learned while executing the experiments in the scope of the competition.

Remote Usage of Testbed Infrastructure: It is clearly a benefit to use a remotely accessible testbed infrastructure for experimentation with RF-based indoor localization algorithms, since it reduces the costs of visiting the testbed environment and simplifies the execution of experiments. The simplification of the execution of evaluation experiments is due to the developed support for generating and monitoring RF interference, and controlling the mobility platform, so the experimental overhead on the user is significantly reduced. The presented evaluation experiments were executed without the presence of a test-person carrying a localized device, thus increasing the comparability of algorithms and objectiveness of achieved results. In our evaluation experiments all SUT nodes were positioned with the same orientation and on the same height, with average error in positioning smaller than 15 cm, which would be hardly achieved by a test-person. Obviously, by not having a test-person we could not capture the effects that that person could have on the performance of the evaluated algorithms.

Except for benefits in terms of increased comparability of evaluation results, using the automated infrastructure for evaluation significantly improves the time needed for performing a evaluation experiment. Namely, one experiment as presented here, i.e. surveying 20 evaluation points, took roughly 20 minutes, and during the execution results were automatically stored and metrics were calculated and reported. Only one testbed operator was necessary during the automated experimentation, and this only for the support purposes, e.g. robotic platform unable to avoid an obstacle, closed doors, drained battery of a SUT mobile node, etc.

An interesting observation is that all algorithms evaluated in the scope of the competition used de-facto the same type of raw input data for location estimation, namely the RSS signal feature. In more general terms, RSS is a widely adopted signal feature in RF-based indoor localization, mostly because of the simplicity of obtaining this feature from the received signals. Since a

Table 3.6: Summarized results for all categories

SUT	Point acc. [m]	Room acc. [%]	Latency [s]	Interference sens. [%]	Point acc. score	Room acc. score	Latency score	Interference sens. score	Final score
				Category 1					
Airplace 1	2.71	80.00	3.07	61.43	8.10	7.50	8.91	0.00	7.13
Quantile	3.87	70.00	20.11	28.09	6.81	5.00	0.00	5.48	5.27
Geo-n	2.64	60.00	0.48	62.50	8.18	2.50	10.00	0.00	5.27
3CoM	2.85	55.00	0.01	37.30	7.94	1.25	10.00	3.17	4.99
Airplace 3	3.77	50.00	3.85	14.17	6.92	0.00	8.50	8.96	4.51
RSS Range	3.75	45.00	0.02	42.75	6.94	0.00	10.00	1.81	3.96
Airplace 2	3.38	50.00	3.51	99.48	7.35	0.00	8.68	0.00	3.81
				Category 2					
Airplace 1	2.71	80.00	3.07	61.43	8.10	7.50	8.91	0.00	7.58
3CoM	2.85	55.00	0.01	37.30	7.94	1.25	10.00	3.17	7.16
Geo-n	2.64	60.00	0.48	62.50	8.18	2.50	10.00	0.00	7.14
RSS Range	3.75	45.00	0.02	42.75	6.94	0.00	10.00	1.81	6.57
Airplace 3	3.77	50.00	3.85	14.17	6.92	0.00	8.50	8.96	6.53
Airplace 2	3.38	50.00	3.51	99.48	7.35	0.00	8.68	0.00	5.81
Quantile	3.87	70.00	20.11	28.09	6.81	5.00	0.00	5.48	2.91
				Category 3					
Airplace 3	3.77	50.00	3.85	14.17	6.92	0.00	8.50	8.96	6.71
Quantile	3.87	70.00	20.11	28.09	6.81	5.00	0.00	5.48	5.10
3CoM	2.85	55.00	0.01	37.30	7.94	1.25	10.00	3.17	4.43
Airplace 1	2.71	80.00	3.07	61.43	8.10	7.50	8.91	0.00	4.01
RSS Range	3.75	45.00	0.02	42.75	6.94	0.00	10.00	1.81	3.29
Geo-n	2.64	60.00	0.48	62.50	8.18	2.50	10.00	0.00	3.14
Airplace 2	3.38	50.00	3.51	99.48	7.35	0.00	8.68	0.00	2.34

large number of RF-based localization algorithms share the same input data, offering the raw RSS could potentially be used for a very fast, simple, and highly objective evaluation of different RSS-based localization solutions. Additionally, offering the collected raw RSS would in the same fashion allow optimization of the parameterization of various RSS-based solutions. This idea is further investigated in the following section of this work.

Results of Evaluation: The evaluation results show good performance of the evaluated algorithms, with the best performance being 1.77 m in average localization error, 80 % in room level accuracy and less then 1 sec in response time in the reference scenario. WiFi-based fingerprinting algorithms showed the best results in all three categories, suggesting that such algorithms are promising and worth further investigation. Interestingly, due to a fairly dense deployment of low-power sensor nodes, the proximity-based algorithm performed unexpectedly well. This indicates that, with enough dense deployment, similar technologies such as iBeacon based on

Bluetooth LE can be adequate and simple to use for indoor localization purposes.

The spatial distributions of localization errors show that generally higher errors seem to be achieved at the margins of the environment, i.e. closer to the outside walls. This is most pronounced for the algorithms based on low-power sensor nodes, due to the larger number of reachable infrastructural nodes in the center of the evaluation environment. Presumably due to the higher transmission power of WiFi, this trend is not to the same level emphasized in fingerprinting-based algorithms, i.e. such algorithms generally provide smaller spatial variabilities of localization errors, which is generally a desired feature of localization solutions, again suggesting that WiFi-based fingerprinting is worth further investigation.

Obtained results show that certain interference patterns highly influence the performance of the evaluated algorithms, which motivates further in-depth analysis on how to mitigate the observed negative impact. Specifically, jamming as type of interference has a strong impact on all algorithms, while the usual interference mitigation mechanisms help avoiding the effects of normal WiFi traffic as interference. Spatial distributions of localization errors in the interference scenarios indicate that higher errors usually occur in locations closer to the sources of interference, which is specially emphasized for the algorithms based on the low-power sensor nodes.

3.3 Platform for Evaluation of RF-based Indoor Localization Solutions

Using RSS feature of RF signals is one of the most popular approaches for indoor localization, as illustrated by the numerous works done in this area, with [30, 141, 142] being just a few of many examples. Although rapid progress has been achieved in the baseline performance of these algorithms, their practical application still requires extensive experimentation efforts for tuning the performance at the specific conditions of the deployment scenarios and environments, which are associated with high time and cost overheads.

We aim at reducing these hindrances by providing: (i) detailed public datasets of raw RSS data that can be used as input to RSS-based indoor localization algorithms, collected in a wide range of conditions (different types of devices, different types of interference, different environments) and (ii) a web-based platformfor streamlined usage of these datasets for the purpose of performance evaluation of indoor localization algorithms.

The provided platform supports evaluation of RF-based indoor localization algorithms without the need for experimentation on dedicated physical experimental infrastructure. As shown in Figure 3.18, the platform encompasses two core services. The first one is a service for storage and management of raw measurement data that can be used as an input to the localization algorithms that the experimenters want to evaluate. The second service offers automated calculation of a standardized set of indoor localization performance metrics like localization accuracy, response time, and power consumption. The platform includes a tool for visualization of the raw data that supports experimenters in the process of reviewing the offered datasets. We also provide a set of Software Development Kits (SDKs) for Python and MATLAB, providing functions for streamlined access to raw data and for calculation of performance metrics, which further simplifies the evaluation process.

From the experimenter's perspective, with the help of the raw data visualization tool, the experimenter is able to visualize and “browse” the available datasets. After selecting the desired raw dataset, the experimenter can use it as input to an algorithm to be evaluated. Based on the results of the performance evaluation, the experimenter is able to parametrize or fine-tune the algorithm for improving its performance.

The currently offered datasets are collected on three testbeds, giving experimenters the possibility to evaluate their algorithms in diverse environments. The raw data that can be used as input to the evaluated algorithms is collected under different scenarios, with different densities of evaluation points, using different hardware and with different sources of controlled RF interference, providing a wide range of experimentation possibilities. This data was collected with the help of the previously discussed automated testbed infrastructure.

3.3.1 Platform Overview

This section presents the design goals, as well as the design and implementation of the platform for virtual evaluation of RF-based indoor localization algorithms.

Design Goals

Main requirements for the platform for virtual evaluation of RF-based indoor localization algorithms are the simplicity of usage for the experimenters, extensibility, reliability, fast data flow, remote usage, and programming language and platform independence.

Usability: The platform should allow for easy scoping and filtering of the raw data, so that experimenters can selectively request a record at a specific location coordinates and for a given technology, and consequently get the desired dataset in an efficient way. The advantage of that approach, in comparison to plain “downloading” alternative, lies in the fact that experimental raw datasets for evaluation of indoor localization algorithms can be very large. Especially for “universal” datasets that can be used for evaluation of different localization algorithms, the aim is to collect data at high spatial sampling densities and using diverse hardware equipment. But any particular algorithm would likely use only a small subset of this data in a given evaluation campaign. The approach of disseminating the whole raw data sets as simple “downloadable” archives is thus very inefficient and leaves to the experimenter the burden of local filtering. The alternative that we aim at offering, an online service for accessing and managing of this data, is much more convenient for the experimenters and reduces the requirements for local computational resources.

Extensibility: One of the main concerns for the development of the service for management of the raw data is extensibility. Although at the moment the platform predominantly offers RSS features of RF signals (as in more details discussed in the following sections), we envision experimenters contributing with their raw datasets that potentially contain other types of raw data useful for indoor localization purposes. We therefore aim at developing a service where experimenters are able to add or remove parameters to the data records according to their specific needs. Furthermore, the overhead of storing these customized data records into the database should be minimal and should not require complex database or table/schema adaptations.

Fast and reliable remote access: We aim at a cloud-based deployment of platform which will support public remote access for a large number of users, without the need of downloading and running the whole platform on a local machine. This kind of deployment is expected to provide simple, efficient, and reliable access to the platform for the end users.

Language and platform independence: Finally, the design goal is to make the platform available for various programming languages and operating systems. In other words, in order to make the platform usable for a large number of experimenters with expertise in different software platforms and languages, we aim at developing it so that it is independent of a language and platform used by the experimenter.

Platform Design

The overview of the platform indicating the relation among individual components is given in Figure 3.18. The platform consists of a raw data storage service used for storing the raw data that can be inputed into an indoor localization algorithm to be evaluated. The platform further provides a raw data visualization tool, that enables experimenters to easily visualize collected information stored in the provided measurement collections. Furthermore, the experimenter is able to fetch and use a desired raw dataset though a set of software SDKs. Finally, a set of standardized metrics for characterizing the performance of an algorithm can be obtained through one function call to a metrics calculation service.

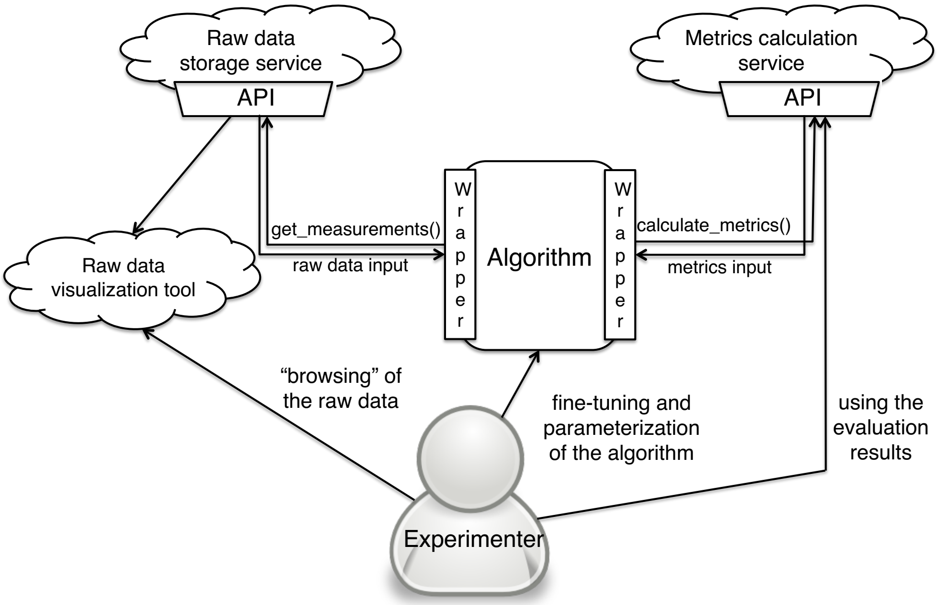

Raw data storage service
API
Metrics calculation service
API
Raw data visualization tool
get_measurements()
raw data input
Wrapper
Algorithm
Wrapper
calculate_metrics()
metrics input
"browsing" of the raw data
fine-tuning and parameterization of the algorithm
using the evaluation results
Experimenter

Figure 3.18: Components of the cloud-based evaluation platform

The raw data storage service stores collected measurements that can be used for indoor localization evaluation purposes. The measurements are stored together with the locations where they are taken, annotated with the locations of transmitting devices, the metadata describing the environment and the hardware used for collection of the raw data. The service provides a publicly available API for managing the stored data, where the experimenter can "search" the stored datasets and select a desired one.

The platform further consists of a set of software SDKs or wrappers developed for Python and MATLAB programming languages, due to their popularity in rapid prototyping of algorithms, including those for indoor localization. The experimenter can use the wrappers to fetch measurements through a single function call. The experimenter is then able to input the fetched data to an algorithm to be evaluated. The SDKs provide interaction with the cloud service for the data storage as shown in Listing 3.2. Using the *get_measurements* command, the experimenter is able to fetch the data from an experiment or a specific measurement. With the *filtering* command, it is possible to filter the fetched data based on desired parameters, such as number of measurements, wireless channel or transmitting device.

The output of an algorithm, *i.e.* estimated locations, and the matching ground-truth coordinates where the measurements were taken, can be passed to a metrics calculation service using the *calculate_metrics* command provided by the wrappers. By doing that the experimenter is able to obtain a standardized set of performance metrics, allowing him to easily evaluate the performance of an algorithm, using the experimentally collected data as the input and receiving a set of standardized metrics as the output of the procedure.

Listing 3.2: Important Python SDK functions

```
def get_measurements(database ,experiment ,measurement);
def filtering(measurement,num_meas,channel ,sender);
def calculate_metrics(data);
```

Platform Implementation

As described previously, the main components of the platform are the raw data storage service, service for calculation of the performance metrics, a set of SDKs for simple interaction with the services using Python and MATLAB programming languages, and a visualization tool for convenient data browsing. While the other components provide straightforward functions and simplifications for the experimenters, due to its central role, the implementation of the raw data storage service is worth explaining in more detail. The raw data storage service is implemented as a web service developed in Python 2.7 using the Flask module. The Flask module provides a simple way of creating RESTful web services. We leveraged this to develop multiple functions for supporting raw data storage and management. Raw data is stored in a MongoDB database, an open-source document database and the leading Not only SQL (NoSQL) database written in C++. It can store JSON based messages using Binary JSON (BSON) format of data. Our data messages are defined as Protocol Buffer structures, a way of encoding structured data using an efficient and extensible binary format. The service we developed, together with the MongoDB database, is running on a EC2 instance in AWS. Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides scalability and platform independence of our services. The experimenter is able to communicate with the services through properly defined HTTP requests.

The design goal of extensibility of the platform was achieved by using Protocol Buffer (version 2) for defining message types and MongoDB for storing those messages. In Protocol Buffers, each data structure that needs to be encoded is encapsulated in the form of a message. The specification of the internal structure of the messages is done in special protocol files that have the *.proto* extension. The specification is performed using a simple, but powerful, domain specific data specification language that allows easy description of the different message fields, their types, optionality, etc. Using the Protocol Buffer compiler *protoc* the *.proto* specification files can be compiled to generate data access classes in number of languages like C++, Java and Python. These classes provide simple accessors for each field (like query() and set_query()) as well as methods to serialize/parse the whole structure to/from raw bytes. NoSQL databases employ less constrained consistency models than traditional relational databases. By using a NoSQL or schema-less type of database, *i.e.* MongoDB, the service enables the storage of any type of defined message, without the need of changing the code and/or the database itself.

RESTful web services enable remote access to the data using simple HTTP requests. Protocol Buffers serialize messages into binary streams which support fast communication between the experimenters and the platform. RESTful web service as a part of service for management of the raw data provides remote access to the service. High availability is supported by running the service in the Amazon cloud. Finally, the fast data flow from experimenter to database is achieved by using Protocol Buffer serialization and MongoDB schema-less database for storing the binary data.

With the Protocol Buffer compiler, message specification files (*.proto*) can be compiled to generate data access classes in various programming languages, e.g. C++, Java, JavaScript, and Python. Due to the fact that communication with the cloud is done using HTTP requests, it is possible to manage data from different users' platforms, and also using different programming languages (all modern programming languages provide support for HTTP requests). To conclude, Protocol Buffers and a RESTful web service allow managing raw data from different platforms and programming languages.

Data Structure

NoSQL databases usually provide the following hierarchy. On the top level is a set of databases. Each database consists of a number of collections (the equivalent to tables in relational databases). Finally, each collection consists of a set of messages which contain data. We follow the same principle in our raw data storage service. The data storage hierarchy of the raw data storage service is given in Figure 3.19. The database for storing the raw data from the indoor localization experiments consists of a set of collections (called "Experiments" in our case), where a collection stores all raw data from one round of measurements, *i.e.* from one measurement survey. Each collection consists of messages that are related to physical locations where the measurements are taken, where each location is defined with coordinates (x, y, z) in relation to a predefined zero-point. Besides the database for storing raw data, there is also a database for storing the metadata describing the raw data. The explicit split between the data and metadata is done to reduce the amount of data that has to be fetched from the platform, *i.e.* in case where a lot of data shares the same metadata, the metadata has to be fetched only once and applied to all the data. Also, the explicit split enables reusage of one metadata message for multiple repetitions of the same measurement survey, which decreases the redundancy in the stored information. This database contains descriptions of different experiments or measurement surveys performed for collecting the raw data.

A snapshot of the raw data in JSON representation is given in Listing 3.3. This sample data consists of the RSSI measurements, information about the measurements (timestamp, transmitter ID, run number, etc.) and of the location of the transmitting and receiving devices. Alternatively, instead of storing RSSI values, it is also possible to store ToF, AoA or Link Quality Indicator (LQI) measurements, or any combination of those measurements.

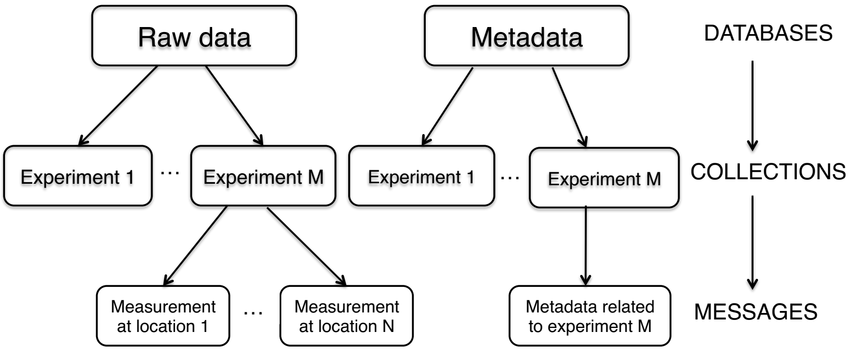


Figure 3.19: Hierarchy of the raw data storage service

Listing 3.3: Raw data format

```
{
 receiver_id: "MacBook Pro",
 run_nr: 1,
 timestamp_utc: 1373126790,
 sender_id: "tplink08",
 sender_bssid: "64:70:02:3e:aa:11",
 rssi: −42,
 channel: "11",
 receiver_location: {
   room_label: "FT226",
   coordinate_z: 9.53,
   coordinate_y: 1.67,
   coordinate_x: 23.9},
 sender_location: {
   room_label: "FT226",
   coordinate_z: 10.9,
   coordinate_y: 0.7,
   coordinate_x: 31}
}
```

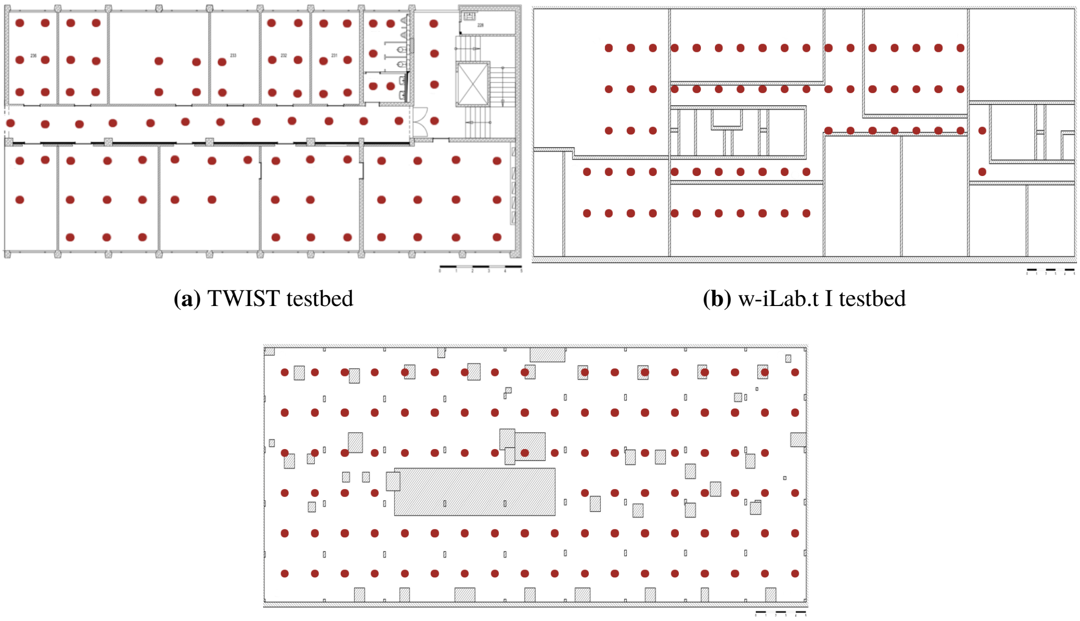
3.3.2 Overview of Available Datasets

The current datasets primarily consist of IEEE 802.11b/g and IEEE 802.15.4 RSSI measurements collected using different devices at different locations in various environments and scenarios. In addition, for some environments the datasets also include IEEE 802.15.4 ToF and IEEE 802.15.4 LQI measurements. A device for collecting the measurements has been carried to each measurement point using an autonomous mobility platform, which is a robotic mobility platform that can carry a system for collecting the raw data. In the following we overview the publicly available datasets and discuss how they help to solve open questions in RF-based indoor localization.

Diversity of environments: One open research question in RF-based indoor localization is related to having a solution that provides reasonably accurate and real-time performance in diverse environments [143, 144]. The provided measurements collected in various environments give experimenters a possibility to evaluate the stability of their solution in diverse conditions, with an aim on solving the aforementioned research challenge. Current datasets have been collected in three different testbeds (TWIST [138] and w-iLab.t I/II [56]). The footprints of the testbed environments are given in Figure 3.20, with the locations of measurement points for the densest measurement surveys indicated with red dots. The first two testbeds represent typical office environments. The measurements were performed in static environments, *i.e.* with no people moving, without introducing additional unaccounted sources of RF interference, etc. The third testbed represents a big open space environment, such as a warehouse or an underground garage. This environment is shielded with no people moving and with effectively no RF interference.

Diversity of densities of measurements points: The measurements have also been collected with different densities of measurement points, which has various advantages. Firstly, some of the indoor localization algorithms, e.g. fingerprinting-based ones, need a training dataset with typically high density. The most dense dataset can in that case be used as the training dataset for the algorithm. Furthermore, it is still unclear how to select the measurement points for the objective evaluation of RF-based indoor localization algorithms [139]. By providing, in terms of realistic possibilities and time constrains, a highly dense set of measurement points, we provide the opportunity of evaluating the influence of the number and locations of measurement points on a large set of algorithms, which could address the mentioned research question. Finally, an open research question than can be addressed with the offered datasets is the trend in spatial distribution of localization errors [140, 145]. For the TWIST testbed environment as an example, the footprints with different densities of measurements points and with measurement points indicated with red dots, are given in Figure 3.21 (measurement points densities of 22.5, 11, and 5 m^2 per point, respectively).

Diversity of interference scenarios: It is shown that RF interference can highly influence the performance of RF-based indoor localization algorithms [99, 140]. There is, however, no work on improving the interference robustness of indoor localization algorithms, and by providing the measurements collected in the environment with high interference, we hope to provide users the possibility of improving the interference robustness of their algorithms. The generated interference scenarios range from a scenario in which no artificial interference is generated and the uncontrolled interference is minimized, to scenarios with various types of RF interference. In the interference scenarios, the interference types vary from jamming using IEEE 802.11 or IEEE 802.15.4 devices to IEEE 802.11 traffic as the interference source.

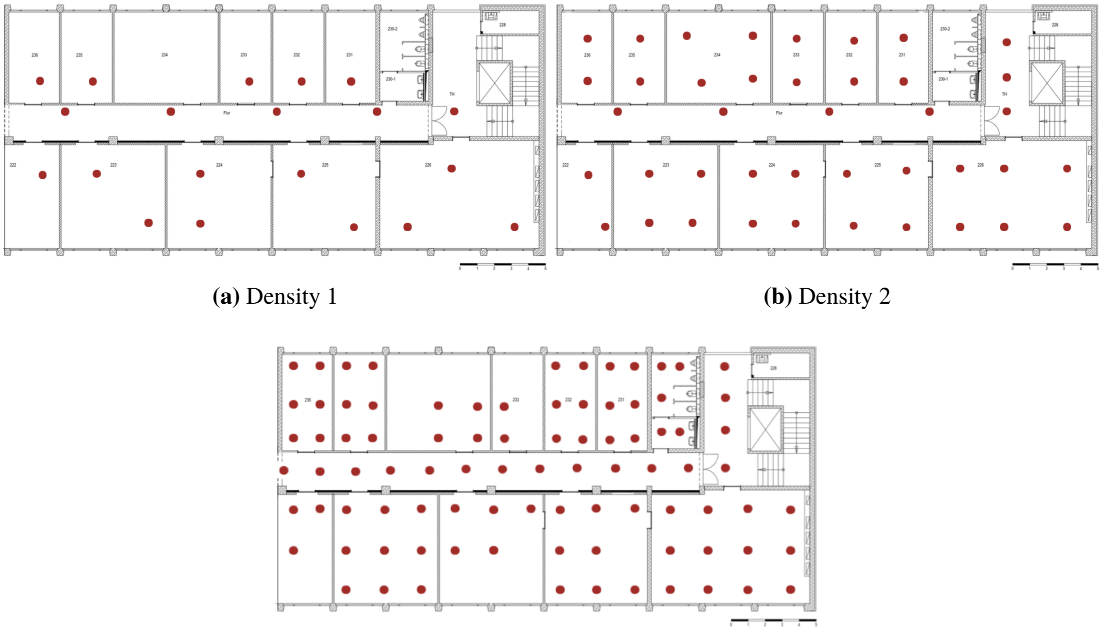


(a) TWIST testbed

(b) w-iLab.t I testbed

(c) w-iLab.t II testbed

Figure 3.20: Different testbed environments used for the collection of raw data



(a) Density 1

(b) Density 2

(c) Density 3

Figure 3.21: Different densities of evaluation points used for the collection of raw data

Diversity of localization devices: While it is known that the device diversity is one of the biggest obstacles in having widely used indoor localization service [146], this problem has not been fully addressed yet. By providing measurements from various devices we hope to provide possibility of improving the performance of RF-based indoor localization in this area. We have collected measurements from two extensively used technologies for RF-based indoor localization, namely IEEE 802.11 and IEEE 802.15.4. The full datasets for different technologies are available for the TWIST testbed environment. In Figure 3.22 we show the locations of controllable IEEE 802.11 (blue squares) and IEEE 802.15.4 (red points) nodes that have been used as anchors in TWIST testbed [138]. In our data collections the measurements from these devices are annotated with with the location of the transmitters. In case of the IEEE 802.11 anchor nodes, we have used six TP LINK 4300 wireless routers, set on WiFi channel 11 and with a transmission power of 20 dBm (100 mW). Apart from these devices, we also stored the data from all other visible APs in the environment, but without their locations. For the IEEE 802.15.4 technology we have been using TelosB nodes deployed in our TWIST testbed as anchor nodes. We used various devices as receivers for obtaining the provided datasets. Namely, for the IEEE 802.11 technology we used the MacBook Pro and Lenovo ThinkPad laptops, Google Nexus 7 tablet and Nexus S smartphone. For the IEEE 802.15.4 technology we have been using the TelosB and stm32w nodes as measuring devices.

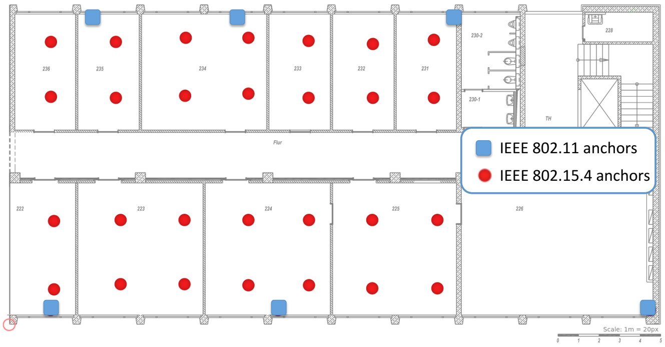


Figure 3.22: Different transmitting devices used for the collection of raw data

3.3.3 Evaluation Results

In order to evaluate the time efforts of using the virtual platform for evaluation of RF-based indoor localization algorithms, as well as the applicability of the collected raw datasets for the evaluation of RF-based indoor localization algorithms, we have executed a set of experiments described in this section. The aim of the evaluation is to demonstrate that the achieved performance metrics of an evaluated algorithm, i.e. the accuracy and delay (response time), do not practically differ in case the proposed platform is leveraged, in comparison to the case when testbed infrastructure is used for evaluation. Secondly, while it is clear that, in comparison to using the infrastructure for experimentation, adopting the proposed platform yields lower-complexity of experimentation for the experimenter, we aim on demonstrating that there is also a significant reduction in time efforts needed for experimentation.

We used a WiFi beacon packets RSSI-based fingerprinting algorithm that uses quantiles of beacon packets RSSI values and PH distance to calculate similarities between training and user-generated fingerprints. The algorithm is in details discussed in the next chapter of this work. Further, we evaluated the performance of the algorithm firstly by using the proposed infrastructure for evaluation of RF-based indoor localization algorithms and secondly by using the raw dataset collected under similar conditions by using the virtual platform.

Since the impact of the "staleness" of recorded data depends on the specific conditions in the environment, in the comparison of performance of the evaluated fingerprinting algorithm we have collected a raw dataset and performed an experimental run using the infrastructure on the same day and in a static environment. The presented results are not intended to prove that recorded data can replace many runs with real hardware, due to changes in the conditions that can occur with time. In other words, the same raw datasets collected in different time snapshots will differ, due to an intrinsic randomness in wireless environment, and for capturing this difference different runs with a real hardware are necessary. On the contrary, the presented results capture realistically the specific conditions in a time snapshot and can serve as basis for fair comparison between the results obtained by using the raw dataset as an input to the algorithm and the results obtained by performing an experimental run leveraging the evaluation infrastructure, i.e. requesting the algorithm to provide an estimate at each evaluation point.

In the experiments using the infrastructure we evaluated the performance of a given algorithm, which was running on a MacBook Pro laptop, in four evaluation scenarios described in Section 3.2.2 with 20 evaluation points for each evaluation scenario, as depicted in Figure 3.21b). Using the platform for virtual evaluation, we selected appropriate datasets in which a MacBook Pro was used as the device for collecting raw data, the raw data was collected in the same 20 evaluation locations, in the same testbed, and in the same four evaluation scenarios. The functions of the Python wrapper were used to fetch the raw data, input it to the algorithm, and to calculate the performance metrics. The delay metric in this case is the sum of the time needed to collect the measurements and the time needed for the processing of the raw data and providing the location estimates.

The summarized results for 20 evaluation points in each evaluation scenario are given in Table 3.7. As indicated in the table, the results of the evaluation using the virtual platform are comparable to the results using the infrastructure specifically designed for evaluation purposes. Small differences in the localization errors and room accuracies are caused by an unavoidable randomness in wireless environments.

From the time perspective, the evaluation using the virtual platform takes only a few (tens of) seconds, once the evaluation chain (fetch the raw data - input the data to an algorithm - request metrics - get metrics) is set. In comparison, one run of an experiment with 20 evaluation points using the automated infrastructure for experimentation takes around 1 min per point, or around 20 min in total. From our experience, we can argue that manual experimentation takes even more time, in comparison to using the automated infrastructure, which again favors using the virtual platform for evaluation. If we also take into account the time necessary for setting up a comparable experimental setup (on the order of days or weeks), the benefit of using the proposed virtual platform is even higher.

Table 3.7: Evaluation results obtained using the infrastructure *vs.* the cloud platform

Scenario	**Reference**			**Scenario 1**			**Scenario 2**			**Scenario 3**		
Metric	Infra.	Platf.	Diff.	Infra.	Platf.	Diff.	Infra.	Platf.	Diff.	Infra.	Platf.	Diff.
Mean error [m]	2.82	2.91	0.09	3.84	3.72	0.12	5.20	4.91	0.29	4.02	4.26	0.24
Median error [m]	2.61	2.45	0.16	3.11	2.87	0.24	4.81	4.41	0.40	3.56	3.61	0.05
RMS error [m]	3.14	3.07	0.07	4.75	4.45	0.30	5.88	5.53	0.35	4.80	4.55	0.25
75 percentile error [m]	3.87	3.74	0.13	5.14	4.87	0.27	6.46	6.22	0.24	5.36	5.45	0.09
90 percentile error [m]	4.19	4.56	0.37	5.87	5.44	0.43	9.49	8.81	0.68	7.37	7.44	0.07
Min. error [m]	0.47	0.22	0.25	1.26	1.17	0.09	0.88	0.56	0.32	0.79	1.02	0.23
Max. error [m]	5.76	6.41	0.65	6.70	6.53	0.17	11.46	10.31	1.15	10.93	11.65	0.72
Room accuracy [%]	70.00	65.00	5.00	50.00	55.00	5.00	45.00	50.00	5.00	45.00	45.00	0.00
Mean delay [s]	20.08	21.02	0.94	19.57	21.05	1.08	20.58	20.76	0.18	20.02	20.55	0.53
Median delay [s]	20.05	21.20	1.15	19.83	21.09	1.26	20.45	20.55	0.10	20.05	20.63	0.58
RMS delay [s]	20.09	21.81	1.72	19.88	21.54	1.66	20.58	20.85	0.27	20.02	20.66	0.64
75 percentile delay [s]	20.11	21.32	1.21	19.95	21.78	1.83	20.71	20.91	0.20	20.11	20.76	0.65
90 percentile delay [s]	20.25	21.51	1.26	20.03	21.95	1.92	21.64	22.08	0.44	20.16	20.87	0.71
Min. delay [s]	19.66	20.12	0.46	19.71	19.56	0.15	19.92	20.03	0.11	19.67	20.12	0.45
Max. delay [s]	21.02	22.23	1.21	20.11	22.04	1.93	21.94	22.65	0.71	20.26	21.04	0.78

3.4 Conclusions

As the first contribution in this thesis we presented the testbed and cloud infrastructures for evaluation of RF-based indoor localization solutions in environments with artificially generated RF interference context. The usage of the proposed testbed infrastructure enables repeatable and comparable evaluation, without the need of a test-person carrying a device to be localized to different evaluation locations. The cloud infrastructure enables calculating, storing, and presenting the results of evaluation experiments in a unified way by using a set of standardized performance metrics. The results of the experiments are publicly available in a form of a portfolio of evaluated solutions, hence people are able to compare the performance of their localization solution with the performance of already evaluated ones. The portfolio currently includes more than 30 evaluated RF-based indoor localization solutions, ranging from WiFi-based fingerprinting to IEEE.15.4 multilateration and proximity-based ones. The developed testbed and cloud infrastructures have been used extensively, not just throughout this thesis, but also by numerous other researchers. This resulted in a large number of published scientific results, some examples being [147, 148, 149, 150, 151].

As stated previously, the proposed infrastructures were built with the aim of enabling unified, fast, and reproducible evaluation of RF-based indoor localization solutions. Enabling these features of the evaluation has a trade-off in terms of the realism of the obtained results. That is to say, the performance results obtained by leveraging the proposed infrastructures will certainly be overly optimistic in comparison to the ones obtained in case a localization solution is deployed in a realistic environment (e.g. hospital, airport, grocery store). In realistic environments, there is a variety of external influences that will potentially degrade the performance of RF-based indoor localization solutions. These influences include the orientation, hight, and type of a device used for localization, the effects of person's body and the position of a device on a person (e.g. being held vs. in a pocket) and other people in an environment, and the effects of changes in an environment (moving furniture, open/closed doors, etc.). In order to enable objective comparable evaluation of different RF-based indoor localization solutions, we aimed at minimizing these influences to the level that is practically possible. The performance results obtained by using the proposed infrastructures can therefore be viewed as practical lower bounds of the performance of an evaluated solution. Our infrastructures also enable "isolating" a particular effect that can influence the performance of an RF-based indoor localization solution. We have shown that on an example of the effects of RF interference and provided means for isolating other effects. Having the possibility of isolating one effect, one can then focus on trying to mitigate that particular effect. We will demonstrate how this can be done in the next chapter of the thesis, where we will for example focus on the effects of RF interference on the performance of WiFi-based indoor fingerprinting and show how the observed negative effects can be successfully mitigated. Without the means of objective and repeatable evaluation, it would be questionable if a particular mechanism for mitigating a certain effect really serves its purpose, or the observed difference in the evaluation results is due to a chance in the conditions between different runs of an experiment. Finally, as a contribution for enabling evaluation in realistic conditions, some of the datasets that we provide have been collected in realistic environments, in particular in a hospital and in an underground mine. We also encourage and provide support for other people to contribute with their datasets for other types of environments.

Certainly, there are limitations to the capabilities of the proposed infrastructures. Our robotic mobility platform serves as an accurate source of ground truth information and enables reproducible positioning of a device to be localized. However, our robotic mobility platform is not able to mimic a person's body, hence this effect cannot be isolated. Our infrastructure is not able to recreate the movement of a person, e.g. walking or running, meaning that it is not possible to evaluate IMU-based indoor localization solutions, which are very promising, in particular in complementing RF-based indoor localization approaches [152, 153, 18]. Furthermore, our testbed infrastructure is limited to the evaluation of RF-based indoor localization solutions in the 2.4 GHz ISM frequency band. This is because we can guarantee that there are no other potential sources of interference only in those frequencies. Our testbed infrastructure is able to support experimentation in the sub-GHz and 5 GHz ISM frequency bands, however for those frequencies uncontrolled interference should be expected (e.g. the university's EDUROAM network), which can hinder the objectiveness of the obtained results. Our testbed infrastructure currently does not include any devices for experimentation in higher frequencies. Given the tendency of shifting the whole wireless communication paradigm to higher frequencies [154, 155, 156], this is a definite limitation of our testbed infrastructure.

With the initial goal of demonstrating the feasibility and usefulness of the developed infrastructures, we have organized and executed a remote localization competition in which several RF-based indoor localization algorithms have been evaluated in scenarios with artificially generated RF interference context. In the thesis, we therefore discussed the scenarios in which different algorithms have been evaluated and the interference patterns that have been generated for evaluating the impact of interference on the evaluated algorithms. We also presented the followed performance evaluation methodology, which includes the used performance metrics, the procedure of their calculation, and the procedure of obtaining the final scores for three different evaluation categories. These three categories were primarily focused on localization accuracy, latency, and interference robustness of location information provisioning, respectively. Finally, we presented the results of the evaluation for the three evaluation categories.

Our results demonstrated the feasibility and usability of the developed infrastructures for performing the evaluation experiments without or with controlled RF interference context. Furthermore, the results confirmed the well-known claims from the literature stating that WiFi-based fingerprinting is a promising solution for indoor localization, e.g. [75, 157]. Our experiments also showed that RF interference can indeed degrade the performance of the evaluated algorithms, which provided the ground for improving the interference robustness of WiFi-based fingerprinting. The competition revealed that a large number of solutions use the same type of raw data for estimating location. This observation lead to our next contribution - a platform for streamlined evaluation of indoor localization algorithms by leveraging precollected raw datasets.

The proposed a platform for streamlined experimental evaluation of RF-based indoor localization algorithms enables a simple, yet effective experimentation without the need of a testbed infrastructure. We firstly provided an overview of the design ideas that guided the development of the platform, followed by the selected implementation tools and frameworks. We further described the currently publicly available raw datasets that can be used for the evaluation, together with the experimentation possibilities that they enable. Finally, we demonstrated the usage of the platform and assessed the time overhead of using the proposed platform by performing the same set of experiments using the platform and the previously proposed testbed infrastructure. The performance metrics achieved by using the platform show high level of similarity to the metrics achieved by physically performing the evaluation experiments, which demonstrates the correct performance of the platform. The advantage of using the platform is visible in its usage simplicity for the experimenters and in the reduction of experimentation time. In addition, when using the platform, the experimenter is provided with easy to use, low-complexity access to the infrastructure, which removes the burden of handling the components of the infrastructure for evaluation. In comparison to manual evaluation, when using the platform the experimenter does not have to carry a measuring device and does not have to care about finding the exact coordinates for the ground-truth locations, which are both labor and time consuming. By using the platform it is possible to use exactly the same dataset for comparing different algorithms or different parameterizations of an algorithm, which increases the comparability of the achieved performance results.

The platform has been leveraged for the different evaluations performed in this thesis, but also by various other researchers in the scientific community, with examples being [158, 149, 159, 151, 160]. This platform has also been used during the IPSN/Microsoft Indoor Localization Competition 2014, where it supported the calculation, storage, and presentation of performance results [61]. The results of this competition differed significantly from the results that we and other researchers usually generated using our platform. However, the platform successfully supported the IPSN/Microsoft Indoor Localization Competition 2014, demonstrating that the initial design goal of extensibility was fulfilled. Finally, the platform is currently being used as the core engine for data storage and performance results calculation in the ongoing NIST competition. In this competition, the participants are invited to evaluate their indoor localization algorithms on a basis on extensive measurements from various sensors collected by four different types of smart-phones [161, 162]. More specifically, for this competition readings from WiFi, cellular, GNSS, and environmental, position, and motion sensors were collected using LG G4, Motorola Nexus 6, OnePlus 2, and Samsung Galaxy S6 smart-phones. The evaluation procedure in this competition is envisioned to follow the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 18305 standard “Test and evaluation of localization and tracking systems” [1]. Our platform successfully supported storing of the measurements from this larger number of sensors and from multiple smart-phones. Furthermore, our platform was able, with small modifications, to support another procedure for metrics calculation. Both of these facts demonstrate the fulfillment of the “extensibility” as a design goal for the proposed cloud platform. The participants of the competition are accessing our platform remotely and using different types of devices, which demonstrates the fulfillment of the “language and platform independence” and “remote accessibility” design goals. Finally, the researchers are, based on their needs, able to scope the collected sensor readings, access only certain types of readings, of access only readings from a particular smart-phone. This demonstrates the fulfillment of the “usability” design goal for the developed cloud platform.

[1] https://www.iso.org/obp/ui/#iso:std:iso-iec:18305:dis:ed-1:v1:en

4 Advancing WiFi Fingerprinting-based Localization Solutions

In the previous chapter, we outlined the infrastructure for objective and fast evaluation of RF-based indoor localization solutions. With the help of the developed infrastructure, we have a means of carrying out further research toward advancing RF-based indoor localization.

As suggested repeatedly in the literature and discussed previously in the thesis, fingerprinting-based localization leveraging the widely deployed WiFi infrastructures is one of the most popular classes of indoor localization approaches. However, there are still multiple challenges associated with WiFi fingerprinting-based localization solutions. An in-depth understanding of WiFi fingerprinting-based localization solutions is currently missing. Furthermore, the results obtained in the competition show that certain interference patterns highly influence the performance of WiFi-based fingerprinting, which motivates further in-depth analysis on how to mitigate the observed negative impact, thus increase the interference robustness of fingerprinting. Moreover, the deployments of fingerprinting solutions are generally time consuming, which poses a challenge in their widespread usage. Since the deployment of fingerprinting is time consuming, the performance of fingerprinting should ideally be predictable for new environments because, if the expected performance in a new environment is not going to be satisfactory for some envisioned use-cases, then why going through the trouble of deploying it? We aim at addressing these challenges in this chapter of the thesis.

Parts of this chapter have been published previously. A systematic experimental evaluation of the performance of WiFi-based fingerprinting solutions has been discussed in [144]. The effects of interference on RSS features of RF signals are discussed in [99]. An approach for mitigating the effects of interference in WiFi RSS-based fingerprinting is overviewed in [163]. An approach for reducing the deployment overheads in WiFi-based fingerprinting is given in [164]. Finally, the possibility of extrapolation of the performance of WiFi fingerprinting-based localization solutions across environments has been discussed in [165].

4.1 Decomposition of WiFi Fingerprinting-based Localization

Despite their popularity and importance of WiFi fingerprinting-based indoor localization solutions, the current praxis of their comparative experimental evaluation is lacking rigor, with studies typically following an ad-hoc evaluation process and focusing on black-box comparison of complete algorithms. Usually, the algorithms are presented and evaluated as monolithic solutions, although several well-defined phases can be identified: collection of raw data, creation of fingerprints, matching of fingerprints and training data and post-processing for which specific procedures can be separately defined. Thus, the existing evaluation approaches fail to offer deeper insight into the relative contribution of these individual phases to the overall performance of the solutions.

To address the shortcomings, we perform systematic experimental evaluation of the performance of WiFi-based fingerprinting solutions that is based on previously developed approach for objective evaluation of RF-based indoor localization algorithms. We study the relative contribution of various procedures used in the individual algorithm phases on the overall performance, confirming and extending the insights of earlier, less systematic attempts in this direction [166, 68, 75]. In a comprehensive experimental case-study, we evaluate 3 WiFi fingerprinting algorithms with 4 raw RSSI collection procedures, 3 fingerprint creation and pattern matching procedures, 4 different post-processing procedures in 3 testbeds and 4 evaluation scenarios, resulting in 36 individual experiments.

4.1.1 Fingerprinting using WiFi Beacon Packets RSSI Measurements

General workflow of WiFi fingerprinting algorithms can typically be divided in three phases, while the execution of a fingerprinting procedure has two steps, as shown in Figure 4.1. The first step, called a training step, is executed *off-line* for obtaining a footprint of a wireless environment. The localization area is divided in a number of segments or cells and each cell is scanned a certain number of times for raw data (phase I). Furthermore, a fingerprint of each cell is created (phase II) following some methodology for processing the collected raw data. Using the obtained training fingerprints, the training database is created and stored on a fingerprinting server. The second step, known as a runtime or *on-line* step, is consists of three individual phases. Two of them are mirroring the phases of the first step: a number of scans are first created by the user at an unknown location, and the runtime fingerprint for the location is created following the same procedure as in the training step. On the server's side, the runtime fingerprint is compared with the training dataset following some predefined matching procedure (phase III). The output of the matching phase is then typically post-processed using a procedure like the kNN method. Actions within each phase are specified by so called procedures, in the following we will discuss examples of them.

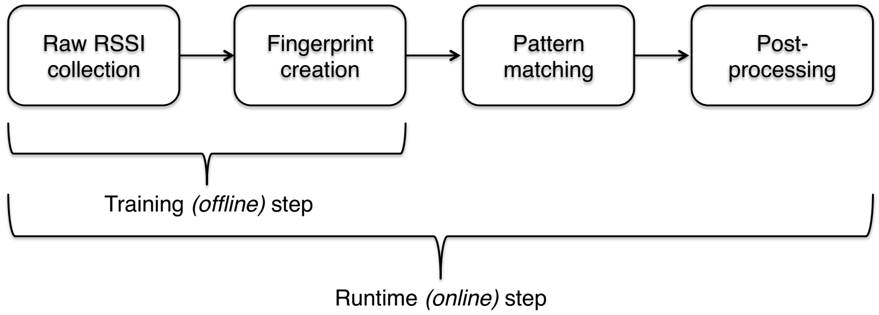


Figure 4.1: General workflow of WiFi fingerprinting algorithms

Collection of Raw RSSI Measurements

The first phase of fingerprinting algorithms is constituted by the collection of raw data used generating location information. While the sources of raw data can be different, we will, due to their particular interest for this thesis, talk only about RSSI measurements from WiFi beacon packets. Three basic decisions define this phase: the locations of measurements, the number of measurements that should be collected in each location, and the set of beacon sources (WiFi APs) which are considered. Let N be the number of environmental scans taken at one measurement point. During each scan the vector of RSSI measurements from WiFi APs is collected. By repeating the measurements at one location N times a matrix of RSSI measurements from different APs is created. Similar procedure is followed in both the training and runtime steps, with N being usually smaller in the runtime step due to time, power, resource utilization constrains. To limit the size of raw dataset and to improve the robustness, different methods can be used for selecting the set of contributing APs. Typical examples that we consider are:

All Visible APs: In this case not only APs being under control of the localization service provider, thus expected to operate continuously with stable parameters, but also all available APs are included (e.g. APs deployed in the served environment and in neighboring buildings).

All Visible APs with Threshold: In this case only RSSI measurements above certain threshold are used as raw data. The goal of this scoping is to reduce the time variability of RSSI measurements by discarding those below certain threshold. We consider two such threshold values: -90 dBm and -80 dBm.

Dedicated APs: The third investigated approach for the collection of raw RSSI measurements is selective collection of RSSI measurements from only a "dedicated" set of APs. Albeit their primary function is obviously providing wireless connectivity, this set is assumed to be "under control" of the localization service provider. Therefore the stability of their operation in time, as well as sufficient coverage of the served environment can usually be expected.

Fingerprint Creation

In the fingerprint creation phase, certain characteristic values are derived from the raw RSSI measurements according to some procedure. Note that a selection made in this procedure is independent of the decisions made in Phase I. These procedures might relate to time series of measurements from a single AP, or involve whole vectors of APs. For simplicity we will further present examples of the fingerprints relating to each AP separately, listed in the sequence of increasing computational complexity.

Averaged RSSI Vectors: This procedure [68] aims at computing an average value of the RSSI measurements obtained from each AP used for localization purposes. The fingerprint is an average value of the RSSI measurements obtained from each AP in both training and runtime steps, where $K_{r,t}$ is the length of the vector. Let $\boldsymbol{\mu}_{t,m} = [\overline{RSSI_{t,1}}, ..., \overline{RSSI_{t,k}}, ..., \overline{RSSI_{t,K_{r,t}}}]$

be the vector of averaged RSSI values $\overline{RSSI_{t,i}}$ from an AP i obtained in the training step at the location m ($m \in 1, ..., M_t$). Similarly, let $\boldsymbol{\mu}_r = [\overline{RSSI_{r,1}}, ..., \overline{RSSI_{r,k}}, ..., \overline{RSSI_{r,K_{r,t}}}]$ be the vector of averaged RSSI values $\overline{RSSI_{r,i}}$ from each AP i obtained in runtime step, i.e. runtime fingerprint.

RSSI Quantiles: This procedure [54] uses q quantiles of the RSSI values from each AP as fingerprints, which are calculated in two steps. First the CDF of the RSSI measurements from each AP is computed. Second, the quantiles, i.e. RSSI values with probabilities $k/(q-1)$, where $k = 0, 1, ..., q-1$, are calculated. The result of the quantile calculation in both training and runtime steps is a quantile matrix $Q_{K,q}$, where K is the number of APs visible at the given location and q is a number of quantiles.

MvG Distributions of RSSIs: This fingerprint creation procedure uses the Multivariate Gaussian (MvG) distributions of RSSI measurements from each AP used for localization [40]. The procedure assumes that the RSSI measurements from different APs are distributed according to the MvG distribution. In other words, the distribution of the RSSI measurements from each AP at one cell can be written as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the vector of averaged RSSI values from each AP used for localization and $\boldsymbol{\Sigma}$ is the covariance matrix describing the relation between RSSI measurements from different APs.

Pattern Matching

The pattern matching phase is generally used for capturing similarities between fingerprints in a training dataset and a runtime fingerprint [167]. The training fingerprint that is most similar to a runtime fingerprint according to a given pattern matching procedure is then reported as an estimated location. From the various procedures for pattern matching we select three samples and from the point of view of computational complexity, we present them sequentially. Our choice of presented procedures is not completely free, while it has to be constrained dependent on the fingerprinting creation.

Euclidean distance: This pattern matching procedure [40] uses the Euclidean distance between training fingerprint at the cell m and the runtime fingerprint and it is given as:

$$D_E(\overline{\boldsymbol{X}_{t,m}}, \overline{\boldsymbol{X}_r}) = |\overline{\boldsymbol{X}}_{t,m} - \overline{\boldsymbol{X}}_r| \tag{4.1}$$

where $\boldsymbol{X}_{t,m}$ and $\boldsymbol{X}_r$ are fingerprint vectors in training and runtime step, respectively. The cell with the smallest distance (also called smallest weight) is reported as an estimated location. In the rest of the thesis, we will use ED as pattern matching procedure for the Averaged RSSI fingerprints.

Pompeiu-Hausdorff Distance: This pattern matching procedure uses PH metric for capturing similarities between fingerprints in training dataset and runtime fingerprint [54]. The PH distance between two sets is given as follows:

$$D_{PH}(\boldsymbol{X}_{t,m}, \boldsymbol{X}_r) = \max_{x_{t,k} \in \boldsymbol{X}_{t,m}} \min_{x_{r,k} \in \boldsymbol{X}_r} d(x_{t,k}, x_{r,k}) \tag{4.2}$$

where $d(x_{t,k}, x_{r,k})$ is the Euclidean Distance measurement between elements of the runtime fingerprint $\boldsymbol{X}_r$ and training fingerprint $\boldsymbol{X}_{t,m}$ at point m. The training point with the smallest PH distance with the runtime fingerprint is reported as an estimated location. In the rest of the thesis, we wil use PH as pattern matching procedure for RSSI quantile fingerprints.

Kullback-Leibler Distance: This pattern matching procedure uses the Kullback-Leibler (KL) distance metric between training fingerprint at the point m and the runtime fingerprint for capturing the similarities between them [40]. In this procedure training and runtime fingerprints are Probability Density Functions (PDFs) given by $p_{t,m}$ and p_r, respectively. We apply KL distance as pattern matching for MvG fingerprints. The Kullback-Leibler (KL) distance metric between the probability distributions in this case is given as:

$$\begin{aligned} D_{KL}(p_{t,m}, p_r) = &\frac{1}{2}((\boldsymbol{\mu}_r - \boldsymbol{\mu}_{t,m})^T (\boldsymbol{\Sigma}_r)^{-1} (\boldsymbol{\mu}_r - \boldsymbol{\mu}_{t,m}) \\ &+ tr((\boldsymbol{\Sigma}_r)^{-1} \boldsymbol{\Sigma}_{t,m} - \boldsymbol{I}) - ln|(\boldsymbol{\Sigma}_r)^{-1} \boldsymbol{\Sigma}_{t,m}|) \end{aligned} \tag{4.3}$$

where *tr*$(\cdot)$ denotes the trace of a matrix (sum of its diagonal elements) and **I** is the identity matrix. This pattern matching procedure reports the cell with the smallest KL distance to the runtime fingerprint as an estimated location.

Post-processing

A post-processing method, like the application of kNN, is usually the last phase of the fingerprinting procedure. The input from previous phase is the location estimate, which can be easily extended to a set of best estimates. Namely, instead of using the best match between training dataset and runtime fingerprint, one can use the set of *k* best matches. Based on this set of training fingerprints considered as best matches, different kNN methods for estimating the final location can be applied. From the numerous possible kNN methods for the simplicity reasons we select two well known and widely used ones and use them as representatives for evaluating the influence of kNN methods.

Non-weighted k-Nearest Neighbors: Non-weighed kNN methods give the same weight to each of the *k* fingerprints reported as best matches with the runtime fingerprint. We use 3 or 4 as parameters of non-weighted kNN method for evaluation of influence of kNN method on the accuracy of indoor fingerprinting.

Weighted k-Nearest Neighbors: Weighted kNN neighbors assign different weights to the training fingerprints reported as the closest training fingerprints in the previous phase. We use two sets of weights, namely for $k = 3$ the weights are $[4/7, 2/7, 1/7]$, while for $k = 4$ the weights are $[8/15, 4/15, 2/15, 1/15]$. Given that our aim is to evaluate the influence of kNN methods in general, we select these fixed values as weights for simplicity reasons only, acknowledging that a more intelligent selection could yield more optimal performance of fingerprinting.

4.1.2 Testbeds Environments Used in the Evaluation Procedure

In this section we present our approach for addressing the problem of objective comparison of WiFi fingerprinting-based indoor localization algorithms by following the previously developed comprehensive methodology and by leveraging the developed streamlined evaluation platform and the collected raw datasets. We have selected the geometrical and room level accuracy metrics as the focus of the presented study. We perform our experiments in three testbeds: TWIST, w-iLab.t I, and w-iLab.t II.

TWIST testbed: The first testbed used in the evaluation is the previously presented TWIST testbed. All three floors of the testbed were used in the evaluation. The footprints of the 2^{nd} floor of the testbed are given in Figure 4.2. Black dots in the figure present the locations of the dedicated APs, while red dots present the locations of training points used in the fingerprinting procedure. The dedicated wireless APs used for localization are TL-WDR4300 wireless routers, with a fixed channel allocation scheme set on channel 11 (2462 MHz) and with the transmission power of dedicated APs set to 20 dBm (100 mW). All experiments are performed during weekend afternoons, in an environment with minimized and monitored interference. The wireless environment was monitored using WiSpy 2.4x spectrum scanners, and all samples taken in the presence of interference above the threshold of -80 dBm were repeated. As a mobile device to be localized we used a MacBook Pro laptop with the AirPort Extreme Network Interface Card (NIC).

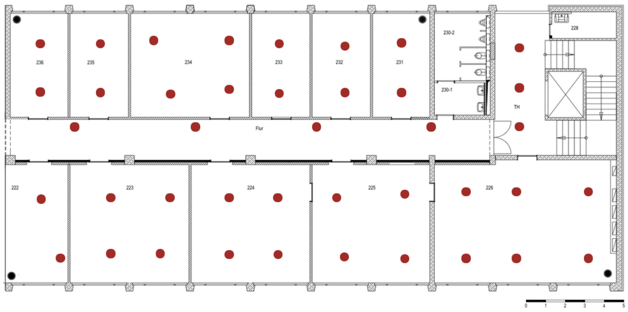


Figure 4.2: Footprint of the 2^{nd} floor of the TWIST testbed

w-iLab.t I testbed: The w-iLab.t I wireless testbed used for the evaluation purposes is located at De Zuiderpoort in Ghent, Belgium. The testbed area used for the evaluation is more than $800m^2$ large office area (approx. 40 x 20 m) with plywooden walls. The footprint of the w-iLab.t I tesbed is given in Figure 3.20 b), together with locations of training points (red dots) and locations of dedicated APs (black dots) used for fingerprinting. The testbed nodes are Alix 3C3 devices running a Linux distribution and in essence these are mini PC's equipped with two IEEE 802.11 a/b/g interfaces. They have been configured to a fixed channel allocation scheme set on channel 11 (2462 MHz), with the transmission power of dedicated APs set to 20 dBm (100 mW). Similarly as for the TWIST testbed, the wireless environment was monitored using WiSpy 2.4x spectrum scanners and samples taken in the presence of interference above the threshold of -80 dBm were repeated.

w-iLab.t II testbed: The third testbed use for the evaluation purposes is the w-iLab.t II testbed [56]. The testbed is located in Ghent and it is a part of Future Internet Department of iMinds. The testbed has the size of more than 1000 m^2 (50 x 25 m) and can be characterized as an open-space industrial environment. The footprint of w-iLab.t II testbed is given in Figure 3.20 c). The black dots in the figure represent locations of the APs used for localization, while the red dots depict the locations of training points used in the fingerprinting procedure. Other objects in the figure represent the locations of obstacles in the testbed environment. Obstacles are mostly made of metal so a lot of shielding and reflection in the environment is expected. The APs used for localization are Zotac embedded PCs with IEEE 802.11n wireless cards. The same type of device is used as a mobile device to be localized. Other parameters of the experiment are similar as the ones from the TWIST testbed. The transmission power of the dedicated APs is set to 20 dBm (100 mW) and the fixed channel allocation scheme is set on channel 11 (2462 MHz). The environment is shielded so there is no external wireless interference to be expected.

4.1.3 Evaluation Scenarios

In the following we present four evaluation scenarios and their instantiations in the testbeds. The scenarios have been selected so that different sizes and types of environments are covered, e.g. office vs. open-space, small vs. large building, plywooden vs. concrete walls, one vs. multiple floors. For each of those scenarios we made a certain number of fingerprints both in training and runtime steps. One fingerprint of the environment in the training step consists of 40 scans of the RF environment, where each one is made by scanning all the available WiFi channels (1 to 11) for 1 s. The same procedure is used for gathering the runtime fingerprints, but here one fingerprint consists of only 10 scans of the WiFi environments. This reflects the more stringent constraints on the latency and power consumption in the runtime step.

First scenario: The first scenario can be characterized as “Small size office environment”, hence this scenario is instantiated on the 2nd floor of TWIST testbed. The testbed environment has been scanned for fingerprints as presented in Figure 4.2 and the training dataset has been created. The training dataset for this scenario then consists of 41 training points and each point consists of 40 scans of the RSSI measurements. Some APs are not visible at some training points or some scans. In this case, these RSSI measurements are given a default RSSI value (-100 dBm). The measurement points of the second scan of the environment, used as the runtime dataset, are shown in Figure 4.3.

Second scenario: Second scenario can be characterizes as the “Medium size office scenario”, therefore it is instantiated in the w-iLab.t I testbed in Ghent. The training dataset consists of 56 training fingerprints, as presented in Figure 3.20 b). The set of 20 measurement points, used as the evaluation points in the runtime step of fingerprinting, are presented in Figure 4.4. Some APs are possibly not visible on all measurement points and in this case the RSSI measurements from these points are given the default values of -100 dBm.

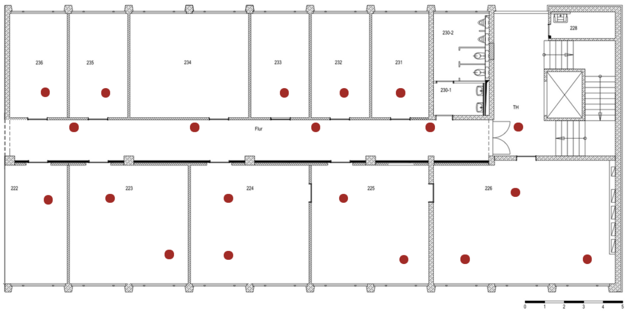


Figure 4.3: First scenario instantiation

Third scenario: The third scenario is characterized as a “'Big size office environment” and it is instantiated on 2^{nd}, 3^{rd}, and 4^{th} floor of TWIST testbed. Each floor is supplied with 4 dedicated APs, so altogether 12 APs are set up in the localization area and used as the dedicated localization APs. The training dataset is filled with the scans of the environment at each training point, and finally it consist of 123 fingerprints. Same as for the previous scenarios, each training point is scanned 40 times. The RSSI measurements that are not visible at particular point are given a default value of -100 dBm. The runtime dataset consists of the measurements at the same positions as in the scenario 1, but extended on the 3^{rd} and 4^{th} floors. The runtime points of the 2^{nd} floor are depicted in Figure 4.3, while the points on other two floor have the same locations in terms of x and y coordinate, while coordinate z differs at each floor.

Fourth scenario: The fourth scenario can be characterized as the “Big open-space scenario” and it is instantiated in the w-iLab.t II testbed in Ghent. Only 4 dedicated APs were used for localization. The training dataset consists of 100 fingerprints with locations given in Figure 3.20 c). The runtime dataset consists of 27 fingerprints equally distributed in the testbed environment, shown with dots in Figure 4.5. The rest remains as in the previous scenarios.

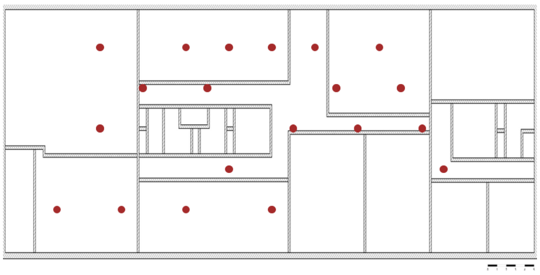


Figure 4.4: Second scenario instantiation

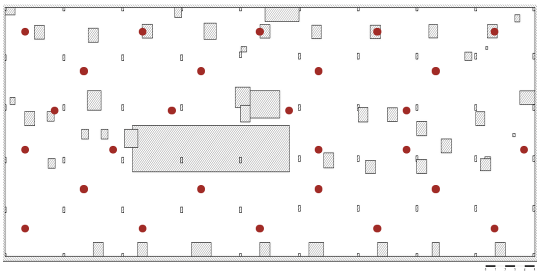


Figure 4.5: Fourth scenario instantiation

4.1.4 Evaluation Results

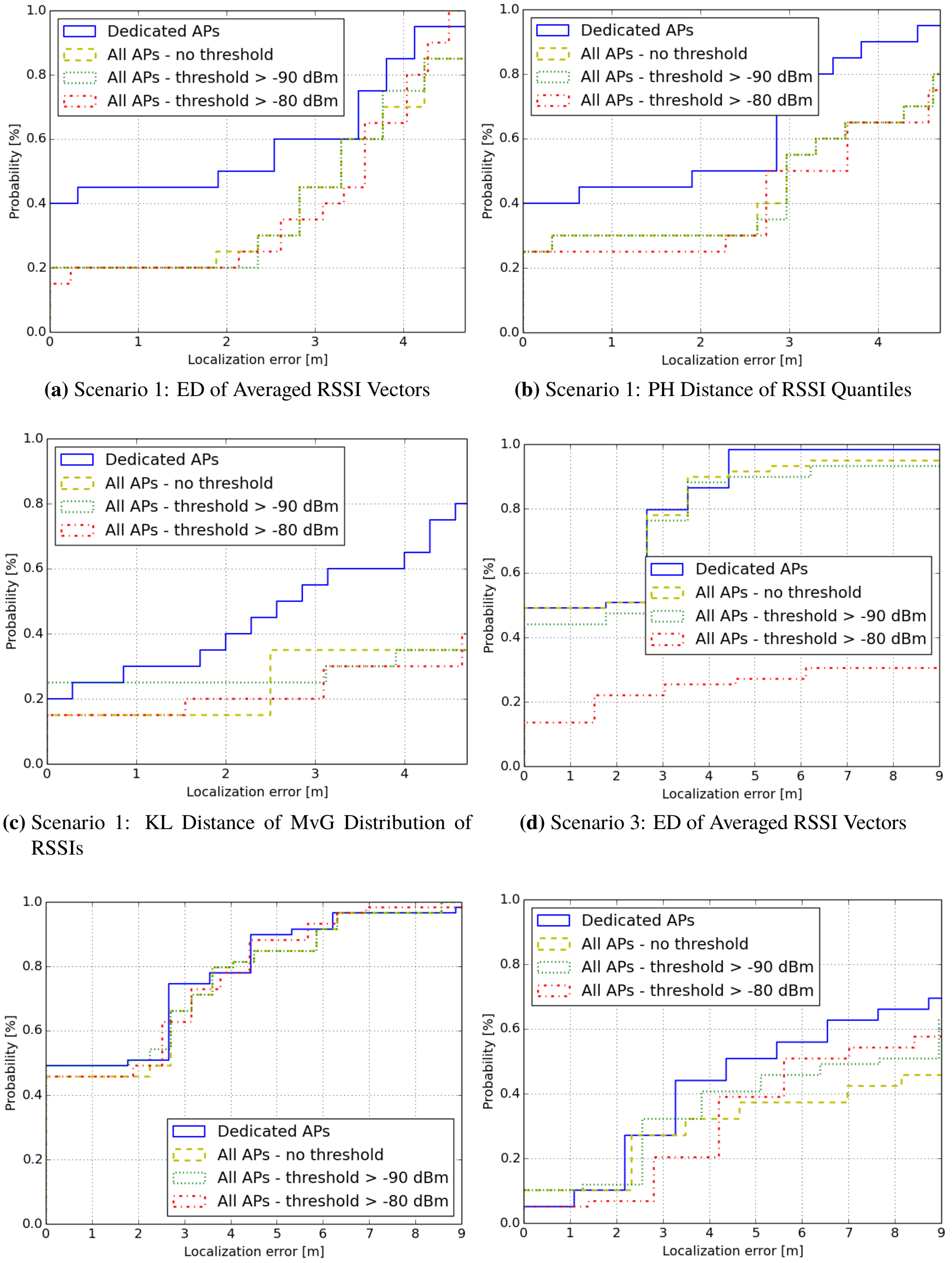
This section presents the comparison of algorithms' performance in terms of geometrical and room level accuracy of localization. Firstly, we present the influence of different procedures for raw RSSI measurement collection on the performance of different algorithms. To do so, we use different procedures for the collection of raw RSSI measurements, while fixing the fingerprint creation and pattern matching procedures. For combinations of fingerprint creation and pattern matching procedures we choose the ones in which both fingerprint creation and pattern matching have the same complexity level. For example, we match the least complex fingerprint creation procedure with the least complex pattern matching procedure. Finally, we evaluate the influence of different kNN procedure on the accuracy of different fingerprinting algorithms.

We perform the first part of the evaluation, i.e. the one concerned with the collection of raw RSSI measurements, in office scenarios in the TWIST testbed, because this environment is not shielded and number of uncontrollable APs, beside the dedicated APs used for localization, are expected. Secondly, for all presented scenarios, we evaluate the influence of different procedures for fingerprint creation and pattern matching on the overall performance of fingerprinting algorithms. This is done for a fixed procedure for raw RSSI collection. Finally, we apply different kNN procedures on the outputs of a pattern matching procedure to evaluate their influence on the final performance of different fingerprinting algorithms.

Results using Different Procedures for Raw RSSI Collection

The influence of procedures for collection of raw RSSI measurements on the final performance of fingerprinting algorithms for scenarios 1 and 3 is given in Figure 4.6. We collected raw RSSI measurements from all APs visible in the environment and filtered them in order to create different raw RSSI collection procedures. The evaluation scenarios 2 and 4 have been excluded from this evaluation, since only dedicated APs are the w-iLab.t I and w-iLab.t II testbeds, hence there was no possibility of generating different raw RSSI collection procedures. By performing filtering on top of the collected data, we were able to reuse the data from the same experiment for all raw RSSI collection procedures. On the differently obtained raw RSSI measurements we applied the same fingerprint creation and pattern matching procedures for estimating location, so the influence of a procedure of raw RSSI collection is captured in the final location estimation.

Summarized results in terms of average localization errors and room level accuracies achieved by different algorithms in scenarios 1 and 3 are given in Tables 4.1 and 4.2, respectively. The results show that the best performance is achieved in case RSSI measurements from only dedicated APs are used. In some cases the difference in performance is only marginal, but generally at least similar performance is achieved using dedicated APs, compared to using all visible APs, with or without a threshold. This observation holds for both evaluated scenarios in the testbed environment and for all used fingerprint creation and pattern matching procedures. This result is interesting because it demonstrates that, at least for the evaluated scenarios, it is better to leverage RSSI measurements from a smaller number of stable and controllable APs, in contrast to using all APs available in a certain environment.



(a) Scenario 1: ED of Averaged RSSI Vectors

(b) Scenario 1: PH Distance of RSSI Quantiles

(c) Scenario 1: KL Distance of MvG Distribution of RSSIs

(d) Scenario 3: ED of Averaged RSSI Vectors

(e) Scenario 3: PH Distance of RSSI Quantiles

(f) Scenario 3: KL Distance of MvG Distribution of RSSIs

Figure 4.6: Localization errors for different raw data collection procedures

Table 4.1: Summary results for scenario 1

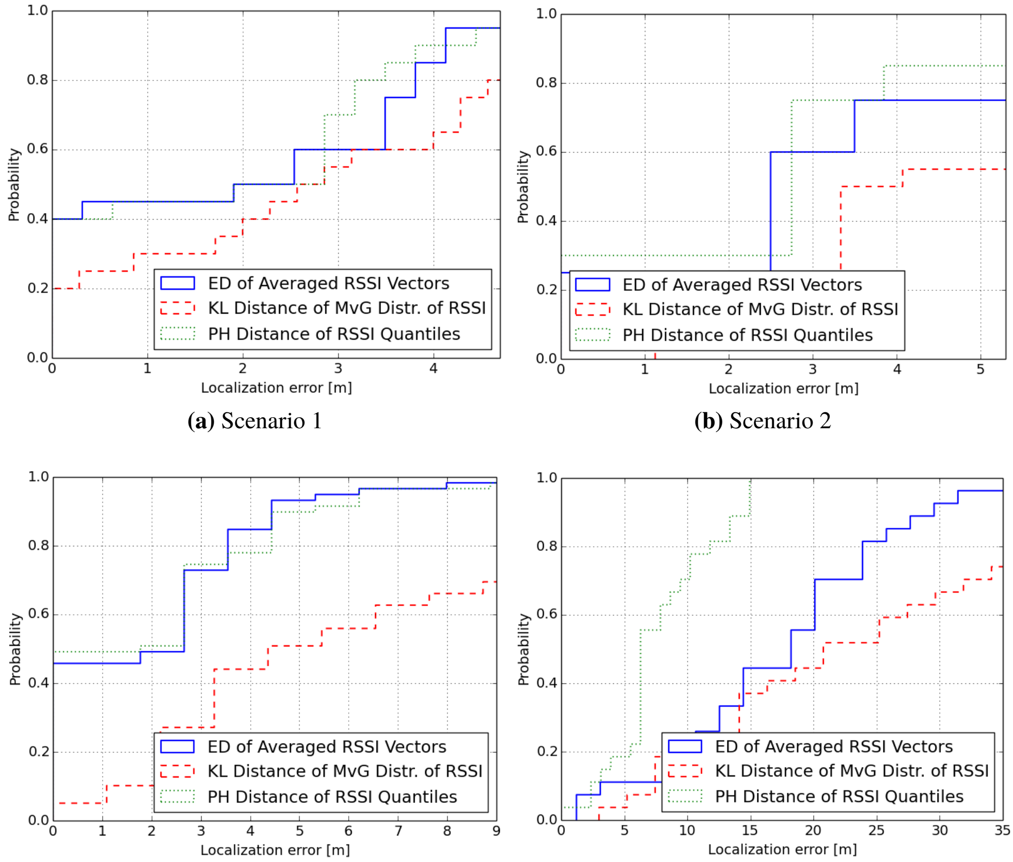
	4 APs	All APs	RSSI>-90	RSSI>-80
Euclidean Distance of Averaged RSSI Vectors				
Mean error [m]	2.21	3.56	3.55	2.98
Error variance [m^2]	4.05	6.42	6.35	2.53
Error median [m]	2.48	3.46	3.46	3.665
Min. error [m]	0.00	0.00	0.00	0.00
Max. error [m]	6.35	9.42	9.42	4.75
Room acc. [%]	80.0	70.0	70.0	75.0
PH Distance of RSSI Quantiles				
Mean error [m]	2.02	3.13	3.14	3.14
Error variance [m^2]	3.73	5.15	5.15	5.15
Error median [m]	2.52	3.19	3.20	3.20
Min. error [m]	0.00	0.00	0.00	0.00
Max. error [m]	6.35	6.60	6.60	6.60
Room acc. [%]	85.0	65.0	65.0	65.0
KL Distance of MvG Distributions of RSSIs				
Mean error [m]	2.77	7.41	6.71	8.88
Error variance [m^2]	3.85	32.99	24.57	54.84
Error median [m]	2.98	6.78	6.78	8.26
Min. error [m]	0.00	0.00	0.00	0.00
Max. error [m]	5.71	25.01	15.62	30.97
Room acc. [%]	50.0	35.0	30.0	35.0

Table 4.2: Summary results for scenario 3

	12 APs	All APs	RSSI>-90	RSSI>-80
Euclidean Distance (ED) of Averaged RSSI Vectors				
Mean error [m]	2.09	2.33	2.63	14.61
Error variance [m^2]	7.74	11.16	12.64	25.64
Error median [m]	2.18	2.63	2.73	17.43
Min. error [m]	0.00	0.00	0.00	0.00
Max. error [m]	17.74	17.74	17.74	25.11
Room acc. [%]	81.6	81.6	78.6	23.3
PH Distance of RSSI Quantiles				
Mean error [m]	2.05	2.31	2.31	2.31
Error variance [m^2]	7.16	6.35	2.34	6.99
Error median [m]	2.15	2.72	6.49	2.58
Min. error [m]	0.00	0.00	0.00	0.00
Max. error [m]	17.21	9.01	9.01	12.61
Room acc. [%]	84.4	68.3	68.3	71.3
KL Distance of MvG Distributions of RSSIs				
Mean error [m]	6.84	9.75	8.30	9.62
Error variance [m^2]	22.35	34.99	26.18	34.12
Error median [m]	5.28	9.47	8.33	6.72
Min. error [m]	0.00	0.00	0.00	0.00
Max. error [m]	21.82	23.27	25.57	28.06
Room acc. [%]	26.6	23.3	23.3	26.6

Results using Different Fingerprint Creation and Pattern Matching Procedures

A comparison of different fingerprint creation and pattern matching procedures for all evaluated scenarios is presented in Figure 4.7. For the collection of raw RSSI measurements only dedicated APs were used because of the best performance of this procedure, as demonstrated previously. The average localization errors of the algorithm named ĶL Distance of MvG Distributions of RSSIs is 2.77, 17.57, 6.84 and 17.63 m for scenarios 1, 2, 3 and 4, respectively. Furthermore, the average localization error of the algorithm ED of Averaged RSSI Vectors is significantly smaller for the office scenarios, namely 2.21, 2.67 and 2.09 m for scenarios 1,2 and 3 respectively. As for the open-space scenario, the localization error of ED Distance of Averaged RSSI Vectors is 14.9 m. Finally, the PH Distance of RSSI Quantiles performs similarly as the ED Distance of Averaged RSSI Vectors in terms of the average localization error in office scenarios, achieving average localization errors of 2.02 and 2.75 and 2.05 m for scenarios 1,2 and 3, respectively. For the open-space scenario, i.e. the evaluation scenario 4, the achieved average localization error is 8.01 m, which is an improvement of more than 6 m or 40% in comparison to the other two algorithms. Table 4.3 presents the summarized results of the performance evaluation.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

Figure 4.7: Localization errors for different fingerprint creation and pattern matching procedures

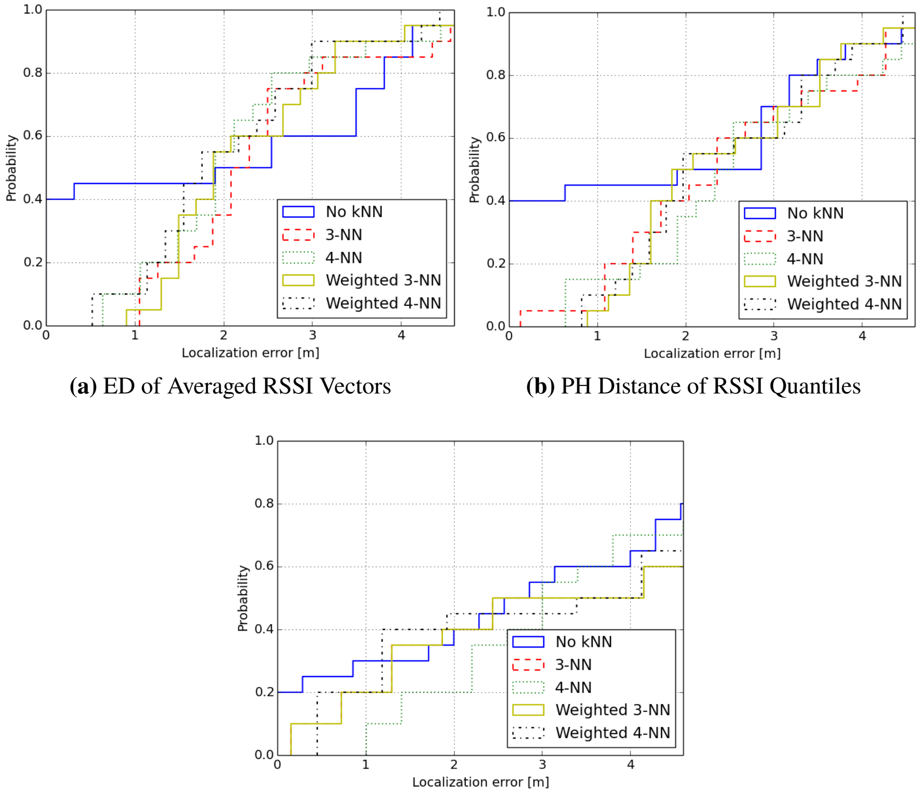
The obtained localization errors show that at least for the selected fingerprint creation and pattern matching procedures, more complex algorithms do not necessarily translate into more accurate localization. On the contrary, to a certain degree counter intuitive, less and medium complex procedures achieve better performance in the presented environments and scenarios. This is an accordance to some results suggested in the literature, e.g. [89], but also contrasting some others, e.g. [40]. These results also show that the performance of fingerprinting algorithms highly depends on an environment in which fingerprinting is deployed. Similar observation can be made by comparing the evaluation results for the office and open-space scenarios. As visible from the obtained results, promising results have been observed for all algorithms for small and medium sizes office-like environments. As the environment type changes to an open-space-like environment, the performance of fingerprinting degrades significantly. Two conclusions can be made from this observation. First, environment characteristics play a role in the performance of fingerprinting algorithms, which is also supported by the results reported in the literature, e.g. [148]. Hence, there is a need for predicting the performance of WiFi fingerprinting in a new environment, before its deployment in that environment. This issues has been discussed in Section 4.4. Second, diverse approaches for localization are needed for environments with different characteristics, with the possibility of their overlapping deployments in a certain environment. However, from the end-user's perspective, a seamless provisioning of location information is needed. Hence, for supporting seamless provisioning there is a need for a middleware service for handover, fusion, and integration of different sources of location information and its provisioning to the end-users and applications.

Table 4.3: Results for different fingerprint creation and pattern matching procedures

	ED Distance	PH Distance	KL Distance
		Scenario 1	
Mean error [m]	2.21	2.02	2.77
Error variance [m^2]	4.05	3.73	3.85
Error median [m]	2.48	2.52	2.98
Room acc. [%]	80.0	85.0	50.0
		Scenario 2	
Mean error [m]	2.67	2.75	17.67
Error variance [m^2]	4.01	6.58	64.31
Error median [m]	2.75	2.75	15.81
Room acc. [%]	80.0	85.0	25.0
		Scenario 3	
Mean error [m]	2.09	2.05	6.84
Error variance [m^2]	7.74	7.16	22.35
Error median [m]	2.18	2.15	5.28
Room acc. [%]	81.1	84.4	26.6
Floor acc. [%]	100.0	100.0	58.2
		Scenario 4	
Mean error [m]	14.92	8.01	17.63
Error variance [m^2]	10.16	4.28	9.73
Error median [m]	12.71	8.48	18.25

Results using Different k-NN Procedures

This section presents the results of evaluation using different kNN procedures. We applied two different kNN procedures, weighted and non-weighted, with two sets of parameters, as described previously. Tables 4.4, 4.5, 4.6 and 4.7 summarize the evaluation results for the scenarios 1, 2, 3 and 4, respectively. Figure 4.8 depicts the influence of different kNN procedures on the average localization errors for the evaluation scenario 1. For these experiments we used two different procedures for fingerprint creation and pattern matching and only dedicated APs for the collection of raw RSSI measurements.



(a) ED of Averaged RSSI Vectors

(b) PH Distance of RSSI Quantiles

(c) KL Distance of MvG Distribution of RSSIs

Figure 4.8: Localization errors for different kNN procedures in the evaluation scenario 1

From the results obtained in all evaluation scenarios it is visible that kNN procedure, when applied on the KL Distance of MvG Distribution of RSSIs procedures for fingerprint creation and pattern matching, generally increases localization errors, which is caused by inaccurate set of best estimates, save the best one. However, for the other two fingerprint creation and pattern matching procedures, applying the kNN procedure generally reduces the variability of localization errors. More specifically, the results suggest that the variances and maximum values of localization errors are reduced in case kNN procedure is applied after these two fingerprint creation and pattern matching procedures. However, the accuracy of WiFi based fingerprinting algorithms, when different kNN procedures are applied, seems to depend on the type and size of an environment. In the office scenarios, it seems that applying the kNN procedure in the small environment does not significantly improve the accuracy of fingerprinting algorithms, while the improvement is more visible in case kNN procedures are applied in office environments of bigger sizes.

In all scenarios and for all fingerprint creation and pattern matching procedures the weighted 4-NN procedure generally yields the best results, in comparison to the other evaluated kNN procedures. Depending on an environment, as well as on fingerprint creation and pattern matching procedures, the results achieved by the 4-NN procedure are either comparable or better than the results obtained without applying kNN procedure or by applying any other kNN procedure.

In conclusion, the results confirm the relatively large impact of the procedures in the early phase of collection of raw RSSI measurements, something that is usually only marginally considered in the design of many fingerprinting algorithms. They show that, somewhat counterintuitively, a lower number of WiFi APs and simpler fingerprint creation and pattern matching procedures can sometimes match and even outperform more sophisticated alternatives, in terms of localization accuracy. They also highlight the importance of the post-processing and filtering step, like kNN, in reducing the localization error variability and extremes. Our results also illustrate the importance of customized selection of procedures for the individual phases of fingerprinting algorithms, in accordance to the specific conditions in different deployment environments and in light of the relevant performance metrics. Although the presented study is just a snapshot, we believe that when applied to a wider class of procedures and their combinations, our evaluation methodology supported by the proposed infrastructure can lead to significant improvement in the fundamental understanding of the process of composing efficient fingerprinting algorithms for specific deployment scenarios.

Based on the results of the evaluation, in the following sections of the thesis we will aim at improving selected aspects of two fingerprinting algorithms. Both algorithms will use dedicated APs in the raw RSSI collection phase because we demonstrated that this selection maximizes the room and point level accuracy of WiFi fingerprinting among the evaluated raw RSSI collection procedures. Furthermore, both algorithms will leverage the weighted 4-NN method in the post-processing procedure, since this method yield the lowest variability of localization errors across the evaluated post-processing methods. The algorithms will differ in the fingerprinting creation and pattern matching methods. Namely, we will use both Euclidean Distance of Averaged RSSIs and Pompeiu-Hausdorff Distance of RSSI Quantiles for fingerprint creation and pattern matching.

Table 4.4: Summary results for scenario 1

	No kNN	3-NN	4-NN	w3-NN	w4-NN
ED distance of Averaged RSSIs					
Mean error [m]	2.16	2.52	2.26	2.38	2.19
Error variance [m^2]	4.05	1.25	1.21	0.97	1.13
Error median [m]	2.48	2.33	2.07	1.98	1.86
Min. error [m]	0.00	1.05	0.64	0.90	0.52
Max. error [m]	6.35	5.17	4.87	4.82	4.64
Room acc. [%]	85.0	70.0	65.0	70.0	65.0
PH Distance of RSSI Quantiles					
Mean error [m]	2.02	2.70	2.67	2.61	2.54
Error variance [m^2]	3.73	2.20	1.60	1.48	1.29
Error median [m]	2.52	2.49	2.49	2.12	2.14
Min. error [m]	0.00	0.13	0.64	0.88	0.82
Max. error [m]	6.35	6.49	4.87	5.67	4.65
Room acc. [%]	75.0	85.0	75.0	85.0	75.0
KL Distance of MvG Distributions of RSSI					
Mean error [m]	2.77	3.86	3.85	3.86	4.00
Error variance [m^2]	3.85	7.82	4.71	7.82	11.40
Error median [m]	2.98	3.66	3.34	3.66	3.83
Min. error [m]	0.00	0.15	1.00	0.15	0.45
Max. error [m]	5.71	11.58	8.99	11.58	15.15
Room acc. [%]	55.0	50.0	50.0	50.0	50.0

Table 4.5: Summary results for scenario 2

	No kNN	3-NN	4-NN	w3-NN	w4-NN
ED distance of Averaged RSSIs					
Mean error [m]	2.67	2.92	2.79	2.49	2.44
Error variance [m^2]	4.01	5.63	4.05	2.31	1.96
Error median [m]	2.75	2.11	1.99	2.35	2.23
Min. error [m]	0.00	0.00	0.73	0.00	0.18
Max. error [m]	5.50	8.25	8.24	5.35	5.53
Room acc. [%]	80.0	85.0	95.0	85.0	95.0
PH Distance of RSSI Quantiles					
Mean error [m]	2.75	2.58	2.68	2.53	2.28
Error variance [m^2]	6.58	4.46	1.79	4.24	2.74
Error median [m]	2.75	2.10	2.35	1.59	1.74
Min. error [m]	0.00	0.00	1.37	0.00	0.36
Max. error [m]	11.00	8.25	6.87	7.46	7.15
Room acc. [%]	85.0	85.0	85.0	85.0	85.0
KL Distance of MvG Distributions of RSSI					
Mean error [m]	17.67	15.62	14.31	15.62	16.25
Error variance [m^2]	64.31	61.95	55.80	58.95	62.77
Error median [m]	15.81	12.44	10.58	12.44	11.92
Min. error [m]	2.75	0.96	0.99	0.96	2.19
Max. error [m]	36.21	39.46	39.29	39.46	37.49
Room acc. [%]	25.0	25.0	35.0	25.0	35.0

Table 4.6: Summary results for scenario 3

	No kNN	3-NN	4-NN	w3-NN	w4-NN
			ED distance of Averaged RSSIs		
Mean error [m]	2.31	2.77	2.47	2.77	2.41
Error variance [m^2]	8.61	2.92	1.61	4.30	2.69
Error median [m]	2.73	2.32	2.15	2.10	2.04
Min. error [m]	0.00	0.00	0.50	0.89	0.72
Max. error [m]	17.74	8.54	6.53	12.14	11.31
Room acc. [%]	86.4	72.9	76.2	72.9	76.2
			PH Distance of RSSI Quantiles		
Mean error [m]	2.32	2.46	2.38	2.31	2.24
Error variance [m^2]	9.51	1.62	1.45	2.41	2.17
Error median [m]	2.18	2.36	2.19	1.87	1.94
Min. error [m]	0.00	0.00	0.30	0.44	0.29
Max. error [m]	17.74	7.62	6.09	11.07	10.28
Room acc. [%]	79.6	71.2	72.9	71.2	72.9
			KL Distance of MvG Distributions of RSSI		
Mean error [m]	6.84	7.04	6.57	7.12	7.06
Error variance [m^2]	22.35	25.05	19.75	26.63	25.13
Error median [m]	5.28	5.99	5.64	5.50	6.01
Min. error [m]	0.00	0.63	0.39	0.13	0.47
Max. error [m]	21.82	18.86	17.66	18.32	18.71
Room acc. [%]	31.2	34.2	36.7	34.2	36.7

Table 4.7: Summary results for scenario 4

	No kNN	3-NN	4-NN	w3-NN	w4-NN
			ED distance of Averaged RSSIs		
Mean error [m]	14.92	13.37	12.24	13.39	12.88
Error variance [m^2]	60.16	58.14	52.40	59.01	53.84
Error median [m]	12.71	11.41	11.32	11.43	11.26
Min. error [m]	3.00	1.00	0.75	1.21	1.07
Max. error [m]	39.00	33.47	31.88	33.76	33.60
			PH Distance of RSSI Quantiles		
Mean error [m]	8.01	8.07	8.93	9.07	8.63
Error variance [m^2]	44.28	44.54	41.22	42.54	41.82
Error median [m]	8.48	8.89	9.15	9.89	8.22
Min. error [m]	0.00	2.12	2.37	2.12	2.23
Max. error [m]	23.68	22.51	22.36	22.52	22.74
			KL Distance of MvG Distributions of RSSI		
Mean error [m]	17.63	14.73	13.38	14.84	14.32
Error variance [m^2]	75.81	74.89	58.32	74.34	68.85
Error median [m]	18.25	12.37	11.47	13.05	12.59
Min. error [m]	3.00	0.00	2.12	2.14	1.41
Max. error [m]	47.43	33.06	31.15	32.85	30.87

4.2 A Service for Reducing Deployment Efforts of WiFi Fingerprinting-based Localization

As mentioned previously, for enabling a WiFi fingerprinting-based indoor localization service, the service provider is required to generate a training dataset, i.e. to survey an environment at predefined locations. The users' generated fingerprints are then compared with fingerprints from the generated training dataset, and based on their similarities location estimates are reported. The required training survey is time and labor consuming, which is further accentuated by the fact that the same survey, due to the changes in an environment or due to the collected measurements getting staled, has to be repeated after a certain period of time. The density of collected fingerprints in a training survey has a direct relation with the accuracy of fingerprinting [89], i.e. a higher density of training fingerprints yields a better accuracy of location estimation until a certain density threshold. A well-known concept of increasing the density of a training dataset, without an increase in the amount of time and labor required for surveying an environment for additional fingerprints, is to leverage a propagation model to create additional fingerprints at locations not surveyed, i.e. to generate virtual training fingerprints. This process, however, yields executional overhead in terms of defining locations of virtual training fingerprints, implementing a propagation model, generating virtual training fingerprints and storing them in a training database.

To address these shortcomings, we aim on simplifying the process of storage and management of training fingerprints, as well as the generation and storage of virtual training fingerprints in a training database. Thus, we propose the Enriched Training Service (ETS), a web-service that enables storage and management of WiFi RSSI-based training fingerprints. The adjective "enriched" indicates that the ETS provides a functionality that enriches its main purpose and which can be leveraged for generating virtual training fingerprints. Given the original set of training fingerprints of a certain density is stored in the ETS, the enriching functionality can firstly be leveraged for defining virtual training points. Secondly, based on a propagation model, the power levels from different WiFi APs at the defined virtual training points can be modeled and virtual training fingerprints can be generated and stored in the ETS. Leveraging this functionality results in the improved accuracy of fingerprinting with the same density of physically collected training fingerprints, i.e. without increase in the efforts of performing a training survey. The ETS features a modular design, where different types of fingerprinting algorithms can be supported. The modular design enables different methods to be implemented for both defining virtual training points and modeling of WiFi signal strengths at the defined virtual training points.

In the current implementation, we support our claim of modularity by enabling two fingerprinting algorithms to work with the ETS. Moreover, the definition of virtual training points can be performed based on Voronoi diagrams or based on the localization service provider's input. As for the propagation modeling, the current implementation allows the usage of Inverse Distance Weighted Interpolation (IDWI) and Multi-Wall (MW) propagation models, although, due to its modular design, additional models can be easily introduced.

4.2.1 Overview of Enriched Training Service

The usual procedure of surveying an environment for enabling WiFi RSSI fingerprinting is the following. The localization service provider firstly defines a set of training points, usually in a fairly regular grid fashion, although that is not a general requirement for WiFi fingerprinting. At each of the defined points, the localization service provider samples the WiFi environment and this sample, with a corresponding location coordinate, is stored in a training database. The procedure is repeated until WiFi RSSI measurements from all defined points are collected and stored in the training database. This usual procedure can be performed using the ETS, as depicted in Figure 4.9. Based on the requirements of an algorithm to be used, the collected raw RSSI measurements that are stored in the ETS can be processed to generate training fingerprints. In the following step, the generated training fingerprints can again be stored in the ETS. The training fingerprints can be stored separately from the raw RSSI measurements, which enables an easy replacement of the fingerprinting algorithm by generating a new set of training fingerprints from the originally stored raw RSSI measurements.

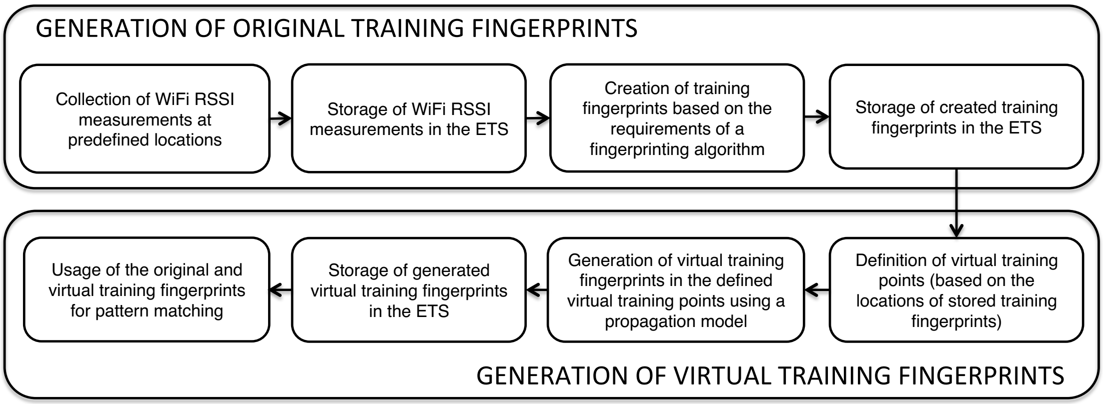

GENERATION OF ORIGINAL TRAINING FINGERPRINTS
Collection of WiFi RSSI measurements at predefined locations
Storage of WiFi RSSI measurements in the ETS
Creation of training fingerprints based on the requirements of a fingerprinting algorithm
Storage of created training fingerprints in the ETS
Definition of virtual training points (based on the locations of stored training fingerprints)
Generation of virtual training fingerprints in the defined virtual training points using a propagation model
Storage of generated virtual training fingerprints in the ETS
Usage of the original and virtual training fingerprints for pattern matching
GENERATION OF VIRTUAL TRAINING FINGERPRINTS

Figure 4.9: Envisioned usage of the Enriched Training Service

In order to increase the density of training fingerprints without increasing the number of physical measurements, the enriching ETS functionality allows the generation of virtual training fingerprints. The envisioned procedure of generating virtual training fingerprints is further depicted in Figure 4.9. Based on the locations of the original training fingerprints stored in the ETS, a set of virtual training points is defined by leveraging a specific method for virtual training points definition selected by the localization service provider. In the next step, in the defined virtual training points the RSSI values from different WiFi APs are modeled based on a propagation model selected by the localization service provider. Finally, the generated virtual fingerprints are stored in the ETS together with the original set of training fingerprints.

Design and Implementation of Building Blocks

A modular design of the enriching ETS functionality allows easy replacement of a fingerprinting algorithm, as well as the usage of different methods for defining virtual training points and generating virtual training fingerprints, depending on the localization service provider's preferences.

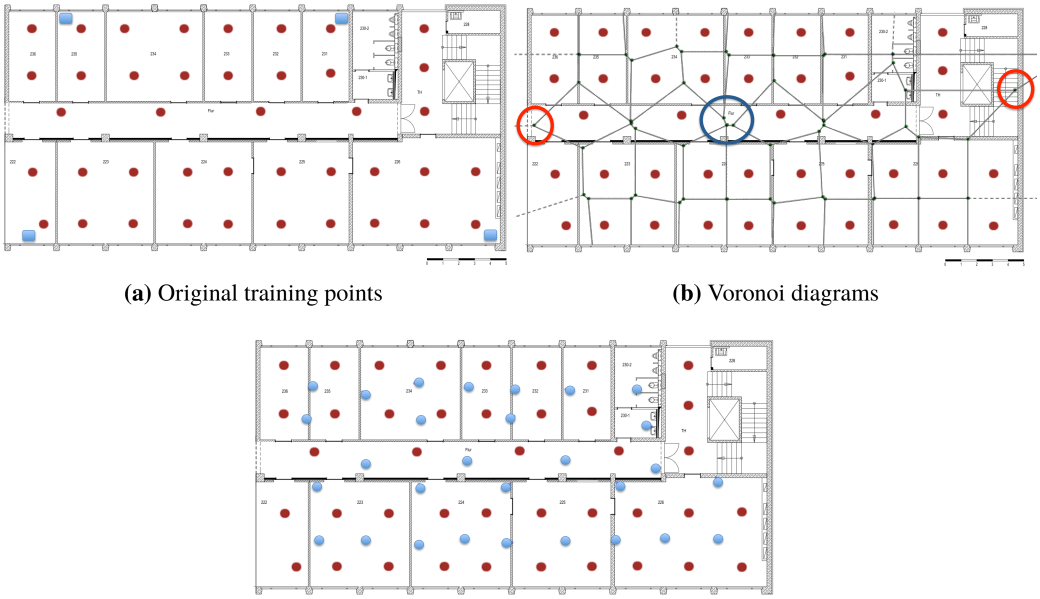
Furthermore, additional methods can be implemented for both the definition of virtual training points and for the generation of virtual training fingerprints. In this section, we overview the currently supported fingerprinting algorithms and methods implemented in modules for defining virtual training points and for generating virtual training fingerprints based on propagation modeling.

Defining Virtual Training Points

The current implementation of the enriching functionality of ETS features two methods for defining virtual training points.

Localization Service Provider's Input: This method allows the localization service provider to define coordinates of intended virtual training points as an input to the following processing block. The coordinates are envisioned to be provided in <*x,y,z*> format in a designated text file. Defining virtual training points based on localization service provider's input is a traditional method adopted in most of the literature, which motivated us to implement it as a method in the current implementation of the enriching functionality of the ETS. However, the drawback of this method is that it puts the burden of defining virtual training points on the localization service provider. Also, by leveraging this method, it is possible to (by mistake) define virtual points that are close to one another or to some of the original training fingerprints, which only results in increased latency of providing location estimates, without benefiting the accuracy of fingerprinting.

Modified Voronoi Diagrams: We designed and implemented this method for defining virtual training points based on the concept of Voronoi diagrams [168]. For a given set of training points, the Voronoi diagrams create regions in an environment, where each region consists of all points that have the smallest distance from one of the original training points. An example is given in Figure 4.10. In Figure 4.10a), the locations of the original training points are depicted with red dots. Figure 4.10b), in addition to the original set of training points, depicts the Voronoi decomposition of the environment. As visible in the figure, the environment is partitioned in a set of Voronoi regions, and those regions intersect in points that are known as Voronoi vertices. Voronoi vertices are points equidistant to three or more of the original points, and here we use them as virtual training points. However, due to the irregularities in the original training grid, the Voronoi vertices can be relatively close to one another, as the ones inside of a blue circle depicted in Figure 4.10b). This would result in multiple virtual training points that are relatively close to each other, which would ultimately result in increased latency of generating location estimates without benefiting the accuracy of fingerprinting. Due to that, on top of the obtained Voronoi vertices we apply the following modification. We firstly calculate the minimum nearest neighbor distance of the original set of training points. We use the calculated distance as a metric for detecting the Voronoi vertices that are relatively close to one another. If the distance between two or more virtual training points is less than half of the minimum nearest neighbor distance between the original training points, we consider these Voronoi vertices as relatively close to one another. In case two or more Voronoi vertices are relatively close to one another, we merge these Voronoi vertices based on their average value. Intuitively, one Voronoi vertice should be defined in the region between two or more adjacent original training points. In case more than one vertice is defined, the modification based on minimum nearest neighbor distance will detect that and merge the defined vertices into one virtual training point. Moreover, we limit the area of the Voronoi vertices to the minimum and maximum coordinates of the original training points. FOr example, the Voronoi vertices inside the red circles in Figure 4.10b) are not included in the set of virtual training points. The final results of defining virtual training points based on the modified Voronoi diagrams is given in Figure 4.10c), where blue dots indicate locations of virtual training points. The benefits of this method are that it does not require localization service provider's input and that it merges fingerprints that are close to one another according to the minimum nearest neighbor distance criteria.



(a) Original training points

(b) Voronoi diagrams

(c) Final training points (original + virtual)

Figure 4.10: Defining virtual training points using modified Voronoi diagrams

Propagation Modeling

This section provides an overview of the two propagation models currently implemented as part of the ETS. More precisely, the first presented "model" is in fact an interpolation procedure, while the second one is a propagation model in its full meaning. This shows the capability of ETS in supporting both interpolation procedures and propagation models for generation of virtual training fingerprints.

Inverse Distance Weighted Interpolation Model: The first propagation model is a simple and well-known IDWI model [169], in which the modeled values are based on the previously collected ones. The benefit of this model lies in the fact that it only depends on the originally collected fingerprints, i.e. it does not require any additional input from the service provider. However, due to its simplicity, substantial errors are anticipated in the modeled power levels, which makes this model less beneficial in terms of improving the accuracy of fingerprinting. This shortcoming is specifically emphasized for a small number of original training fingerprints, since the inputs are in that case highly limited.

In this model, weights are given to the measurements according to the inverse of their distance to a point in which WiFi signals are to be interpolated. Finding the interpolated value Iv at a point p is based on fingerprints $Iv_i = Iv\left(p_i\right)$ for $i = 1, 2, ..., N$ (N being the number of points, d being the distance function) and given as follows:

$$Iv\left(p\right) = \sum_{i=1}^{N} \frac{w_i\left(p\right) Iv_i}{\sum\limits_{j=1}^{N} w_j\left(p\right)}, \text{ where: } w_i\left(p\right) = \frac{1}{d\left(p, p_i\right)} \tag{4.4}$$

Multi-Wall Model: The second propagation model is the COST 231 multi-wall and floor model for indoor radio propagation [170], with its applicability for generation of virtual training fingerprints being demonstrated in [171]. In comparison to the previous model, this model takes into account the type and number of walls, floors or obstacles in the environment, as well as the locations of transmitting WiFi devices, which increases the burden on the localization service provider that has to specify both. However, this model is anticipated to capture in a better way a propagation environment, which is more beneficial for the accuracy of fingerprinting, in comparison to the IDWI model. The attenuation in the MW model is modeled with the following equation:

$$L(d) = l_0 + 10\gamma \log(d) + M_w. \tag{4.5}$$

The first attenuation contribution in the model is a generic and widely known one-slope term that relates the difference between transmitted and received power to the distance d. Two parameters influence the attenuation in this term: the constant l_0 (the path-loss at 1 m distance and at the center frequency of 2.4 GHz) and the path-loss exponent γ. The second attenuation contribution is the linear wall/floor/obstacle term. The number of obstacles N in the direct path between the transmitter and the receiver is counted and for each type of obstacle an attenuation contribution l_w is assumed, resulting in the total attenuation of all walls/obstacles in the direct path between the transmitter and the receiver M_w. Given the model and the site-specific measurements collected in an environment, a simple least square fitting procedure can be leveraged, which allows minimization of the differences between powers P_m, measured in each m-th ($m = 1, 2, ..., M$) training point from all used APs, and the model estimated received power $EIRP - L\left(d_m\right)$, where $EIRP$ denotes the effective isotropic radiated power at the transmitter. The equation is given by:

$$\{l_c, \gamma, l_w\}_{opt} = \arg\min_{l_c,\gamma,l_w} \{\sum_{m=0}^{M-1} |P_m - (EIRP - L(d_m))|^2\} \tag{4.6}$$

where $L(d_m)$ contains both the attenuation from a power-distance relation and the attenuation of each wall/ floor/obstacle. Further, l_c is a constant used for optimizing the minimization of the cost in a multi-wall model, which includes the influence of the parameter l_0. Using the calculated parameters $\{l_c, \gamma, l_w\}_{opt}$, a footprint of an environment, and locations of WiFi APs as inputs, the WiFi signal power levels at virtual training points can be modeled.

Fingerprinting Algorithms

The ETS currently enables the usage of two previously discussed WiFi RSSI-based fingerprinting algorithms summarized in Chapter 2 of this thesis. These algorithms leverage different types of fingerprints, which illustrates the capability of ETS to accommodate different types of fingerprinting algorithms.

4.2.2 Implementation of Enriched Training Service

In this section, we shortly overview the implementation of the ETS which enables the following features: extensibility, fast and reliable remote access, and language and platform independence. The ETS implementation is based on the implementation of our web-based platform for streamlined evaluation of RF-based indoor localization algorithms (Chapter 3), where similar features have been selected to support data storage.

The ETS is a web service implemented in Python 2.7 using the Flask module, which provides a simple way of creating RESTful web services. The training fingerprints are stored in a MongoDB database, an open-source document database and the leading NoSQL database written in C++. A fingerprint is presented as a Protocol Buffer structure, a way of encoding structured data using an efficient and extensible binary format. The extensibility of the stored fingerprints is achieved using the Protocol Buffer for defining a fingerprint structure and MongoDB database for storing those fingerprints. This feature enables an easy storage of different types of training fingerprints, burdening the localization service provider only with the necessary modification of the Protocol Buffer message reflecting a new type of fingerprint. By using a NoSQL type of database, the ETS enables storage of any type of defined message, without a need of changing the schema and/or the database itself. The RESTful design and the implementation as a web-service enable remote access to the ETS using only HTTP requests. Protocol Buffers serialize messages into binary streams which support fast communication between the localization service provider and the ETS service. Due to the fact that communication with the ETS service is done using HTTP requests, it is possible to manage data from different platforms, and also using different programming languages, since most of the modern languages provide libraries enabling HTTP requests.

4.2.3 Experimental Evaluation

The ETS provides two clear benefits for the localization service providers: a possibility of straightforward storage and management of raw data and different types of training fingerprints, and an automatic generation of virtual training fingerprints without localization service provider' efforts. Both of these benefits reduce the overheads of deploying a fingerprinting-based localization solution. In the following, we aim on showing that it is beneficial to use the ETS and its enriching functionality for improving the accuracy of a deployed WiFi fingerprinting-based localization solution.

The experimental evaluation in this work was performed by following the methodology for experimental evaluation of RF-based indoor localization algorithms an solutions discussed in Chapter 3 of this thesis. The result of the evaluation is a set of metrics characterizing the performance of indoor localization algorithms or solutions. This set includes point accuracy, room-level accuracy and processing time. Point accuracy is represented by the Euclidean distance between the estimated and ground-truth coordinates. Room-level accuracy is a binary metric that states the correctness of the estimated room. Finally, processing time is defined as the time needed for an algorithm to produce a location estimate, given the raw RSSI measurements are collected and fingerprints are generated.

A training survey in this work has been collected using a specifically designed testbed infrastructure for evaluation of RF-based indoor localization algorithms presented in the previous chapter of this thesis. The leveraged testbed infrastructure offers the capability of collecting highly accurate measurements with reduced external influences, such as uncontrolled interference or movements, influence of experimenter's body, etc. Moreover, the measurements collection was performed on a weekend afternoon, further minimizing the external influences.

The environment used for the evaluation is given in Figure 4.7a), with locations of the original training fingerprints labeled with red dots and with locations of the WiFi APs labeled with blue squares. The WiFi APs were configured to operate on the IEEE 802.11b channel 11 (2462 MHz), with transmission power of 20 dBm (100 mW). For the evaluation purposes two sets of measurements were collected at 20 evaluation points with their locations indicated in Figure 4.3.

Two repetitions of measurements at the same locations provide additional insights in the temporal stability of the obtained results, which strengthens the reliability of our observations. While only two repetitions of the same experiment are not sufficient to provide statistical benefits, the comparability of results obtained in these repetitions excludes the possibility of a sudden change in the performance (e.g. due to interference, movements, changes in the environment), which could lead to errors in the conclusions. The collected measurements were stored in the previously discussed web-based platform for streamlined experimental evaluation of RF-based indoor localization algorithms using previously collected raw data traces. The platform provides a simple way of reusing the same datasets for multiple evaluations, and by leveraging this functionality we were able to reuse the same set of measurements for generation of different types of fingerprints and for the evaluation of different algorithms.

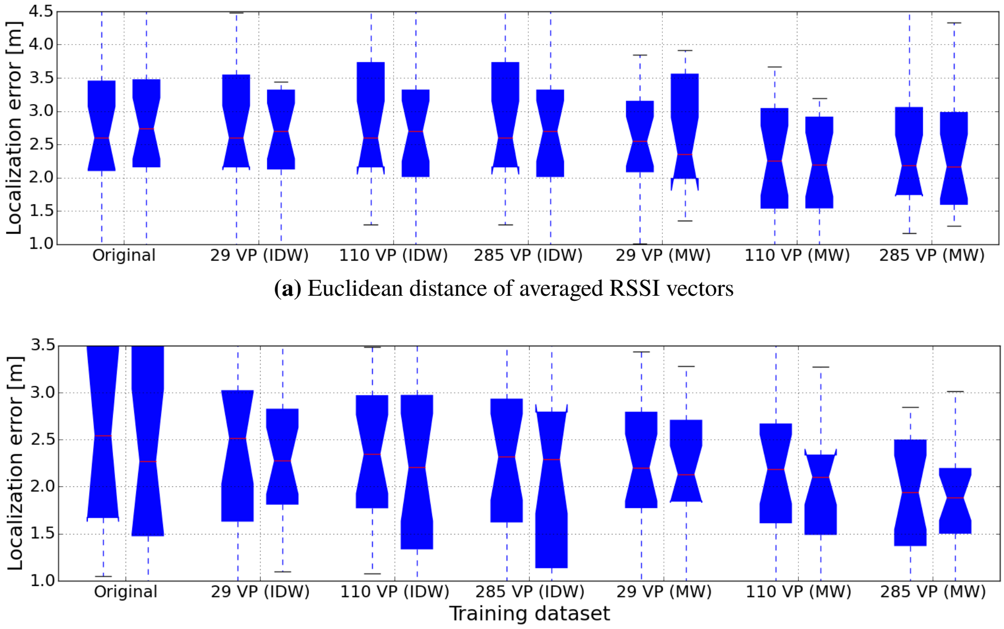
4.2.4 Evaluation Results

Firstly, we collected the original set of training fingerprints in our evaluation environment, with locations of fingerprints indicated in Figure 4.10a), and we stored them in the ETS. Then, by leveraging the enriching ETS functionality, additional virtual training fingerprints were generated in three iterations. By leveraging the original set of 41 training fingerprints, in the first iteration 29 virtual training fingerprints were generated based on modified Voronoi diagrams and leveraging both propagation models. In the second iteration, the original set of training fingerprints and the 29 previously generated virtual training fingerprints were used as an input to the procedure, which resulted in all together 110 virtual training fingerprints (including the 29 fingerprints generated in the first iteration). Similarly, the third iteration yielded in summary 285 virtual training fingerprints. Defining the locations of those 285 virtual training fingerprints manually would take a substantial amount of time and efforts, while defining them by leveraging the proposed procedure based on modified Voronoi diagrams was performed automatically without burdening the localization service provider, which demonstrates the benefit of the proposed procedure.

In Figure 4.11, the accuracy of the used fingerprinting algorithms is depicted in case when the original training set is used and in case when additional virtual training fingerprints are generated based on the modified Voronoi diagrams and leveraging the two described propagation models. As presented in the figure in a regular box-plot fashion, for both repetition of the experiments and for both used algorithms, in case a simple IDWI model is used, there is almost no improvements in the accuracy of fingerprinting. However, in case a more complex MW model is used, the improvement in accuracy is visible and for both fingerprinting algorithms it increases with a number of virtual training fingerprints. The summary of achieved results is given in Table 4.9. In the best case scenario, the evaluation results show that the average localization error of the algorithm "Pompieu-Hausdorff distance of RSSI quantiles" decreases from roughly 2.5 m to less than 1.8 m, due to the generation of 285 virtual fingerprints, which is an improvement of roughly 28%. The improvement in the accuracy is less emphasized for the algorithm "Euclidean distance of averaged RSSI vectors", which indicates that different gains in accuracy can be expected for different algorithms.

An increase in the number of training points generally increases the processing time of a fingerprinting algorithm, since the user's generated fingerprint has to be compared with a larger number of training fingerprints. We evaluated the processing time of the used fingerprinting algorithms by requesting for each of the 20 evaluation points 100 times the location estimates. The time needed for providing each location estimate was measured and afterwards the statistical information about the processing time needed for providing one location estimate was calculated. The increase in the processing time of the used fingerprinting algorithms, due to the generation of the virtual training fingerprints, is given in Table 4.8. As visible from the table, the increase in the accuracy comes at the cost of an increased processing time. For the aforementioned example, the increase of 28% in the accuracy of fingerprinting comes at the cost of roughly 45% increase in the processing time of the algorithm. However, this time is not a dominant factor in the latency of fingerprinting, since the sampling of the WiFi environment takes 2-3 s, depending on the hardware and device drivers.

The obtained results show that the ETS' enriching functionality for generating virtual training fingerprints improves the accuracy of WiFi fingerprinting-based localization. An improvement in accuracy of fingerprinting can also be achieved by physically collecting additional fingerprints and storing them into the database. However, such an approach poses an additional overhead, which is removed by leveraging the ETS' enriching functionality. Hence, the improvement in accuracy achieved by leveraging the ETS' enriching functionality can alternatively be perceived as a reduction in deployment overheads in WiFi-based fingerprinting. Example-wise, if the performance of a deployed fingerprinting algorithm in a given environment is not satisfactory, the localization service provider could resort to generating virtual training fingerprints using the enriching functionality of the ETS, instead of physically collecting additional fingerprints.



(a) Euclidean distance of averaged RSSI vectors

(b) Pompieu-Hausdorff distance of RSSI quantiles

Figure 4.11: Localization accuracies with and without virtual training fingerprints

Table 4.8: Algorithms' processing time with and without virtual training fingerprints

	Original	**29 VTP**	**110 VTP**	**285 VTP**
	Euclidean distance of averaged RSSI vectors			
Mean time [s]	0.37	0.39	0.43	0.76
Median time [s]	0.36	0.37	0.40	0.72
Min/max time [s]	0.32/ 0.46	0.33/0.69	0.36/0.85	0.58/1.39
	Pompeiu-Hausdorff distance of RSSI quantiles			
Mean time [s]	0.53	0.58	0.76	1.20
Median time [s]	0.50	0.59	0.74	1.18
Min/max time [s]	0.42/0.69	0.43/0.75	0.57/0.97	0.88/1.63

Table 4.9: Summarized localization accuracies with and without virtual training fingerprints

	Original		**IDWI 29 VTP**		**IDWI 110 VTP**		**IDWI 285 VTP**		**MW 29 VTP**		**MW 110 VTP**		**MW 285 VTP**	
	Rep. 1	Rep. 2	Rep. 1	Rep. 2	Rep. 1	Rep. 2	Rep. 1	Rep. 2	Rep. 1	Rep. 2	Rep. 1	Rep. 2	Rep. 1	Rep. 2
Euclidean distance of averaged RSSI vectors														
Av. error [m]	2.73	2.97	3.09	3.14	3.12	3.25	3.12	3.25	2.69	2.83	2.46	2.46	2.48	2.38
Median error [m]	2.59	2.74	2.59	2.70	2.59	2.70	2.59	2.70	2.54	2.35	2.25	2.19	2.17	2.16
Min error [m]	0.78	0.98	0.92	0.98	1.29	0.98	1.29	0.98	1.01	1.35	0.97	0.14	1.16	1.27
Max error [m]	5.24	5.68	6.88	6.31	6.18	6.02	6.18	6.02	6.06	6.29	6.42	5.07	4.77	4.33
Room acc. [%]	65.0	65.0	65.0	65.0	70.0	65.0	70.0	65.0	70.0	70.0	75.0	75.0	80.0	80.0
Pompeiu-Hausdorff distance of RSSI quantiles														
Av. error [m]	2.98	2.54	2.37	2.31	2.35	2.26	2.28	2.11	2.21	2.13	2.22	1.94	1.82	1.82
Median error [m]	2.54	2.26	2.51	2.27	2.34	2.20	2.31	2.29	2.20	2.13	2.18	2.10	1.93	1.88
Min error [m]	1.05	0.23	0.77	1.09	1.07	0.77	0.92	0.57	0.52	0.48	0.62	0.68	0.10	0.06
Max error [m]	6.22	5.20	3.38	3.67	3.48	3.83	3.63	3.63	3.43	3.27	4.87	3.27	2.84	3.01
Room acc. [%]	75.0	75.0	75.0	75.0	75.0	80.0	80.0	80.0	80.0	80.0	80.0	85.0	85.0	90.0

4.3 Increasing Interference Robustness of WiFi Fingerprinting-based Localization

The number of wirelessly connected devices is constantly increasing, as well as the wireless traffic generated by each device [172]. Naturally, with the increase in the number of connected devices and in the overall traffic among those devices, the probability of wireless interference is also increasing. We have shown previously in the thesis that RF interference can significantly degrade the performance of RF-based indoor localization solutions, including WiFi fingerprinting-based ones. This negative effects of RF interference of WiFi-based fingerprinting should obviously be mitigated, which is still an unresolved challenge.

We address this challenge by proposing a procedure for reducing the effect of RF interference on WiFi RSSI-based fingerprinting algorithms. The proposed procedure, to which we refer as *interference mitigation procedure*, can be applied after the RSSI measurements collection procedure of fingerprinting algorithms in order to adjust the RSSI measurements before their further processing for estimating location. Our procedure is based on the assumption that a fingerprinting system leverages a certain level of environmental awareness, i.e. spectrum power levels in the 2.4 GHz ISM frequency band, in order to estimate the average interference power levels at different WiFi channels in the 2.4 GHz ISM frequency band. By using the estimated average interference power levels it is possible to calculate the additive variability of the RSSI measurements due to RF interference. Additive variability of RSSI measurements is here defined simply as the change in RSSIs due to RF interference and this term should not be confused with statistical variance-like terms. Once this variability is estimated, the proposed procedure leverages this information for adjusting RSSI measurements in both phases of fingerprinting, which improves the performance of fingerprinting algorithms. For adjusting the RSSI measurements we adopt a theoretically characterized effect that RF interference has on RSSI measurements [99]. This effect, depending on the interference power, ranges from no practical influence, to a linear correlation with RSSI variability, and finally to a full packet-loss.

4.3.1 A Theoretical View on Interference Effects on RSSI Measurements

In this section, a theoretical framework from [99] is adopted to study the effect of interference on RSS values. Note that this section is not the author's sole contribution and is given here for completeness purposes.

We assume an Additive White Gaussian Noise (AWGN) channel, a signal source with transmission power P_X', a destination with noise variance N, and an interferer. The interferer is a continuously transmitting source with fixed transmission power P_I'. The transmission rate of the source is r and it is also assumed that RSS values can only be obtained if the destination can correctly decode the source messages. This assumption is used to model the packet-based nature of RSS values, where RSS values, reported by some technologies, can only be obtained if the packet is correctly decoded. If all powers are measured in mW, it is shown in [99] that the effect of interference on the RSS value is additive, until the moment that the destination Signal to Interference plus Noise Ratio (SINR) is below the necessary level for decoding source messages. To observe the change of δ dB in RSS value at the receiver, the interference power should increase above certain threshold which is dependent on δ. As shown in [99], the change of RSS value in dB due to RF interference can be described as follows:

$$\delta = 10 \log \left(\frac{P_I + P_X + N}{P_X + N} \right), \tag{4.7}$$

where P_X and P_I are signal and interference power *at the receiver*, respectively. However, increasing interference power P_I at the receiver leads to decreasing SINR, which eventually falls below the threshold γ_{snr} needed for correctly decoding the message, hence no RSS value can be reported. In this situation, the maximum observable change of RSS value at the receiver is as follows:

$$\delta_{max} = 10 \log \left(1 + \frac{1}{\gamma_{snr}} \right). \tag{4.8}$$

The interesting point is that if δ_{max} is too small, with respect to the previously set threshold, then no change in RSS value is observed as long as the source message is correctly decoded. The visibility threshold δ represents the change in RSS value at the receiver that can be observed and interpreted as an increase by interference. Based on this threshold, we can distinguish three different operational regimes. In the noise-limited regime, RSS values at the receiver are not affected by RF interference. In the interference-limited regime, RSS values at the receiver are changed with interference power according to Equation 4.7. Finally, in the collision regime, RSS values at the receiver cannot be reported, since the source messages can not be decoded correctly, which corresponds to the case when the full packet-loss occurs. A graphical presentation of different regimes detected in our previous work is given in Figure 4.13. In the figure, red and blue lines show the increase in variability of received RSS values and decrease in the Packet Reception Rate (PRR) with the increase in the interference power at the receiver, respectively.

As it can be seen in Equation 4.7, the amount of change in RSS value can be measured if one has access to P_I and $P_X + N$. The destination can measure total received power $P_{RX} = P_I + P_X + N$, which corresponds to the numerator of the fraction inside the logarithm in Equation 4.7. If the interferer transmits continuously and if the destination can measure the power before the start of the transmission, the destination can also find $P_I + N$. But these two values are not enough to find the change δ in RSS value. In general, when the noise power N is comparable with received interference power P_I, nothing can be said about δ based on two measurements done by the destination. However, if P_I is much bigger than N, the value $P_I + N$ can be roughly approximated by P_I and hence δ is given as follows:

$$\delta = 10 \log \left(\frac{P_{RX}}{P_{RX} - P_I} \right), \tag{4.9}$$

where $P_X = P_{RX} - P_I$. The previous equation provides the change of RSS in dB in terms of two measurements done by the destination. Let us assume that the RSSI measurements, reported by different technologies and specifically WiFi, are the quantized versions of RSS values discussed here. In this case, Equation 4.9 can be used to "correct" the RSSI measurements in interference-limited regime, under the assumption that the WiFi receiver provides the P_{RX} power level and the P_I is provided by leveraging spectrum information at the receiver.

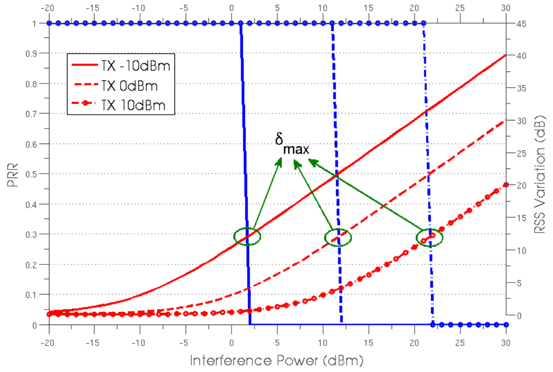


Figure 4.12: RSS and PRR change with interference power

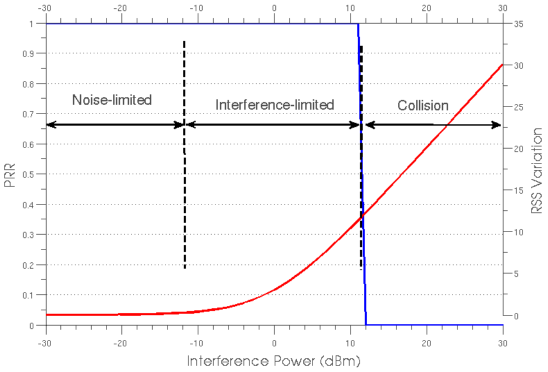


Figure 4.13: Operational regimes for RSSI-PRR variation

4.3.2 Extended Workflow of Fingerprinting Algorithms

The overview of the general workflow of WiFi RSSI-based fingerprinting is given in Figure 4.1 and in details discussed in Section 4.1 of this thesis.

Interference Mitigation Procedure

We extend this general workflow of fingerprinting algorithms by inserting a procedure for reducing the effect of RF interference on the RSSI measurements. This procedure is applied on both the RSSI measurements collected in a training survey of an environment of interest and on user's generated RSSI measurements at an unknown location. The reason lies in the fact that, in both training and runtime phases of fingerprinting algorithms, collected RSSI measurements can be influenced by RF interference. An overview of the extended workflow of fingerprinting is given in Figure 4.14.

Similar to the general fingerprinting workflow, in the workflow with inserted interference mitigation procedure, the user collects RSSI measurements at an unknown location in the environment of interest. In addition, in the raw data collection procedure, the user at the same time samples the spectrum in the 2.4 GHz ISM frequency band and sends it to the fingerprinting server. Firstly, using the spectrum scan provided by the user, an interference power estimation method is leveraged for estimating the average interference power levels using the spectrum power levels. Using the user's provided beacon packets RSSI measurements, which also indicate the operating IEEE 802.11 channel of each AP, and using the estimated average interference power level on each IEEE 802.11 channel, according to Equation 4.9 the additive variability for RSSI measurements from each AP can be estimated. The estimated variability for each AP is then subtracted from original RSSI measurements and these adjusted measurements are used as an input to the fingerprint creation procedure. The fingerprinting workflow continues further in the same way as in the general workflow of fingerprinting, with the exception that the same interference mitigation procedure has to be performed in the training step of fingerprinting to adjust RSSI measurements before storing them in a training database.

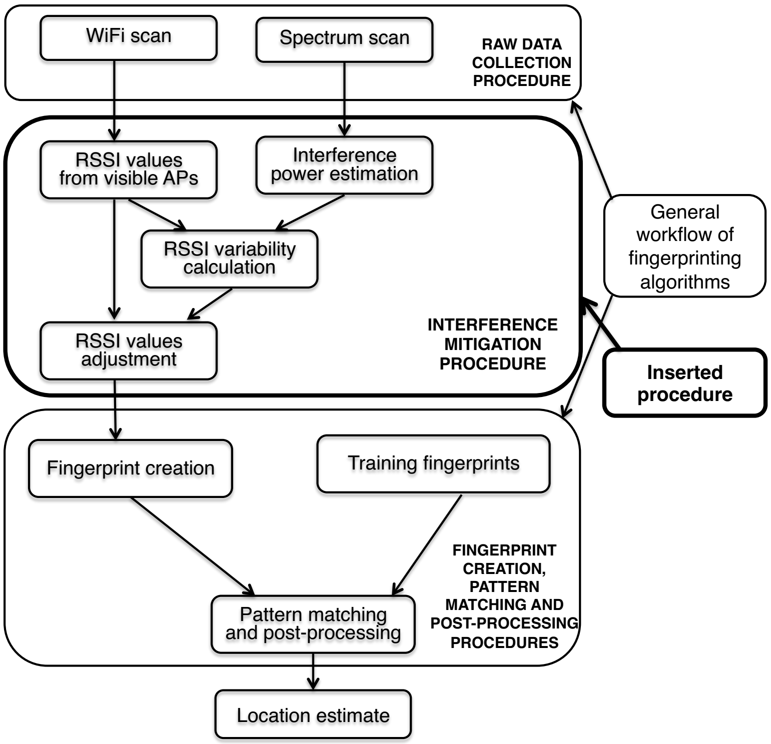

WiFi scan
Spectrum scan
RAW DATA COLLECTION PROCEDURE
RSSI values from visible APs
Interference power estimation
RSSI variability calculation
RSSI values adjustment
INTERFERENCE MITIGATION PROCEDURE
General workflow of fingerprinting algorithms
Inserted procedure
Fingerprint creation
Training fingerprints
FINGERPRINT CREATION, PATTERN MATCHING AND POST-PROCESSING PROCEDURES
Pattern matching and post-processing
Location estimate

Figure 4.14: Extended workflow of WiFi fingerprinting algorithms

Interference Power Estimation

To this end, we propose two methods for average interference power estimation using the sampled power levels on different sampling frequencies in the spectrum.

Interference Power Estimation Method 1: The first method uses sampled spectrum power levels on different sampling frequencies for estimating average interference power levels at a receiver on each IEEE 802.11 channel in the 2.4 GHz ISM frequency band. The method first filters all samples that are considered as noise, i.e. smaller than the threshold of -99 dBm. This threshold is selected because it is a standard noise level for WiFi devices [53] and because the power level below this threshold has no practical influence on RSSI measurements, as discussed previously. Second, the method clusters the samples sampled on different sampling frequencies into groups corresponding to IEEE 802.11 channels. For example, all samples taken at sampling frequencies between 2400 and 2422 MHz are arranged in a group that corresponds to IEEE 802.11 channel 1. The frequencies are clustered corresponding to IEEE 802.11 channels because RSSI measurements used in fingerprinting algorithms are related to a IEEE 802.11 channel, making it reasonable to estimate the average interference power level corresponding to IEEE 802.11 channels. Finally, the method averages (both over time and over all sampling frequencies in a particular group) all the samples in each group and the result of averaging is a number that represents the estimated interference power level on a given IEEE 802.11 channel.

Interference Power Estimation Method 2: The second interference power estimation method, additionally to the sampled spectrum power levels, uses the probability of interference occurrence for estimating the interference power at the receiver. Instead of only averaging samples in each group, this method takes into account the ratio between the number of samples in which interference occurs and a total number of samples in one group (including noise samples). The method uses the calculated ratio as a weight with which the averaged samples in each group are multiplied. This ratio is a rough estimate of the probability of interference occurrence, and is used with the following intuition. If one IEEE 802.11 channel is highly used and the other one is less used, but they both have the same spectrum power levels when interference occurs, this methods will provide different average interference power level estimates for these two channels, while the previous method will result in the same estimate of the average interference power.

The proposed average interference power estimation methods at the receiver can reasonably be used under the condition that a time during which the signal occupies the spectrum is much lower than a time when there is interference in the spectrum. Otherwise, the methods would provide the estimated signal level from the spectrum information. Since for WiFi fingerprinting RSSI measurements from beacon packets are used, and beacon packets are by default transmitted periodically every 100 ms and have a short duration of around 50 μs depending on the transmission rate, the described methods can reasonably be used and will provide an estimate of the average interference power levels at different IEEE 802.11 channels in the 2.4 GHz frequency band.

The accuracy of the average interference power estimation based on the spectrum information depends on the spectrum sampling frequency. Naturally, if the sampling frequency is higher than Nyquist rate and therefore it is high enough to realistically capture the changes in spectrum power levels, the interference power can be obtained using the second method. If the sampling frequency is lower than Nyquist rate, it is possible to envision the usage of sparse signal processing methods, e.g. [173] to estimate precisely the average interference power. However, these methods need a specific spectrum sensing design. Current hardware, with its limitations, generally cannot support such high sampling frequencies and specific sensing design. Due to that, our average interference power estimation methods are based on two different heuristics. The first method considers a worst case scenario, i.e. the interference is present for the whole duration of signal transmission and therefore the average interference power level in the spectrum corresponds to the average value of powers during the measurement period. In other words, if the beacon packet is hit by interference, it endures the interference in its whole transmission duration. This provides the justification for filtering the spectrum samples below certain threshold. Worst case scenario is reasonable when an interference source is designed to transmit whenever the beacon packet is transmitted. In the second proposed average interference power estimation method, the main assumption is that interference can be present or absent during signal transmission. Particularly in this case, a beacon packet in its duration can be equally affected or not affected by interference. Therefore in the long run, the average interference effect on received power is dependent on the probability that the interference is present. The second proposed average interference power estimation method then estimates this probability by calculating the ratio of samples in which interference is transmitting to total number of samples.

The knowledge about the spectrum power levels at an unknown location is expected to be available for mobile devices to be localized. In the era of cognitive radios and seamless cooperation between heterogeneous devices this knowledge will become essential in enabling all the envisioned capabilities. Either in a form of a connectivity brokerage as a central entity for providing the information [174, 175], or by embedding the spectrum sensing capabilities in the end-devices [176], the information about the spectrum power levels is expected to be available. It is already possible to leverage the functionalities of various WiFi chipsets to measure 2.4 GHz spectrum information. Under this assumption we use the information about the spectrum as given, in order to show the feasibility of our system as a proof-of-concept.

4.3.3 Experimental Evaluation

The aim of the experimental evaluation is to evaluate if and to which extent the developed interference mitigation procedure helps in mitigating the previously observed negative impact of RF interference on the performance of WiFi RSSI-based indoor fingerprinting.

The evaluation methodology follows the previously discussed methodology for experimental evaluation of RF-based indoor localization solutions under RF interference. The result of the evaluation is a set of performance metrics, i.e. the point accuracy, room-level accuracy and processing time of the evaluated fingerprinting algorithms. In the evaluation, we used the two WiFi RSSI-based fingerprinting algorithms whose overviews are given in Section 2 of this thesis.

The evaluation experiments were performed using the previously discussed testbed infrastructure specifically designed for the evaluation and benchmarking of RF-based indoor localization algorithms in the scenarios with artificially generated RF interference context. The leveraged testbed infrastructure provides the possibility of collecting accurate measurements with minimized external influences. The experiments were performed during weekend afternoons, so the influence of uncontrolled interference, people walking and slight movements of objects (chairs, tables) in the testbed premises have been minimized.

As described previously, fingerprinting algorithms require a training step in which the localization environment is surveyed for a set of distinguishable features (training fingerprints). The locations of training points for both fingerprinting algorithms used in the evaluation are given in Figure 4.2. The testbed infrastructure was used for collecting the raw RSSI measurements in different interference scenarios. For each interference scenario two sets of measurements were collected at 20 evaluation points with locations indicated in Figure 4.3.

Two repetitions of the same experiment provide additional insight into the temporal stability of the obtained results, which strengthens the reliability of our observations. The collected raw RSSI measurements were stored in the previously discussed platform for streamlined experimental evaluation of RF-based indoor localization algorithms using precollected raw datasets. This web-based platform provides a simple way of reusing datasets for evaluation of different RF-based indoor localization algorithms. By leveraging the platform, we were able to use the same raw datasets for the evaluation of the two described fingerprint algorithms without and with the procedure of mitigating the influence of RF interference and with two methods for estimating interference power levels in the interference mitigation procedure.

The evaluation was performed in four evaluation scenarios in details discussed in Chapter 3 of this thesis (a reference scenario and three interference scenarios). The spectrum power levels at an example location in the environment for each scenario is depicted in Figure 3.15. The leveraged interference scenarios do not represent a real-life interference context in which multiple sources of different types of interference are expected. On the contrary, the generated interference scenarios are simplistic on purpose, which gives an opportunity to evaluate if and to what extent interference of a particular type can influence the performance of evaluated algorithms. Note that the raw RSSI collection that used as a training dataset was collected under no artificial interference, while only the datasets that were used for the runtime phase of fingerprinting were collected under different evaluation scenarios (a reference and three interference scenarios).

As a short reminder, in the reference scenario no artificial interference was generated, making the performance of fingerprinting algorithms achieved in this scenario a "reference" for evaluating the impact of interference generated in the interference scenarios. The first interference scenario was comprised of several interference sources that are typical for office or home environments. Interference was emulated using 4 WiFi embedded PCs having the roles of a server, access point, data client, and video client. In the second interference scenario, the interference was created using the IEEE 802.15.4 Tmote Sky nodes. The interference type was jamming on one IEEE 802.15.4 channel with a constant transmit power equal to 0 dBm. Five of these jamming nodes were present in the testbed environment. For the third interference scenario, a signal generator was used to generate synthetic interference with an envelope that reflects WiFi modulated signals, but without CS. The transmission power was set to 20 dBm, while the wireless channel was set on WiFi channel 11.

The environment used for evaluation is given in Figure 4.2, in which also the locations of training points are indicated with red dots. Fingerprinting algorithms leverage the information from all visible WiFi APs in the 2.4 GHz ISM frequency band. In addition to that, we set-up four additional WiFi APs in the corners of the environment, with their locations marked in figure with black dots. Those APs are configured to operate on IEEE 802.11 channel 11 (2462 MHz) with the transmission power of 20 dBm (100 mW). The used traffic model is IEEE 802.11b. These APs are working constantly and generally provide the highest RSSI measurements in the environment of interest, meaning that they have a high relevance for the fingerprinting procedure.

As a mobile device for collecting the raw RSSI measurements a MacBook Pro laptop with an AirPort NIC was used. We leveraged the RSSI measurements provided by the AirPort driver's method *airport –scan*, which provided us a scan of all visible WiFi APs every 0.5 sec. Twenty of these scans were used in the training phase, and eight in the runtime phase of fingerprinting.

For the collection of the spectrum measurements we used a WiSpy 2.4x device, which provides power levels in 2.4 GHz spectrum. The WiSpy device samples the spectrum by sweeping through the frequencies from 2400 MHz to 2495 MHz with a frequency step of 333 kHz, with the sweep time of 507 ms. The example spectrum information provided by a WiSpy device is graphically presented in Figure 3.15. This type of information was used for estimation the interference power levels at each evaluation point. While a WiSpy device is limited in the granularity of spectrum and sweeping frequency, the obtained data is sufficient for use as a proof-of-concept.

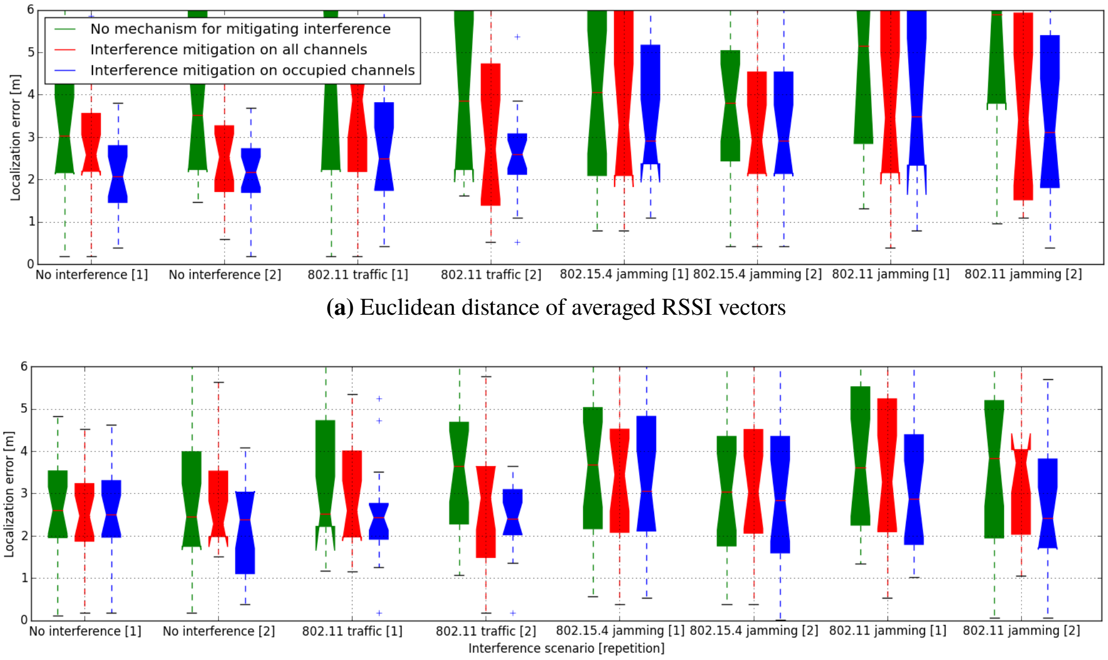
4.3.4 Evaluation Results

The distributions of localization errors achieved by two fingerprint algorithms in four interference scenarios are depicted in Figure 4.15 in a regular box-plot fashion. Furthermore, statistical information about the point and room-level accuracies achieved by two algorithms is given in Table 4.10. In the following we discuss three major observations.

As for the first observation, the results show that RF interference can increase the achieved localization errors and, thus, decrease the accuracy of the evaluated fingerprinting algorithms. The effect of interference can be observed by comparing the localization errors obtained by the algorithms without applying the interference mitigation procedure. In other words, the localization errors obtained in the evaluation in different interference scenarios are generally higher than the errors obtained in the scenario where no artificial interference is generated. It can also be observed that the smallest influence of interference on the accuracy of the algorithms occurs when the interference source is IEEE 802.11 traffic. The reason for this effect is the standard Carrier Sense Multiple Access (CSMA) mechanism in the IEEE 802.11 APs, which reduces the possibility of interfering with the beacon packets used for the localization purposes. The influence of interference is generally higher when interference source is jamming, since the CSMA mechanism does not exist. Finally, in the jamming scenarios, higher degradation of the accuracy of fingerprinting algorithms is observed when jamming is performed on a IEEE 802.11 channel. The reason for that is the higher transmission power (20 dBm) and wider channel (20 MHz), in comparison to the scenario where jamming is done on a IEEE 802.15.4 channel with transmit power of 0 dBm and 2 MHz wide channels.

The second observation concerns the stability of the obtained results. While it is impossible to have an entirely stable evaluation results, due to the intrinsic randomness of wireless environments, our results show that two repetitions (in two sequential days) of the same experiment result in highly comparable localization errors and their relevance and reliability is for that reason increased. This can be seen by comparing the localization errors obtained in two different runs of the same experiment, as shown in Figure 4.15 and Table 4.10. The reason for high repeatability of the obtained results is also related to using the previously discussed testbed infrastructure for experimental evaluation of RF-based indoor localization solutions. While we cannot draw statistical claims on the stability based on only two repetitions, it does strengthen our other conclusions, i.e. that the interference can degrade the performance of fingerprinting algorithms and the interference mitigation methods are useful in improving the robustness of fingerprinting algorithms to RF interference.

The third observation is related to the improvements in the accuracy of evaluated fingerprinting algorithms by leveraging the spectrum information and applying the interference mitigation procedure. As the obtained results show, the accuracy of fingerprinting algorithms is improved in the interference scenarios when the interference mitigation procedure is used. At the same time, in the scenarios where the influence of interference on the accuracy of fingerprinting algorithms is not high, the interference mitigation procedure has no influence, i.e. it does not degrade the performance. For some scenarios, even in case no interference is generated, some improvements are visible. The reason is the uncontrolled interference in the environment, which, although minimized in our experiments, was present in the environment. The uncontrolled interference to a certain level degraded the performance of fingerprinting algorithms, thus the interference mitigation procedure in that case improved the performance of fingerprinting. It is also visible that, in general, comparable or better performance is achieved when the average interference power estimation method used in the interference mitigation procedure takes into account the probability of the occurrence of interference, in comparison to the other method for interference power estimation. We believe that the reason for that lies in the fact that the first method only performs



(a) Euclidean distance of averaged RSSI vectors

(b) Pompeiu-Hausdorff distance of RSSI quantiles

Figure 4.15: Evaluation results for both fingerprinting algorithms in different scenarios

averaging of a set of spectrum samples on a given IEEE 802.11 channel, which, for the case when interference does not occur with high probability, results in too high estimates. For that reason, the difference in performance of methods 1 and 2 is clearly visible in the interference scenario 1, where the interference is IEEE 802.11 traffic. For the jamming scenarios the probability of interference is high, so both methods achieve comparable results. The improvements in the interference robustness due to leveraging an interference mitigation method are the smallest for the case when the source of interference is different technology (interference scenario 2). The reason is the difference in signal bandwidth for the two technologies, which indicates that a more accurate estimation of the interference power would result in further improvements in the interference robustness of WiFi-based fingerprinting algorithms.

Naturally, one consequence of introducing a new interference mitigation procedure in a fingerprinting algorithm is the increase in the processing time of an algorithm, i.e. increase in time needed for reporting a location estimate. We evaluated the processing time of the two used fingerprinting algorithms by requesting the algorithms to provide 100 times the location estimates for the 20 evaluation points in the reference scenario. The time needed for providing each location estimate was measured and afterwards the statistical information about the processing time needed for providing one location estimated was calculated. The same procedure was repeated for the case when interference mitigation procedure is introduced in the fingerprinting algorithms. In Table 4.11, the obtained results are presented. Note that the presented processing times do not include the time needed for collecting the raw data, which is constant for both cases and depends on the measuring device and used device driver. Furthermore, the processing times in the

Table 4.10: Summarized evaluation results - point and room-level accuracy

	Reference		**Interference 1**		**Interference 2**		**Interference 3**	
Metrics	Rep 1	Rep 2	Rep 1	Rep 2	Rep 1	Rep 2	Rep 1	Rep 2
Euclidean distance of averaged RSSI vectors - no interference mitigation								
Median error [m]	3.03	3.51	4.73	3.85	4.05	3.81	5.16	5.89
Average error [m]	4.48	4.69	6.63	5.73	4.41	4.29	6.37	6.85
Minimum error [m]	0.17	1.46	0.17	1.61	0.79	0.41	1.31	0.96
Maximum error [m]	14.4	14.72	17.28	15.67	10.11	11.01	19.32	16.77
Room-level accuracy [%]	55.0	40.0	50.0	50.0	30.0	50.0	40.0	25.0
Euclidean distance of averaged RSSI vectors - method 1								
Median error [m]	2.57	2.52	3.87	2.71	3.27	2.91	3.45	3.40
Average error [m]	3.06	3.00	4.21	3.69	4.33	3.86	4.55	4.46
Minimum error [m]	0.17	0.58	0.17	0.52	0.79	0.41	0.38	1.09
Maximum error [m]	8.23	8.51	17.82	14.84	10.11	11.01	12.72	12.75
Room-level accuracy [%]	60.0	70.0	65.0	65.0	40.0	50.0	40.0	35.0
Euclidean distance of averaged RSSI vectors - method 2								
Median error [m]	2.06	2.17	2.49	2.59	2.91	2.91	3.48	3.12
Average error [m]	2.42	2.46	2.93	2.72	3.84	3.86	4.94	4.32
Minimum error [m]	0.38	0.18	0.41	0.52	1.01	0.41	0.79	0.38
Maximum error [m]	7.11	6.57	7.60	6.19	10.27	1.01	11.76	12.72
Room-level accuracy [%]	75.0	80.0	80.0	75.0	40.0	50.0	45.0	50.0
Pompeiu-Hausdorff distance of RSSI quantiles - no interference mitigation								
Median error [m]	2.60	2.45	2.52	3.63	3.62	3.04	3.61	3.83
Average error [m]	2.81	2.91	3.38	3.59	3.81	3.67	4.66	3.90
Minimum error [m]	0.11	0.19	1.17	1.07	0.56	0.38	1.34	0.05
Maximum error [m]	6.10	6.41	6.36	6.45	8.31	11.20	14.66	12.07
Room-level accuracy [%]	70.0	70.0	45.0	55.0	50.0	40.0	55.0	60.0
Pompeiu-Hausdorff distance of RSSI quantiles - method 1								
Median error [m]	2.47	2.28	2.59	2.89	3.44	3.03	3.26	3.71
Average error [m]	2.60	2.81	2.98	2.82	3.58	3.77	3.78	3.89
Minimum error [m]	0.19	1.50	1.16	0.18	0.38	0.38	0.53	1.06
Maximum error [m]	6.07	5.64	5.34	7.31	8.31	11.20	9.30	14.66
Room-level accuracy [%]	75.0	80.0	70.0	60.0	50.0	45.0	70.0	65.0
Pompeiu-Hausdorff distance of RSSI quantiles - method 2								
Median error [m]	2.49	2.38	2.42	2.41	3.06	2.83	2.86	2.41
Average error [m]	2.88	2.49	2.48	2.44	3.28	3.34	3.39	3.08
Minimum error [m]	0.19	0.38	0.19	0.19	0.54	0.03	1.03	0.05
Maximum error [m]	8.05	6.10	5.25	3.64	6.04	11.20	7.39	8.39
Room-level accuracy [%]	80.0	85.0	65.0	70.0	55.0	40.0	75.0	70.0

interference scenarios are statistically the same as the processing times obtained in the reference scenario, so only results for the reference scenario are presented.

The selection of method for estimation of the interference power also has a small influence on the overall processing time of an algorithm. For that reason, we only present results in which the used interference power estimation method is "method 2", i.e. the one that provides higher accuracy of indoor localization. The obtained results show that the increase in processing time of the algorithms due to the interference mitigation method is in average around 200 ms, which is an increase of around 20% in the processing time. However, the whole latency of providing location estimates consists of the time needed for RSSI collection and of the processing time of an algorithm. Depending on a hardware and a device driver, the time needed for obtaining the measurements from one WiFi scan is around 2-3 sec, making it a dominant factor in the overall latency. For that reason, the increase in the overall latency of providing location estimates due to the interference mitigation method is practically of a small importance.

Table 4.11: Summarized evaluation results - processing time

Metric	No mitigation	With mitigation
Euclidean distance of averaged RSSI vectors		
Average processing time [s]	0.7249	0.9649
Median processing time [s]	0.7255	0.8523
Minimum processing time [s]	0.6967	0.8138
Maximum processing time [s]	0.8024	2.2834
Pompeiu-Hausdorff distance of RSSI quantiles		
Average processing time [s]	0.7340	0.8951
Median processing time [s]	0.6931	0.8603
Minimum processing time [s]	0.6705	0.7805
Maximum processing time [s]	1.0091	1.5935

4.4 Extrapolation of WiFi Fingerprinting-based Localization Performance Across Environments

As discussed previously, WiFi RSSI-based fingerprinting is one of the promising candidates for an indoor localization service. Consequently, performance evaluations of such algorithms in different environments are becoming publicly available [61, 35]. Thus, the interested localization service providers are able to get insights in the performance of various algorithms in different environments through these standardized benchmarks. However, it is unclear if the performance results, in terms of localization errors, achieved in one environment and for one parameterization of an algorithm, can be representative for another environment and parameterization of the same algorithm.

The extrapolation of the performance of fingerprinting algorithms across environments would be beneficial from two perspectives. From the technological and research perspective, it would allow categorization of algorithms based on their suitability for different types of environments. From the localization service providers' perspective, the extrapolation would give them the possibility of speculating about the performance of different algorithms for a particular environment without the need for extensive experimentation. Collection of such measurements is particularly problematic for buildings that are usually not accessible for an extensive experimentation (e.g. hospitals or buildings under construction).

The possibility of performance extrapolation for different systems has been addressed in various research domains. For example, in network experimentation the need for having realistic experimentation conditions in testbeds has been addressed in [177], while for evaluation of artificial intelligence the question of extrapolation of results achieved in testbeds to reality has been addressed in [178]. Moreover, in [179] the authors aim on predicting the performance of GPU applications by correlating them to existing benchmarks, while in [180] the authors qualify the similarity of computer systems and then use this similarity for predicting the performance of a new application. However, in the domain of WiFi fingerprinting, the question of performance extrapolation is, to the best of our knowledge, still open.

To address this shortcoming, we systematically analyze the performance achieved by a set of WiFi RSSI-based fingerprinting algorithms in a set of environments with different characteristics. Our goal is to establish a link between the similarities among environments and parameterizations of algorithms in these environments, with the possibility to extrapolate the performance of such algorithms across environments. Our contributions include a hypothesis about the possibility of extrapolation of the performance of WiFi fingerprinting algorithms across environments, demonstrating the feasibility of the hypothesis, and outlining a methodology for its further evaluation.

4.4.1 Extrapolation Hypothesis

We hypothesize that for the accurate extrapolation of the performance of fingerprinting algorithms across environments one has to be careful to select both environments and parameterizations of an algorithm that can be characterized as "similar". We consider two environment as "similar" if there are similarities pertaining to their physical shape, *w.r.t.* their outer size and their inner spaces. In addition, in order for two environment to be "similar", there has to be a similarity in propagation characteristics in these two environments.

Apart from the environments *per-se*, we hypothesize that the parameterizations of a fingerprinting algorithm in these environments have to be similar for being able to accurately extrapolate the performance of the algorithm across the environments. The "similarity" in the parameterization of an algorithm pertains to the type of algorithm used, the number, density and deployment locations of APs, their trsansmission powers and operating frequencies, the number of measurements taken at each measurement point, and the number and density of training points. Given that the parameterizations of an algorithm in two environments are comparable and the environments are "similar" according to our definition, we hypothesize that a reliable extrapolation of the algorithm's performance across environments is possible.

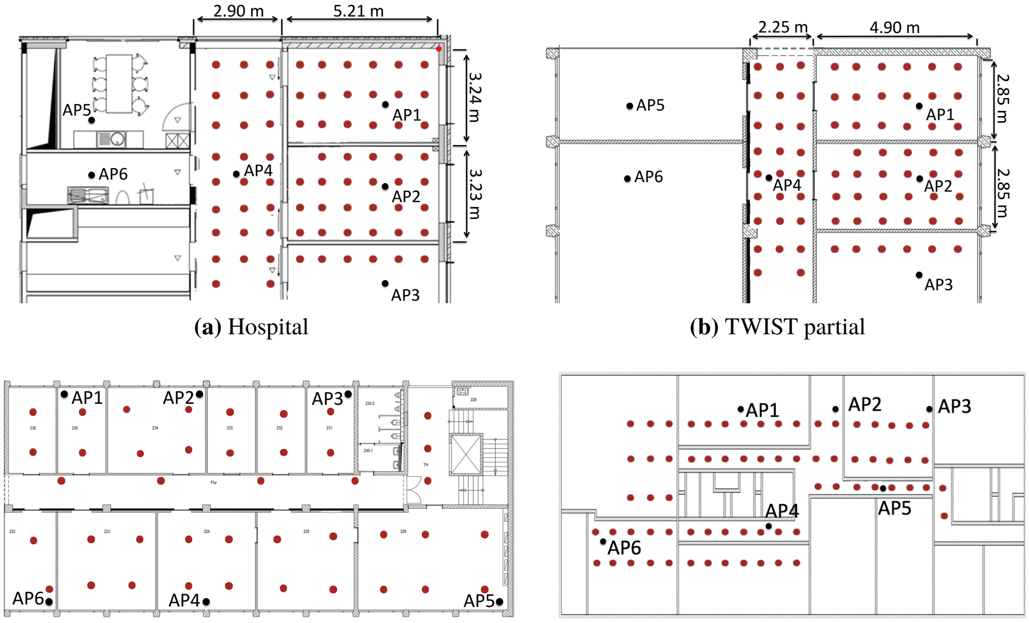
We evaluate the hypothesis by comparing the performance achieved by two fingerprinting algorithms in a baseline, real-life hospital environment with their performance in three other environments with different levels of similarities to the hospital environment. Our similarity characterization yields that one environment is highly similar to the hospital baseline, in both physical and propagation characteristics, while the other two environments have increasing dissimilarities with the baseline. As discussed in Chapter 3 of this thesis, the performance results of fingerprinting algorithms in these three additional environments are publicly available and they serve as standardized benchmarks of the performance of fingerprinting algorithms. The fact that the results in those two environments serve as public standardized benchmarks of the performance of fingerprinting algorithms provides motivation for their usage.

4.4.2 Environments: Physical Shape and Experimental Setup

In this section, we present the physical shape and the experimental setup in the four used environments, with the relevant environment related parameters given in Table 4.12.

Table 4.12: Parameters of the environments

Parameter	Outer size	Mean room size	Wall type
Hospital	8.1 x 7.4 m	15.0 m^2	plywood
TWIST partial	7.2 x 6.8 m	12.5 m^2	concrete
TWIST	30.0 x 15.0 m	27.0 m^2	concrete
w-iLab.t I	45.0 x 17.0 m	51.0 m^2	plywood



(**a**) Hospital

(**b**) TWIST partial

(**c**) TWIST

(**d**) w-iLab.t I

Figure 4.16: Floorplans of the four environments

Hospital: The hospital serves as a baseline environment for which we would like to extrapolate the performance of fingerprinting algorithms. The measurement campaign was performed in the "chirurgic day" ward of the Sint-Jozefskliniek Izegem in Belgium, with footprint depicted in Figure 4.16a). For the measurement collection the end section of a corridor was used, while the rest of ward was in "normal operation", meaning people were present in the hospital. In the given environment, six IEEE 802.11g APs (WNDR 4300) were deployed, one in each room in the environment of interest, with their locations indicated in the figure. The APs were configured in beaconing mode, with beacon transmission period of 100 ms. The transmission power of each AP was set to 20 dBm. The APs 1, 3 and 4 were operating on WiFi channel 6, while for the others (AP 2, 5 and 6) the operational channel was WiFi channel 11. In the environment of interest, which included three rooms and a hallway, a total number of 73 measurement points were defined in a relatively dense grid with cell size equal to 1 m, as indicated by the red dots in Figure 4.16a). In each of the defined points, the wireless environment was sampled for WiFi beacon packets using a laptop with external wireless adapter (TL-WN823N). At each location, four scans of the wireless environment were performed, meaning that at each location, from each visible AP, at most 4 beacon packets could be obtained. Due to a limited availability of the environment and in order not to interrupt the normal hospital activities, the measurement campaign was constrained in both size and duration. However, it represents one of the first publicly available datasets that capture a realistic hospital environment for the purpose of evaluation of WiFi-based fingerprinting.

TWIST partial: The standardized benchmark obtained from this environment, which is a part of the previously discussed TWIST testbed, has high levels of similarity with the hospital environment, *w.r.t.* similar sizes of the offices/rooms, outer sizes, and similar activities as in the hospital (people moving in the testbed premises, since it is an office building). We believe that this environment mimics the hospital environment to the level that is practically possible, as shown in Figure 4.16b). In this part of the environment, six WiFi APs were deployed at similar locations as for the hospital. The APs were configured as in the hospital, although they were of a different type (TP LINK N750). Similarly, 73 measurement points were selected and in each point the wireless environment was scanned four times, same as for the hospital, but using an Airport Extreme network interface card.

TWIST: The TWIST testbed environment in its entirety is an office environment, with room sizes slightly bigger than the hospital environment and with an outer dimensions of roughly 30x15 m^2. Also in this environment, six dedicated WiFi APs were deployed with their locations shown in Figure 4.16c). They were configured to operate in beaconing mode with beacon transmission period of 100 ms and transmission power of 20 dBm. In this environment, 41 measurement points were defined and in each of the points four scans of the wireless environment were performed.

w-iLab.t I: The w-iLab.t I testbed is an office environment with outer dimensions of roughly 45x 17 m^2. In comparison to the TWIST testbed, in this environment the office sizes are less similar to the hospital, as shown in Figure 4.16d). For this measurement campaign also six “dedicated” WiFi APs were deployed with the same configuration as in the TWIST environment. For the measurement campaign 69 measurement points were defined and in each of them a scan of the wireless environment was performed. In this environment also four measurements per point were taken.

4.4.3 Propagation Characteristics

For further understanding of the similarities among environments, we modeled the RF propagation parameters by leveraging the collected measurements. The applied propagation model is the previously in details discussed COST 231 multi-wall model for indoor radio propagation [170]. The applicability of this model for indoor localization purposes has been demonstrated in [171]. In the following we provide only a rough overview of the COST 231 propagation model. The first attenuation contribution in the model is a well-known one-slope term that relates the received power to the distance. Two parameters influence the attenuation in this term: the constant l_0 (the path-loss at 1 m distance and at the center frequency of 2.45 GHz) and the path-loss exponent

γ. The second attenuation contribution is a linear wall/obstacle term. The number of obstacles in the direct path between transmitter and receiver is counted and for each type of obstacle an attenuation contribution is assumed. Given the model and the site-specific measurements from different environments, we leveraged a least square fitting procedure that allows minimizing the cost function between the measured received power and the modeled one.

The parameters to be optimized are the constant l_c related to the least square fitting procedure, the γ path-loss exponent, and the wall attenuation factor. Although $\gamma = 2$ is a usual assumption for the propagation in the free space, coming from Friis equation, due to the obstacles in our environments we also estimate γ. Moreover, we do multiple prediction tests on (rolling) 10% of sampling points using a model fitted on the remaining 90%, to be sure that we do not have unexpected interactions between the linear constant and the γ. The modeled averaged propagation parameters are given in Table 4.13, since small discrepancies are obtained by performing the rolling tests. As indicated in the table, for all environments l_c and γ variables have comparable values. Furthermore, in terms of wall attenuation, the similarity between the hospital and the partial TWIST environments is high, while smaller similarities are observed between the hospital, TWIST and w-iLab.t I environments.

Table 4.13: Parameters of the propagation model

Parameter	l_c [dB]	γ	l_w $[dB]$
Hospital	57.38	1.46	2.60
TWIST partial	58.36	1.25	2.78
TWIST	53.73	1.64	4.51
w-iLab.t I	60.23	1.29	1.12

4.4.4 Evaluation Scenarios and Approaches

This section gives an overview of the scenarios and approaches used in the evaluation. The algorithms used in the evaluation are the same as the ones used in previous sections of the thesis, with their summarizes given in Chapter 2.

Evaluation Scenarios

To create multiple scenarios for evaluating the possibility of performance extrapolation across environments, for both algorithms we filtered the collected raw data (i.e. we used different parameterizations of algorithms) as follows. Firstly, we evaluated the localization errors in case measurements from all APs were used as inputs to an algorithm (including the APs that we have not deployed and using both 2.4 and 5 GHz ISM frequency bends). Secondly, we used only measurements from the 2.4 GHz ISM frequency band. Thirdly, we used only measurements from "dedicated" APs, i.e. the six APs we deployed for localization purposes. Furthermore, we filtered measurements from only some dedicated APs to evaluate if the performance degradation due to a removal of particular APs is consistent across environments. Our decision on which AP to include was not driven by the goal of optimizing the APs deployment, but merely to increase the diversity of evaluation scenarios.

However, in case the environments are less similar under the assumption that the relative difference in performance results across environments is preserved, the questions that we were able to address are more limited. According to that limitation, we designed our evaluation scenarios. In the first scenario, we used only six dedicated APs. In the following scenario, we used the APs 1, 2 and 4, which corresponds to a scenario in which two APs are on one side and one AP is in the center of the other side of an environment. In the third scenario, we used the APs that are all on one side of an environment. Finally, in the last two scenarios, we filtered one AP from the corner and from the center of an environment, respectively. The goal is to evaluate the possibility of preserving the relative difference in performance across environments for different scenarios, despite the fact that differences in environments and parameterizations exist.

Evaluation Approaches

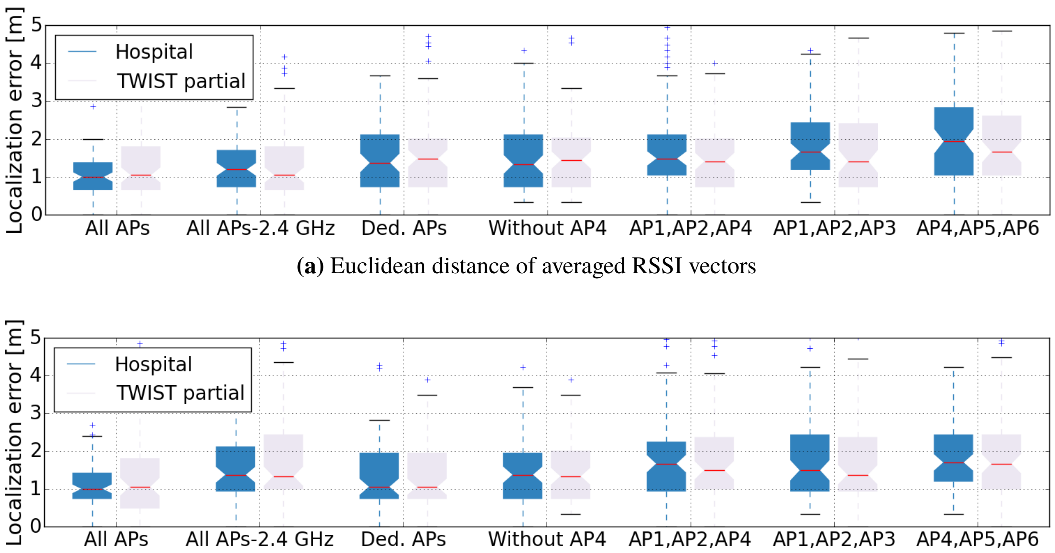
The localization errors in fingerprinting generally have two aspects: spatial and temporal [68]. The temporal one is related to changes of the relevant values in time domain, which usually causes inconstancies between training and runtime fingerprints. The spatial aspect is related to a particular environment and respective algorithm's parameterization. We aimed on removing the temporal aspect and focusing on the spatial one, so we do not clearly differentiate between the training and runtime phases of fingerprinting, i.e. we use measurements from the same measurement campaign in both phases. We leveraged two approaches to obtain the split between the training and runtime phases. First, we used first three RSSI measurements from each AP at each location for training, while the fourth measurement was used in the runtime phase. We will refer to this approach as the "evaluation approach 1". Second, while one measurement point was used as a runtime fingerprint, measurements from all the other points were used for training, and we will refer to this approach as the "evaluation approach 2". Two approaches were selected to increase the number of evaluation scenarios, thus increasing the reliability of our findings.

4.4.5 Evaluation Results

In this section, we present the results of evaluating the possibility of extrapolating the performance of fingerprinting algorithms across environments. Given the observations made previously, we classify one environment and algorithms' parameterization as "similar" (TWIST partial), while the other two (TWIST and w-iLab.t I) are "less similar" to the baseline hospital environment.

Extrapolation in Similar Environments

The distributions of localization errors for the two used algorithms (with their summaries given in Chapter 2 of the thesis) in similar environments are given in Figure 4.17 and Figure 4.18 for the evaluation approaches 1 and 2, respectively. As visible in the figures, for both algorithms and for both evaluation approaches the localization errors in similar environments show small differences. The results show that it is possible to give a statement about the performance of the evaluated fingerprinting algorithms in one environment by using the performance results from another, similar environment.



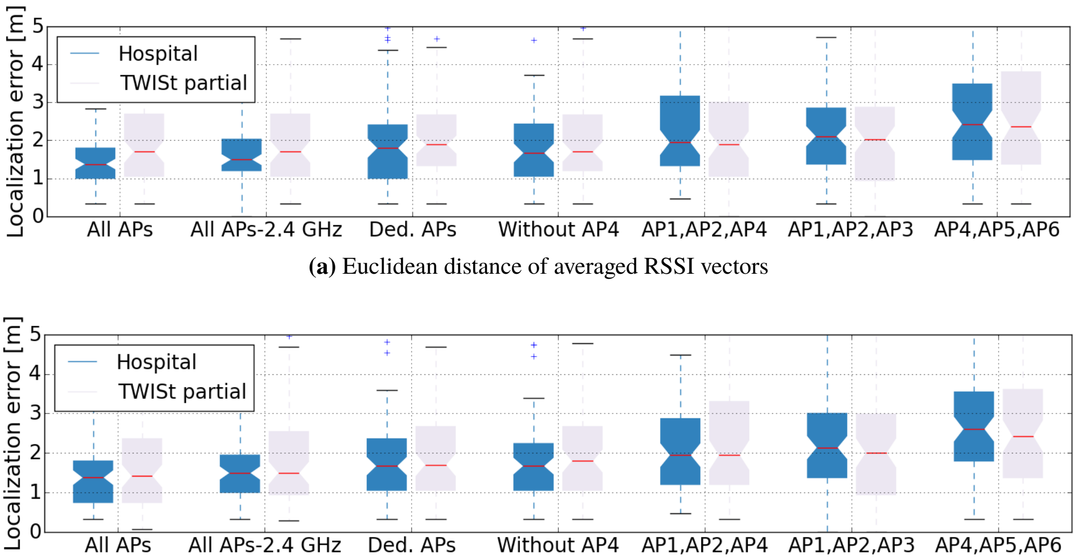
(a) Euclidean distance of averaged RSSI vectors

(b) Pompieu-Hausdorff distance of RSSI quantiles

Figure 4.17: Comparison of localization errors for similar environments - evaluation approach 1

We evaluated the similarities in distributions of localization errors across environments using Pearson's Chi-Squared tests. Pearson's Chi-Squared test tests a null hypothesis stating that one distribution observed in a sample is consistent with another particular distribution [181]. The results of the tests yield that, for all evaluation scenarios, both evaluation approaches, and both algorithms, the hypothesis of localization error distributions across environments being consistent is true with the probability of more than 95%. Furthermore, we used the Cohen's d tests to evaluate the magnitude of the difference of mean localization errors across environments. The result of a Cohen's d test is Cohen's d value, which is a scale-free indication of the size of an effect between two observations [182]. As a rule of thumb, Cohen's d values smaller than 0.2 represent small effect, values smaller than 0.5 represent medium size effect, while higher values than 0.5 represent high effect size. Specifically, in our evaluation Cohen's d values represent the magnitude of the effect that the change of an environment has on the observed localization errors. Table 4.14 gives the Cohen's values for localization error distributions observed by the two used algorithms for different evaluation scenarios across two similar environments. As visible in the table, the calculated Cohen's values are smaller than 0.2, meaning that the change of environments in this case has a small effect on the localization errors. Higher Cohen's d values are obtained in case all APs in the environments were used in the evaluation ("All APs" and "All APs-2.4 GHz"), since in these scenarios also uncontrollable APs (e.g. visible APs from neighboring buildings) were used as inputs to the algorithms, which resulted in a higher effect size that a change in environments has on the observed localization errors.

In order to get a clearer view on the possibility of performance extrapolation of fingerprinting algorithms across environments, in Figure 4.19 we depict the localization error distributions observed by the used algorithms in different rooms in two similar environments. Furthermore, in Table 4.15 we present the Cohen's d test results for the distributions of localization errors



(a) Euclidean distance of averaged RSSI vectors

(b) Pompieu-Hausdorff distance of RSSI quantiles

Figure 4.18: Comparison of localization errors for similar environments - evaluation approach 2

Table 4.14: Cohen's d test results in similar environments

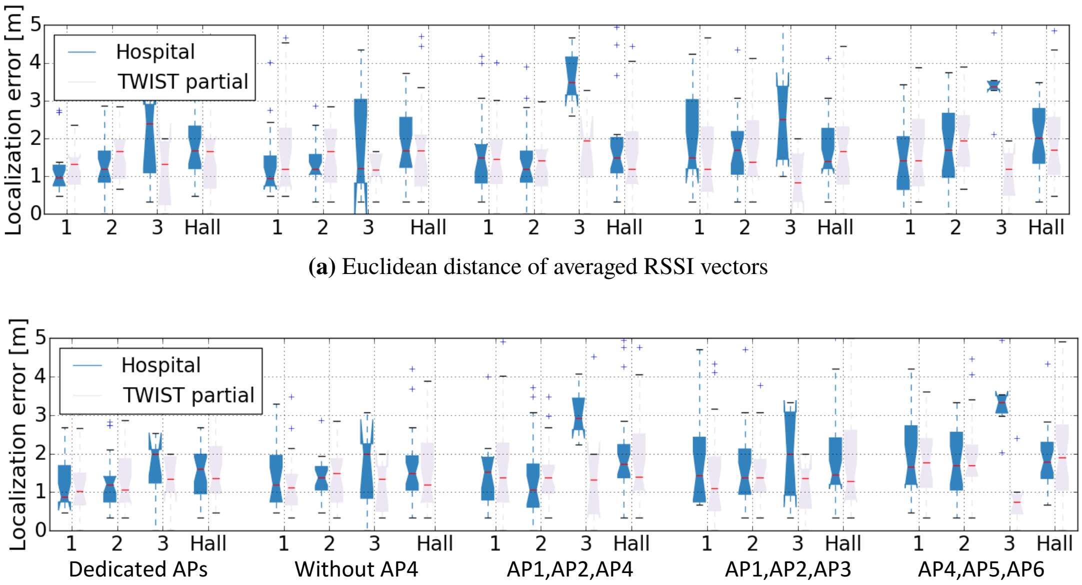
All APs	All APs 2.4GHz	Dedicated APs	Without AP4	AP1,2,4	AP1,2,3	AP4,5,6
Euclidean distance of averaged RSSI vectors						
0.45/0.58	0.06/0.39	0.03/0.18	0.09/0.06	0.11/0.12	0.08/0.07	0.09/0.07
Pompieu-Hausdorff distance of RSSI quantiles						
0.16/0.35	0.14/0.34	0.02/0.19	0.06/0.24	0.02/0.02	0.07/0.12	0.06/0.03

Table 4.15: Cohen's d test results for room-level accuracies in similar environments

Dedicated APs	Without AP4	AP1,AP2,AP4	AP1,AP2,AP3	AP4,AP5,AP6
Euclidean distance of averaged RSSI vectors				
0.11/0.07/0.38/0.09	0.19/0.01/0.29/0.15	0.01/0.11/1.15/0.12	0.22/0.07/0.79/0.09	0.23/0.16/2.79/0.11
Pompieu-Hausdorff distance of RSSI quantiles				
0.17/0.15/0.85/0.03	0.12/0.22/0.60/0.11	0.05/0.09/2.00/0.08	0.16/0.11/1.30/0.19	0.21/0.16/2.91/0.14

per room for different evaluation scenarios across similar environments. Note that we present only the results obtained by leveraging the evaluation approach 1, since the trends are consistent for both evaluation approaches. The rooms in Figure 4.19 are labeled with "1", "2", "3" and "Hall", and those are respectively the rooms where APs 1, 2 and 3 are deployed and the hallway (Figure 4.16a, Figure 4.16b). It is clear from Figure 4.19 and Table 4.15 that the change of environments has a small effect on the observed localization errors per room, except for the errors observed in the room labeled with "3".

The reason for this trend lies in the fact that only 6 measurement points have been defined in the room labeled with "3". Due to that, no meaningful statistics about the localization errors in this room could be extracted. In other words, it is not our intention and we believe it is not possible



(a) Euclidean distance of averaged RSSI vectors

(b) Pompieu-Hausdorff distance of RSSI quantiles

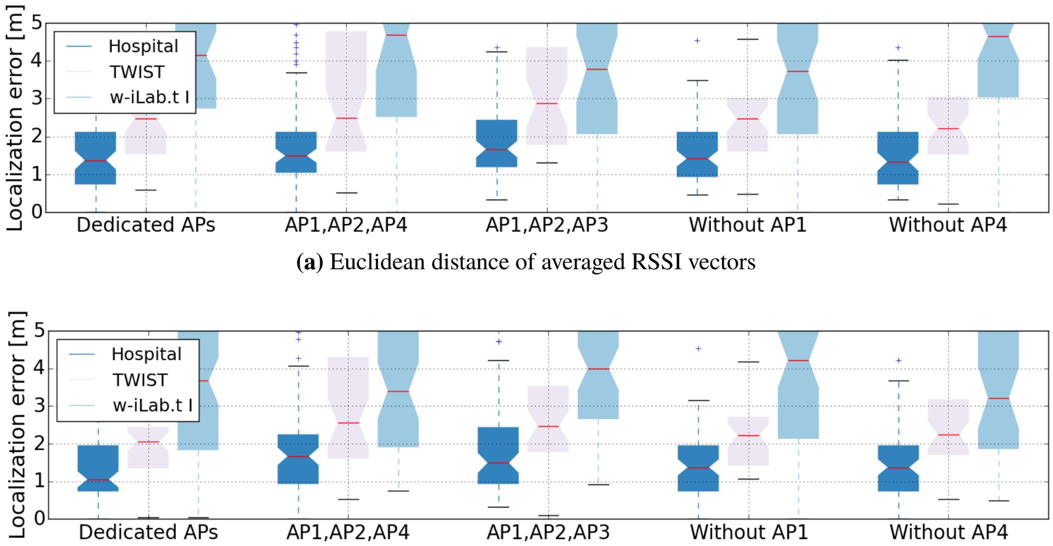
Figure 4.19: Comparison of room-level accuracies for similar environments

to extrapolate the localization errors for any single measurement point from one environment to another. The reason why a single evaluation point is not sufficient lies in the instability of the RF-based indoor localization algorithms and solutions, which is related to an intrinsic randomness in each wireless environment. This means that, even for the same environment and exactly the same measurement point, if a localization estimate is requested twice, these estimates are usually not the same. On the contrary, we believe it is only possible to accurately extrapolate the statistics of errors, and for obtaining a statistic a meaningful number of samples has to be used. This requirement is not fulfilled for the samples in the room labeled with "3", thus the performance of algorithms is not statistically comparable in this case.

Extrapolation in Less Similar Environments

In case less similar environments are used for the performance evaluation, the results are depicted in Figure 4.20. Despite a smaller level of similarity between those environments and the hospital, we aim on evaluating the possibility that the relative difference between the observed localization errors for different parameterizations of algorithms can be preserved. Note that for this case we do not evaluate scenarios in which all measurements are used, but only scenarios where measurements from the six dedicated APs are leveraged. The reason is that, due to larger sizes of these environments in comparison to the hospital, the number of visible APs is expected to be increased, which could lead to wrong conclusions about the extrapolation possibility.

As visible in Figure 4.20, the evaluation results show that a relative difference between various parameterizations of algorithms is preserved for the TWIST environment, while for the w-iLab.t I environment that is not the case. More specifically, it is visible in the figure that, for the hospital and the TWIST environment, higher localization errors are observed in case APs from one side



(a) Euclidean distance of averaged RSSI vectors

(b) Euclidean distance of averaged RSSI vectors

Figure 4.20: Comparison of localization errors for less similar environments

of the environment are used as anchors for location estimation (AP1,AP2,AP3 in figure), in comparison to the scenario when two APs are used on one side and one anchor on the other side of the environment (AP1,AP2,AP4). This observation, however, does not hold for the w-iLab.t I environment, since in this case the localization performance is better when only APs from one side of the environment are used. Similarly, for both algorithms, and for both hospital and TWIST environments, comparable localization errors are observed in case an AP from the center of an environment is not used for the localization purposes (Without AP4), in comparison to a case when an AP from the corner of an environment is not used (Without AP1). As it can be seen in the figure, in this case the ranking is again not preserved for the w-iLab.t I environment. Similar observation can be made from Table 4.16, since Cohen's d values are similar across different evaluation scenarios in case the hospital is compared to the TWIST environment, while they have larger discrepancies in case the hospital is compared with the w-iLab.t I environment.

Clearly, the hospital environment and the two other, less similar environments differ in the outer sizes, as well as in the number of inner walls. However, the hospital and TWIST environments have comparable room sizes, while that is not the case for w-iLab.t I environment. Furthermore, the wall attenuation factor is different for all three environments. This factor is much higher for the TWIST environment, in comparison to the hospital, while for the w-iLab.t I this factor is much smaller than for the hospital environment. A relatively high wall attenuation indicates that a wireless environment has higher spatially distinguishable features, which benefits fingerprinting in general [68]. This is possibly a reason for substantially smaller localization errors in TWIST, in comparison to w-iLab.t I environment, in case a small number of APs is used. In all three environments, the same number and configuration of APs are used, although their deployment locations and densities are different. Finally, the number and density of measurement points differ, while the number of measurements taken at each point is the same for all three environments.

Table 4.16: Cohen's d test results in less similar environments

Dedicated APs	AP1,AP2,AP4	AP1,AP2,AP3	No AP1	No AP4
		Euclidean distance of averaged RSSI vectors		
0.68/1.41	0.69/1.04	0.64/1.39	0.79/1.67	0.84/1.32
		Pompieu-Hausdorff distance of RSSI quantiles		
0.74/1.56	0.72/1.27	0.77/1.04	0.82/0.47	0.72/1.84

4.5 Conclusions

Due to its well-known advantages, we selected WiFi-based fingerprinting as a promising candidate for an accurate and robust indoor localization service. We then aimed at enhancing the understanding of WiFi-based indoor localization solutions by formalizing their common phases and evaluating the contributions of procedures in the individual phases on the final performance of such solutions. We have done that by using the developed infrastructure and following the proposed systematic evaluation methodology. We illustrated how the method for raw RSSI collection influences the final accuracy and showed that scoping the collection of raw measurements only to a set of dedicated and controlled APs yields the best results for the presented environments and scenarios. We further compared three different methods for fingerprint creation and pattern matching and showed that the ones with low or medium computational complexity achieved the best performance results. Moreover, we evaluated the influence of post-processing on the example of k-Nearest Neighbors (kNN) procedures, showing that applying kNN procedure in fingerprinting decreases the variability and maximum localization errors, i.e. improving the stability of fingerprinting algorithms. We showed that this influence, however, depends on the type of deployment environment. The accuracy of algorithms with and without kNN is comparable in small environments, while the improvement when applying the kNN is more visible in environments with larger sizes. Finally, based on the results of our experiments in three different office scenarios, we observed that in similar environments, in terms of the environment size, type and density of APs, the evaluated fingerprinting algorithms maintain their relative ranking, while the localization error, as expected, generally scales with the size of an environment.

We further aimed at reducing the deployment overheads of WiFi-based fingerprinting. Hence, we presented the Enriched Training Service (ETS), a web-service that can be used for storing and managing WiFi RSSI measurements and training fingerprints for various types of fingerprinting algorithms. The enriching functionality of the ETS can be used to automatically generate virtual training fingerprints and store them together with the original training dataset. We have shown that the usage of the enriching functionality of the ETS results in increased accuracy of fingerprinting, without additional time overheads and deployments efforts. The improvement is clearly dependent on the evaluation environment and the used algorithm, and it is expected that different propagation models will be optimal for different environments and algorithms, as indicated in [183]. The modular design of the ETS, apart from supporting the integration of different fingerprints algorithms, allows easy implementation of additional propagation models, and this feature can, in the longer run, serve as a basis for creating a framework for comparing the feasibility of different propagation models for generating virtual training fingerprints for different environments and algorithms.

Due to the fact that RF interference can substantially degrade the performance of WiFi-based fingerprinting algorithms, as demonstrated previously, we also proposed a procedure for mitigating this negative effect by leveraging the knowledge of spectrum power levels in the 2.4 GHz ISM frequency band. The proposed procedure assumes the availability of spectrum information, which to a certain level limits its practical usage. This is because the majority of current WiFi chipsets do not have the capability of measuring spectrum information. However, for other purposes (e.g. for enabling opportunistic networking [184, 185] and cognitive radio paradigms [186, 187]) there is a tendency of incorporating this capability in WiFi chipsets and some of the current chipsets already provide this functionality [188, 189]. We therefore believe that the proposed procedure will in the future be usable in practice. The proposed procedure uses the estimated average interference power levels for removing the additive variability in WiFi beacon packets RSSI measurements due to interference. We further proposed two methods for estimating the average interference power levels from the sampled spectrum power levels. By using the developed testbed infrastructure, we evaluated the performance of two fingerprinting algorithms in four RF interference scenarios with and without applying the interference mitigation procedure. In our experimental setup, we firstly demonstrated that our evaluation results are statistically stable in time, which increases their reliability. Secondly, the results show and confirm our previous findings that RF interference can reduce the accuracy of WiFi fingerprinting algorithms. Finally, we demonstrated that by leveraging the proposed procedure for reducing the effects of RF interference the accuracy of evaluated algorithms is notably improved. The processing time of evaluated algorithms increases by roughly 200 ms in average when interference mitigation procedure is introduced to the system, which is of a small practical importance. The cost of our system, in comparison to the usual fingerprinting procedure, is the necessity of having sampled spectrum power information.

Finally, we contributed by postulating a hypothesis for the possibility of extrapolation of the performance of WiFi-based fingerprinting algorithms across environments. We demonstrated that the observed localization errors of a set of fingerprinting algorithms in one environment have very small statistical differences to their localization errors in an environment that is, according to our hypothesis, considered as similar to the baseline environment. We have also shown that, even in case environments and parameterizations of algorithms are less similar, the accuracy-based ranking of fingerprinting algorithms can be preserved across environments. In other words, we demonstrated that the performance extrapolation across environments is a feasible concept, which depends on the similarities among environments.

5 Proposal for Standardized Localization Service

With the contributions made in the previous chapter, various performance aspects of WiFi-based indoor fingerprinting have been enhanced. Nevertheless, WiFi-based fingerprinting is still a single location information provisioning service able to provide location information for only a limited set of served environments. Similar observations can be made for other types of provisioning services [147, 148]. In general, the majority of provisioning services, especially in indoor environments, have limited applicability because each of them satisfies only certain use-cases and provides desirable performance only in environments with specific characteristics. For some other use-cases or in environments with different characteristics, other provisioning services can offer better performance [139, 190]. These provisioning services differ substantially in the types of signals used, processing of these signals, and in the way the location information is delivered to the applications, i.e. in the APIs for providing location information.

For seamless provisioning of location information (i.e. everywhere and at any time), there is a need for an entity that integrates different location information provisioning services. At the same time, this entity should support provisioning of location information to the interested applications in a standardized way, i.e. independently of the provisioning service that is at a certain moment generating this information. Therefore, as the mobile user moves from an area serviced by one provisioning service to another area serviced by some other provisioning service, this entity should support a handover of provisioning of location information across these provisioning services. The most acknowledged example of such a scenario is seamless provisioning from outdoor to indoor environments and vice-versa, as discussed in e.g. [191, 192, 193, 194].

Moreover, the served areas for different provisioning services can significantly overlap. To enhance the accuracy and robustness of location information provisioning, the fusion of location information from different provisioning services should be performed [195, 196]. As mentioned in Section 2.2.3, this fusion can be done either on the level of fusion of raw data, i.e. resources used for generating location information, or on the level of fusion of location estimates generated and reported by different provisioning services. The former introduces organizational issues, an example being that many sources of raw data for localization are part of other systems to which a given provisioning service does not have full access. Furthermore, this approach raises ownership and privacy issues, e.g. it is hard to put a price on the access to resources for generating location information and it is difficult to prevent their leakage. We envision a future where provisioning services are not going to leak out raw data, but will rather be black-box services with a price tag for providing location estimates. Hence, we believe that a more convenient and practical approach for improving localization performance is the handover between and the fusion of black-box provisioning services that can generate and report location estimates of a mobile device in a given space at a certain time.

In this chapter, we contribute with an approach for enabling dynamic addition and/or exchange of provisioning services, as well as fusion of their results in a seamless way from the perspective of location-based applications. In the following, we first propose a modular standardized localization service architecture that aims at a comprehensive solution of the above mentioned interoperability and integration problems. The proposed architecture features composition of location information provisioning services by a newly introduced component called an "integrated location service". The integrated location service offers to the location-based applications a single interface that is independent from the used individual provisioning services and their possible fusion. The proposed modular standardized localization service architecture has been published in [197].

The integrated location service selects and invokes provisioning services based on their provisioning features and the requirements for location information from the applications. The selection of provisioning services can be subject to different optimizations of the relevant metrics, i.e. accuracy, latency, and power consumption of provisioning. As the second contribution in this chapter, we provide a set of example solutions to the problem of selection of provisioning services. This contribution has been published in [198].

As the final contribution, as a proof-of-concept we provide the Standardized Localization Service (SLSR), an example prototypical implementation of proposed architecture, followed by its instantiation and evaluation in the previously discussed testbed infrastructure for performance evaluation of RF-based indoor localization solutions. This contribution has been published in [199].

5.1 Standardization of Localization Service Architecture

In this section, we first outline the functionalities of the today's state of the art localization services, followed by proposing a standardized localization service architecture.

5.1.1 State of the Art Localization Services

Location awareness is an integral part of today's leading smartphone operating systems, Android and iOS.

In Android, location information is exposed to the consuming applications through the Google Play services location APIs. The *LocationRequest* class is used for requesting location information with certain Quality of Service (QoS) from the operating system. These parameters pertain to the desired location information accuracy (best, "block", and "city" level accuracy), provisioning duration (expiration duration or time), provisioning frequency, maximum wait time for location updates, number of updates to be reported, smallest displacement for location information updates, and priorities for reporting location information (which have impact on the accuracy vs. power consumption trade-off). Current location information can be requested with the *getLastLocation* method. Accessing location information updates requires implementation of the *LocationListener* interface. The interface provides the *onLocationChanged* method for receiving updates upon a change of location information. The *Location* object represents location information consisting of a latitude, longitude, timestamp, and additional information (e.g. bearing and altitude). Location information context, such as an environmental map or a translation of location information to an address can be requested using the Google Maps Geocoding API.

Similar functionalities are provided in the iOS's Core Location framework. The standard location service provides location information and its changes for a specified level of accuracy and distance filter parameters. Current location information can be requested using the *requestLocation* method. Provisioning of location information can be requested using the *startUpdatingLocation* method. Furthermore, location information updates can be based on region monitoring service. Region monitoring can be started using the *startMonitoringForRegion* for geographical and the *startRangingBeaconsInRegion* method for beacon regions. Moreover, location information updates can be based on visit monitoring, where such updates are delivered upon entering or exiting frequently visited locations, such as home or work. Visit monitoring can be started using the *startMonitoringVisits* method. The significant-change location service's method *startMonitoringSignificantLocationChanges* provides location information updates upon specified significant change of such information, where the change is larger than the one covered with region or visit monitoring. Contrary to Android's API where location information updates are requested for a certain time period and upon its expiration have to be requested again, provisioning of location information updates in the iOS's framework has to be stopped explicitly using appropriate terminating methods. Similar to the Android's Geocoding API, iOS provides its own Geocoder for converting location information into names, addresses, placemarks, etc. Finally, iOS provides applications access to environmental maps though the Map Kit framework.

The localization service architectures in both Android and iOS are tightly integrated with the operating system and provide a well-defined interface for the location-based applications to access location information. Clearly, due to differences in the interfaces, portability of the application across operating systems from the location information provisioning perspective is not straightforward. In both architectures, the operating system is an entity that fuses location information provided by different location information provisioning services. In particular, the provisioning services are based on GNSS, cellular, and WiFi sensors, with the addition of iBeacons for the iOS's framework. In general, provisioning services in both architectures evolve over time (e.g. addition of iBeacons in the iOS's framework), but their addition is slow and depends on releasing new versions of the operating system or of the core services. Therefore, the addition is not happening "on-demand" and it is hence unable to keep pace with the rapidly evolving provisioning services.

These provisioning services could be continuously available (e.g. running on a mobile device), but could as well have a locality feature, i.e. provide service for a given environment only. An example of the latter is a WiFi fingerprinting-based provisioning service, where a training database, i.e. a sampling of a served WiFi environment at known locations, is required for generating location information of a mobile device [144]. In this case, a fingerprinting server containing the training database is usually deployed in a served environment, while a fingerprinting client deployed on a mobile device interacts with the server for obtaining location information. Useful resources (i.e. raw data) for generating location information can also be embedded in an environment potentially surrounding a mobile device. For example, surveillance cameras can be used for distinguishing a case of a person (hence device) being in that space, in contrast to nobody being there.

5.1.2 Standardized Localization Service Architecture

Our goal is to provide a localization service architecture that captures the outlined functionalities of the state of the art localization services, but also by design enables dynamic usage of both provisioning services and resources for generating location information in case of their availability. This includes a scenario in which a mobile device enters an environment served by an provisioning service or in which a resource is available in an environment. In that case, the provisioning service should be able to start providing location information of the device and this information should be fused with location information provided by other services or, in case of a resource, this resource should by available to the provisioning services for generating location information. Furthermore, the goal is a straightforward addition of new provisioning services in a way that does not require versioning of an operating system. Our aim is to achieve that in an unified way that enables portability of location-based applications, provisioning services, and resources.

The proposed standardized localization service architecture is depicted in Figure 5.1. Specifications of architectural modules and their functionalities are given as follows. A *location-based application* (shortened to an *application*) is a module that requires location information of a mobile device. A *location information provisioning service* (or shortly a *provisioning service*) is an abstraction for different instances of localization solutions that can determine mobile devices' location information. An *integrated location service* represents a module that supplies the consuming applications with location information on one side, and requests such information from the provisioning services on the other. The integrated location service manages the selection of provisioning services to be invoked, given the requirements from the applications and the accuracy/cost trade-off of each provisioning service. The integrated location service provides a setting for fusion of location information provided by different provisioning services. It further allows caching of such information and its provisioning to many applications at once, as well as a setting for calculating parameters related to location information, such as the speed or direction of movement. It also offers redundancy of such information in case some provisioning services fail to provide it. The applications are thus agnostic from a specific provisioning localization service. Furthermore, the provisioning services are agnostic from the perspective of location information fusion, which simplifies their introduction in a mobile device. A *mobile device and environmental resources for generating location information* (or shortly *resources*) is an abstraction of input data needed by the provisioning services for generating location information.

A naive procedural overview of requesting and reporting location information is given as follows. The procedure begins by the application requesting location information of certain characteristics from the integrated location service. In case the requested location information is not available at the integrated location service level (i.e. not cached), the integrated location service issues a request for location information provisioning of desired features to one or more registered provisioning services. Each provisioning service responds with an "offering", i.e. it bids a set of location information features that can be provided to the integrated location service. This offering is based on the availability and quality of resources needed for generating location information. These resources can be available both on a device and in an environment. In case the provided offering is satisfactory, the integrated location service requests a service from the provisioning service. The service is then acknowledged or rejected by the referred provisioning service for a certain time period. The procedure continues by the provisioning service access-

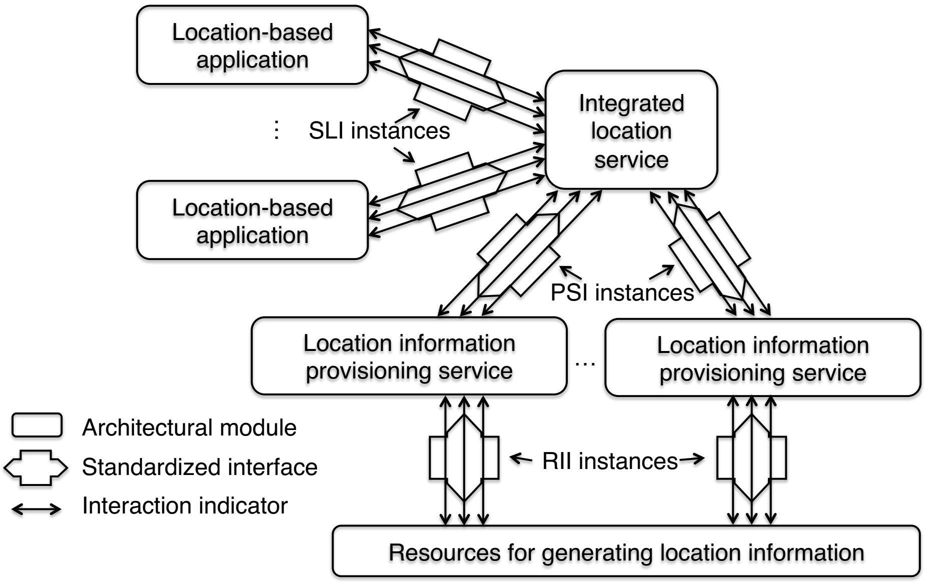

Location-based application
SLI instances
Location-based application
Integrated location service
PSI instances
Location information provisioning service
...
Location information provisioning service
Architectural module
Standardized interface
Interaction indicator
RII instances
Resources for generating location information

Figure 5.1: Localization service architecture

ing the resources, followed by generating and reporting location information to the integrated location service. The integrated location service potentially fuses the received information and reports final location information to the application.

Standardized interaction between the modules of the proposed localization service architecture allows portability of applications across operating systems. Having a well defined interface between the application and the integrated location service simplifies the development of applications by exposing to the developers only a small set of common primitives needed for accessing location and appropriate context information. As mentioned previously, the integrated location service serves as the central point for both service selection and distributing location information. Therefore, through caching, potentially multiple applications can benefit from one location or context information available on the integrated location service level. Moreover, supporting fusion of location information from multiple provisioning services is enabled by design because of the availability of all such information at the integrated location service level. Upon request, provisioning services report location information to the integrated location service through a standardized interface. Developers of provisioning services only have to deal with the registration of their service with the integrated location service and, by following the standardized interaction, their service can provide location information to many applications at once. On the lower level, through standardized access and arbitration to the underlying resources, multiple provisioning services can benefit from obtaining the same resource in a given time instance. This also makes the provisioning service reusable across OS ecosystems. In the following, we provide detailed overviews of the standardized interaction primitives between different architectural modules.

5.1.3 Standardized Interfaces

As shown in Figure 5.1, the Standardized Localization Interface (SLI) defines primitives and their parameters required for enabling interaction between the location-based application and the integrated location service. The Provisioning Service Interface (PSI) standardizes interaction between the integrated location service and the provisioning service. The Resource Interaction Interface (RII) defines a set of standardized primitives and their parameters for accessing resources needed by provisioning services for generating location information.

Standardized Localization Interface (SLI)

The *Specify policy* interaction primitive of the SLI is used by the applications for specifying the desired priority of location information provisioning. The only parameter of the interaction primitive is *policy*, which can either specify “accuracy”, “latency” or “power” as the desired provisioning priority. This priority specification is then used by the integrated location service maximizing the utility function of provisioning services invoking based on the policy specified by each application and the accuracy, latency, and power consumption parameters of each provisioning service. The policies of provisioning will be in more details discussed in the next section of the thesis.

Furthermore, the SLI’s interaction primitives cover the requesting and reporting location information of desired features, i.e. the location type and dimensionality (2D or 3D), its desired accuracy (in *meters*), the period of its provisioning (in *seconds*), and the desired provisioning duration (in *seconds*). Furthermore, the SLI includes interaction primitives for exchanging the location information context. A summary of the SLI’s interaction primitives, with an indication of the directionality of each interaction primitive, i.e. from the location-based application (LA) to the integrated location service (IL) and vice versa, is given in Table 5.1.

The application can request location information of certain features from the integrated location service using the *Request location* interaction primitive. If such information is available, the integrated location service can report it using the *Report location* interaction primitive. In case such information is absent at the integrated location service level, the integrated location service initiates the interaction with the provisioning services for obtaining it, which is standardized though the PSI and discussed in the next paragraph.

Location information can be reported as a global, local or semantic location, which is defined with the *location type* parameter of the *Request location* primitive, its type 0, 1, and 2 referring to global, local, and semantic location information, respectively. The global location, comprised of a longitude, latitude, and altitude information, defines a location in the WGS 84 global coordinate system. The local location, represented in a (x, y, z) notation (in *meters*), is understood as a location in relation to some predefined location coordinate, i.e. zero-point, usually in indoor spaces. In addition to their 3D notations, both types are in practice also often used for 2D localization, so the altitude/z-coordinate is not compulsory. In case the application requires location information in a 2D space, the *dimensionality* parameter of the *Request location* primitive is set to 0, otherwise it is set to 1. If location information is requested for a 3D space, and only 2D location information is available to the integrated location service (after consolidating the provisioning services), this should be reported back to the application to be handled in application-specific manner. In case of local and global location, the confidence interval parameter of the *Report location* primitive is expressed as the expected average error (in *meters*), i.e. the average offset from the ground-truth location. In contrast to geometrical coordinates that express location information as a point in a particular coordinate system, the semantic location type is a string object used for expressing location information as being inside of a space or in the proximity of an object. This format gives a contextual meaning to the provided location information. The *inside(space)* expression is used for expressing a location being inside of a space, while for the proximity to an object the *proximity_to(object)* expression is used. In case of the semantic location, the *confidence interval* of the *Report location* primitive is defined as the correctness of location information and it is expressed as an expectancy (ratio) of the number of correct and all possible information. The *distance* parameter in the *Report location* primitive is used exclusively in case of the semantic location type, particularly in the *proximity_to(object)* expression to specify the proximity to the object more precisely. This parameter defines the distance (in *meters*) implied by the proximity to an object. In addition to reporting location information, the integrated location service can report the *movement vector* parameter, which specifies the movement speed (in *meters per second*) and orientation (in azimuth and (optionally) elevation *degrees*) of a mobile device. This parameter is reported if the integrated location service can calculate it from the historical location data and if the binary parameter *movement* of the preceding *Request location* primitive is set to its non-default value 1.

The *accuracy* and *period* parameters of the *Request location* primitive define the desired accuracy of location information and the period of its provisioning. The *on_event* parameter is a binary indicator, with its default value 0 indicating that location information should be provided periodically, with the period specified with the *period* parameter. The value 1 indicates that location information should be provided when it changes from a preceding location information, with the size of change (in *meters*) indicated by the *step* parameter of the primitive. These parameters are defined to better support the broad range of different requirements that the applications may have. For example, the application may benefit more from receiving location information with a shorter period and/or power consumption, given that its accuracy is above a certain threshold, rather than obtaining a more accurate information, but with a larger period and/or power consumption [35, 139]. E.g. for asset tracking scenarios periodically receiving location information is required, while for some other, e.g. person walking in assisted living and health-care scenarios, such information is required on event, e.g. the person left the room [63, 147].

The *duration* parameter defines a time interval for which the application requires location information updates. For some applications and usage scenarios location information is needed just once, which can be defined by setting the *duration* parameter to 0. Setting this parameter to -1 implies that location information should be provided until further notice. Location information can also be requested for a defined amount of time by setting the *duration* parameter to a desired value (in seconds). For the defined time interval, location information updates should, either periodically or on event, be reported to the application. In case location information updates are required longer than the originally defined duration, the application can request a renewal. This is defined by the *Request renewal* primitive, with the *duration* of the new provisioning as the only parameter in the primitive. By setting the *duration* parameter to 0, the application indicates that the service is no longer needed and can be terminated.

The SLI also defines a set of primitives for requesting and reporting a context of reported location information. Two type of context information are envisioned, i.e. provisioning of the environmental map and translation from one location type to another. The *Request context* primitive is used by the application for requesting the context, while the *Report context* primitive provides the requested context. The *context type* parameter in *Request context* primitive defines the type of context requested by the application, with two possible context types envisioned at the moment. If the *context type* parameter is set to 1, in the *Report context* primitive the integrated location service will report the utilized *map*. Together with the map, this response has to define the zero-point, i.e. the point used as a reference location, the mapping between map sizes in its default resolution and physical sizes of an environment (in *meters*). If the *context type* parameter is set to 2 in the *Request context* primitive, a translation between location types is requested from the integrated location service. In this case, additional *parameters* resource has to be defined in the *Request context* primitive, i.e. the input and output location types and input location information. The integrated location service responds with the *location types translation* parameter, i.e. location information of the requested output location type. If the map is not cached or the translation cannot be carried by the integrated location service, the integrated location service requests this context information from the provisioning service. This is standardized through the PSI and discussed in the next section. The sequential numbering of the *context type* parameter is done to simplify the extension in case additional context information is required in the future.

In case of a translation to the semantic location, multiple semantic locations can be representative of one local or global location, hence the response can consist of multiple semantic locations. Similarly, if a translation is from the semantic location to the local/global location type, reported location information can be a compound of multiple location coordinates. These segments can then, in the application, be connected to create shapes, such as rooms, buildings, etc. The limitation is that such shapes have to be of a polygonal type, i.e. without curved parts, which is sufficient for most of the space shapes used in practice, noting that curved parts can be accurately approximated by polygonals with small “steps”.

Provisioning Service Interface (PSI)

The PSI defines primitives and parameters exchanged between the integrated location service (IL) and the provisioning services (PS). An overview of the PSI is given in Table 5.2, with an indication of the directionality of primitives.

Firstly, the provisioning service has to register itself to the integrated location service as defined by the *Register service* primitive. The parameters are the *address* where the provisioning service can be invoked and the *location information type* that the provisioning service can provide. The integrated location service can request the “discovery” of provisioning service’s features with the *Discover service* primitive, with the only parameter being the *address* of the provisioning service.

Table 5.1: Summary of the Standardized Localization Interface

Primitive	Description	Direction	Parameters
Specify policy	Specify trade-off policy	LA→IL	policy
Request location	Request location information	LA→IL	type, dimensionality, accuracy, period, on_event, step, duration, movement
Report location	Report location information	IL→LA	location information, distance, confidence interval, movement vector
Request renewal	Request provisioning renewal	LA→IL	duration
Request context	Request context of location information	LA→IL	context type, parameters
Report context	Report context of location information	IL→LA	map (zero-point, map vs. physical sizes) or location types translation

Table 5.2: Summary of the Provisioning Service Interface

Primitive	Description	Direction	Parameters
Register service	Register location provisioning service	PS→IL	address, location information type
Discover service	Discover provisioning service's features	IL→PS	address
Offer service	Offer provisioning service of certain features	PS→IL	dimensionality, accuracy, period, power consumption
Request service	Request provisioning service	IL→PS	address, period, on_event, step, duration
Report service	Report provisioning service	PS→IL	service_granted, duration
Request renewal	Request service renewal	IL→PS	address, duration
Request location	Request location information	IL→PS	address
Report location	Report location information	PS→IL	location information, distance, confidence interval
Request context	Request context of location information	IL→PS	address, context type, parameters
Report context	Report context of location information	PS→IL	context_indicator, address, map or location types translation

With the *Offer service* primitive, the invoked provisioning service reports its ability to provide location information and the anticipated features of this information. A registered service may not be available or may not be able to provide location information of desired features. For example, a GNSS-based provisioning service will usually not adequately provide location information in indoor environments. Moreover, the WiFi sensor may be disabled, thus a WiFi-based provisioning service will not be able to provide location information. The parameters exchanged in the *Offer service* primitive are the expected *accuracy*, *period*, and *power consumption* (in *Watts*) of location information provisioning. Except for the power consumption, these are the same parameters as in the *Request location* primitive of the SLI. This is done so that the integrated location service can assume a central role in the provisioning service invoking decisions, based on the accuracy, period, and power consumption trade-offs. In other words, while the *accuracy* and *period* parameters are essential for applications and should be specified by the applications, we believe the power consumption related optimizations should be performed by the integrated location service.

If a satisfactory provisioning service has been selected, the integrated location service requests the service with the *Request service* primitive. In this primitive, the integrated location service specifies the address of the provisioning service using the *address* parameter. Similar to the SLI case, the *on_event* parameter is used to specify if location information should be provided periodically, with the period defined with the *period* parameter, or on event, in case location information changes more than specified with the *step* parameter. The *duration* parameter of the primitive is used for specifying a time interval for which the service is required. Setting the *duration* parameter to 0 implies that location information should be provided once. Setting this parameter to -1 implies that location information should be provided for a maximum time allowed for service provisioning, as defined by the provisioning service. The provisioning service accepts the service using the *Report service* primitive, with the only parameter being the granted provisioning *duration*. In case the service is required for a longer period of time, it has to be renewed by the integrated location service using the *Request renewal* primitive, with the parameter being the *address* of the provisioning service and the renewal *duration*. In that case the service is either rejected or granted anew for a certain duration using the *Report service* primitive. The service can be rejected by setting the *service_granted* parameter to its non-default binary value 1. The *Request renewal* primitive with the *duration* parameter set to 0 can be used to terminate service provisioning before the expiration of previously defined duration.

Once an agreement for a service has been established, the interaction continues by the integrated location service requesting location information provisioning. This is defined by the *Request location* primitive, with the only parameter being the *address* of the evoked provisioning service. The provisioning service responds using the *Report location* primitive, with the parameters being the *location information* in the pre-specified format, the *distance* in case of the semantic location type, and the *confidence interval* of the provided information. These parameters are defined in the same way as for the SLI's *Report location* primitive.

Similar to the SLI's primitives, the integrated location service can request the context from the provisioning service using the *Request context* primitive, while the requested context is reported using the *Report context* primitive. The integrated location service can request the utilized map or location types translation from the provisioning service. In contrast to the SLI, the *Request* *context* primitive also specifies the address of the provisioning service where the request for context is directed. The provisioning service can either *per-se* provide the context, the context provisioning cannot be granted, or the provisioning service can redirect the request to some other context provisioning service. This is defined using the *context_indicator* parameter of the *Report context* primitive. In case this parameter is set to 0, the context is provided to the integrated location service by the provisioning service, while in case this parameter is set to 1 the context information cannot be provisioned. If the *context_indicator* is set to 2, the requested context can be provisioned by another service, with its address defined with the *address* parameter of the *Report context* primitive. In this case, the *Request context* primitive has to be directed to a new address and the addressed context provisioning service will report the context using its own *Report context* primitive. This scenario is anticipated in case one context provisioning service is a representative for a set of provisioning services.

The providers of provisioning services are expected to provide the context information either through their provisioning services or through a dedicated context provisioning service. This includes the used zero-point and the utilized map of a served environment together with the mapping between map sizes and physical sizes of the environment. In addition, the providers are expected to specify semantic locations in the environment, as well as the translation function between different location information types.

Resource Interaction Interface (RII)

The RII defines a set of primitives for exposing the resources needed by the provisioning services for generating location information. The primitives are related to either getting or configuring a resource by the provisioning service. The *technology* parameter defines the technology used by a particular service for generating location information. The *feature* parameter defines a specific feature of the technology used by the provisioning service. These two parameters can also specify a non-technology specific resource (e.g. an action required from the user of a mobile device). A summary of the RII parameters and primitives is given in Table 5.3, with an indication of primitives' directionality (resources (RS) to the provisioning service (PS) or vice versa). Provisioning services can require resources from multiple technologies or multiple features of the same technology [200, 151]. In this case multiple *Get resource* and/or *Configure resource* primitives should be issued.

The *Get resource* primitive serves as a request for a resource of specific technology and feature, while the *result* of the *Report resource* primitive contains the requested resource. Some examples are provided in Table 5.4. The requested resource can for instance be RSSI values from the WiFi beacon packets or a Bluetooth ID of a mobile device. Similarly, the *Configure resource* primitive can be used for configuring a resource by the provisioning service. Example wise, a request can be directed toward calibrating a gyroscope by shaking a mobile device or synchronizing a mobile device to the UWB anchor in a given environment. In this case, the *Report success* primitive is issued to report the *success* of the requested operation. The resources can be available on a mobile device's and accessible though a request to the device's operating system. The resources can also be available in an infrastructure and accessible remotely.

Since a plethora of provisioning services' types exist and they are in general based on a large variety of technologies and features required for generating location information [27], the parameters in the RII primitives are left abstract, defined only as a one or set of tuples *(technology, feature)*. The exact specification of the tuples is left for the future, because at this point it would be merely a speculation. The reason is that the ubiquitous deployments of provisioning services in different types of environments are still not widely available. Once such ubiquitous deployments are in place, this will yield a set of parameter tuples that are practically used, hence these tuples will thereafter be standardized through the RII. A non-exhaustive set of examples used in practice is given in Table 5.4. An exhaustive set of smart-phone sensors that can be used for generating location information is given in [161].

Table 5.3: Summary of the Resource Interaction Interface

Primitive	Direction	Parameters
Get resource	PS→RS	technology, feature
Report resource	RS→PS	result
Configure resource	PS→RS	technology, feature
Report success	RS→PS	success

Table 5.4: Example resources for generating location information

Technology	Feature
WiFi	RSSI values of observed beacon packets
Bluetooth	Mobile device's Bluetooth identifier
Cellular	Observed cellular base-stations
GNSS	Observed GNSS fixes
None	Calibrate gyroscope by shaking a mobile device
UWB	Synchronize a mobile device to a UWB anchor node

5.1.4 Additional Considerations

If the integrated location service is not desired, the SLI can still be used for interaction directly between the application and the provisioning service. This direct interaction can be achieved because the SLI parameters and actions are designed as a subset of the PSI ones. This assumes a constrained interaction, where location and context information can be requested, but the features of provisioning (e.g. periodically/on event, 2D/3D, etc.) have to be defined ahead.

Both the SLI and PSI are in essence similar to the location information provisioning APIs of the two previously discussed state of the art localization services. Both Google Play services location APIs and iOS's Core Location framework specify the functionality of getting the last known location, as well as the functionality of receiving location updates. The SLI's and PSI's actions, with a specific configuration of parameters, can be used for achieving the same functionalities. Android's framework defines the duration of provisioning, where after its expiration the service has to be requested again. The same functionality can be achieved with the SLI's and PSI's actions, same as the termination of service provisioning which is in line with the iOS's explicit termination functionality. Moreover, both Android's and iOS's interfaces provide functionalities of either displaying a location address or translation of coordinates to regions. These are specific types of translation between the global and the semantic location type in context information exchange in the proposed architecture. Both interfaces support provisioning of maps to the application, which is equivalent to the map provisioning in the proposed architecture. This means that the Google Play services location APIs and iOS's Core Location framework can functionally be "plugged" into the integrated location service and therefore can be viewed as an provisioning service, as well as a valuable source of context information in the proposed architecture.

Actions defined by the SLI are asynchronous, i.e. an action does not block further execution until a response to the action has been received. Asynchronous interaction is utilized since the provisioning of location information can be requested on event, hence the information can be provided only once in a relatively long period of time. It is reasonable to allow further processing and timely reaction on external events. The PSI's actions are also asynchronous, since e.g. a request for service can be issued by the integrated location service to many provisioning services. Not all parameters are needed for the integrated location service to be able to provide this information to the application, hence the asynchronous interaction. The RII's actions can be both synchronous and asynchronous, depending on the needs of the provisioning services.

Provisioning service developers have to define the expected accuracy, period, and power consumption of provisioning such information. Therefore, a means for objectively accessing these parameters is required. This can be done by evaluating provisioning services in existing testbed environments or by using publicly available data traces for such evaluation, as discussed previously in the thesis. Also, predictions of accuracy, period, and power consumption in dependence to environment type will be necessary, where one such prediction on an example of WiFi-based indoor fingerprinting has been discussed previously in the thesis. Finally, the internal mechanisms of the provisioning service for dynamically predicting their accuracy, period, and power consumption based on the conditions in an environment and the availability and quality of the resources used for generating location information will be needed (e.g. [151]).

5.2 Selection and Aggregation of Location Information Provisioning Services

Although the functionalities of the integrated location service have been specified, the question of how to select the provisioning services remained open. One way of performing the selection is to address a request for location information immediately after receiving it. Considering requests individually results in a loss of optimization potential, in case there is a possibility of delaying the addressing of requests. Assuming that it is possible to delay the addressing of requests, the selection of provisioning services can be performed in a time bucketed fashion. Time bucketed selection opens a space for non-trivial optimizations because potentially multiple requests can be simultaneously considered.

In the following, we first define a model for a time bucketed selection of provisioning services. We then provide two algorithms for selection and aggregation of location information provisioning services. In general, the objective of an algorithm for selection and aggregation of provisioning services is to minimize a certain cost of location estimation while meeting the applications' requirements. For the algorithms we propose, the objectives are specific - the primary objective is latency requirements satisfaction and the secondary objective is satisfaction of accuracy requirements. One algorithm is subject to per-request power minimization, while the other is subject to per-time bucket power minimization. We show that by bundling requests per-time bucket and not focusing on individual ones, one can achieve roughly 25% better performance in terms of power consumption, while trading-off accuracy satisfaction.

5.2.1 A Setting for Selection and Aggregation of Location Information Provisioning Services

We assume that each request for location information specifies the desired accuracy and latency of location information. The latency requirement is specified as the maximum acceptable delay time of receiving location information after the request is sent, while the accuracy requirement specifies the maximum acceptable mean localization error of the provided information. We also assume that each provisioning service is, upon request, able to report it provisioning features to the integrated location service.

We further assume that that there is a potential for delaying the addressing of the arriving requests for location information from the applications. Such delaying is in many cases possible, for example in periodic provisioning of location information or in provisioning “on event”, i.e. if the current location information to a certain degree changes from the previous location of a device. The feasibility of delay-tolerant localization has also been discussed and acknowledged in [201]. Due to the feasibility of delaying the addressing of requests for location information and to be able to perform sophisticated optimizations, we focus on a time bucketed operation of the integrated location service.

In the time bucketed model, the time interval by which the addressing of the request can be delayed, with the purpose of optimization by possibly addressing multiple requests, is dependent on the latency requirement specified by the request. Furthermore, the discovery of provisioning features of available provisioning services is in practice not instantaneous. Additionally, due to mobility that results in dynamically appearing and performance-wise variable provisioning services, the discovery has to be performed periodically. To align the need for potentially aperiodic time bucketing from the applications’ perspective and periodic time bucketing from the provisioning services’ perspective, we design a periodic time-bucketed model for the selection of provisioning services. Note that the algorithms proposed in the following section would also be able to operate correctly for the case of aperiodic time buckets.

Periodic time bucketing means that the up-to-date information about the availability and features of different provisioning services are available to the integrated location service only at certain periodic time instances. The decision of which provisioning services to select and invoke should hence be made immediately after this information becomes available. The developed algorithms work under the assumption of a periodic time bucket, where at the beginning of each time bucket the requests arrived during the previous bucket are addressed, as shown in Figure 5.2.

The integrated location service also serves as a location information caching entity, therefore some of the requests for location information can be resolved in the same bucket through caching if the location information available in the current time bucket is satisfactory for addressing the requirements of a given request. The caching step would essentially come before the decision on which provisioning services to invoke. Upon receiving a request for location information, the integrated location service would immediately check if the cached information can be used

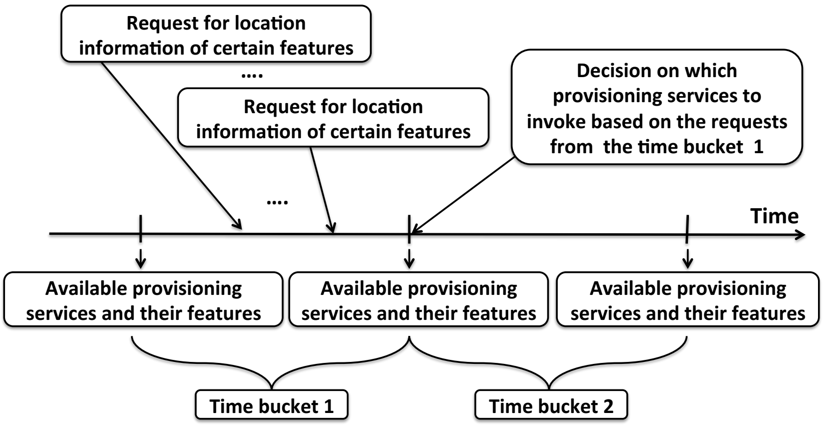

Request for location information of certain features
....
Request for location information of certain features
Decision on which provisioning services to invoke based on the requests from the time bucket 1
....
Time
Available provisioning services and their features
Available provisioning services and their features
Available provisioning services and their features
Time bucket 1
Time bucket 2

Figure 5.2: Timeline of the decision-making process

to resolve the request. If the cached information is not sufficient, the request is addressed at the beginning of the next time bucket through selection and invocation of suitable provisioning services. Since caching can be separated from the selection and aggregation of provisioning services, we consider only requests that cannot be resolved by caching.

Specification of Input Parameters

In this section, we provide a specification of the input parameters required by the developed algorithms for selection and aggregation of provisioning services. The first set of input parameter consists of the desired accuracy and latency features of location information provisioning from all requests that were received by the integrated location service during a time bucket. In addition, the integrated location service notes the decision delay for each request, which is the time interval between the time when the request was received and the end of the time bucket. This decision delay is then subtracted from the latency requirement, resulting in a so-called service time requirement that takes into account the delay caused by waiting until the end of the $N-1^{st}$ time bucket. Note that the resulting service time can be a negative number. Both algorithms will in that case select the provisioning service with the smallest latency of provisioning. The requirements issued by the applications can then be written as:

$$\begin{aligned} requirements &= (accuracy, service\_time) \\ where: \quad service\_time &= latency - decision\_delay \end{aligned} \tag{5.1}$$

At the beginning of the N^{th} bucket, the integrated location service, through discovery, obtains the currently available provisioning services and their features. The features of provisioning include latency, accuracy, and power consumption. The latency is defined as the mean expected delay of providing, while the accuracy is specified as the mean expected localization error. The power consumption specifies the estimate of power required for generating location information by a provisioning service. Such an “offering” can be written as:

$$offering = (accuracy, latency, power) \tag{5.2}$$

Generation of Virtual Provisioning Services

The fusion of location information provided by different provisioning services in average results in a more accurate location information than any of the individual location information. Improved accuracy due to fusion of provisioning services can obviously be beneficial in satisfying the accuracy requirements from the applications. This expected accuracy of a fused provisioning service, as well as the latency and power consumption of such a fused provisioning service, have to be known to the integrated location service for making a decision about the invocation of such a service. We call the estimation of parameters of fused provisioning services the generation of "virtual provisioning services". In this following, we provide an algorithm for the generation of virtual provisioning services.

Fusion of provisioning services yields a more accurate virtual provisioning service in comparison to each provisioning service in the combination if these individual provisioning services have similar accuracy features [196]. In the proposed algorithm for generation of virtual provisioning services, we base this similarity on the accuracy difference threshold acc_{th} between the accuracy features of different provisioning services. Only in case the similarity between services exists, the virtual provisioning services are created as all combinations of similar provisioning services.

The final location information, i.e. the one provided by a virtual provisioning service, is the arithmetic mean of location information provided by all provisioning services that compose a given virtual provisioning service. Let us assume that N provisioning services reported location information (x_i, y_i), $i = 1, ..., N$. The location information (x_F, y_F) reported by the virtual provisioning service is then given as follows:

$$(x_F, y_F) = \left(\frac{x_1 + ... + x_N}{N}, \frac{y_1 + ... + y_N}{N} \right) \tag{5.3}$$

The latency of provisioning of such a fused location information then equals the highest latency among the latencies of all services in the combination, while the power consumption feature equals the sum of power consumption features of all provisioning services in the combination. The accuracy $acc_{virtual}$ of the generated virtual service can, assuming uncorrelated location information from the provisioning services, be approximated as follows, where the $acc_{pr,i}$ specifies the accuracy of the i^{th} provisioning service:

$$acc_{virtual} = \frac{\frac{1}{N} \sum_{i=1}^{N} acc_{pr,i}}{\sqrt{N}} \tag{5.4}$$

Example-wise, let us assume three provisioning services A, B, and C with the features as given in (5.5). Furthermore, let us assume acc_{th} of 0.1, meaning the following combinations of services can be generated as virtual provisioning services: (A), (B), (C), (A,B), (A,C), (B,C), and (A,B,C).

$$\begin{aligned} offering_a &= (accuracy, latency, power) = (1.00, 1.0, 1.0) \\ offering_b &= (accuracy, latency, power) = (1.05, 1.1, 0.9) \\ offering_c &= (accuracy, latency, power) = (0.95, 1.2, 1.3) \end{aligned} \tag{5.5}$$

It follows that for the virtual service (A,B,C) the accuracy feature equals $(1.00 + 1.05 + 0.95)/3\sqrt(3) = 0.58$. Furthermore, the latency feature equals $max(1.0, 1.1, 1.2) = 1.2$, while the power consumption feature equals $\sum(1.0, 0.9, 1.3) = 3.2$.

5.2.2 Decision Making Algorithms

In the following, we outline the two algorithms for the selection and aggregation of location information provisioning services. As mentioned, their primary objective is meeting the latency, followed by meeting the applications' accuracy requirements. The Per-Request Satisfaction Algorithm (PRSA) is subject to per-request power consumption minimization, while the Per-Time-bucket Satisfaction Algorithm (PTSA) is subject to minimizing the power consumption over a time bucket. While we acknowledge the relevance of optimizing for accuracy or Pareto optimizing for both requirements for certain scenarios, we decide to focus primarily on the latency satisfaction. This is because for many scenarios it is better, from the user's perspective, to receive location information timely, even a less accurate one, than to interrupt the provisioning periodicity for a more accurate location information [202, 147]. Compared to accuracy, latency also has higher significance for novel use-cases related to location-aided improvements in wireless networks [11]. Moreover, we focus on the power consumption minimization because of its relevance for mobile devices with limited power sources [9].

Per-Request Satisfaction Algorithm

Intuitively stated, the aim of the PRSA algorithm is to meet the service time and accuracy requirements of each request in a time bucket in a way that minimizes per request power consumption. The execution diagram of the PRSA is given in Figure 5.3. In this algorithm, requests received in a previous time bucket are sequentially addressed at the beginning of a current bucket. From the set of virtual services generated at the beginning of a current time bucket, a service with minimum power consumption that satisfies the (accuracy, service time) requirements of the first request is stored. In case such a service does not exist, the algorithm tries to find a service that satisfies the service time requirement of the request and that is closest to meeting the accuracy requirement. If such a service again does not exist, the algorithm finds the service with the smallest latency and stores it. The algorithm then addresses the next request, until all requests are addressed. Upon addressing all requests, the algorithm finds a unique set of provisioning services from the set of stored virtual provisioning services. These services are then invoked and upon the reception of location information this information is, first and where appropriate, fused and, second, reported to the applications based on their requirements.

Per-Time-bucket Satisfaction Algorithm

The aim of the PTSA algorithm is to satisfy the requirements from the applications, while trying to minimize power consumption over a time bucket. Figure 5.4 depicts the execution diagram of the PTSA. The first step of the execution is ordering by serving time the requests received during a previous time bucket. In the next step, a request with the strongest accuracy requirement is selected and all virtual provisioning services that can satisfy its (accuracy, service time)

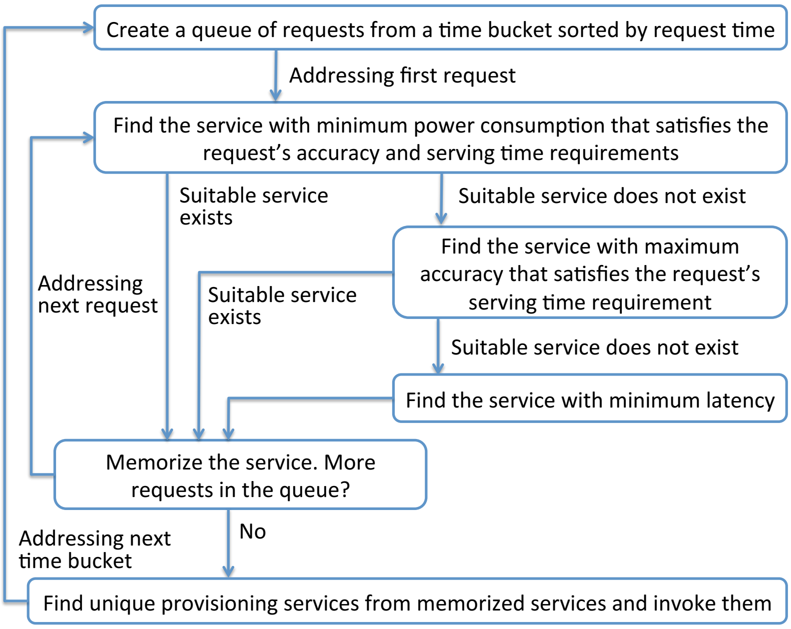

Create a queue of requests from a time bucket sorted by request time
Addressing first request
Find the service with minimum power consumption that satisfies the request's accuracy and serving time requirements
Suitable service exists
Suitable service does not exist
Find the service with maximum accuracy that satisfies the request's serving time requirement
Addressing next request
Suitable service exists
Suitable service does not exist
Find the service with minimum latency
Memorize the service. More requests in the queue?
Addressing next time bucket
No
Find unique provisioning services from memorized services and invoke them

Figure 5.3: Execution diagrams of the Per-Request Satisfaction Algorithm (PRSA)

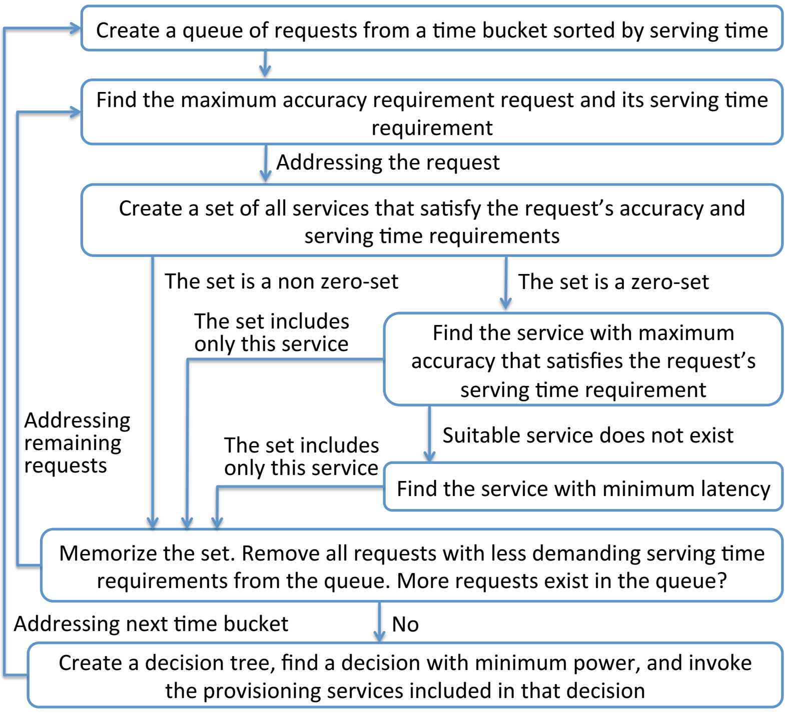

Create a queue of requests from a time bucket sorted by serving time
Find the maximum accuracy requirement request and its serving time requirement
Addressing the request
Create a set of all services that satisfy the request's accuracy and serving time requirements
The set is a non zero-set
The set is a zero-set
The set includes only this service
Find the service with maximum accuracy that satisfies the request's serving time requirement
Addressing remaining requests
Suitable service does not exist
The set includes only this service
Find the service with minimum latency
Memorize the set. Remove all requests with less demanding serving time requirements from the queue. More requests exist in the queue?
Addressing next time bucket
No
Create a decision tree, find a decision with minimum power, and invoke the provisioning services included in that decision

Figure 5.4: Execution diagrams of the Per-Time-bucket Satisfaction Algorithm (PTSA)

requirements are stored as one set of services. In case none of the services can satisfy these requirements, the set consists only of a service that can satisfy the service time requirement in a way that is closest to meeting the accuracy requirement. If none of the services can satisfy the service time requirement, the set consists only of a service that comes closest to the service time requirement. Once the set is created, the algorithm continues by finding the strongest accuracy requirement of the requests that were, by serving time, preceding the previously addressed request, i.e. the requests that have stronger service time requirements. The execution stops when the requests with the strongest service time requirement in a time bucket is addressed.

The described procedure yields a number of sets of virtual provisioning services that can address the requests with strongest accuracy and service time requirements, while the rest is implicitly addressed. The execution of the algorithm continues by creating a decision tree where each decision includes exactly one element from each set as an input and the overall power consumption of that decision as an output. The decision with minimum power consumption is then selected, and the provisioning services that are part of that decision are invoked. The decision tree is built in a following way, given that the objective is the power minimization of the decision. The reference branch includes the service with the smallest power consumption in each set. The following branches include only combinations in which there is at least one repetition of provisioning services across sets, since all other combinations necessarily have higher power consumption than the reference branch and are thus irrelevant. Following this procedure reduces the execution time of the algorithm by reducing the number of branches in the decision tree.

Example of Execution

Let us assume that in the $N-1^{st}$ time bucket the integrated location service received 4 requests for location information, as given in Table 5.5. As stated previously, each request specifies the (accuracy, latency) requirements, while the integrated location service logs the decision delay. The service time can be calculated out of latency and decision delay, and together with the accuracy defines the final set of requirements for each request. Furthermore, let us assume that at the beginning of the N^{th} time bucket the integrated location service discovers 4 available provisioning services with their offerings as given in Table 5.6. The initial step of both algorithms is to generate virtual provisioning services. For that purpose, let us also assume the accuracy threshold acc_{th} of 0.15, which yields the virtual provisioning services as given in Table 5.7. The next step of the PRSA and PTSA execution is ordering of the requests sequentially and by service time, respectively. The ordering of requests by service time is given in Figure 5.5.

The PRSA continues by taking the first arriving request, i.e. the request #4. That request defines the accuracy and service time requirements of 1.99 and 0.39, respectively. The virtual services that can satisfy its requirements are the services II, III, IV, and VI. Among them, the service IV has the smallest power consumption, hence that service is stored. The algorithm then continues by addressing the request #2. The requirements of the request #2 can be satisfied by the virtual service VI only, therefore the algorithm stores that service. Following the same principle, the virtual service I is stored for the request #1. For the request #3, the only virtual service that can satisfy its requirements is the virtual service IV, thus this service is stored. The algorithm then yields the virtual services I, IV, and VI which translates to the provisioning services A, C, and D. Hence, the provisioning is requested from these services.

Table 5.5: Requirements of all requests

Request	Accuracy	Latency	Decision delay	Service time
#1	1.09	0.70	0.21	0.49
#2	1.39	0.50	0.21	0.29
#3	1.59	0.60	0.41	0.19
#4	1.99	0.50	0.11	0.39

Table 5.6: Offerings of all provisioning services

Offering	Accuracy	Latency	Power consumption
A	1.0	0.4	1.5
B	1.1	0.3	2.0
C	1.4	0.2	2.1
D	1.5	0.1	1.6

The PTSA starts by finding a request with the strongest accuracy requirement. For the given example, that is the request labeled with #1. The algorithm stores a set of all services that can satisfy the requirements of the request #1, i.e. the set (I,V,VI). The algorithm continues by finding the request with the strongest accuracy requirement among the requests with lower service time requirement than the previously addressed request. In the example, these requests are #2, #3, and #4, among which #2 has the strongest accuracy requirement. The set of virtual services that can address the requirements of the request #2 includes only one service, i.e. the virtual service VI. Note that addressing the request #2 implicitly addresses the request #4, since that request has lower requirements on both accuracy and service time than the request #2. For the last request i.e. the request #3, the only service that can address its requirements is the virtual service IV. The algorithm therefore yields three sets of virtual services as follows: set 1 - (I, V, VI), set 2 - (VI), and set 3 - (IV). In the notation using provisioning services these set are: set 1-(A,[A,B],[C,D]), set 2-([C,D]), and set 3-(D).

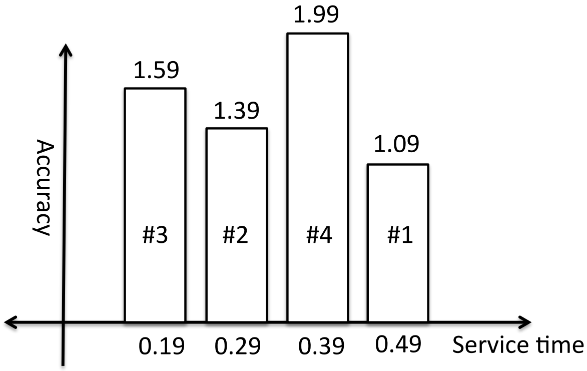


Figure 5.5: Requests ordered by service time

Table 5.7: Virtual provisioning services

Virtual service	Accuracy	Latency	Power	Elements
I	1.00	0.4	1.5	A
II	1.10	0.3	2.0	B
III	1.40	0.2	2.1	C
IV	1.50	0.1	1.6	D
V	0.74	0.4	3.5	[A,B]
VI	1.03	0.2	3.7	[C,D]

In the following step, the algorithm builds a decision tree as shown in Table 5.8. The first possible decision in the tree is built by selecting, from each set, a service with the smallest power consumption. Other decisions are then build only of elements that are repeated across sets, since all other combinations necessarily have higher power consumptions than the first decision. Following this procedure reduces the number of decisions and therefore improves the execution time of the algorithm. In the example, the decision ([A,B],[C,D],[D]) is not taken into consideration, since the virtual service [A,B] does not occur in other sets. The decision 2 from the table is, however, taken into consideration, since the virtual service [C,D] drawn from the set 1 also appears in the set 2. This decision also turns out to be the one that minimizes the overall power consumption and it is therefore selected as the final decision. In other words, the algorithm yields that the services C and D should be invoked for provisioning.

Table 5.8: Decision tree for the PTSA

Decision #	Set 1	Set 2	Set 3	Unique services	Power
1	A	C,D	D	A,C,D	5.2
2	C,D	C,D	D	C,D	3.7

5.2.3 Evaluation Procedure, Scenarios, and Parameters

For the performance evaluation, we simulate the realistic operation of two instances of the integrated location service, where the two instances differ only in the algorithms for the selection of provisioning services. The goal of the evaluation is to capture the difference in the performance of the integrated location service due to contrasting algorithms for the selection of provisioning services.

Evaluation Procedure

The integrated location service receives requests for location information of certain features. The three relevant parameters for modeling each request are the decision delay, and accuracy and latency requirements. The decision delay has been modeled using a uniform distribution from a time bucket duration interval because the probability of receiving a request is constant across the duration of a time bucket. The latency and accuracy requirements have also been modeled using uniform distributions. In the same way, the offerings of each provisioning service, i.e. the latency, accuracy, and power consumption, have been modeled using uniform distributions.

Uniform distributions have been selected because it is hard to give more accurate predictions of the requirements and provisioning features, given that both of them can have significant variations [203]. The numbers of requirements and offerings in a time bucket have been derived from discrete uniform distributions. Uniform distributions for the numbers of requirements and offerings have been selected because requirements and offerings are expected to be relatively small and equally probable numbers [3, 204]. Given the generated requirements and offerings as inputs, the two algorithms were used to select the provisioning services for each time bucket.

As the performance evaluation criteria, we specify the following set of metrics: power consumption, latency and accuracy requirements satisfaction, and joint satisfaction of both requirements. The metrics concerning satisfaction of certain requirements are obtained as the ratio between satisfied requirements and the total number of requirements generated in a simulation run. We simulate 1000 times the execution of each algorithm over a time-bucket.

Evaluation Scenarios and Parameters

Our aim is to generate the evaluation scenarios in a way that qualitatively captures a wide parameter space, with parameters being the numbers of requirements and offerings, latency and accuracy requirements from the applications, and latency, accuracy, and power consumption features of the provisioning services, as depicted in Figure 5.6. The first classification of scenarios has been made based on the number of offerings from the provisioning services and requirements from the applications. We therefore distinguish a set of scenarios in which the number of offerings is statistically larger than the number of requirements, and vice-versa. For both sets of scenarios differentiated by the number of offerings and requirements, we further distinguish scenarios based on latency requirements from the applications and latency offerings from the provisioning services. Hence, we differentiate between scenarios in which the latency requirements are statistically “weaker”, “comparable” or “stronger” than the latency offerings. Finally, for each such scenario, we similarly specify scenarios based on accuracy requirements and offerings. Thus, we also distinguish scenarios in which the accuracy requirements are weaker, comparable or stronger than the accuracy offerings. For this qualitative distinction between the requirements and offerings we utilize the Cohen’s coefficient, which characterizes a statistical relationship between two distributions. Cohen’s d coefficients smaller than 0.2 represent a small difference, values smaller than 0.5 represent a medium difference, while higher values than 0.5 represent a high difference between distributions. We use Cohen’s d coefficient of 0 for the comparable latency/accuracy requirements and offerings and 0.4 and -0.4 for the stronger and weaker requirements than offerings, respectively.

In regards to the parameterizations of latency and accuracy requirements, we make use of the claim that such requirements highly depend on the use-case scenarios [205, 139]. One set of use-cases have a stronger requirement on the accuracy of provided location, in contrast to the latency of provisioning. For the others, it is more important to get a location estimate fast, but with a lower accuracy. We therefore define two sets of requirements issued by the applications in a certain time bucket, with a main differentiation being that the first set is more focused on accuracy, but with a smaller latency requirement, while the other has higher latency, but smaller accuracy requirement. For the first set of requirements, i.e. the ones focused on accuracy, we draw the accuracy requirement as a random number between 0.75 and 1.25 m, while the latency

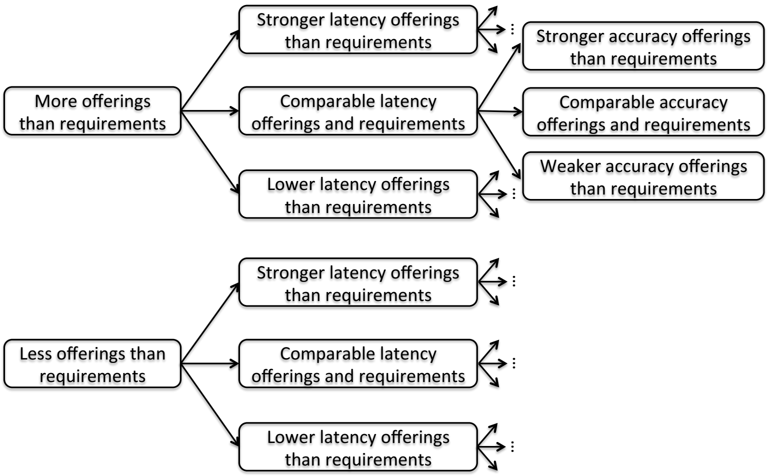

More offerings than requirements
Stronger latency offerings than requirements
Comparable latency offerings and requirements
Lower latency offerings than requirements
Stronger accuracy offerings than requirements
Comparable accuracy offerings and requirements
Weaker accuracy offerings than requirements
Less offerings than requirements
Stronger latency offerings than requirements
Comparable latency offerings and requirements
Lower latency offerings than requirements

Figure 5.6: Differentiation of the evaluation scenarios

is drawn as a number between 0.75 and 1.25 s. For the second set of requirements we draw accuracy and latency requirements as uniform numbers between 1.4 and 2 m, and 0.65 and 1.05 s, respectively. This specification of requirements is constant across all evaluation scenarios.

Suggestions given in [35] have been utilized for the parameterization of the offerings. The authors in [35] provide the performance evaluation of various localization solutions in the same environment, using a unified evaluation procedure, and an automated human-free infrastructure for evaluation of localization solutions [136]. The work distinguishes different classes of location provisioning services with a trade-off of either providing location information with higher accuracy, but with higher latency and power consumption, or providing less accurate location information, but faster and with lower power consumption. We therefore, in a similar fashion as for the requirements parameterization, select two sets of location information provisioning services distinguishable by the accuracy vs. latency and power consumption trade-off. The contributions in [190, 148] suggest that the performance of localization solutions highly depends on the environment characteristics (e.g. size, building material). Since the performance varies, we are fairly free in the parameterization of the offerings, as long as such offerings are consistent with the accuracy vs. latency and power consumption trade-off. In order to create the described evaluation scenarios and to be compliant with the specified Cohen's coefficients, we define the offerings that are statistically either higher, comparable, or lower than the requirements from the applications. The latency requirements of the first set of offerings are drawn as uniform numbers between 0.65 and 1.15, 0.75 and 1.25, and 0.85 and 1.35 s for the scenarios with respectively strong, comparable, and weak latency offerings in comparison to requirements. Similarly, for the first set of offerings the accuracy requirements are drawn as uniform numbers between 0.65 and 1.15, 0.75 and 1.25, and 0.85 and 1.35 m for the scenarios with strong, comparable, and weak latency offerings in comparison to requirements, respectively. The second set of offerings is latency-wise parameterized as a uniform number from the intervals 0.55 and 0.95, 0.65 and 1.05, and 0.75 and 1.1 s for respectively the same scenarios [35], while the accuracies are drawn from intervals 1.3 and 1.9, 1.4 and 2.0, and 1.5 and 2.1 m for the same scenarios. The accuracy offerings are compliant with the state-of-the-art accuracies of a large variety of localization solutions reported in [61], while the selected latencies also adhere to the usual applications' requirements reported in the literature [206, 207]. The power is, for the first set of offerings, drawn as a uniform number between 1.2 and 1.8, while for the second set this interval was between 0.8 and 1.2. These intervals for power consumption represent a normalized power consumption with a ratio 3:2, which is consistent with the accuracy and power consumption trade-off, as for example reported in [147].

In scenarios with more offerings than requirements, per time bucket we select the number of requirements from each set, distinguished by the accuracy and latency requirements, as a random number between 1 and 4, while the number of offerings is, for each set of offerings, derived as a random number between 1 and 7. In scenarios with less offerings than requirements, the numbers of requirements and offerings are for each set selected as uniform numbers between 1 and 7, and 1 and 4, respectively. Such a selection roughly fits the practice, as for example suggested in [3] and, although the number of both requirements and offerings is expected to rise in the future, this parameterization is sufficient to qualitatively capture the behavior of algorithms in two distinguishable scenarios based on the requirements and offerings relation.

As discussed previously, the integrated location service operates in a periodic time bucketed fashion. In the evaluation, we set the duration of each time bucket to 0.5 s. This value has been selected since it is fairly smaller that the update period of roughly 1-2 s needed for users' tracking scenarios [147]. Periodic time bucketing simplifies the performance analysis of the integrated location service and provides a lower bound on the performance, since the time buckets start independently of the latency requirements from the applications. In other words, the duration of a time bucket can exceed the latency specified by a request addressed in that time bucket, which would result in a unsatisfied latency requirement for that request.

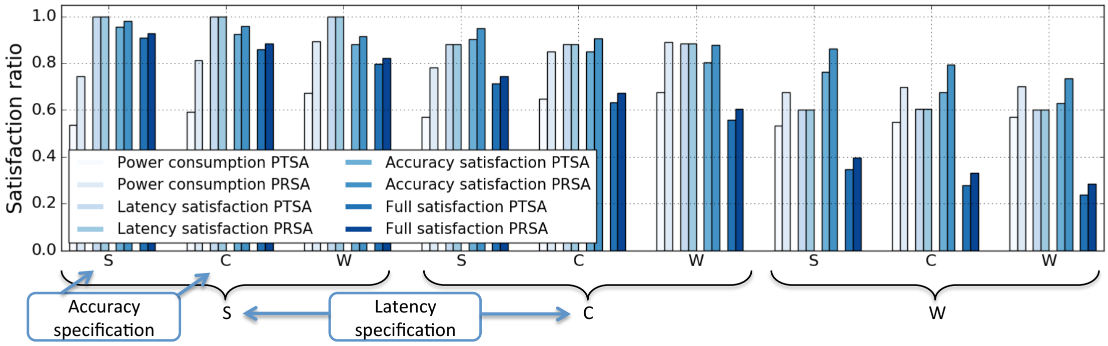
5.2.4 Evaluation Results

The evaluation results are summarized in Table 5.9. The first three columns specify different evaluation scenarios differentiated based on relations between the number of offerings and requests, statistical difference between the latency features of the offerings and latency requirements of the requests, and statistical difference between accuracy feature of the offerings and accuracy requirements of the requests, respectively for columns 1, 2, and 3. The fourth column specifies the overall number of requests generated in a simulation run. The numbers of requests slightly differ per simulation run in the same set of evaluation scenarios differentiated based on the relation between the numbers of offerings and requests. The reason for this behavior is related to the used simulation model in which the number of requests for each time bucket is drawn as a random number from a certain interval of numbers. A larger discrepancy in the numbers of requests is visible across different evaluation scenarios differentiated based on the relation between the number of offerings and requests. This is because of different intervals for drawing random numbers (i.e. generating the number of requests) for such scenarios. The next two columns of the table give the overall power consumption, while the following column provides the relative difference in the power consumption of the algorithms. The following columns represent the satisfaction on respectively latency, accuracy, and both requirements, together with the relative difference in performance of two algorithms for each metric. An exception is in the latency satisfaction, where due to exactly the same performance of the algorithms, their relative difference is not provided.

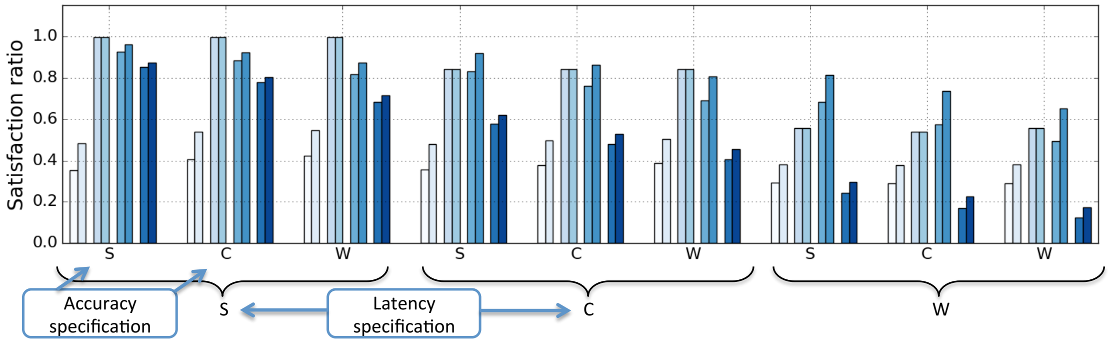
Table 5.9: Summarized results of the evaluation

Offerings vs. requests			# requests	Consumed power			Satisfaction of requests							
Number	Latency	Accuracy					Latency-wise		Accuracy-wise			Both requirements-wise		
				PTSA	PRSA	Diff.[%]	PTSA	PRSA	PTSA	PRSA	Diff.[%]	PTSA	PRSA	Diff.[%]
More	Strong	Strong	4870	2615.4	3627.6	27.9	4860	4860	4649	4772	2.6	4424	4521	2.2
More	Strong	Comp.	4959	2943.3	4032.3	27.0	4947	4947	4577	4751	3.8	4255	4386	3.1
More	Strong	Weak	4961	3338.8	4428.3	24.6	4950	4950	4368	4539	3.9	3956	4076	3.0
More	Comp.	Strong	4970	2842.9	3885.5	26.8	4382	4382	4494	4714	4.9	3542	3694	4.2
More	Comp.	Comp.	4947	3207.1	4205.6	23.7	4352	4352	4201	4487	6.9	3134	3327	6.1
More	Comp.	Weak	5060	3428.2	4511.5	24.0	4466	4466	4069	4437	9.0	2822	3064	8.6
More	Weak	Strong	4986	2654.8	3367.5	21.1	3006	3006	3810	4305	13.0	1737	1982	14.1
More	Weak	Comp.	4994	2736.1	3487.2	21.5	3025	3025	3382	3971	17.4	1386	1655	19.4
More	Weak	Weak	5019	2857.8	3522.8	18.8	3025	3025	3159	3696	17.0	1194	1428	19.6
Less	Strong	Strong	8076	2847.1	3913.9	27.2	8055	8055	7475	7767	3.9	6875	7058	2.7
Less	Strong	Comp.	7897	3211.7	4259.0	24.6	7872	7872	6982	7288	4.4	6159	6332	2.8
Less	Strong	Weak	7877	3334.9	4306.1	22.6	7845	7845	6436	6867	6.9	5392	5639	4.6
Less	Comp.	Strong	8087	2885.8	3879.9	25.6	6802	6802	6730	7421	10.2	4675	5026	7.5
Less	Comp.	Comp.	7956	3001.6	3951.7	24.0	6700	6700	6055	6857	13.2	3825	4218	10.3
Less	Comp.	Weak	8139	3164.1	4099.0	22.8	6846	6846	5617	6575	17.0	3289	3714	12.9
Less	Weak	Strong	7930	2319.3	3031.9	23.5	4428	4428	5408	6453	19.3	1935	2353	21.6
Less	Weak	Comp.	8055	2322.5	3037.6	23.5	4337	4337	4636	5920	27.7	1381	1825	32.1
Less	Weak	Weak	8151	2349.7	3108.9	24.4	4541	4541	4035	5322	31.9	1024	1417	38.4

The per-request normalized performance results are given in Figure 5.7. The figure depicts the relation between power consumptions, latency requirements satisfactions, latency requirements satisfactions, and both requirements satisfactions. As visible from the figure, for all evaluation scenarios the PTSA algorithm consumes less power per request than the PRSA algorithm. Similar observation is given in Table 5.9, showing that across all evaluation scenarios the PTSA algorithm consumes roughly 25 % less power than the PRSA algorithm. The reason for lower power consumption comes from the fact that the PTSA performs optimization on all requests in a time bucket, while the PRSA tries to minimize power consumption individually. Due to that, the number of possible combinations of mapping the offerings to the requirements is higher for the PTSA. To demonstrate the difference, let us assume that N is the number of requests that have to be addressed and M is the number of available provisioning services for a certain time bucket. The number of virtual provisioning services that can be generated then has O(2^M) combinations. The number of possible mappings in the PRSA is then O($N \times 2^M$) because for each of the N requests one of the O(2^M) virtual provisioning services can be selected. The asymptotic number of combinations for the PTSA is O($\binom{N\times 2^m}{N}$) because for each of the N requests a set of possible offering is generated and in a later step exactly one virtual service per set can be selected.



(a) Scenarios with more offerings than requests



(b) Scenarios with less offerings than requests

Figure 5.7: Summarized per request normalized evaluation results (S-stronger, C-comparable, W-weaker offerings vs. requests)

Furthermore, both algorithms generally yield exactly the same performance it terms of latency satisfaction, as can be seen in Table 5.9. The reason for this behavior is that both algorithms have the same strategy for addressing the requests where the offerings cannot support the latency requirements of the request, i.e. selecting an offering that comes closest to meeting the latency requirement, without regarding the accuracy or power consumption of such an offering.

The PRSA yields better performance in terms of the accuracy requirement satisfaction and consequently in terms of the satisfaction of both requirements. The reason is that the PTSA, after addressing a request, assumes that all requests with lower accuracy and latency requirements are implicitly addressed as well. That means that, in case the latency and/or accuracy requirements of the request that is being addressed cannot be fulfilled by the available offerings, the algorithm finds the most acceptable offering, but also assumes the implicit fulfillment of requests that have lower latency and accuracy requirements than the current request, which is in that case not necessarily correct. The PRSA will, on the other hand, sequentially address all requests, so even if some requests cannot be fulfilled, the others with lower latency and accuracy requirements will be explicitly addressed, which increases the probability of satisfying their requirements. The relative difference in accuracy and both requirements satisfaction in this case depends on the evaluation scenario. In scenarios where the latency offerings are statistically stronger than, or even comparable to, the latency requirements, the relative difference in the performance is rather small, i.e. mostly smaller than 10%. However, in case the latency offerings are statistically weaker than the latency requirements, the PRSA performs significantly better than the PTSA in terms of accuracy and both requirements satisfaction.

5.3 Implementation of the Standardized Localization Service

In the following sections, we provide the Standardized Localization Service (SLSR), a prototypical implementation of the standardized localization service architecture specified previously. We further instantiate and evaluate the developed service in the previously discussed testbed specifically designed for performance evaluation of RF-based indoor localization solutions [136]. Our results demonstrate the benefits of the SLSR compared to a single localization solution in terms of localization accuracy enhancements. We also characterize the influence of different functional properties on the overall performance of the developed service, with the aim of demonstrating that each of these properties benefits the overall performance of the SLSR.

5.3.1 Overview of the Standardized Localization Service

Previously described components of the standardized localization service architecture could be stand-alone applications on a mobile device to be localized, but could as well be deployed in an environment surrounding the device. The latter is tightly related to the IoT vision of smart environments providing services to the users and devices in such environments [208]. Before discussing the implementation of the SLSR, we have to position each component of the standardized localization service architecture in these two classes.

The majority of location-based applications are stand-alone entities on a mobile device, primarily used for providing some location-based service. However, as discussed in the introduction of the thesis, location information is an important part of context information with high potential for improving wireless networks operation. Such scenarios often require location information of a device to be shared with other entities surrounding the device. We therefore also envision applications leveraging such information to be deployed in an environment surrounding the device.

The integrated location service can be viewed as an individual per-device entity, but also as a global centralized entity for all mobile devices. The latter would allow for a global optimization of invocation of location information provisioning services and would simplify sharing of location information across devices and applications. However, such a global integrated location service would intrinsically have increased privacy and security risks, while the avoidance of such issues is of immense importance for location information [209]. Furthermore, such a global integrated location service would, because of the large number of devices to be supported, be hardly scalable and would have to be a stateless entity. Therefore, the "intelligence" of requesting location information would have to be pushed to the applications. This means that, for example, periodic or on event-based provisioning of location information could not be explicitly specified, but should be done by periodically issuing requests for current location information. If such advanced knowledge of future requests for location information is available at the integrated location service level, the integrated location service could leverage it for optimizing the invocation of provisioning services, which could potentially yield reductions in power consumption and latency of location information provisioning, as indicated in [198]. We therefore believe the integrated location service should be fully operating on a mobile device to be localized.

As stated previously, we believe that both location information provisioning services and resources for generating location information could be stand-alone entities on a mobile device, but could also, along the IoT vision, be embedded in smart environments surrounding the device to be localized.

Functional Overview of the Standardized Localization Service

A detailed functional overview of the SLSR is given in Figure 5.8. The procedure starts by different location-based applications requesting location information of certain features from the integrated location service. As mentioned, the features of provisioning are desired accuracy and latency of location information, where location information can be requested once, or periodically or on an event for a certain duration. These requests for location information are received at a listener for requests for location information, where their arrival times are logged.

The requests are then forwarded to a long-term requests interpretation entity. The role of the long-term requests interpretation is to translate a request for periodic provisioning or a request for provisioning on events into repeated individual requests for location information. For example, if a request specifies periodic provisioning with a period of 1 s in the duration of 30 s, the long-term requests interpretation will translate such request into 30 requests for location information with the latency of 1 s at appropriate times.

Each request is then potentially resolved with cached location information in case such information is not stale and if it has acceptable accuracy attribute for addressing the request. Similarly, each request is potentially resolved by mapping of location information types. For example, if location information of one type is requested, while a cached information of another type exists and is not stale, the mapping of one type to another can be performed, hence the request can be

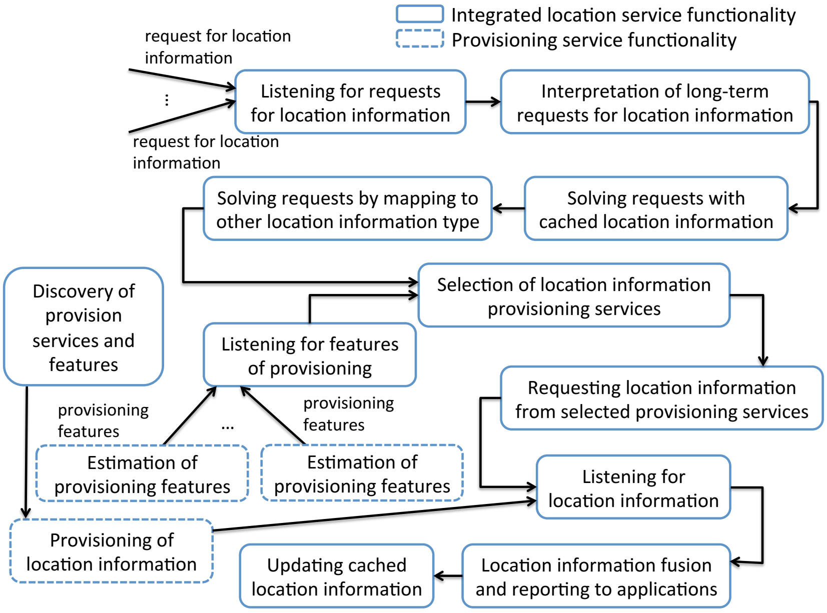

Integrated location service functionality
Provisioning service functionality
request for location information
⋮
request for location information
Listening for requests for location information
Interpretation of long-term requests for location information
Solving requests by mapping to other location information type
Solving requests with cached location information
Discovery of provision services and features
Selection of location information provisioning services
Listening for features of provisioning
provisioning features
...
provisioning features
Requesting location information from selected provisioning services
Estimation of provisioning features
Estimation of provisioning features
Listening for location information
Provisioning of location information
Updating cached location information
Location information fusion and reporting to applications

Figure 5.8: Detailed overview of functional components of the Standardized Localization Service

resolved. All requests resolved by either cached or mapped location information are immediately reported back to the applications.

All unresolved requests are used as an input in an algorithm for selection of location information provisioning services. Furthermore, provisioning features of available location information provisioning services are also used as an input to the algorithm. These features of provisioning are obtained through the discovery of provisioning services and their features, which is initiated by the integrated location service. Upon request, all available provisioning services estimate and report their provisioning features to the integrated location service. Selection of provisioning services to be invoked can be based on the two previously proposed algorithms.

Location information from different provisioning services is received at a listener for location information. Upon receiving each location information, the integrated location service decides, based on the requirements from each application, if this information should be reported back to an application. The reported location information is also used for updating cached information.

5.3.2 Implementation of the Standardized Localization Service

Implementation of the Integrated Location Service

The integrated location service is, in its prototypical implementation, realized as a multi-threaded service, as shown in Figure 5.9. One thread supports listening for requests for location information from applications and for provisioning features from provisioning services, and notifies other threads if a request or a provisioning feature was received. Moreover, the same thread performs the discovery of available location information provisioning services by reading from a register service log and discovery of their provisioning features by writing a discovery request in a proper discovery log. The discovery duration for the provisioning services that are "in vicinity" to a mobile device is fairly small, while this duration increases with the increase in the distance between the mobile device and a particular provisioning service. By proper timestamping of the requests and responses for discovery and provisioning features, we leverage this property for distinguishing provisioning service surrounding a mobile device from the ones that cannot provide location information of the device in a given environment.

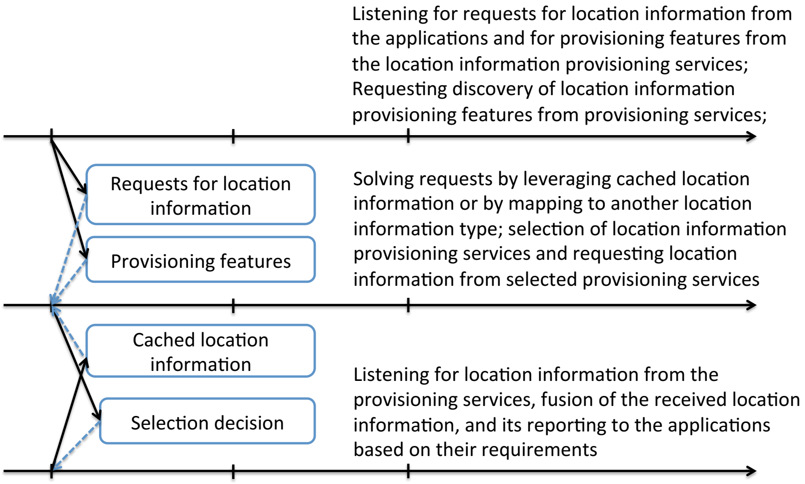

Listening for requests for location information from the applications and for provisioning features from the location information provisioning services; Requesting discovery of location information provisioning features from provisioning services;
Requests for location information
Provisioning features
Solving requests by leveraging cached location information or by mapping to another location information type; selection of location information provisioning services and requesting location information from selected provisioning services
Cached location information
Selection decision
Listening for location information from the provisioning services, fusion of the received location information, and its reporting to the applications based on their requirements

Figure 5.9: Multi-threaded realization of the integrated location service

The second thread aims at addressing newly arrived requests for location information by leveraging cached location information or by performing mapping between different location information types. For leveraging cached location information the second thread requires the parameter specifying cached location information, which is generated in the third thread discussed below. In the current implementation, we use a very simple strategy for determining if cached location information is considered stale, i.e. it is stale if it is older than 0.5 s. With the same intuition as for the previously discussed time-bucket duration, the staleness metric of 0.5 s has been selected because it is two times smaller than the update period of 1 s required for users' tracking scenarios. The second thread also handles the selection of location information provisioning services to be invoked for provisioning. The decision is based on the provisioning features of the available provisioning services and requests from the applications.

The third thread handles listening for location information from location information provisioning services. The same thread also handles fusion of location information and reports location information to applications based on their requirements. To operate properly, the thread requires the selection decision about the provisioning services that have been invoked. The selection decision parameter that is exchanged between these threads is essentially a mapping between each request from the application and the provisioning services that have been invoked for addressing that request. This parameter is required so that the thread can decide if the location information received by a certain provisioning service should be immediately reported to an application or the location information from other provisioning service(s) should still be received, this information should be fused, and then the final location information should be reported to an application. The same thread deals with updating the cached information with the newly arrived location information and reporting the updated cached information to the second thread.

GDP-based Implementation of Distributed Messaging

For the prototype implementation of the distributed messaging in the SLSR we have selected the Global Data Plane (GDP) [210]. The GDP is designed as a universal middleware data distribution platform, offering prominent types of data access. The GDP natively supports a single-writer log-based messaging, while also providing a Representational State Transfer (REST) and publish/subscribe interfaces for accessing data. Logs are encrypted and only applications with proper encryption keys can access their data, which ensures data privacy and security.

Distributed single-writer log is a messaging pattern based on a partitioned and replicated commit log service. The GDP also supports the discovery of such logs and granting reading access to a particular log. A log is named by an opaque 256-bit number and consists of a series of records, each record consisting of a record number, a timestamp, and data. The mapping of the distributed messaging between the components of the SLSR and the GDP is performed by modeling each interaction primitive by a log in which one entity is allowed to write new data. Other entities are allowed to read and receive notification of the data being written in the log. A favorable feature of the GDP-based distributed messaging is the support for caching and historical information storage and retrieval. This feature is used for the implementation of the module for resolving requests for location information by leveraging cached location information. The GDP provides the possibility of addressing services of the same type at once, which is leveraged in the implementation of the module for the discovery of provisioning services and their features.

Interaction between the integrated location service and each location-based applications is implemented as a set of GDP logs. Distributed messaging supported by the GDP allows the application to access location information provided by the integrated location service even in case such application is not deployed on a mobile device. The application is envisioned to be the only writer to its log for requesting location, with the entry data being the parameters defining the desired features of location information (location type (e.g. point or room-level location estimate), accuracy, latency, once/periodically/on events, duration). The integrated location service is subscribed to that log and upon request reports location information by writing it to a log for reporting location information to which the application is subscribed. The registration of an application with the integrated location service is therefore a matter of creating a set of GDP logs and registering them with the integrated location service.

The registration of a provisioning service is performed by granting the integrated location service a read permission to the provisioning service's "reporting location log", where the first entry to that log is the address of the provisioning service (256 bit identifier) and the location information type(s) that can be provided. The integrated location service is the only writer to a discovery log to which all provisioning services are subscribed. Provisioning services subscribed to that log report their availability and provisioning features by writing into a log for reporting provisioning features, where an entry defines an offering (i.e. accuracy, latency, and power consumption). The procedure continues in the same way as for the interaction between the location-based application and the integrated location service.

As mentioned, the GDP provides support for REST and publish/subscribe-based messaging.

Due to the lack of space, in the following we only discuss how the distributed messaging in the SLSR can be supported by the REST messaging paradigm. The REST messaging pattern's key abstraction is a resource, where the resource is any named information. The REST pattern bases messaging on four operations: creating, reading, updating, and deleting a resource. location-based applications can request location information from the integrated location service using a request for location information, with the payload being the desired features of this information. The integrated location service reports this information as the response to the request, where the response consists of the requested location information. In case the consuming application requires periodic or on an event reporting of location information, an observe flag should be set in the request. When the observe flag is set by the application, the integrated location service will continue to reply after the initial resource has been reported, therefore the asynchronous nature of responses can be achieved. Registration of a location information provisioning service with the integrated location service is done by issuing a register service request to the integrated location service, with the payload being the address of the provisioning service and location information type that can be provided. During the discovery of provisioning services, the integrated location service issues a service discovery request to each registered and suitable provisioning service. As the response, the invoked provisioning services report their provisioning features. The integrated location service requests location information provisioning from a provisioning service in the same way as an application requests location information from the integrated location service.

5.3.3 Instantiation of the Standardized Localization Service

The SLSR can be instantiated in any environment served by one or more provisioning services that are registered with the integrated location service and compliant with the previously discussed PSI interface. We instantiated the developed prototype in the previously discussed testbed environment for experimental evaluation of RF-based indoor localization solutions. We used a Lenovo Think-Pad laptop as a mobile device in the instantiation. The components of the SLSR were realized as Python 2.7-based multi-threaded daemon processes. For the instantiation of location information provisioning services we selected a set of WiFi RSSI fingerprinting-based localization solutions.

As discussed in Chapter 3 of the thesis, WiFi RSSI fingerprinting-based localization solutions require a training phase in which a survey of WiFi environment is performed at predefined locations, i.e. at each location a set of scans for WiFi beacon packets' RSSI measurements is taken and stored in the training database [144]. A mobile device whose location is to be determined then generates its own scan of the WiFi environment and sends it to a fingerprinting server. At the server, this scan is compared with the scans from the database and the most similar one is reported as the estimated location of the mobile device. Additionally, fingerprinting can also include a post-processing procedure, e.g. kNN, where instead of reporting just one training location as the estimated location, a set of closest matches is merged and reported as the location estimate, which benefits accuracy and decreases variability of localization errors.

An increase in the density of training locations benefits the accuracy of fingerprinting. Similarly, different procedures for calculating similarities between training fingerprints and the user's generated one yield different accuracies and latencies of provisioning, as discussed in [144]. For generating multiple instances of location information provisioning services with different accuracy and latency features, we therefore used different densities of training locations and different algorithms for calculating similarities between fingerprints.

Different densities of training locations are depicted in Figure 5.10. The first set of provisioning services uses a “small” number of training points, namely one training point per office, with training points as indicated with blue dots in the figure. The second set of provisioning services uses a “medium” number of training points, i.e. two or four training points per office, depending on an office size, as indicated with red dots in the figure. The third set of provisioning services uses a “large” number of training points, i.e. the combination of the small and medium set of training points. Black squares indicate the locations of WiFi APs used as transmitters of beacon packets for all instantiated provisioning services.

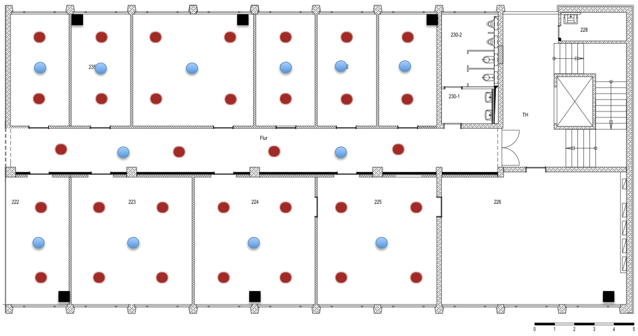


Figure 5.10: Locations of training points and WiFi APs

Same as for the previous contributions in the thesis, here we also used the same two algorithms for calculating similarities between fingerprints. Their overviews are given in Chapter 2.

As the result, we instantiated six provisioning services out of the combination of training dataset densities and algorithms for calculating similarities between fingerprints. These provisioning services were able to generate and provide local or semantic type of location information of a mobile device. Semantic type of location information is in this case a office-level estimate of a mobile device’s location. In addition, we instantiated two provisioning services that were able to provide only semantic location information. This mimics a usual positioning service based on iBeacons that is able to provide only coarse-grained room-level accuracy of localization with lower power consumption in comparison to WiFi-based fingerprinting. Due to the shortage of iBeacons, we mimicked their operation by leveraging fingerprinting, more precisely by using a small training dataset and the two algorithms for calculating similarities between fingerprints, and by limiting the reporting of location information to only a semantic office-level type of provisioning. Hence, in addition to six provisioning services that were able to provide location information of a local or semantic type, we had additional two services that were able to provide only semantic location information.

5.3.4 Evaluation Setup, Scenarios, and Procedure

In the first step of the evaluation, we aim at showing that the composition of provisioning services, as enabled by the SLSR, benefits the accuracy of location information provisioning. In the next step, we aim at demonstrating the relevance of different functional properties specified in the SLSR by showing that each of them benefits the overall performance of the SLSR.

As discussed previously, TWIST testbed is an office environment in its usual operation, while also offering a fully automated testbed infrastructure for the evaluation of indoor localization solutions [136]. The testbed features a mobility platform capable of autonomously carrying a device to be localized to different evaluation locations. The same mobility platform can be used for moving a device to be localized over a predefined trajectory, which is suitable for the evaluation of tracking solutions, in contrast to evaluation at discrete points only. As a reminder, for its positioning in space the autonomous mobility platform matches the highly-accurate floor plan of the testbed environment with depth information provided by a Kinect camera, achieving less than 15 cm in average localization errors in the testbed space. This is an order of magnitude lower localization error than what the current state of the art indoor off-the-shelf indoor localization solutions can achieve [61], hence the location information from the mobility platform can be used as the ground-truth location of a mobile device for the evaluation purposes. The mobility platform with its speed of 0.5 m/s is somewhat slower than a usual walking speed of a person and the trajectory followed by the platform is to a certain level unnatural for a person. However, the platform reduced the disturbances due to a person's body, such as shadowing and subtle shaking, thus increases the objectiveness of the evaluation results. Furthermore, the mobility platform enables extraction of reference locations of a mobile device that are unavoidable for the evaluation. This cannot be achieved by a test-person, especially in tracking scenarios, and we therefore leveraged location information provided by the mobility platform as the reference in our evaluation.

The testbed infrastructure also supports monitoring of RF interference context that can considerably degrade the performance of RF-based localization solutions, thus reduce the objectiveness of evaluation (as demonstrated previously). The evaluation experiments have therefore been performed on weekends with minimized and controlled influence of RF interferences. The raw data from the experiments, i.e. the WiFi beacon packets' RSSI scans at different locations in the testbed, were collected and stored in the previously presented web-platform for streamlined experimental evaluation of localization solutions [137]. This enabled streamlined evaluation of the SLSR by inputing the collected raw data, calculating location information, comparing them with ground truths, and calculating performance metrics. The web-platform enhanced the objectiveness and comparability of our results, for example when typifying the benefits of different functional components of the SLSR. Two repetitions of all experiments were performed to make an argument about the stability of our performance results.

The evaluation was performed in a localization and two tracking scenarios. For the localization scenario a set of 20 discrete evaluation points was defined, as with red dots indicated in Figure 5.11. As the performance metrics we specified point and room-level accuracy. The point accuracy is defined as the Euclidean distance between ground-truth and estimated location, while the room-level accuracy characterizes the correctness of estimated room. Clearly, increasing the number of provisioning services also benefits the availability and robustness of location information provisioning by increasing the redundancy of information, but due to the lack of space these influences have not been explicitly characterized.

For the evaluation in the tracking scenarios, we specified the trajectory of movement of a mobile device, as shown in Figure 5.11. During the course of movement along the trajectory, we constantly collected and stored the raw RSSI scans. We then leveraged the collected scans as input for generating location estimates, while changing the structure and parametrization of the SLSR. The aim of the first tracking scenario was to demonstrate that an increase in the number of location information provisioning services benefits localization accuracy and to characterize the nature of this benefit. In this scenario, we used each scan of WiFi environment taken along the trajectory for estimating the location of the mobile device.

In the second tracking scenario, we aimed at showing the benefits of different functional components of the SLSR: using the long-term requests interpretation in contrast to "pushing the intelligence" of requesting location information to the applications; using cached location information and mapping between location information types for potentially resolving requests for location information; using different algorithms for selection of provisioning services; using different estimators of provisioning features. Metrics used for performance evaluation in the second tracking scenario include satisfaction of accuracy, latency, and both requirements, and the total power consumption of the SLSR. Requirement satisfaction is a binary metrics defined as the capability of the SLSR to provide location information of with a better characteristic than the requirement specified in a given request.

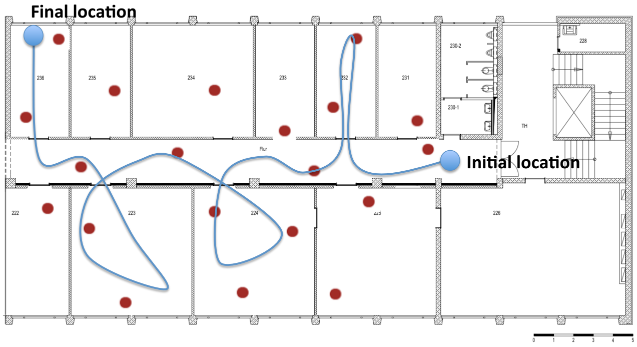


Figure 5.11: Discrete points for localization and movement trajectory for tracking performance evaluation

Requests for location information have been generated according to, what we believe, is one realistic scenario for location information requirements. These requests are summarized in Table 5.10. The first request represents a usual person' tracking scenario with conventional requirements, i.e. 1 m in average localization error provided periodically with the period of 1 s during the course of the trajectory and with the location information type being local location information [211]. The second requirement resembles a light switching scenario, in which based on the presence of a person in a room the light is either turned on or off [212]. The location information type is semantic, i.e. room-based information provisioning required periodically and frequently with the period of 1 s. The third request mimics a smart thermostat scenario, in which the heating in a room in turned on based on the presence of a person in the room [213]. Clearly, this request requires a periodic semantic provisioning of location information with a high requirement for accuracy and low requirement for latency of provisioning (5 s). The fourth scenario mimics a self-configuring printer scenario in which, if the mobile device is in the same room as the printer, a fine-grained periodic provisioning is needed with the accuracy of 1.5 m and latency of 3 s, while in all other rooms only room-level information is needed upon a change of location information. The fifth scenario resembles a High-Definition (HD) uncompressed video steaming using 60 GHz narrow beams [156], in which these narrow beams have to "track" the user as it moves in the environment. In this scenario, fast (1 s) and accurate (1 m) periodic provisioning of location information has to be available for successful tracking of the user [214]. The sixth request mimics a location-aided horizontal WiFi handover mechanism, in which a fine-grained tracking is required in handover areas (1.5 m/1 s), while a course-grained on an event provisioning of location information is needed in non-handover areas [215]. The seventh request represents the D2D link establishment scenario, in which if two devices are in the vicinity a D2D communicational link is established between them [216]. This scenario requires a fairly fast provisioning of location information (1.5 s) when it changes from the previous location, with low requirement on the accuracy of location information (2 m).

Table 5.10: Requests for location information

#	Accuracy	Latency	Provisioning type	Location type
1	1.5 m	2.0 s	periodic	local
2	100 %	1.5 s	periodic	semantic
3	100 %	5.0 s	periodic	semantic
4	100 %/1.5 m	3.0 s	periodic / on event (1 m)	semantic / local
5	1.0 m	1.0 s	periodic	local
6	1.5 m	1.0 s	periodic / on event (3 m)	local
7	2.0 m	1.5 s	on event (1 m)	local

5.3.5 Evaluation Results

In the first step of the evaluation, we aimed at benchmarking the static performance of each location information provisioning service instantiated under "the umbrella" of the SLSR. The results are given in a regular box-plot fashion in Figure 5.12. As visible from the figure, with an increase in the density of used training dataset (from "small" to "medium" and "large"), the average accuracy of location information provisioning increases as well. Furthermore, the algorithm PH Distance of RSSI Quantiles shows marginally better accuracy of location information provisioning. This is due to higher complexity of its pattern matching procedure and due to the fact that RSSI quantiles were used as fingerprints, which provides more information than the usage of average RSSI values in the other algorithm [144]. Both conclusions are consistent for both repetitions of experiments, making an arguments about the reliability of our results.

Similar conclusions for the point and room-level accuracy can be drawn from Table 5.11 which provides summary results merged over two repetitions of the experiment. Furthermore, as shown in the table, with an increase in the number of training points the latency of reporting location estimates also increases, which is consistent with the well-known trade-offs of WiFi fingerprinting [35]. Also, a more complex algorithm has higher latency of location information provisioning. These summarized performance results are used as static provisioning features of each provisioning service in the second tracking scenario. Due to the inability to directly measure the power consumption of location information provisioning, we leverage a simple characterization of a power profile of each provisioning service, i.e. all WiFi-based provisioning services are given a constant power profile of 2. The services that mimic iBeacon-based location information provisioning were given a two times smaller power profile. This is consistent with reported comparisons of the WiFi vs. Bluetooth power consumptions, e.g. [217].

The results for the first tracking scenario are given in Figure 5.13. Each group of box-plots depicts localization errors for two repetitions of an experiment. As visible from the figure, the results demonstrate a high level of stability across the two repetitions, which makes an argument about the reliability of our findings.

Table 5.11: Summary performance results of the instantiated provisioning services

Provisioning service	Accuracy	Room accuracy	Latency	Power consumption
Euclidean small	3.14 m	52.5 %	0.66 s	2.0
Quantile small	3.05 m	55.0 %	0.72 s	2.0
Euclidean medium	2.41 m	62.5 %	0.86 s	2.0
Quantile medium	2.22 m	65.0 %	0.91 s	2.0
Euclidean large	1.96 m	75.0 %	1.05 s	2.0
Euclidean semantic	/	52.5 %	0.66 s	1.0
Quantile semantic	/	55.0 %	0.72 s	1.0

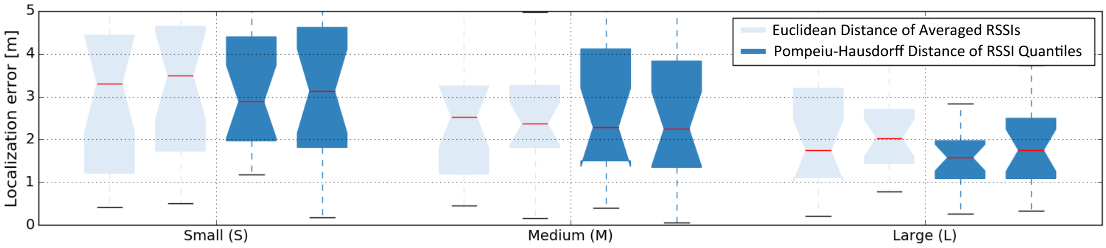


Figure 5.12: Localization errors of the instantiated provisioning services

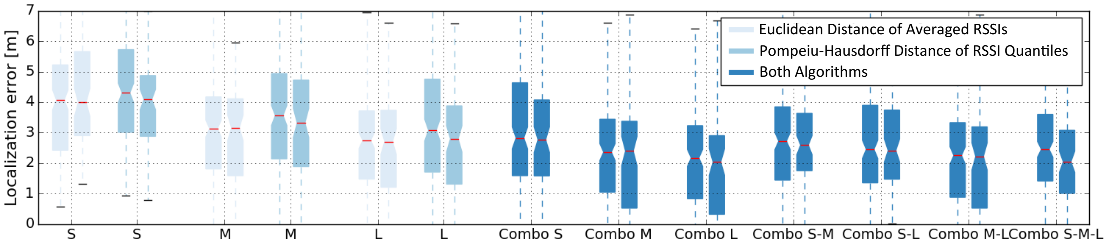


Figure 5.13: Influence of fusion of location estimates on the final accuracy of provisioning

The first six groups of box-plots depict the results achieved by the two algorithms with different densities of a training dataset, namely small (S), medium (M), and large (L). As visible from the figure, there is a substantial increase in average localization errors in comparison to the static scenario. The reason for that is the movement of the device to be localized in this scenario, which influences the quality of the obtained WiFi RSSI measurements. Furthermore, in this scenario the less complex algorithm, i.e. the Euclidean Distance of Averaged RSSIs, achieved marginally better performance. This is due to the fact that in moving scenarios the usage of multiple RSSI scans for estimating location does not benefit, but actually reduces the localization accuracy, since the older scans are essentially taken at some previous locations along the movement trajectory.

The following box-plots depict the localization accuracies in case of fusion of location estimates provided by the two algorithms. For example, the "Combo S" group of box-plots presents the results of fusion of location estimates provided by the two algorithms, both of them using the small training dataset. Similarly, the "Combo S-M" depicts the localization accuracies in case the estimates provided by four instances of provisioning services, with instances being the two algorithms that use small and medium datasets. The figure shows that the fusion of location estimates is beneficial, but only if the fused instances have comparable localization errors. For example, the "Combo S" localization errors are generally smaller than the localization errors of any instance of provisioning services that leverage the small training dataset. The same conclusion holds for instances that use medium and large training datasets, for both algorithms, and for both repetitions of the experiment. However, in case instances with different localization accuracies are fused, the resulting localization errors are generally comparable or worse than the ones obtained by fusion of only instances with the highest accuracies. For example, the accuracies labeled with "Combo S-M" have larger localization errors than the ones labeled with "Combo M". This means that the fusion of location information should be performed only if the fused instances have comparable accuracies, which is consistent with Equation (5.4).

In the second tracking scenario, we generated a realistic load for the integrated location service by using the previously specified requests for location information. We captured the satisfaction of requirements using the following procedure. If the localization error of location information reported to an application is smaller than the localization error (i.e. accuracy) specified in the corresponding request for location information, then the requirement is considered satisfied. Similarly, if the latency of provisioning of location information is smaller than the one specified by the request, the latency requirement is considered satisfied. Both requirements of a request are satisfied if both accuracy and latency exceed the ones specified in the request. The consumed power metric specifies the total power consumed by all previsioning services during the course of a movement along the trajectory, with the power profile of each provisioning service being as specified in Table 5.11.

Table 5.12 summarizes the obtained results for the two algorithms for selection of location information provisioning services. The total number of requirements differs slightly for the two repetitions of the experiment because of small differences in movement of a mobility platform in different repetitions. The "Basic" type of the SLSR includes only basic functionalities of the integrated location service, namely reception of requests for location information, discovery of provisioning services and their features, invocation of provisioning services, reception of location estimates from the invoked provisioning services, their fusion, and their reporting to the applications. The "Caching & mapping" type additionally includes caching and mapping functionalities. Caching functionality enables addressing requests for location information with cached location information if the features of such information exceed the requirements of the request and in case cached location information is not stale.

Mapping functionality supports mapping of local to room-level location information type and vice-versa, if appropriate type of location information exists in the cache. The "Long-term interpretation" type of the SLSR includes the long-term requests interpretation. In other words, previous types assumed that the application will issue a request for location each time this information is needed. The "Long-term interpretation" type assumes that requests for periodic or on an event provisioning for a certain duration can be specified in one request. This request is received by the long-term requests interpretation entity, which does a request translation to multiple requests at appropriate time instances. The listed types of SLSR leverage static features of provisioning of different provisioning services, as specified in Table 5.11. The "Dynamic" type uses a dynamic specification of the latency and accuracy features of provision of different provisioning services. This is possible since we know the accuracy and latency features of provisioning of all provisioning services and for all WiFi scans collected along a trajectory, as discussed in the previous tracking scenario.

The results in Table 5.12 show that the PTSA, which does a more global over a time-bucket optimization of selection of provisioning service consumes roughly 25 % less power in total. However, for the "Basic" type of SLSR the number of satisfied accuracy and both requirements of the PTSA is roughly 25 % smaller that the same satisfactions of the PRSA. Both algorithms utilize the same strategy for meeting the latency requirements from applications, hence the latency satisfaction is the same for both of them. These findings are consistent with the ones derived previously by simulation the algorithms' performance. Introducing the mapping and caching capabilities dramatically reduces the total consumed power, while it also highly benefits both accuracy and latency requirements satisfaction. While it is straightforward that caching and mapping functionalities benefit latency requirement satisfaction, the reason for benefiting accuracy satisfaction is due to the fact that, if an estimate of location is provided faster, due to the moving nature of the device, also the accuracy of the estimated location will be higher. In other words, improvements in latency in tracking scenarios also benefit the accuracy of the SLSR. Pushing "the intelligence" of requests for location information to the integrated location service, as done in the "Long-term interpretation" type, further benefits the latency of location information provisioning, since it removes the latency overhead of sending a request by an application to the integrated location service. However, the benefit in terms of accuracy or/and latency satisfaction is relatively small, since the duration of issuing a request in the prototype implementation is also very small. However, presumably for the more distributed applications the request time will be larger, which will result in higher benefits of using the long-term requests interpretation. Finally, the usage of dynamic prediction of provisioning features benefits the accuracy and both requirements satisfaction because a more accurate selection of provisioning services can be performed. However, these dynamic features generally suggest smaller accuracy of provisioning of provisioning than the accuracy features obtained by performing a static evaluation, as demonstrated by the results of the first tracking scenario and the results of static evaluation. Therefore, in many cases more provisioning services are invoked to satisfy the requirements from the applications. Thus, the power consumption of the "Dynamic" type is considerably increased in comparison to the "Caching & mapping" and "Long-term interpretation" types.

Table 5.12: Requirements satisfaction of different instances of the Standardized Localization Service

Type	Total number of requirements		Accuracy satisfaction		Latency satisfaction		Both requirements satisfaction		Consumed power	
Per-request satisfaction algorithm										
Basic	659	665	141	154	388	391	113	121	2021	2037
Caching & mapping	659	665	352	361	504	511	303	312	1201	1225
Long-term interpretation	659	665	361	371	531	542	313	322	1167	1171
Dynamic	659	665	451	471	528	537	347	354	1387	1376
Per-time-bucket satisfaction algorithm										
Basic	659	665	98	104	388	391	68	70	1510	1536
Caching & mapping	659	665	331	338	504	511	281	299	854	831
Long-term interpretation	659	665	341	351	531	542	294	308	833	812
Dynamic	659	665	433	452	528	537	339	341	1041	1032

5.4 Conclusions

For achieving accurate, seamless, and robust provisioning of location information, we firstly defined a standardized modular localization service architecture. The proposed localization service architecture consists of location-based applications, an integrated location service enabling fusion and handover of the integrated location information provisioning services, and resources for generating location information. Secondly, we standardized the interaction requirements between the distributed modules of the proposed architecture.

We further proposed two algorithms for selection of location information provisioning services. The selection decision is based on the accuracy and latency requirements specified by the location-based applications and the location information provisioning features of the available provisioning services. The decision is subject to per-request power minimization for the Per-Request Satisfaction Algorithm (PRSA), and a time bucket-wise power minimization for the Per-Time-bucket Satisfaction Algorithm (PTSA). We demonstrated analytically and experimentally roughly 25% better performance of the PTSA in terms of power consumption with a trade-off in accuracy satisfaction. In general terms, we showed that the selection of provisioning services influences the final performance of localization services, even for relatively small numbers of requirements from the applications and available provisioning services during a time bucket. Since we believe these numbers are expected to grow, sophisticated selection algorithms will play an important role in optimizing the performance of localization services. We believe that this contribution makes a step in the direction of addressing this issue. Nevertheless, the presented results should be viewed only as a rough indication of the importance of optimizing the selection of provisioning services. This is because we abstracted out and treated at a very high level the aspect of power consumption of individual provisioning services. An experimental evaluation of power consumption of individual provisioning services is needed before making any stronger claims about the performance of different algorithms for selection of provisioning services.

Both power consumption and accuracy are important features of location information provisioning and it is hard to prioritize any one of them globally, since different users have different preferences. We therefore believe that the ultimate selection of location information provisioning services should have two levels. On the "strategic" level, the localization service should outsource to the user high-level optimization preferences. On the "operational" level, i.e. within the selected high-level strategy, different algorithms can be used for achieving these goals, and the two algorithms we proposed are just examples that primary focus on latency requirements satisfaction. Therefore, the ultimate choice on which algorithm to use will depend on the user's preferences, hence the possibility for such a selection is envisioned in the localization service architecture (the *Specify policy* interaction primitive of the SLI).

Furthermore, we contributed with a prototypical implementation, instantiation, and evaluation of the SLSR. Our results demonstrated the benefits of fusion of location information provided by different provisioning services. Furthermore, our results showed a considerable increase in localization errors in tracking scenario in comparison to static localization scenarios. Additionally, our results experimentally demonstrated the influence of different algorithms for selection of location information provisioning services to the overall performance of the standardized localization service. We further showed that even the usage of fairly trivial methods for caching of location information and mapping of location information types highly benefits the performance of the standardized localization service. Finally, having the global view on the provisioning features, in contrast to leveraging well-established static performance benchmarks, resulted in more optimal selection of provisioning services, which in turn benefits the accuracy and latency of location information provisioning.

6 Using Location Information for Improving Wireless Networks Operation

Assuming an accurate, reliable, seamless, and performance-wise foreseeable location information of mobile and nomadic devices is available, this information can be used for improving the performance of wireless networks. This potential has been conceptually discussed in the introductory part of this thesis (Section 1.1). However, the vast majority of discussions on potential location-based improvements of the performance of wireless networks assumes a perfectly accurate location information, which is an unrealistic assumption in practice.

In this chapter of the thesis, we demonstrate that realistic and practically achievable location information can be used for improving the performance of wireless networks. We do that on two emerging concepts in next generation wireless networks, i.e. Device-to-Device (D2D) communication and mobile relaying. Parts of this chapter have been published previously as follows. A mechanism for location-based D2D link establishment has been discussed in [218]. A location-based mechanism for positioning of a mobile relay has been proposed in [219].

6.1 Location-based Decision-making Mechanism for D2D Link Establishment

A massive growth in the amount of wireless traffic is expected in the near future [220]. In comparison to leveraging the available infrastructure (i.e. intermediate nodes) for communication between end-devices, D2D communication represents a potential for reducing the amount of network traffic and improving the latency and energy efficiency of communication [221].

Current D2D link establishment mechanisms usually leverage RSS for deriving if the two devices that want to establish a communication link are in proximity. Mechanisms for RSS-based D2D link establishment assume that communication links between devices can be established if RSS values from probe packets transmitted by one device and observed by the other are above a certain threshold [125, 126]. One drawback of RSS-based D2D link establishment mechanisms comes from the fact that RSS is a highly fluctuating signal feature [222], therefore such mechanisms can be unreliable. Such mechanisms also require constant active probing, i.e. transmission of probe packets by one device and listening for probe packets by the other. This probing has to be done repeatedly and even if the devices are very far from one another. During probing the devices are not able to communicate with the infrastructure, which represents a loss of communication potential in case D2D links cannot be established. Repeated transmissions of probe packets also negatively impact the power consumption of the devices.

At the same time, we are witnessing an increase in location-awareness in wireless networks, as discussed in the previous chapters of this thesis, as well as in e.g. [61, 11]. We believe that the usage of location information of devices that wish to establish a D2D link and some knowledge about the propagation characteristics in a wireless environment would result in a better solution for a decision-making mechanism for D2D link establishment. In contrast to RSS-based mechanisms, such a location-based D2D link establishment mechanism would not need repetitive active probing, but the probing would have to be performed only when the proposed mechanism yields that a high chance of establishing a D2D link exists.

When leveraging location information for D2D link establishment, one needs to keep in mind that location information of mobile devices are generally not ideal, i.e. localization errors are to be expected. In addition, wireless signals intrinsically contain randomness, resulting from both large-scale (shadowing) and small-scale fading (multi-path fading). Hence, even if the distance between devices is small enough to establish a D2D link, this does not necessarily imply that the devices will be able to communicate. Therefore, the imperfections in both propagation and location information and have to be considered while designing a reliable location-based D2D link establishment mechanism.

With those issues issues in mind, we propose a decision-making mechanism for location-based D2D link establishment. The proposed mechanism takes into account inaccuracies caused by randomness in wireless propagation and errors in location information of mobile devices. We firstly show that that the Euclidean distance between the devices, if calculated using their erroneous location information, is a Rice-distributed random variable, assuming that the localization errors per-axis are zero-mean Gaussian distributions. We leverage the derived distance in a multi-wall propagation model for modeling path-loss and large scale fading, while we model small-scale fading based on Rayleigh distribution. We derive a closed form equation for the probability of establishing a D2D link between two devices.

We evaluate the proposed mechanism by comparing the link establishment decisions yielded by the mechanism with the measured SNR between the devices in a complex office-like environment, i.e. in the previously discussed testbed for the evaluation of RF-based indoor localization solutions. We do that for multiple SNR thresholds used as inputs to the mechanism, for different locations of the devices, and for various distances between the devices. We show that by using a 5 dB SNR safety margin, i.e. by setting the SNR threshold 5 dB above the one nominally needed for communication, the mechanism can achieve the rate of false positive link establishment decisions of less than 2%. We further show a relatively small increase in the percentage of false negative link establishment decisions as a function of increasing localization errors.

6.1.1 D2D Link Establishment Scenario

We envision a D2D link establishment scenario as depicted in Figure 6.1. Mobile devices that wish to communicate express their intentions to the infrastructure to which they are connected. The infrastructure is aware of the mobile devices' physical locations, which are to a certain level erroneous. The aim of the mechanism is to calculate the probability that the SNR between two devices is higher than a desired SNR threshold. In case the mechanism yields that there is a high probability of the SNR between two devices being higher than a desired threshold, the infrastructure notifies both devices that a D2D communication link can likely be established.

We assume that the location of each mobile device is specified with its x and y coordinates in a 2D coordinate system. We further assume that all per-coordinate localization errors have the same standard deviation σ. This assumption is justified since at these relatively small distances between devices (i.e. in the same environment) location information of the devices are usually provided by the same localization service.

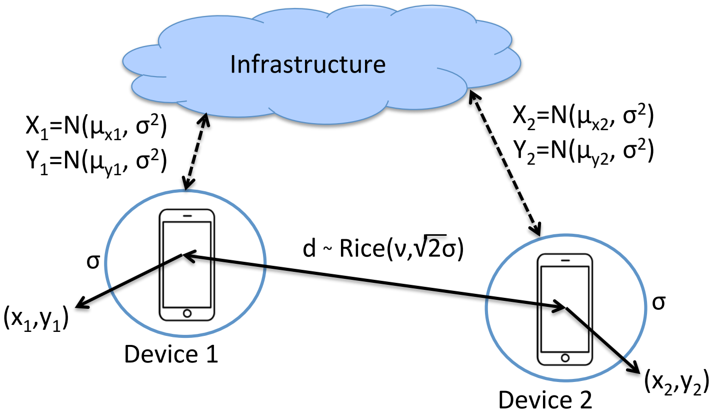

Infrastructure
X_1=N(μ_{x1}, σ^2)
Y_1=N(μ_{y1}, σ^2)
X_2=N(μ_{x2}, σ^2)
Y_2=N(μ_{y2}, σ^2)
d ~ Rice(ν,√2σ)
σ
(x_1,y_1)
Device 1
σ
(x_2,y_2)
Device 2

Figure 6.1: Overview of the envisioned scenario

6.1.2 Location-based Mechanism for Establishing D2D Links

Location information of each device is erroneous with error per-coordinate being a zero-mean Gaussian distributed random variable. Modeling per-coordinate localization errors with Gaussian distribution is a usual and well established procedure in the literature, some examples being [223, 151, 147].

Euclidean Distance Between Devices

Suppose that the location information of the devices 1 and 2 are provided by the localization service as (μ_{x_1}, μ_{y_1}) and (μ_{x_2}, μ_{y_2}). Furthermore, the correct locations of the devices are random variables (X_1, Y_1) and (X_2, Y_2). We assume that each coordinate of the location information provided by the localization service is a normally distributed random variable specified by its mean value μ and standard deviation σ:

$$X_1 \sim \mathcal{N}(\mu_{x_1}, \sigma^2), \quad Y_1 \sim \mathcal{N}(\mu_{y_1}, \sigma^2) \tag{6.1}$$

$$X_2 \sim \mathcal{N}(\mu_{x_2}, \sigma^2), \quad Y_2 \sim \mathcal{N}(\mu_{y_2}, \sigma^2). \tag{6.2}$$

Euclidean distance between the devices is therefore estimated as:

$$d = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}. \tag{6.3}$$

Proposition 6.1.1. *If the true locations of two devices are modeled according to Equations 6.1 and 6.2, the Euclidean distance between the devices is a random variable distributed according to Rice distribution $d \sim \mathrm{Rice}(\nu, \sqrt{2}\sigma)$ with the CDF given as follows:*

$$F(\delta) = \mathbb{P}(d \leq \delta) = 1 - Q_1\left(\frac{\nu}{\sqrt{2}\sigma}, \frac{\delta}{\sqrt{2}\sigma}\right) \tag{6.4}$$

where:

$$\nu = \sqrt{(\mu_{x_1} - \mu_{x_2})^2 + (\mu_{y_1} - \mu_{y_2})^2}. \tag{6.5}$$

Furthermore, Q_1 is the Marcum 1 function given by [224]:

$$Q_1(a,b) = \int_b^{\infty} x \, \exp\left(-\frac{x^2+a^2}{2}\right) I_0(ax) \, \mathrm{d}x \tag{6.6}$$

for $a, b \geqslant 0$, where I_0 is a well-known modified Bessel function of the first kind.

Proof. The proof is given in Appendix B. □

Propagation Model

In the propagation model, path-loss, large-scale fading (shadowing), and small-scale fading (multipath fading) are considered. Small-scale fading in a wireless channel is characterized with a Rayleigh distribution because we are assuming harsh and complex indoor environments with people carrying the devices (hence body shadowing), thus the probability of a Line-of-Sight (LoS) component arriving at the receiver is very small. For modeling path loss and large-scale fading we use the multi-wall model [225]. The applicability of the model for modeling indoor environments has been demonstrated in [171]. In the model, the signal attenuation $L(d)$ in dB, which is a result of path-loss and shadowing, is given by:

$$L(d) = l_c + 10\gamma \log(d) + M_w. \tag{6.7}$$

l_c is a constant value related to the model fitting procedure discussed below. The attenuation $L(d)$ is dependent on the distance d from the transmitting device, path-loss coefficient γ of the environment, and the total attenuation from all walls M_w in the direct path between the transmitter and the receiver. Each wall has its attenuation contribution l_w, hence for the number of walls N_w in the direct path between the transmitter and receiver total attenuation from all walls M_w is given as:

$$M_w = \sum_{i=1}^{N_w} l_w. \tag{6.8}$$

Therefore, the total attenuation in linear units, i.e. in Watts, is given by:

$$\ell(d) = 10^{\frac{L(d)}{10}} = d^{\gamma} 10^{\frac{l_c + \sum_{i=1}^{N_w} l_w}{10}} = \kappa d^{\gamma}. \tag{6.9}$$

If the small scale fading channel gain is denoted by the parameter h, the instantaneous SNR is given by:

$$\mathrm{SNR} = \frac{hP_{tx}}{N\kappa d^{\gamma}} \tag{6.10}$$

where N is the noise power. If the small scale fading is Rayleigh distributed, then the channel gain h is exponentially distributed in Watts [226]. For the rest we assume that h has exponential distribution with mean value 1. Note that the instantaneous SNR is affected by two random parameters h and d. Let $\mathrm{SNR}_{\mathrm{th}}$ represents the SNR threshold above which a D2D communicational link can be reliably established. The probability that a link can be reliably established is denoted by $F_{h,d}(\mathrm{link})$ and characterized in following proposition.

Proposition 6.1.2. *Suppose that small scale fading gain is exponentially distributed and* d *has Rice distribution, i.e.* $d \sim \mathrm{Rice}(\nu, \sqrt{2}\sigma)$*. The link probability* $F(link)$ *is given by:*

$$F_{h,d}(link) = \mathbb{E}\left[\exp\left(-\frac{\mathrm{SNR}_{\mathrm{th}}}{\overline{\mathrm{SNR}}}\right)\right] \tag{6.11}$$

where $\overline{\mathrm{SNR}}$ *is the average SNR given by* $\frac{P_{tx}}{N\kappa d^{\gamma}}$ *and the expectation is taken over* d*. Moreover if free-space path-loss exponent* γ *equals 2, we have:*

$$F_{h,d}(link) = \frac{P_{tx}\exp\left(\frac{-2\nu^2\sigma^2\kappa N\mathrm{SNR}_{\mathrm{th}}}{P_{tx}+4\sigma^2\kappa N\mathrm{SNR}_{\mathrm{th}}}\right)}{P_{tx}+4\sigma^2\kappa N\mathrm{SNR}_{\mathrm{th}}}. \tag{6.12}$$

Proof. See Appendix B. □

The above equation can be computationally complex. With further simplification, we assume that the small scale fading effect is averaged out. In this scenario, SNR is given by $\frac{P_{tx}}{N\kappa d^2}$.

Proposition 6.1.3. *Suppose that small scale fading effect is averaged out and* d *has Rice distribution, i.e.* $d \sim \mathrm{Rice}(\nu, \sqrt{2}\sigma)$*. The link probability* $\overline{F(link)}$ *is then given by:*

$$F_d(link) = 1 - Q_1\left(\frac{\nu}{\sqrt{2}\sigma}, \frac{1}{\sqrt{2}\sigma}\left(\frac{P_{tx}}{N\kappa\mathrm{SNR}_{\mathrm{th}}}\right)^{1/2}\right). \tag{6.13}$$

Proof. See Appendix B. □

Discussion

The proposed decision-making mechanism for location-based D2D link establishment leverages the previously presented multi-wall model for modeling large-scale fading and path-loss. as discussed previously, the multi-wall model assumes that an accurate floor plan of the served environment is available and can be used for determining the number of walls in the direct path between the devices that wish to establish a D2D link. The requirement for the availability of a floor-plan should not pose a challenge for the deployment of the proposed mechanism. On the contrary, the floor-plan will in most cases already be available since the environment is served by an indoor localization service, and such services usually require a floor plan of the served environment.

Furthermore, the multi-wall model requires an estimation of model parameters, i.e. the wall attenuation l_w and the constant l_c related to a model fitting procedure. The model fitting procedure is a procedure of calculating model parameters using measurements collected at different locations in a targeted environment. The used model fitting procedure is a simple least-square fitting procedure given as follows:

$$\{l_c, l_w\}_{opt} = \underset{l_c,l_w}{\mathrm{argmin}}\{\sum_{m=0}^{M-1}|P_m - (P_{tx} - L(d_m))|^2\}. \tag{6.14}$$

The procedure allows minimization of the differences between powers P_m measured in each m-th ($m = 1, 2, ..., M$) measurement location and the model estimated received power $P_{tx} - L(d_m)$, where P_{tx} denotes the transmit power at the transmitter. Obviously, the model fitting procedure requires a certain number of measurements to be collected before the model parameters could be calculated. This again should not pose a challenge for the deployment of the proposed mechanism. The usual procedure of almost effortless collection of the required measurements is crowd-sourcing in which mobile devices in a targeted environment opportunistically collect measurements, in this case RSS scans [227, 19]. These measurements are then, together with the locations where they are collected, reported to the network and can be leveraged for calculating the model parameters l_c and l_w. The locations of collected measurements can be obtained from a localization service that is deployed in the targeted environment.

6.1.3 Evaluation of the Proposed Mechanism

The idea of the evaluation procedure is to compare the D2D link establishment decisions made by the proposed mechanism with measured references in order to obtain the number of correct decisions made by the mechanism. We perform the evaluation of the proposed mechanism using WiFi as an example technology. However, it should be noted that the proposed mechanism is technology independent.

For the evaluation of the proposed mechanism we leveraged precollected measurements from 41 locations in our testbed environment, as shown in Figure 6.2. As discussed previously, the testbed infrastructure includes an autonomous mobility platform for accurate positioning of a measuring device at different measurement locations. The average per-axis localization error of the autonomous mobility platform in the testbed is 10 cm [136]. At each measurement location we collected a set of 40 WiFi scans for WiFi beacon packets. From the obtained packets we extracted RSSI values from four transmitters with locations as indicated in Figure 6.2. The transmitting devices were TL-WDR4300 wireless routers operating in the 2.4 GHz ISM frequency band (channel 11) with their transmission powers set to 20 dBm (100 mW). As the receiving device a MacBook Pro notebook with the AirPort Extreme network interface card was used. The experiments were performed during weekends where there were no people present. All neighboring WiFi nodes are operating in the 5 GHz ISM frequency band, hence the wireless interference was minimized.

Two measurement collections were performed at different days. The first one was leveraged for the least square fitting procedure (Equation 6.14) for calculating the multi-wall parameters. The procedure yielded the wall attenuation l_w of 4.81 dBm and the constant l_c of 43.73 dBm.

The second measurement collection was used for the evaluation of the proposed mechanism. The idea of the evaluation procedure was to leverage Equation 6.13 for calculating the probability of establishing a D2D communication link between the transmitting device (one of the WiFi routers) and the receiving device (MacBook Pro). This was done per location and per transmitting device by setting the SNR threshold $\mathrm{SNR}_{\mathrm{th}}$ to different values around the measured average SNR value. The measured average SNR value was obtained by averaging the obtained RSSI values from certain transmitter and receiver locations and subtracting the network interface card's noise floor of -96 dBm. In our evaluation procedure, we therefore in total leveraged 164 transmitter-receiver pairs (41 receiver x 4 transmitter locations).

In practical deployment scenarios, the SNR threshold is usually going to be set to a certain constant value that is required for achieving the desired quality of a D2D communication link. However, the aim of our experiments was to characterize the behavior of the proposed mechanism for SNR threshold values that are close to the measured average SNR, i.e. for which the proposed mechanism has a higher chance of making wrong decisions. For that reason, different SNR threshold values ranged from values that were 5 dB lower to values 5 dB higher than the measured average SNR value.

We assumed that a D2D communication link can be reliably established if the probability of establishing a link $\mathrm{F}_\mathrm{d}(\mathrm{link})$ is higher than 0.95. We further assumed that the link cannot be established if the probability $\mathrm{F}_\mathrm{d}(\mathrm{link})$ is lower than 0.95. As a reference for the evaluation, for the SNR threshold values strictly higher than the average measured SNR values we assumed the link cannot be reliably established. Similarly, for the SNR threshold values strictly lower than the measured average SNR we assumed that the link can be reliably established.

The evaluation metrics were the number of false positive and false negative decisions about the possibility of establishing a D2D communication link between two devices. False positive decisions made by the mechanism will have the following end-result. The devices that wish to establish a D2D link will initiate its establishment by one of them issuing a probe request that will not be heard by the other. The false negative decisions represent a loss of D2D communication opportunity because the link can be successfully established, but the establishment is not initiated because of the wrong decision made by the mechanism.

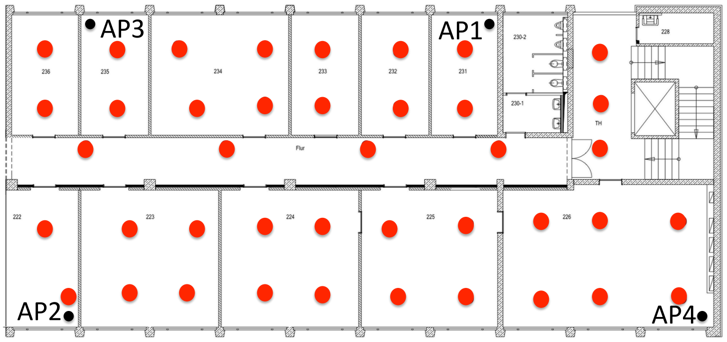


Figure 6.2: Locations of WiFi APs and evaluation points in the testbed environment

The percentage of false positive link establishment decisions for various differences between SNR threshold and measured average SNR is given in Figure 6.3. Similarly, the percentage of false negative decisions is given in Figure 6.4. For deriving the results the per-axis localization error σ of both devices was set to 0.1 m, since that is the positioning error of the used autonomous mobility platform. As visible in Figure 6.3, setting the SNR threshold to a value 5 dB higher then the average measured SNR results in less than 2% of false positive link establishment decisions. Similarly, as visible in Figure 6.4, if the measured average SNR is a value that is more than 4 dB lower than the SNR threshold the probability of false negative link establishment decisions is reduced to less than 2%. In conclusion, the results show that, even for a complex indoor environment, by setting the SNR threshold value to a value that is 5 dB higher than the desired one for establishing a D2D link, the probability of false positive decisions is reduced to less than

2%. However, such a conservative configuration of the SNR threshold could yield false negative link establishment decisions for SNR values that are higher than the set threshold, as depicted in Figure 6.4.

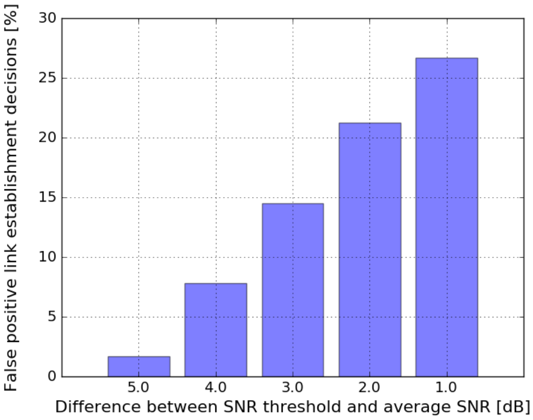


Figure 6.3: Percentage of false positive link establishment decisions

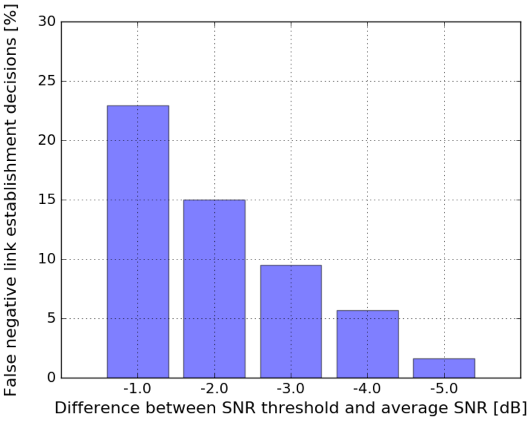


Figure 6.4: Percentage of false negative link establishment decisions

As mentioned, for the previous results we used the per-axis localization error value σ of 0.1 m, since that is the localization accuracy of the used autonomous mobility platform. In the following, we aim at establishing the correlation between the number of false negative D2D link establishment decisions and per-axis localization error σ. We set the SNR threshold to a value that is 5 dB higher than the one nominally needed for establishing a D2D link. We then increase the value of σ from 0.1 to 1.5 m and accordingly change the location coordinates of both devices by adding to their coordinates x and y a number drawn from a zero-mean Gaussian distributions with variance σ^2. By doing that we "artificially" increase the error of location information of both devices. Since the SNR threshold is lower than the average measured SNR, the devices should nominally be able to establish a D2D communication link, hence the number of false negative decisions should be low. However, due to increased localization errors, the proposed mechanism becomes more conservative in its calculation of the probability of establishing a D2D communication link. This "loss of opportunity for D2D communication" is depicted in Figure 6.5. As visible in the figure, with an increase of per-axis localization error σ from 0.1 to 1.5 m the number of false negative decisions increases from roughly 2% to roughly 12%. This relatively small loss of opportunity for D2D communication due to an increase in localization errors comes from the fact that SNR is inversely related to the distance between communication devices. In other words, SNR decreases relatively slowly with an increase in the distance between devices.

An increase in localization errors further results in a decrease in the number of false positive link establishment decisions. Additionally, for a large margin between the average measured SNR and SNR threshold (i.e. larger than 4dB) there is no effect of the increased localization errors because the number of false positives is even for small localization errors very close to zero. A decrease in the number of false positives with an increase in localization errors should not be interpreted as a benefit of larger localization errors in the proposed mechanism. That is to say, an increase in localization errors results in a loss of potential for establishing a D2D communication

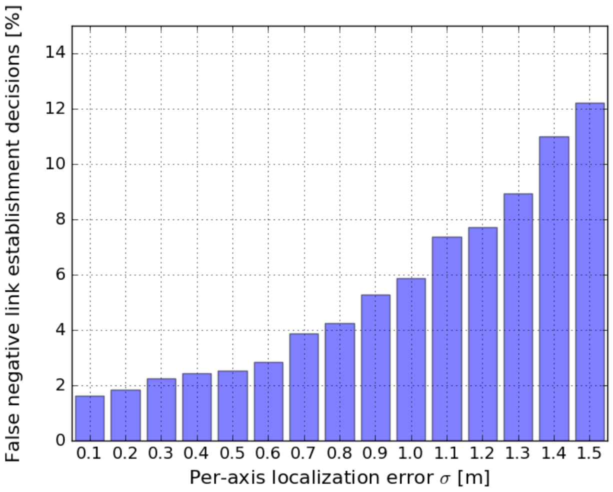


Figure 6.5: Percentage of false negative link establishment decisions vs. localization error per-axis σ

link, as discussed previously. The side effect here is that, due to increased localization errors, the proposed mechanism becomes more conservative in making a positive link establishment decision, hence a decrease in the number of false positive link establishment decision is observed.

6.2 Location-based Mechanism for Positioning of a Mobile Relay

Leveraging a relaying device to transmit information from a source to a destination is shown to enhance throughput, coverage, and energy efficiency of wireless communication [228]. If the destination device is nomadic, however, a fixed relaying device usually significantly constraints the achievable benefits. This fact motivates the use of mobile relaying in which the relaying device is capable of positioning itself in a served environment in a way that optimizes the path quality between the source and the destination [229]. However, it is still unclear how to find an optimal location of the mobile relay for such a scenario.

In the following, we propose a mechanism for positioning of the mobile relay. Under the assumption that an environment is served by a localization service, we ground the decision about the optimal relay location on physical location information of the source and the destination. The proposed mechanism accounts for the errors in location information provided by the localization service, as well as for the propagation characteristics (i.e. path-loss and large-scale fading) of the served environment.

We evaluate the proposed mechanism in a complex indoor environment by leveraging a flexible testbed infrastructure for supporting such experimentation. We do in the 2.4 GHz ISM frequency band using WiFi as an example technology, although the proposed mechanism can be applied for various other technologies for wireless communication. To demonstrate the baseline of the achievable performance of the proposed mechanism, we first evaluate the mechanism under relatively small localization errors, i.e. the smallest ones obtainable in our testbed environment. In case of relatively small per-coordinate localization errors of 10 cm, our results demonstrate less than 4 dB in average difference between the measured SNRs of optimal transmission paths and the measured SNRs of transmission paths through locations yielded by the proposed mechanism. Our results also characterize the loss of communication path quality as a function of increasing localization errors. Finally, we characterize the SNR enhancements due to mobile relaying supported by the proposed mechanism in comparison to direct transmission between end-devices.

6.2.1 Relaying Scenario

The envisioned relaying scenario is presented in Figure 6.6. As depicted in the figure, the aim of the source is to transmit information to the destination, which is achieved through the use of a mobile relay. Since the source is static, perfect location information of the source is assumed to be known to the network infrastructure. The environment is assumed to be served by a localization service deployed in the infrastructure. Location information of all nomadic devices in the environment are specified with their x and y coordinates in a 2D coordinate system. Per-coordinate localization errors of the devices are assumed to have the same standard deviation σ. This assumption is made because location information of the devices in a single environment are usually provided by the same localization service, hence statistically the same errors are expected for all of them.

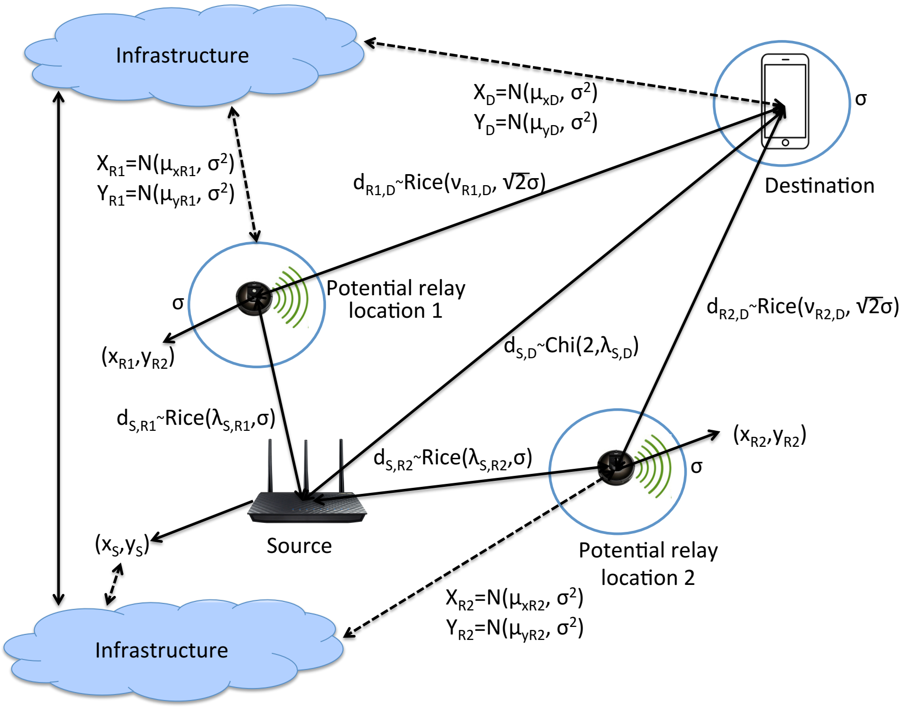

Infrastructure
X_D=N(μ_{xD}, σ^2)
Y_D=N(μ_{yD}, σ^2)
σ
Destination
X_{R1}=N(μ_{xR1}, σ^2)
Y_{R1}=N(μ_{yR1}, σ^2)
$d_{R1,D}$~Rice($v_{R1,D}$, $\sqrt{2}\sigma$)
σ
Potential relay location 1
(x_{R1},y_{R2})
$d_{R2,D}$~Rice($v_{R2,D}$, $\sqrt{2}\sigma$)
$d_{S,D}$~Chi(2,$\lambda_{S,D}$)
$d_{S,R1}$~Rice($\lambda_{S,R1}$,σ)
$d_{S,R2}$~Rice($\lambda_{S,R2}$,σ)
(x_{R2},y_{R2})
σ
(x_S,y_S)
Source
Potential relay location 2
X_{R2}=N(μ_{xR2}, σ^2)
Y_{R2}=N(μ_{yR2}, σ^2)
Infrastructure

Figure 6.6: Overview of the envisioned scenario

The mechanism for positioning of the mobile relay is also deployed in the infrastructure. The goal of the mechanism for positioning of the mobile relay is to decide to position the relay at a location among multiple potential locations so that the path quality between the source and the destination is maximized.

6.2.2 Location-based Mechanism for Positioning of a Mobile Relay

We specify the path quality between the source and the destination by leveraging the Policy 1 from [230]. We select this policy because it yields the best relaying performance among a set of evaluated ones, as reported in [230]. However, instead of taking the instantaneous Channel State Information (CSI), we take the expected SNR between the source and the relay at a given location (i.e. $\overline{\mathrm{SNR}}_{S,R}$) and the expected SNR between the relay at a given location and the destination (i.e. $\overline{\mathrm{SNR}}_{R,D}$). This is because of the assumption that the relay remains at the same location during a communication session between the source and the destination. The path quality for a certain location $i, i = 1, ..., N$, is then given by:

$$\overline{\mathrm{SNR}}_i = \min\{\overline{\mathrm{SNR}}_{S,R_i}, \overline{\mathrm{SNR}}_{R_i,D}\}. \tag{6.15}$$

The optimal relay location among a set of N candidate locations, denoted as l^*, is selected according to the following criterion:

$$l^* = \arg \max_{i=1,...,N} \{\overline{\mathrm{SNR}}_i\}. \tag{6.16}$$

Euclidean Distance Between Devices

A per-coordinate error of location information is modeled by a zero-mean normally distributed random variable, which is, as mentioned previously, a well-established procedure. In the following, we derive expressions for the Euclidean distance between two devices in case one device's location is erroneous and the other's location is perfectly accurate, as well as in case both devices' locations are erroneous.

Euclidean Distance Between Source and Relay Suppose that the correct location information of the source is (x_S, y_S). Furthermore, the potential location of the mobile relay is provided by the localization service is (μ_{x_R}, μ_{y_R}), while its correct location is a random variable (X_R, Y_R). We assume that each 2D coordinate of the location information of the relay provided by the localization service is a normally distributed random variable specified by its mean value μ and standard deviation σ:

$$X_R \sim \mathcal{N}(\mu_{x_R}, \sigma^2), \quad Y_R \sim \mathcal{N}(\mu_{y_R}, \sigma^2). \tag{6.17}$$

Euclidean distance between the source and the mobile relay is therefore given by the following random variable:

$$d = \sqrt{(X_R - x_S)^2 + (Y_R - y_S)^2}. \tag{6.18}$$

Proposition 6.2.1. *If the currect location of the relay is given according to Equation 6.17 and the correct location of the source is known, the Euclidean distance between the source and the relay is a Rice distributed random variable $Rice(\lambda, \sigma)$ with parameters σ and λ with the CDF given as follows:*

$$F_{S,R}(\delta) = \mathbb{P}(d \leq \delta) = 1 - Q_1\left(\frac{\lambda}{\sigma}, \frac{\delta}{\sigma}\right), \tag{6.19}$$

where:

$$\lambda = \sqrt{(\mu_{x_R} - x_S)^2 + (\mu_{y_R} - y_S)^2}. \tag{6.20}$$

Furthermore, Q_1 is the Marcum 1 function given by [224]:

$$Q_1(a, b) = \int_b^{\infty} x \exp\left(-\frac{x^2 + a^2}{2}\right) I_0(ax)\,\mathrm{d}x, \tag{6.21}$$

for $a, b \geqslant 0$, where I_0 is a well-known modified Bessel function of the first kind.

Proof. The proof is given in Appendix B. □

Euclidean Distance Between Relay and Destination Suppose that the potential location information of the relay and of the location information of the destination are provided by the localization service as (μ_{x_R}, μ_{y_R}) and (μ_{x_D}, μ_{y_D}), while the correct locations are random variables (X_R, Y_R) and (X_D, Y_D). Same as before, we assume that each coordinate of the location information provided by the localization service is a normally distributed random variable specified by its mean value μ and standard deviation σ:

$$X_R \sim \mathcal{N}(\mu_{x_R}, \sigma^2), \quad Y_R \sim \mathcal{N}(\mu_{y_R}, \sigma^2) \tag{6.22}$$

$$X_D \sim \mathcal{N}(\mu_{x_D}, \sigma^2), \quad Y_D \sim \mathcal{N}(\mu_{y_D}, \sigma^2). \tag{6.23}$$

Proposition 6.2.2. *If the locations of the relay and the destination are given according to Equations 6.22 and 6.23, the Euclidean distance between the devices is a random variable distributed according to a Rice distribution $d \sim \mathrm{Rice}(\nu, \sqrt{2}\sigma)$ with the following CDF:*

$$F_{R,D}(\delta) = \mathbb{P}(d \leq \delta) = 1 - Q_1\left(\frac{\nu}{\sqrt{2}\sigma}, \frac{\delta}{\sqrt{2}\sigma}\right), \tag{6.24}$$

where:

$$\nu = \sqrt{(\mu_{x_R} - \mu_{x_D})^2 + (\mu_{y_R} - \mu_{y_D})^2}. \tag{6.25}$$

Moreover, Q_1 is again the Marcum 1 function, as specified previously with Equation 6.21.

Proof. The proof is given in Appendix B. □

Propagation Modeling

In the propagation modeling, path-loss and large-scale fading (shadowing) are considered. For the envisioned scenario, we believe that it is important to model only the long-term behavior of the wireless channel. Small-scale (multi-path) fading could, due to destructive interference, create deep fades that will affect the quality of the path, as discussed in e.g. [231]. However, due to small-scale mobility of the destination device (e.g. being held by a person) and due to the fact that we are predominantly considering complex indoor environments where there is a constant change in small-scale fading (e.g. people mobility, opening and closing the doors, etc.), these deep fades are expected to have a short-time span. Additionally, in practice there is a certain time required by the mobile relay to position itself to a location yielded by the proposed mechanism. Since we want to position a relay at a long-term optimal location, we do not model small-scale fading, which is a well-established procedure in location-based mechanisms for the selection of opportunistic relays, e.g. [135, 134]. For propagation modeling we used the previously discussed multi-wall model, the total signal attenuation in the model in Watt at distance d between the transmitter and the receiver, which is the result of path-loss and shadowing, is given by Equation 6.9. The SNR between two devices is then given by:

$$\mathrm{SNR} = \frac{P_{tx}}{N\kappa d^{\gamma}}, \tag{6.26}$$

As mentioned previously, the SNR between the transmitter and the receiver is affected by the random parameter d. To use this SNR values for the positioning of the relay later, one option is to use the expected SNR. However, since d is Rician, d^2 is noncentral Chi squared distribution and the expected value of $\frac{1}{d^{\gamma}}$ does not exist for $\gamma \geq 2$ [232, p. 345]. Instead we assume that SNR is measured in dB. Therefore, in the following we derive expressions for the expected value of logarithm of SNR between two devices in case both devices' location information are erroneous, as well as for the case one device's location information is perfectly accurate, while the other's location information is burdened with errors.

Expected SNR Between Source and Relay

Proposition 6.2.3. *Assuming the Euclidean distance between the source and the relay is a Rician, i.e. a non-central Chi distributed random variable, as specified with Equation 6.19, the expected logarithm of SNR between the source and the relay at a certain location is given as follows:*

$$\mathrm{SNR}_{\mathrm{S,R}} = \ln \frac{P_{tx}}{N\kappa\sigma^{\gamma}} - \frac{\gamma}{2} \ln \left(\frac{\lambda^2}{\sigma^2} g(\frac{\lambda^2}{\sigma^2}) \right), \tag{6.27}$$

where the function $g(.)$ is defined as:

$$g(\xi) = \exp \left(\int_{\xi/2}^{\infty} \frac{e^{-t}}{t} \mathrm{d}t \right). \tag{6.28}$$

Proof. See Appendix B. □

Expected SNR Between Relay and Destination

Proposition 6.2.4. *Assuming the Euclidean distance between the relay at a certain location and the destination is Rice distributed random variable, as given with Equation 6.24, the expected SNR between the relay and the destination is given as follows:*

$$\mathrm{SNR}_{\mathrm{R,D}} = \ln \frac{P_{tx}}{N\kappa(\sqrt{2}\sigma)^{\gamma}} - \frac{\gamma}{2} \ln \left(\frac{\nu^2}{2\sigma^2} g(\frac{\nu^2}{2\sigma^2}) \right), \tag{6.29}$$

where the function $g(.)$ is given by Equation 6.28.

Proof. See Appendix B. □

Discussion

In the two hop transmission through a relay, the expected SNR is a minimum one among the expected SNR between the source and the relay (i.e. $SNR_{S,R}$) and the expected SNR between the relay and the destination (i.e. $SNR_{R,D}$), as specified by Equation 6.15. The proposed mechanism yields a location of the relay by comparing the expected SNRs of transmissions through relays at different potential locations in the served environment, as defined by Equation 6.16. The location yielded by the mechanism is the one that maximizes the expected SNR. The relay can then be instructed to position itself at that location. The potential locations of the relay can be defined in a grid-like fashion or in any other constellation. This decision is currently left to the network administrator. Note that the decision if a direct or a relayed transmission should take place can also be based on the expected SNR by leveraging modified Equation 6.27, where the location of the destination should be used instead of the location of the relay.

Same as for the previously discussed scenario of location-based D2D link establishment, the proposed mechanism for positioning of a mobile relay leverages the multi-wall model for modeling large-scale fading and path-loss. The multi-wall model assumes that a floor plan of the served environment is used for determining the number of walls in the direct path between the devices. The requirement for the availability of a floor-plan does not pose a challenge for the deployment of the proposed mechanism. The floor-plan will usually be available because the environment is serviced by a localization service and such a service usually requires a floor plan of the served environment.

Furthermore, the multi-wall model requires an estimation of model parameters, i.e. the wall attenuation l_w, the path-loss coefficient γ, and the constant l_c related to a model fitting procedure. The model fitting is a procedure of calculating model parameters using measurements collected at different locations in a targeted environment. The used model fitting is a simple least-square fitting procedure, as specified with Equation 6.14. The model fitting procedure requires a certain number of measurements to be collected before the model parameters could be calculated. Same as for the D2D link establishment scenario, this requirement should not pose a challenge for the deployment of the proposed mechanism, because the collection of necessary measurements can be performed through crowd-sourcing.

6.2.3 Evaluation of the Proposed Mechanism

In the evaluation, we aimed at examining the difference between the measured SNRs at optimal locations and the measured SNRs at locations yielded by the proposed mechanism for positioning of the mobile relay. We performed our experimental evaluation using WiFi as an example technology, although the proposed mechanism can conceptually be used with other technologies for wireless communications. Note that, although that would have been a natural first step in the evaluation, we did not evaluate the accuracy of modeling of SNR values using the COST 231 multi-wall model. This evaluation has been carried out previously for the same setup, with the evaluation results reported in [149] (Section 6.1).

For our evaluation we selected the previously discussed testbed infrastructure. Using the autonomous mobility platform we were able to position the relay device at different locations in the environment. The locations were defined in a grid-like fashion, as indicated with red dots in Figure 6.7 (with some small deviations due to obstacles in the environment). As reported in Section 3.1.3, the per-coordinate accuracy of the autonomous mobility platform in the environment is in average roughly 10 cm.

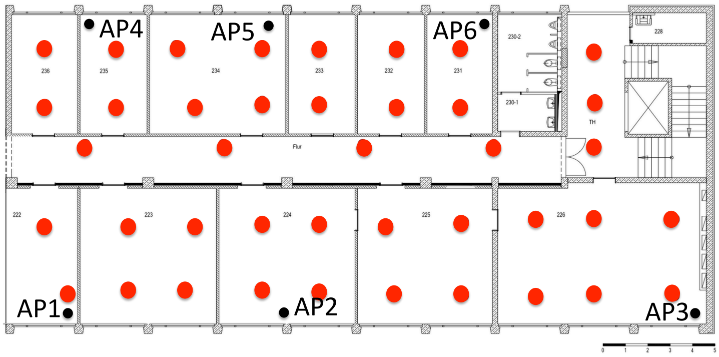


Figure 6.7: Locations of APs and evaluation points in the testbed environment

We used six WiFi APs in our evaluation, with their locations as indicated in Figure 6.7. The locations of the APs were presurveyed using a sophisticated Tachymeter Typ TS 06 Plus (Leica) device and were, hence, known with a very high level of accuracy, i.e. with the average localization errors of less than 2 mm [136]. In the evaluation, each AP was in turn used as the source of information, while all the others were used as destinations. To support the previously discussed scenario, we added a certain level of localization inaccuracies to the location information of APs that were used as destinations. These inaccuracies were introduced by adding a number drawn from a zero-mean Gaussian distribution with a given σ to the perfect location information obtained through presurveying. As the result, the location of the source was in each instance of the experiment perfectly accurate, while the location of the destination and the potential locations of the mobile relay were burdened with inaccuracies characterized by the parameter σ.

At each measurement location we performed 40 scans for WiFi beacon packets from the six APs, followed by extracting RSSI values from the obtained beacon packets. 40 scans were taken to reduce the temporal variability due to small-scale fading from the measurements. The APs were TL-WDR4300 wireless routers operating in the 2.4 GHz ISM frequency band (channel 11) with their transmission powers set to 20 dBm (100 mW). The used routers feature 3 transmitting antennas, thus the spatial variability due to small-scale fading is reduced in the measurements. The receiver of beacon packets transmitted by the APs was a MacBook Pro notebook with the AirPort Extreme network interface card. The experiments were performed during a weekend, when no people were present in the testbed premises. Furthermore, in the testbed all neighboring uncontrollable WiFi nodes are operating in the 5 GHz ISM frequency band [138], thus the interference was minimized during the experimentation.

Two measurement collections were performed at separate days. The first one was used for the least square fitting procedure (Equation 6.14), i.e. for calculating the multi-wall parameters. The procedure yielded wall attenuation l_w of 4.72 dBm, path-loss coefficient γ of 1.74, and constant l_c of 46.73 dBm. The second collection of measurements was used for the evaluation of the proposed mechanism. The evaluation procedure involved use of Equations 6.27 and 6.29 for calculating the expected SNRs between the source and the mobile relay at different locations, and the mobile relay at different locations and the destination, respectively. The expected SNR of each potential path was obtained by leveraging Equation 6.15. The location of a path that maximizes the expected SNR was selected as the optimal one by the proposed mechanism according to Equation 6.16.

This was done for each of 41 mobile relay locations and by leveraging each one of six APs as the source, while all the other APs were in turn used as the destinations. The procedure yielded 15 relaying decisions, where in each of them 41 different transmission paths were considered (one for each potential location of the mobile relay). In our environment, we had a full coverage with all used APs, so in our relaying mechanism and consequently in the evaluation the transmission range did not play an important role. However, the range can be easily introduced into the mechanism by specifying a cut-off SNR value below which the communication cannot take place.

As a reference for evaluating the decisions made by the mechanism, we leveraged the measured averaged SNR values. The measured averaged SNR value for a given AP at a certain measurement location was obtained by averaging the RSSI values from that AP measured at that measurement locations. Furthermore, from the obtained averaged RSSI value from a certain AP at a given location we subtracted the network interface card's noise floor of -96 dBm. This procedure yielded reference SNR values from different APs as observed at each measurement location, which were further used for finding the optimal transmission path, i.e. the one that maximizes the measured averaged SNR between the end-devices.

The used evaluation metric is the loss of path quality, which is specified as a difference between the measured averaged SNR of the optimal path and the measured averaged SNR of the path through the relay at a location yielded by the mechanism. The logic behind defining this metric, instead of specifying a more intuitive one, e.g. the percentage of correct decisions, is the following. With higher probability the mechanism will select a suboptimal transmission path if the optimal location of the relay and the one decided by the mechanism result in communication paths with comparable SNRs. However, the selection of such a location will not significantly influence the quality of a communication path, although it would influence a binary metric such as the percentage of correct decisions about optimal locations made by the mechanism.

For the relatively small localization error σ of 10 cm (i.e. the per-axis localization accuracy of the autonomous mobility platform), the loss of path quality in the proposed mechanism due to the selection of locations that do not maximize the measured averaged SNR is in average around 4 dB, as depicted in Figure 6.8. Furthermore, the loss of communication path quality is increased with the increase in the expected localization errors, as shown in the figure. We selected the range of localization errors σ to capture both the scenarios in which the neighboring potential locations cannot and can be "confused" by the proposed mechanism. For example, with the increase from 10 cm to 1 m in localization errors σ, the loss of communication path quality is doubled, i.e. it increases from roughly 4 to around 8 dB. In addition, there is a very little change in the performance of the proposed mechanism for the per-axis localization errors σ ranging from 0.1 and 0.4 m. This is because those levels of inaccuracies in location information are not enough to "confuse" the proposed mechanism. In other words, the distance between the potential locations of the mobile relay of roughly 2.4 m (Figure 6.7) provides a high enough safety margin for the "drift" in locations of communicating devices in the proposed mechanism caused by the localization errors. For example, let us assume that the expected per-axis location error σ is 0.4 m and the per-axis distance between two potential and neighboring locations is 2.4 m. The vast majority of locations where the relay will position itself (with a certain error in positioning) will be at most 3σ away from the respective potential location where the relay should ideally position itself. The mechanism is taking into account such errors in positioning. However, for the two neighboring locations that are 2.4 m apart, the errors of at most $3 \cdot 0.4\ m = 1.2\ m$ from each potential location are still not enough to confuse one location another. The errors that occur for those small σ values can be accounted mostly to the imperfections of the propagation model or, although scarcely, to the fact that some neighboring locations are less than 2.4 m apart from each other, as visible in Figure 6.7.

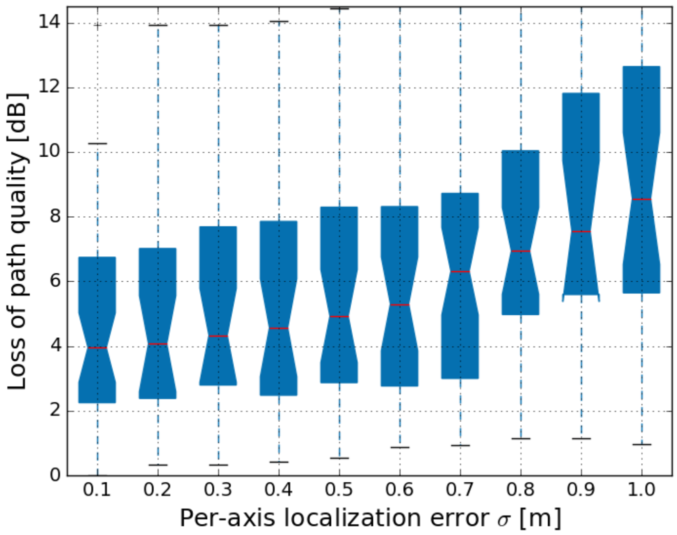


Figure 6.8: Loss of path quality in the proposed mechanism due to the selection of relay locations that do not maximize the measured averaged SNR as a function of increased per-axis localization errors σ

User's Perspective

Despite its aforementioned benefits, using the mobile relay for communication between the end-devices has a set of drawbacks. In practice, it will take a certain amount of time for the relay to position itself at a certain location and establish a path between the source and the destination. It is also possible that the relay is not able to reach the desired location, for example due to an obstacle. Moreover, due to relaying an additional delay, as well as an increase in jittering can be expected [233]. Finally, the mobile relaying imposes an added complexity and it increases the utilization of radio resources [234]. Hence, the potential user could question the initiative for using such a communication paradigm, in comparison to directly transmitting the information from the source to the destination.

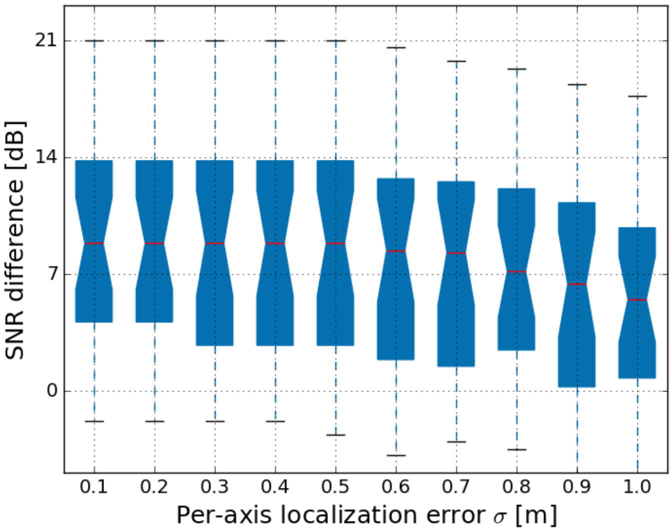


Figure 6.9: Difference between the averaged measured SNR of a path through the mobile relay at a location yielded by the mechanism and the averaged measured SNR of a direct path between end-devices for different per-axis localization errors σ

However, if the primary design goals are throughput and coverage enhancements, then the mobile relaying provides benefits in comparison to direct communication between the source and the destination. To characterize the benefits of the proposed mechanism in terms of throughput enhancements, using the previous setup we evaluate the difference between measured averaged SNR of the path through the mobile relay at a location yielded by the mechanism and the measured averaged SNR of the direct path between the source and the destination. We do that for different per-axis localization errors σ. The results are depicted in Figure 6.9. As visible from the figure, for the relatively small per-axis localization errors σ of 10 cm, the SNR of a path through the mobile relay is in average more than 9 dB higher than the SNR of a direct path between the source and the destination. As the expected localization errors increase, this difference becomes smaller, since the locations yielded by the proposed mechanism become less optimal in terms of SNR maximization. Example-wise, for the per-axis localization error increase from 10 cm to 1 m, the averaged difference between in the SNR of a relayed and of a direct path reduces from roughly 10 to around 5 dB. Note that, as depicted in the figure, this difference can be a negative number. This can happen for multiple reasons. First, the proposed mechanism for positioning of the mobile relay can yield a non-optimal location, hence the achieved SNR through the relayed path can be smaller than the SNR of the direct path. Second, it can happen that the distance between the source and the destination is smaller than the distance between either the source or the destination and the potential location of the mobile relay. Third, if all paths through a relay are heavily obstructed and only the direct provides a LoS connectivity (e.g. in a hallway), it can happen that the SNR of the direct path is larger than the SNR of any path through the relay.

6.3 Conclusions

Our final contributions pertained to using location information for improving selected aspects of wireless networks. First, we presented a decision-making mechanism for location-based D2D communication link establishment. The proposed mechanism takes into account the imperfect location information of the devices that wish to establish a D2D link, as well as the intrinsic randomness in wireless propagation. We evaluated the proposed mechanism in a complex indoor environment and showed that, even for such a complex environment, the mechanism can be reliably used (almost without false positive decisions) if the SNR threshold used as an input to the mechanism is 5 dB higher than the nominal SNR required for establishing a D2D link. We have also shown that the imperfections in localization errors cause only a relatively small loss of communication potential. Hence, today's off-the-shelf indoor localization systems with average localization errors of roughly $\sim$0.5 m localization errors per-axis in a 2D coordinate system [61] can already support the operation of the proposed decision-making mechanism.

Second, we proposed a location-based mechanism for positioning of a mobile relaying device. The presented contribution can initially seem unrealistic, given that currently wireless infrastructures enhanced with mobile devices are not common. However, the concept of enhancing wireless infrastructures with mobility is shown to be beneficial in terms of path quality optimization and coverage extension [229]. Given the fast advancing field of robotics, in particular in terms of Unmanned Aerial Vehicles (UAVs), we believe that this concept is not far-fetched and that in the near future wireless infrastructures will indeed be enhanced with mobile relaying devices.

For the assessment of propagation characteristics of the environment, the proposed mechanism for positioning of a mobile relaying device accounts for the localization inaccuracies in potential locations of the communicating devices, which yields the location where the relay should be positioned. The proposed mechanism estimates the expected SNR, hence for positioning of the relay it does not rely on a feedback information about the path quality from the end-devices participating in communication. Since there is no signaling from the source or the destination to the relay, there is no transmission interruption between the source and the destination until the relay positions itself to a location yielded by the proposed mechanism. Furthermore, our evaluation results demonstrate a small difference between the SNR of optimal paths and the SNR of paths through locations yielded by the proposed mechanism. We further demonstrate the benefits of relaying using the prosed mechanism in terms of the SNR enhancements, in comparison to direct communication between the source and the destination. For the current COTS localization approaches, the mechanism can already provide very good performance in terms of selecting a close-to-optimal location where the relay should be positioned and in terms of enhancements in the SNR, in comparison to direct transmission between the end-devices.

Note that the proposed mechanism for positioning of a mobile relay is not limited to the scenario we assumed, but can be utilized more broadly, e.g. in case of multi-hop relaying, in case the location information of the source is burdened with errors, in case the location information of the destination is perfectly accurate or in case the location information of both the source and the destination devices are burdened with errors. Furthermore, the proposed mechanism works for both downlink and uplink transmission paths. In addition, although we discussed the mechanism in terms of mobile relaying, it is straightforward to apply the same mechanism for the selection of potential relays among a set of opportunistic candidates, as for example discussed in [235, 135, 236]. Although we assumed a simple repeater-type relay, in combination with our location-based mechanism for positioning of the mobile relay, more advanced relaying schemes are also applicable. The mobile relay is considered to be a part of the wireless infrastructure, hence it is a trusted entity. Hence, the usual security and privacy related pitfalls of relaying in wireless networks [237] do not apply here, in contrast to opportunistic relaying through generally untrusted mobile devices, e.g. [133].

7 Conclusion and Future Work

Infrastructures and methodologies for evaluation of indoor localization solutions are being developed rapidly in the recent years. This can be seen by different standards for evaluation of indoor localization solutions, originating from large standardization bodies, becoming publicly available. One such example is the previously mentioned ISO/IEC 18305 standard "Test and evaluation of localization and tracking systems" that was published by NIST and became publicly available in 2015. Additionally, various consortia of large industrial players interested in indoor localization have been formed recently. For example, the InLocation Alliance[1], a consortium founded in December 2013 by the mobile industry to accelerate the adoption of indoor position solutions, includes large industrial members such as Broadcom Corporation, Cisco Systems, Nokia Corporation, Qualcomm Incorporated, Samsung Electronics, Sony Mobile Communications AB, Telecom Italia, etc. With the standardization of the methodologies for evaluation of indoor localization, and with an increased interest of large industrial players in the evaluation infrastructures and methodologies, the possibility of further contributions by the academic community in this direction is becoming smaller.

However, the increase in the objectiveness of such evaluation results will provide additional insights in the performance of various indoor localization solutions. We have shown this on an example of WiFi fingerprinting, but similar statements can most probably be made for other solutions, which is yet to be investigated. In general, the aim should be a robust, timely, accurate, and performance-wise assessable indoor localization solution, regardless of its type.

In relation to WiFi-based indoor fingerprinting, there is an untapped research opportunity around minimizing the efforts involved in training required for enabling a fingerprinting-based solution. In particular, we believe that a crowd-sourcing-based collection of training fingerprints, combined with the ETS for the generation of virtual training fingerprints, could substantially reduce the efforts needed for collecting training fingerprints and at the same time benefit the accuracy of WiFi fingerprinting-based localization.

In regards to the predictability of the performance of WiFi-based fingerprinting, we contributed by demonstrating that the performance extrapolation of fingerprinting algorithms across environments is possible. However, at this point we cannot make any final conclusions on how similar the environments and respective parameterizations of fingerprinting algorithms have to be. In order to give a reliable answer, insights from additional environments and algorithms will be necessary. Hence, we believe future work should be oriented toward this goal, where the support from the community is welcome. The envisioned contribution from the community is twofolds. First, the interested parties can contribute by extending the available raw datasets with data from additional environments. These data-traces can be provided and used in a simple way through the proposed publicly accessible platform for streamlined evaluation of RF-based

[1] http://inlocationalliance.org/

indoor localization solutions. Second, the interested parties can contribute by evaluating their WiFi-based fingerprinting algorithms using the offered raw data-traces and provided evaluation infrastructure. By increasing the number of environments and evaluation results, more insights will be gained about the correlation between the similarity of environments and parameterizations of algorithms on one side, and the achieved performance of such algorithms on the other. These insights would serve to evaluate with higher confidence our hypothesis about the possibility of extrapolating the performance of WiFi-based fingerprinting algorithms across environments. Furthermore, these insights would serve to assess the change in the performance of fingerprinting algorithms due to changes in some of the detected important factors for characterizing the similarities between two environments and parameterizations of algorithms.

Future work could also aim at extending the predictability of performance beyond WiFi fingerprinting-based solutions. We believe this should be done by identifying a smaller number of environmental similarity parameters that have a broader impact on the possibility of extrapolation of different localization approaches across environments. One clear drawback of our approach is that, at the moment, we have based the assessment of the similarity in a manual way, through the physical shape and propagation characteristics of different environments. To scale the approach future efforts could resort to automatizing it.

For enabling seamless provisioning of location information to the applications, we proposed a localization service architecture. However, we believe the same principles apply for other sensor-based distributed service architectures, e.g. activity recognition and ambient sensing. A module with similar functionalities as the integrated location service, i.e. selection of services, caching and processing of information, and information distribution to applications, can serve as a central point in such generalized architecture. This clear decoupling between modules also allows for global optimization and management of resources, which could yield substantial improvements in comparison to today's architectures in terms of operating delays, power consumption, and resource utilization. Future work could be oriented toward further assessment of this hypothesis.

As specified by the Resource Interaction Interface (RII) in the proposed localization service architecture, resource types to be reported to a provisioning service are currently defined in an abstract way, i.e. using the parameters tuple (technology, feature). Future work could include defining a full set of such resources, once when in becomes clear what are the technologies and their signal features with practical utilization for generating location information. Moreover, the context of location information could, in addition to device's movement speed and direction, possibly be extended with other relevant contextual information once when it becomes clear what such information could be. This possibility is already envisioned in the designs of Standardized Localization Interface (SLI) and Provisioning Service Interface (PSI). Therefore, the APIs are potentially subject to change and additional standardization efforts should be anticipated.

We further proposed two algorithms for selection of provisioning services focused on primarily on the satisfaction of latency requirements. We believe that future efforts should be oriented toward designing selection algorithms that are primarily focused on accuracy requirements satisfaction, as well as algorithms focused on joint optimization of both accuracy and latency requirements. Moreover, one aspect that has not been taken into account is the complexity of the algorithms, which is going to be important if the numbers of requirements and offerings increase, which is expected in the future. This issue could be considered as a part of the future work.

Furthermore, as demonstrated previously in the thesis, having the global view on the provisioning features, i.e. a more accurate prediction of provisioning features, improves the selection of provisioning services, in comparison to using well-established static performance benchmarks. In practical terms, we envision that the global view could be realized by accumulating and leveraging the knowledge of historical provisioning features in a given environment or by dynamically predicting the performance of a provisioning service by correlating the features of raw data used by that service for generating location information with the expected provisioning features of that service. Although some contributions in this directions already exist (e.g. [101, 102, 103, 238, 239]), we believe future work should be oriented toward further investigation of both potentialities for dynamically predicting provisioning features. Additionally, future work on the standardized localization service could be oriented toward designing more sophisticated methods for caching of location information and for mapping of different location information types.

We further proposed a location-based mechanism for positioning of a mobile relay in wireless networks. Currently, the specification of potential locations of the mobile relay is left to the network administrators, which could result in either over-provisioning on the number of locations or in failing to specify locations that could maximize the benefits of relaying. Obviously, this specification should define potential relay locations so that the whole environment is considered for positioning of the relay. However, the "drifts" in potential locations of the mobile relay should not occur. To avoid these drifts, we believe the expected localization accuracy in the served environment should be taken into account in the specification of potential relay locations. We speculate that the per-axis distance between neighboring potential locations of the mobile relay should be three times larger than the per-axis localization error, so that the loss of path quality is contributed only to the inaccuracies of the propagation model. Potential future work could be oriented toward evaluating this hypothesis by examining an interplay between the density of potential relay locations, expected localization accuracy, and the performance of the proposed mechanism. We believe these future efforts would yield a methodology for optimal specification of potential locations of the mobile relay.

While in this work we considered a scenario with one relay and one destination, future work could also consider scenarios in which one or multiple relays have to be positioned in a way that optimizes path qualities for multiple destinations. In general, enhancing the wireless infrastructure with mobile terminals is a promising approach for improving their capacity, coverage, and energy efficiency, as discussed in e.g. [229, 240]. Hence, future work could be oriented toward addressing the challenges related to mobile relaying, for example addressing clients' mobility scenarios, extending the location-based positioning of a relay to a 3D environment, or interleaving the mobility of a relay with channel allocation and power control.

The proposed location-based mechanisms for D2D link establishment and for positioning of a mobile relay show a potential for improving wireless networks based on location-awareness. We believe this is just one among many potential improvements that can be made in this domain, where a more exhaustive set of of examples can be found in [11]. Future work in this domain should be oriented toward enhancing the understanding on where the benefits of location-awareness lie. Finally, the erroneous nature of location information should be considered, in contrast to the majority of current contributions that rely on its idealized version.

Appendix A

Evaluated Indoor Localization Algorithms

This section gives a short description of the indoor localization algorithms evaluated in the competition. The first two algorithms presented below are based on WiFi fingerprinting, followed by algorithms based on multilateration and proximity using low-power IEEE 802.15.4 nodes.

Quantile-based Indoor Fingerprinting using Dedicated APs: One of the most promising approaches in indoor localization is fingerprinting using WiFi infrastructure. This fingerprinting-based indoor localization algorithm [54] makes use of the RSSI values from WiFi beacon packets for estimating location. For generating fingerprints this algorithm uses the quantiles of RSSI values from beacon packets transmitted from various WiFi APs in the premises. Furthermore, the algorithm uses Pompeiu-Hausdorff distance for calculating the difference between training fingerprints and ones generated by user to be localized. Finally, the algorithm uses the kNN procedure with the parameter k set to 3.

Indoor Geolocation for Android Smartphones with Airplace: This indoor localization algorithm, named Airplace, is an indoor geolocation platform developed for Android tablets and smartphones [241]. Airplace exploits the available WiFi infrastructure and monitors RSSI values from the surrounding APs to determine the unknown user location. The system utilizes a number of RSSI fingerprints collected prior to localization and stored in the radiomap. Location is then estimated by finding the best match between the currently measured fingerprint and fingerprints in the radiomap. The algorithm relies on the Radial Basis Function Network (RBFN) algorithm, which is a neural networks based algorithm that addresses localization as a regression problem. Essentially, it leverages the collected data in the RSS radiomap to build a mapping from the RSS fingerprint space to the 2D physical space. The algorithm has been instantiated in three different versions. In the first version the Android smartphone Nexus S is used as the mobile node for the deployment of the algorithm (Airplace 1). The fingerprinting procedure in this case uses only a set of dedicated APs. Similarly, the second version uses only dedicated APs, but here the mobile node for deploying the algorithm is an Android tablet Nexus 7 (Airplace 2). Finally, in the third version the fingerprinting procedure is done using all APs visible in the environment, while the algorithm is deployed on the same mobile node, Android tablet Nexus 7 (Airplace 3).

Geo-n Localization Algorithm: The Geo-n algorithm [242] is a highly precise, distance based, general purpose localization algorithm. It was developed based on evaluation of large experiments and extensive simulation of the impact of the spatial anchor distribution in an indoor localization setting. The algorithm uses multilateration and is able to deal with outliers as well as with heavily error prone distance measurements. Geo-n uses a two stage filtering technique. First, the most representative intersection points between every pair of circles induced by anchor coordinates and distance measurements are obtained. This is done by removing intersection points that do not contribute to the localization or are suspected to increase the positioning error. Geo-n then uses the residual intersections to estimate the position of the unlocalized node. The node types used for both mobile and infrastructural nodes are TelosB low-power sensor nodes.

RSS Range-based Positioning Using Grid-based Likelihood Estimation: Ranging using RSSI values from indoor networks is a convenient technique for positioning purposes. Thus, the algorithm [243] explores RSSI ranging from all reachable reference low-power sensor nodes (TelosB) in the environment. For the nonlinear and non-Gaussian positioning problem under highly imprecise RSSI ranging measurements, a simple grid-based likelihood estimation for obtaining location estimates is used. The probabilistic distribution of the target's position is represented by a grid obtained from the Bounding-box algorithm. Afterwards each grid cell is weighted by the residual of current observed ranging measurements and, at last, the state estimation is the likelihood expectation. This algorithm requires no history measurements and no assumption of the measurement model, which can be applied to other positioning scenarios. Furthermore, the grid size is very small, bounded to 36 grid cells for each positioning trial, resulting in a low computation and memory complexity.

3CoM (3 Centers of Mass): For signal strength based indoor localization it is a well known approach to assume the position of the anchor node with the highest signal strength to be the location of the node to be localized. This approach works well in environments with a very dense anchor placement. The approach is extended in order to use the center of mass of the positions of the three strongest anchor nodes (TelosB) to estimate the location [244].

Appendix B

Proof of Propositions 6.1.1 and 6.2.2

Note that $X_1 - X_2$ (or $X_R - X_D$ for Proposition 6.2.2) is a difference of two independent Gaussian random variable which is by itself a Gaussian random variable with variance $2\sigma^2$ and mean value $\mu_{x_1} - \mu_{x_2}$. Similarly, one can see that $Y_1 - Y_2$ (or $Y_R - Y_D$ for Proposition 6.2.2) is a Gaussian random variable with $2\sigma^2$ and mean value $\mu_{y_1} - \mu_{y_2}$. Therefore, the distance d follows Rice distribution $d \sim Rice(\nu, \vartheta)$ with parameters $\nu = \sqrt{(\mu_{x_1} - \mu_{x_2})^2 + (\mu_{y_1} - \mu_{y_2})^2}$ and $\vartheta = \sqrt{2}\sigma$. Note that a similar proof of this proposition is given in [245], however for the distribution of Euclidean distance between two univariate Gaussian distributions.

Proof of Proposition 6.1.2

Note that for a fixed distance d, the average SNR, $\overline{\mathrm{SNR}}$, is given by $\frac{P_{tx}}{N\kappa d^\gamma}$. Using a simple Bayesian argument and assuming small scale fading is exponentially distributed, the link probability is first conditioned on d:

$$\begin{aligned}
\mathbb{P}(\mathrm{SNR} > \mathrm{SNR}_{\mathrm{th}}|d) &= \mathbb{P}\left(\frac{hP_{tx}}{N\kappa d^\gamma} > \mathrm{SNR}_{\mathrm{th}}|d\right) \\
&= \mathbb{P}\left(h > \mathrm{SNR}_{\mathrm{th}}\frac{N\kappa d^\gamma}{P_{tx}}|d\right) \\
&= \exp\left(-\mathrm{SNR}_{\mathrm{th}}\frac{N\kappa d^\gamma}{P_{tx}}\right) \\
&= \exp\left(-\frac{\mathrm{SNR}_{\mathrm{th}}}{\overline{\mathrm{SNR}}}\right).
\end{aligned} \tag{7.1}$$

The link probability is then simply obtained by taking the expectation of last expression with respect to d:

$$F_{h,d}(\mathrm{link}) = \mathbb{P}(\mathrm{SNR} > \mathrm{SNR}_{\mathrm{th}}) = \mathbb{E}\left[\exp\left(-\frac{\mathrm{SNR}_{\mathrm{th}}}{\overline{\mathrm{SNR}}}\right)\right]. \tag{7.2}$$

Suppose that $\gamma = 2$. Since d has Rice distribution with, $D = \left(d/\sqrt{2}\sigma\right)^2$ has noncentral χ^2-distribution with degrees of freedom equal 2 and noncentrality parameter $\left(\nu/\sqrt{2}\sigma\right)^2$. It follows:

$$\begin{aligned}
F_{h,d}(\mathrm{link}) &= \mathbb{E}\left[\exp\left(-d^2\frac{N\kappa\mathrm{SNR}_{\mathrm{th}}}{P_{tx}}\right)\right] \\
&= \mathbb{E}\left[\exp\left(-D\frac{N2\sigma^2\kappa\mathrm{SNR}_{\mathrm{th}}}{P_{tx}}\right)\right].
\end{aligned} \tag{7.3}$$

However the last expression is the moment generating function of noncentral χ^2-distribution for $t = -\frac{N2\sigma^2\kappa\mathrm{SNR_{th}}}{P_{tx}}$ and is given by:

$$\mathbb{E}\left[\exp\left(tD\right)\right] = \frac{\exp\left(\frac{\nu^2 t}{1-2t}\right)}{1-2t}. \tag{7.4}$$

Proof of Proposition 6.1.3

The proof follows the standard approach:

$$\begin{aligned}
\mathbb{P}(\mathrm{SNR} > \mathrm{SNR_{th}}) &= \mathbb{P}\left(\frac{P_{tx}}{N\kappa d^2} > \mathrm{SNR_{th}}\right) \\
&= \mathbb{P}\left(\frac{P_{tx}}{N\kappa\mathrm{SNR_{th}}} > d^2\right) \\
&= \mathbb{P}\left(d < \left(\frac{P_{tx}}{N\kappa\mathrm{SNR_{th}}}\right)^{1/2}\right) \\
&= 1 - Q_1\left(\frac{\nu}{\sqrt{2}\sigma}, \frac{1}{\sqrt{2}\sigma}\left(\frac{P_{tx}}{N\kappa\mathrm{SNR_{th}}}\right)^{1/2}\right).
\end{aligned} \tag{7.5}$$

Proof of Proposition 6.2.1

Note that $(X_R - x_S) \sim \mathcal{N}(\mu_{x_R} - x_S, \sigma^2)$ and $Y_R - y_S \sim \mathcal{N}(\mu_{y_R} - y_S, \sigma^2)$. Therefore the Euclidean distance d between the source and the relay follows:

$$d = \sqrt{(X_R - x_S)^2 + (Y_R - y_S)^2} \sim \mathit{Rice}(\lambda, \sigma), \tag{7.6}$$

where λ is given by Equation 6.20. Note that $\frac{d}{\sigma}$ has unit variance and is indeed a noncentral chi distribution with degrees of freedom equal to 2 and the noncentrality parameter is given by $\frac{\lambda}{\sigma}$, denoted by $\chi(2, \frac{\lambda}{\sigma})$. A similar proof of this proposition is given in [245], however for the distribution of Euclidean distance between two univariate Gaussian distributions.

Proof of Proposition 6.2.3

From Equation 6.26, d being Rice distributed random variable, the expected logarithm of SNR is given as follows:

$$\mathbb{E}\left(\ln\frac{P_{tx}}{N\kappa d^\gamma}\right) = \ln\frac{P_{tx}}{N\kappa\sigma^\gamma} - \frac{\gamma}{2}\mathbb{E}\left(\ln\left(\frac{d}{\sigma}\right)^2\right). \tag{7.7}$$

Note that $\frac{d}{\sigma}$ has noncentral Chi distribution $\chi(2, \frac{\lambda}{\sigma})$. Therefore $\left(\frac{d}{\sigma}\right)^2$ is a non-central Chi square distributed random variable. In [246, Theorem 3], the expected value of logarithm of general noncentral Chi square random variable $X \sim \chi^2(2, \xi)$ was derived as:

$$\mathbb{E}\left(\ln X\right) = \ln \xi g(\xi), \tag{7.8}$$

where:

$$g(\xi) = \exp\left(\int_{\xi/2}^{\infty} \frac{e^{-t}}{t} \mathrm{d}t\right).$$

The proof is finished by using $X = \left(\frac{d}{\sigma}\right)^2$ and replacing $\xi = \left(\frac{\lambda^2}{\sigma^2}\right)$.

Proof of Proposition 6.2.4

The proof follows a similar procedure as above with different normalization. The expected SNR is given as:

$$\mathbb{E}\left(\ln \frac{P_{tx}}{N\kappa d^{\gamma}}\right) = \ln \frac{P_{tx}}{N\kappa(\sqrt{2}\sigma)^{\gamma}} - \frac{\gamma}{2}\mathbb{E}\left(\ln\left(\frac{d}{\sqrt{2}\sigma}\right)^2\right).$$

The expected value is evaluated using Equation 7.8 with $\xi = \frac{\nu^2}{2\sigma^2}$. The proposition follows accordingly.

Bibliography

[1] Iris A Junglas and Richard T Watson. Location-based services. *Communications of the ACM*, 51 (3):65–69, 2008.

[2] Kathryn Zickuhr. Location-based services. *Pew Research*, pages 1–25, 2013.

[3] Jochen Schiller and Agnés Voisard. *Location-based services*. Elsevier, 2004.

[4] Bernhard Kölmel and Spiros Alexakis. Location based advertising. In *Proceedings of the First International Conference on Mobile Business, Athens, Greece*, pages 1–7, 2002.

[5] Gary J Grimes. Cellular terminal for providing public emergency call location information, December 26 1995. US Patent 5,479,482.

[6] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[7] Upkar Varshney. Pervasive healthcare. *Computer*, 36(12):138–140, 2003.

[8] Juhi Ranjan, Yu Yao, and Kamin Whitehouse. An rf doormat for tracking people's room locations. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 797–800. ACM, 2013.

[9] M Kjaergaard. Minimizing the Power Consumption of Location-based Services on Mobile Phones. *IEEE Pervasive Computing*, 8(4), 2010.

[10] Dominique Guinard, Vlad Trifa, Stamatis Karnouskos, Patrik Spiess, and Domnic Savio. Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *IEEE transactions on Services Computing*, 3(3):223–235, 2010.

[11] Rocco Di Taranto, Srikar Muppirisetty, Ronald Raulefs, Dirk Slock, Tommy Svensson, and Henk Wymeersch. Location-aware communications for 5g networks: How location information can improve scalability, latency, and robustness of 5g. *IEEE Signal Processing Magazine*, 31(6):102–112, 2014.

[12] H Birkan Yilmaz, Tuna Tugcu, Fatih Alagöz, and Suzan Bayhan. Radio environment map as enabler for practical cognitive radio networks. *IEEE Communications Magazine*, 51(12):162–169, 2013.

[13] Shree Krishna Sharma, Symeon Chatzinotas, and Björn Ottersten. Satellite cognitive communications: Interference modeling and techniques selection. In *2012 6th Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC)*, pages 111–118. IEEE, 2012.

[14] Tao Cai, Jaap van de Beek, Berna Sayrac, Sebastien Grimoud, Jad Nasreddine, Janne Riihijärvi, and Petri Mähönen. Design of layered radio environment maps for ran optimization in heterogeneous lte systems. In *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 172–176. IEEE, 2011.

[15] Marco Hagting, Michel Godefridus Jansen, and Petrus Hubertus Gerardus Van De Berg. Method for performing handover and roaming in a radio communication environment, May 22 2001. US Patent 6,236,860.

[16] Doru Calin, Holger Claussen, and Huseyin Uzunalioglu. On femto deployment architectures and macrocell offloading benefits in joint macro-femto deployments. *IEEE Communications Magazine*, 48(1):26–32, 2010.

[17] Youping Zhao, Bin Le, and Jeffrey H Reed. Network support–the radio environment map. *Cognitive radio technology*, pages 325–366, 2006.

[18] Valentin Radu and Mahesh K Marina. Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pages 1–10. IEEE, 2013.

[19] Kyungmin Chang and Dongsoo Han. Crowdsourcing-based radio map update automation for wifi positioning systems. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information*, pages 24–31. ACM, 2014.

[20] Thanasis Korakis, Gentian Jakllari, and Leandros Tassiulas. A mac protocol for full exploitation of directional antennas in ad-hoc wireless networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 98–107. ACM, 2003.

[21] John G. Markoulidakis, George L. Lyberopoulos, Dimitrios F. Tsirkas, and Eystathios D. Sykas. Evaluation of location area planning scenarios in future mobile telecommunication systems. *Wireless Networks*, 1(1):17–29, 1995.

[22] Yun Zhou, Yanchao Zhang, and Yuguang Fang. Llk: a link-layer key establishment scheme for wireless sensor networks. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 4, pages 1921–1926. IEEE, 2005.

[23] Dominique Tschopp, Suhas Diggavi, Matthias Grossglauser, and Jörg Widmer. Robust geo-routing on embeddings of dynamic wireless networks. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 1730–1738. IEEE, 2007.

[24] Kristian Evensen, Andreas Petlund, Haakon Riiser, Paul Vigmostad, Dominik Kaspar, Carsten Griwodz, and Pål Halvorsen. Mobile video streaming using location-based network prediction and transparent handover. In *Proceedings of the 21st international workshop on Network and operating systems support for digital audio and video*, pages 21–26. ACM, 2011.

[25] Haakon Riiser, Tore Endestad, Paul Vigmostad, Carsten Griwodz, and Pâl Halvorsen. Video streaming using a location-based bandwidth-lookup service for bitrate planning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 8(3):24, 2012.

[26] Hongseok Kim, Gustavo De Veciana, Xiangying Yang, and Muthaiah Venkatachalam. Distributed-optimal user association and cell load balancing in wireless networks. *IEEE/ACM Transactions on Networking*, 20(1):177–190, 2012.

[27] Hakan Koyuncu and Shuang Hua Yang. A survey of indoor positioning and object locating systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(5):121–128, 2010.

[28] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, 2007.

[29] Guanling Chen, David Kotz, et al. A survey of context-aware mobile computing research. Technical

report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[30] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784 vol.2, 2000.

[31] Andreas Savvides, Chih-Chieh Han, and Mani B Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001.

[32] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 173–184. ACM, 2010.

[33] Jie Yang and Yingying Chen. Indoor localization using improved rss-based lateration methods. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE, 2009.

[34] Mortaza S Bargh and Robert de Groote. Indoor localization based on response rate of bluetooth inquiries. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, pages 49–54. ACM, 2008.

[35] Filip Lemic, Vlado Handziski, Adam Wolisz, Timotheos Constambeys, Christos Laoudias, Stephan Adler, Simon Schmitt, and Yuan Yang. Experimental evaluation of rf-based indoor localization algorithms under rf interference. In *2015 International Conference on Location and GNSS (ICL-GNSS)*, pages 1–8. IEEE, 2015.

[36] Masashi Sugano, Tomonori Kawazoe, Yoshikazu Ohta, and Masayuki Murata. Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard. *Target*, 538: 050.

[37] Zhe Xiang, Song Song, Jin Chen, Hao Wang, Jian Huang, and Xingxin Gao. A wireless lan-based indoor positioning technology. *IBM Journal of research and development*, 48(5.6):617–626, 2004.

[38] Kaishun Wu, Jiang Xiao, Youwen Yi, Dihu Chen, Xiaonan Luo, and Lionel M Ni. Csi-based indoor localization. *IEEE Transactions on Parallel and Distributed Systems*, 24(7):1300–1309, 2013.

[39] Dian Zhang, Yanyan Yang, Dachao Cheng, Siyuan Liu, and Lionel M Ni. Cocktail: An rf-based hybrid approach for indoor localization. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[40] Dimitris Milioris, George Tzagkarakis, Artemis Papakonstantinou, Maria Papadopouli, and Panagiotis Tsakalides. Low-dimensional signal-strength fingerprint-based positioning in wireless lans. *Ad hoc networks*, 12:100–114, 2014.

[41] Anindya S Paul and Eric A Wan. Rssi-based indoor localization and tracking using sigma-point kalman smoothers. *IEEE Journal of Selected Topics in Signal Processing*, 3(5):860–873, 2009.

[42] K. Kaemarungsi. Efficient design of indoor positioning systems based on location fingerprinting. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, volume 1, pages 181–186 vol.1, 2005.

[43] Phongsak Prasithsangaree, Prashant Krishnamurthy, and Panos Chrysanthis. On indoor position location with wireless lans. In *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, volume 2, pages 720–724. IEEE, 2002.

[44] Utpal Paul, Anand Kashyap, Ritesh Maheshwari, and Samir R Das. Passive measurement of inter-

ference in wifi networks with application in misbehavior detection. *IEEE transactions on mobile computing*, 12(3):434–446, 2013.

[45] Ritesh Maheshwari, Jing Cao, and Samir R Das. Physical interference modeling for transmission scheduling on commodity wifi hardware. In *INFOCOM 2009, IEEE*, pages 2661–2665. IEEE, 2009.

[46] Lei Lei, Zhangdui Zhong, Chuang Lin, and Xuemin Shen. Operator controlled device-to-device communications in lte-advanced networks. *IEEE Wireless Communications*, 19(3), 2012.

[47] Yanying Gu, Anthony Lo, and Ignas Niemegeers. A Survey of Indoor Positioning Systems for Wireless Personal Networks. *Communications Surveys & Tutorials, IEEE*, 11(1):13–32, 2009.

[48] Rainer Mautz and Sebastian Tilch. Survey of optical indoor positioning systems. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–7. IEEE, 2011.

[49] H Lim, L-C Kung, JC Hou, and H Luo. Zero-configuration, robust indoor localization: Theory and experimentation. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*.

[50] Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen. A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pages 46–51. IEEE, 2007.

[51] Radu Stoleru, Tian He, and John A Stankovic. Range-free localization. In *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, pages 3–31. Springer, 2007.

[52] Filip Lemic, James Martin, Christopher Yarp, Douglas Chan, Vlado Handziski, Robert Brodersen, Gerhard Fettweis, Adam Wolisz, and John Wawrzynek. Localization as a feature of mmwave communication. In *Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1033–1038. IEEE, 2016.

[53] Brian P Crow, Indra Widjaja, LG Kim, and Prescott T Sakai. Ieee 802.11 wireless local area networks. *IEEE Communications magazine*, 35(9):116–126, 1997.

[54] Filip Lemic. Benchmarking of quantile-based indoor fingerprinting algorithm. TKN Technical Report Series TKN-14-001, Telecommunication Networks Group, Technische Universität Berlin, 2014.

[55] Simon Schmitt, Heiko Will, Thomas Hillebrandt, and Marcel Kyas. A virtual indoor localization testbed for wireless sensor networks. In *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pages 239–241. IEEE, 2013.

[56] Stefan Bouckaert, Wim Vandenberghe, Bart Jooris, Ingrid Moerman, and Piet Demeester. The w-ilab. t testbed. In *International Conference on Testbeds and Research Infrastructures*, pages 145–154. Springer, 2010.

[57] Mert Bal, Henry Xue, Weiming Shen, and Hamada Ghenniwa. A test-bed for localization and tracking in wireless sensor networks. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3581–3586, 2009.

[58] R Fish, M Flickinger, and J Lepreau. Mobile emulab: A robotic wireless and sensor network testbed. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[59] Alvin F Martin, Craig S Greenberg, John M Howard, George R Doddington, and John J Godfrey. Nist language recognition evaluation–past and future. In *Proceedings of Odyssey: The speaker and*

language recognition workshop, pages 145–151, 2014.

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[61] Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski, Souvik Sen, Büsch Jasper Lemic, Filip, et al. A realistic evaluation and comparison of indoor location technologies: experiences and lessons learned. In *14th international conference on information processing in sensor networks*, pages 178–189. ACM, 2015.

[62] Dario Salvi, Paolo Barsocchi, Maria Teresa Arredondo, and Juan Pablo Lazaro Ramos. Evaal, evaluating aal systems through competitive benchmarking, the experience of the 1st competition. In *International Competition on Evaluating AAL Systems through Competitive Benchmarking*, pages 14–25. Springer, 2011.

[63] Paolo Barsocchi, Stefano Chessa, Francesco Furfari, and Francesco Potorti. Evaluating aal solutions through competitive benchmarking: the localization competition. *IEEE Pervasive Computing Magazine*, 12(4):72–79, 2013.

[64] Juan Antonio Álvarez-García, Paolo Barsocchi, Stefano Chessa, and Dario Salvi. Evaluation of localization and activity recognition systems for ambient assisted living: The experience of the 2012 evaal competition. *Journal of Ambient Intelligence and Smart Environments*, 5(1):119–132, 2013.

[65] Seng-Yong Lau, Tsung-Han Lin, Te-Yuan Huang, I-Hei Ng, and Polly Huang. A measurement study of zigbee-based indoor localization systems under rf interference. In *Proceedings of the 4th ACM international workshop on Experimental evaluation and characterization*, pages 35–42. ACM, 2009.

[66] Apostolia Papapostolou and Hakima Chaouchi. Rfid-assisted indoor localization and the impact of interference on its performance. *Journal of Network and Computer Applications*, 34(3):902–913, 2011.

[67] Eddie CL Chan, George Baciu, and SC Mak. Properties of channel interference for wi-fi location fingerprinting. *Journal of Communications Software and Systems. Split: Elsevier SCOPUS*, 2010.

[68] Ville Honkavirta, Tommi Perala, Simo Ali-Loytty, and Robert Piché. A comparative survey of wlan location fingerprinting methods. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, pages 243–251. IEEE, 2009.

[69] Shixiong Xia, Yi Liu, Guan Yuan, Mingjun Zhu, and Zhaohui Wang. Indoor fingerprint positioning based on wi-fi: An overview. *ISPRS International Journal of Geo-Information*, 6(5):135, 2017.

[70] Siddhesh Doiphode, JW Bakal, and Madhuri Gedam. Survey of indoor positioning measurements, methods and techniques. *International Journal of Computer Applications*, 140(7), 2016.

[71] Philipp Bolliger. Redpin-adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, pages 55–60. ACM, 2008.

[72] Yifei Jiang, Xin Pan, Kun Li, Qin Lv, Robert P Dick, Michael Hannigan, and Li Shang. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 441–450. ACM, 2012.

[73] AKM Mahtab Hossain, Hien Nguyen Van, Yunye Jin, and Wee-Seng Soh. Indoor localization

using multiple wireless technologies. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–8. IEEE, 2007.

[74] Moustafa Youssef and Ashok Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 205–218. ACM, 2005.

[75] Arsham Farshad, Jiwei Li, Mahesh K Marina, and Francisco J Garcia. A microscopic look at wifi fingerprinting for indoor mobile phone localization in diverse environments. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pages 1–10. IEEE, 2013.

[76] Mohd Nizam Husen and Sukhan Lee. Indoor human localization with orientation using wifi fingerprinting. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, page 109. ACM, 2014.

[77] Vahideh Moghtadaiee and Andrew G Dempster. Wifi fingerprinting signal strength error modeling for short distances. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–6. IEEE, 2012.

[78] Teemu Pulkkinen and Petteri Nurmi. Awesom: Automatic discrete partitioning of indoor spaces for wifi fingerprinting. In *International Conference on Pervasive Computing*, pages 271–288. Springer, 2012.

[79] Reza Farivar, David Wiczer, Alejandro Gutierrez, and Roy H Campbell. A statistical study on the impact of wireless signals' behavior on location estimation accuracy in 802.11 fingerprinting systems. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8. IEEE, 2009.

[80] Gennady Berkovich. Accurate and reliable real-time indoor positioning on commercial smartphones. In *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, pages 670–677. IEEE, 2014.

[81] Guenther Retscher, Eva Moser, Dennis Vredeveld, Dirk Heberling, and Joerg Pamp. Performance and accuracy test of a wifi indoor positioning system. *Journal of Applied Geodesy jag*, 1(2):103–110, 2007.

[82] Qideng Jiang, Yongtao Ma, Kaihua Liu, and Zhi Dou. A probabilistic radio map construction scheme for crowdsourcing-based fingerprinting localization. *IEEE Sensors Journal*, 16(10):3764–3774, 2016.

[83] Valentin Radu, Lito Kriara, and Mahesh K Marina. Pazl: A mobile crowdsensing based indoor wifi monitoring system. In *Network and Service Management (CNSM), 2013 9th International Conference on*, pages 75–83. IEEE, 2013.

[84] Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. Growing an organic indoor location system. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 271–284. ACM, 2010.

[85] He Wang, Souvik Sen, Alexander Mariakakis, Romit Roy Choudhury, Ahmed Elgohary, Moustafa Farid, and Moustafa Youssef. Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 499–500. ACM, 2012.

[86] Zheng Yang, Chenshu Wu, and Yunhao Liu. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proceedings of the 18th annual international conference*

on Mobile computing and networking, pages 269–280. ACM, 2012.

[87] Sungwon Yang, Pralav Dessai, Mansi Verma, and Mario Gerla. Freeloc: Calibration-free crowdsourced indoor localization. In *INFOCOM, 2013 Proceedings IEEE*, pages 2481–2489. IEEE, 2013.

[88] Qi Zhang, Yutian Wen, Xiaohua Tian, Xiaoying Gan, and Xinbing Wang. Incentivize crowd labeling under budget constraint. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 2812–2820. IEEE, 2015.

[89] Martin Klepal, Dirk Pesch, et al. Influence of predicted and measured fingerprint on the accuracy of rssi-based indoor location systems. In *2007 4th Workshop on Positioning, Navigation and Communication*, pages 145–151. IEEE, 2007.

[90] Binghao Li, Yufei Wang, Hyung Keun Lee, Andrew Dempster, and Chris Rizos. Method for yielding a database of location fingerprints in wlan. *IEE Proceedings-Communications*, 152(5):580–586, 2005.

[91] Zhu Dali, Zhang Huihui, and Feng Weimiao. Research on the construction of radio-map based on support vector regression. In *Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2014 Fourth International Conference on*, pages 77–80. IEEE, 2014.

[92] Xiaoyong Chai and Qiang Yang. Reducing the calibration effort for probabilistic indoor location estimation. *IEEE Transactions on Mobile Computing*, 6(6):649–662, 2007.

[93] Sudhir Kumar, Rajesh M Hegde, and Niki Trigoni. Gaussian process regression for fingerprinting based localization. *Ad Hoc Networks*, 51:1–10, 2016.

[94] Wonsun Bong and Yong Cheol Kim. Reconstruction of radio map from sparse rss data by discontinuity preserving smoothing. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, pages 227–231. ACM, 2012.

[95] Minkyu Lee and Dongsoo Han. Voronoi tessellation based interpolation method for wi-fi radio map construction. *IEEE Communications Letters*, 16(3):404–407, 2012.

[96] Colin Childs. Interpolating surfaces in arcgis spatial analyst. *ArcUser, July-September*, 3235, 2004.

[97] Ning Chang, Rashid Rashidzadeh, and Majid Ahmadi. Robust indoor positioning using differential wi-fi access points. *IEEE Transactions on Consumer Electronics*, 56(3), 2010.

[98] Jie Wang, Qinghua Gao, Hongyu Wang, Hongyang Chen, and Minglu Jin. Differential radio map-based robust indoor localization. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):17, 2011.

[99] Arash Behboodi, Niklas Wiström, Filip Lemic, Thiemo Voigt, and Adam Wolisz. Interference effect on localization solutions: signal feature perspective. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–7. IEEE, 2015.

[100] Arash Behboodi, Filip Lemic, and Adam Wolisz. Interference effect on the performance of fingerprinting localization. *8th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017.

[101] Christian Beder, Alan McGibney, and Martin Klepal. Predicting the expected accuracy for fingerprinting based wifi localisation systems. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–6. IEEE, 2011.

[102] Hendrik Lemelson, Mikkel Baun Kjærgaard, Rene Hansen, and Thomas King. Error estimation

for indoor 802.11 location fingerprinting. In *International Symposium on Location-and Context-Awareness*, pages 138–155. Springer, 2009.

[103] Artyom Nikitin, Christos Laoudias, Georgios Chatzimilioudis, Panagiotis Karras, and Demetrios Zeinalipour-Yazti. Indoor localization accuracy estimation from fingerprint data. In *Proceedings of the 18th IEEE International Conference on Mobile Data Management*, 2017.

[104] Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan, and Rijurekha Sen. Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, Mobicom '12, pages 293–304, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1159-5.

[105] Chenshu Wu, Zheng Yang, Yunhao Liu, and Wei Xi. Will: Wireless indoor localization without site survey. *Parallel and Distributed Systems, IEEE Transactions on*, 24(4):839–848, 2013.

[106] William Wei-Liang Li, Ronald A Iltis, and Moe Z Win. A smartphone localization algorithm using rssi and inertial sensor measurement fusion. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 3335–3340. IEEE, 2013.

[107] Feihu Zhang, Hauke Stahle, Guang Chen, Chao Chen, Carsten Simon, and Alois Knoll. A sensor fusion approach for localization with cumulative error elimination. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 1–6. IEEE, 2012.

[108] Sofıa Aparicio, Javier Pérez, Ana M Bernardos, and José R Casar. A fusion method based on bluetooth and wlan technologies for indoor location. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 487–491. IEEE, 2008.

[109] Sofía Aparicio, Javier Pérez, Paula Tarrío, Ana M Bernardos, and José R Casar. An indoor location method based on a fusion map using bluetooth and wlan technologies. In *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*, pages 702–710. Springer, 2009.

[110] Piotr Mirowski, Tin Kam Ho, Saehoon Yi, and Michael MacDonald. Signalslam: Simultaneous localization and mapping with mixed wifi, bluetooth, lte and magnetic signals. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pages 1–10. IEEE, 2013.

[111] Milan Redzic, Conor Brennan, and Noel E O'Connor. Dual-sensor fusion for indoor user localisation. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1101–1104. ACM, 2011.

[112] Alessio De Angelis, Carlo Fischione, and Peter Händel. A sensor fusion algorithm for mobile node localization. *IFAC Proceedings Volumes*, 44(1):11258–11264, 2011.

[113] Michael Kuhn, Cemin Zhang, Brandon Merkl, Depeng Yang, Yazhou Wang, Mohamed Mahfouz, and Aly Fathy. High accuracy uwb localization in dense indoor environments. In *Ultra-Wideband, 2008. ICUWB 2008. IEEE International Conference on*, volume 2, pages 129–132. IEEE, 2008.

[114] Nader Bargshady, Nayef A Alsindi, Kaveh Pahlavan, Yunxing Ye, and Ferit Ozan Akgul. Bounds on performance of hybrid wifi-uwb cooperative rf localization for robotic applications. In *Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010 IEEE 21st International Symposium on*, pages 277–282. IEEE, 2010.

[115] Alessandro Benini, Adriano Mancini, and Sauro Longhi. An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15. 4a wireless sensor net-

work. *Journal of Intelligent & Robotic Systems*, 70(1-4):461–476, 2013.

[116] Youngjune Gwon and Ravi Jain. Error characteristics and calibration-free techniques for wireless lan-based location estimation. In *Proceedings of the second international workshop on Mobility management & wireless access protocols*, pages 2–9. ACM, 2004.

[117] Jaimyoung Kwon, Baris Dundar, and Pravin Varaiya. Hybrid algorithm for indoor positioning using wireless lan. In *Vehicular technology conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 7, pages 4625–4629. IEEE, 2004.

[118] Binghao Li, Andrew Dempster, Chris Rizos, and Joel Barnes. Hybrid method for localization using wlan. In *Spatial Sciences Conference*, pages 341–350. Citeseer, 2005.

[119] Youngjune Gwon, Ravi Jain, and Toshiro Kawahara. Robust indoor location estimation of stationary and mobile users. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1032–1043. IEEE, 2004.

[120] Sylvie Couronné, Niels Hadaschik, Marc Faßbinder, Maarten Weyn, Thomas von der Grun, Martin Klepal, Widyawan Widyawan, and Tim Denis. Locon: a platform for an inter-working of embedded localisation and communication systems. In *2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, pages 1–3. IEEE, 2009.

[121] Sebastian Lempert and Alexander Pflaum. Towards a reference architecture for an integration platform for diverse smart object technologies. *Proceedings of MMS*, pages 53–66, 2011.

[122] Jeffrey Hightower, Barry Brumitt, and Gaetano Borriello. The location stack: A layered model for location in ubiquitous computing. In *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on*, pages 22–28. IEEE, 2002.

[123] Widyawan Widyawan. A platform for an inter-working of embedded localisation and communication systems. *Proceeding LocON –a Platform for an Inter-Working of Embedded Localisation and Communication Systems*, 2009.

[124] Hasari Celebi and Huseyin Arslan. Utilization of location information in cognitive wireless networks. *IEEE Wireless Communications*, 14(4), 2007.

[125] Mikio Hasegawa, Hiroyuki Morikawa, Masugi Inoue, Udana Bandara, Homare Murakami, and Khaled Mahmud. Cross-device handover using the service mobility proxy. *Proceedings of WPMC2003, Yokosuka, Japan*, 2003.

[126] Daniel Camps-Mur, Andres Garcia-Saavedra, and Pablo Serrano. Device-to-device communications with wi-fi direct: overview and experimentation. *IEEE wireless communications*, 20(3):96–104, 2013.

[127] Nandish P Kuruvatti, Andreas Klein, Lianghai Ji, Chan Zhou, Omer Bulakci, Josef Eichinger, Raja Sattiraju, and Hans D Schotten. Robustness of location based d2d resource allocation against positioning errors. In *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, pages 1–6. IEEE, 2015.

[128] Klaus Doppler, Mika Rinne, Carl Wijting, Cássio B Ribeiro, and Klaus Hugl. Device-to-device communication as an underlay to lte-advanced networks. *IEEE Communications Magazine*, 47 (12), 2009.

[129] Salam Doumiati, Hassan Artail, and David M Gutierrez-Estevez. A framework for lte-a proximity-based device-to-device service registration and discovery. *Procedia Computer Science*, 34:87–94,

2014.

[130] Junyi Feng. *Device-to-device communications in LTE-advanced network.* PhD thesis, Télécom Bretagne, Université de Bretagne-Sud, 2013.

[131] Van Sreng, Halim Yanikomeroglu, and David D Falconer. Relayer selection strategies in cellular networks with peer-to-peer relaying. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 3, pages 1949–1953. IEEE, 2003.

[132] Jimmy Jessen Nielsen, Rasmus L Olsen, Tatiana K Madsen, and Hans-Peter Schwefel. Optimized policies for improving fairness of location-based relay selection. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5. IEEE, 2013.

[133] Jimmy Jessen Nielsen, Rasmus L Olsen, Tatiana K Madsen, and Hans-Peter Schwefel. On the impact of information delay on location-based relaying: A markov modeling approach. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 3045–3050. IEEE, 2012.

[134] Jimmy Jessen Nielsen, Tatiana K Madsen, and Hans-Peter Schwefel. Location-based mobile relay selection and impact of inaccurate path loss model parameters. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1–6. IEEE, 2010.

[135] Jimmy Jessen Nielsen, Rasmus Løvenstein Olsen, Tatiana K Madsen, Bernard Uguen, and Hans-Peter Schwefel. Location-quality-aware policy optimisation for relay selection in mobile networks. *Wireless Networks*, 22(2):599–618, 2016.

[136] Filip Lemic, Jasper Büsch, Mikolaj Chwalisz, Vlado Handziski, and Adam Wolisz. Infrastructure for benchmarking rf-based indoor localization under controlled interference. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2014*, pages 26–35. IEEE, 2014.

[137] Filip Lemic, Vlado Handziski, Niklas Wirström, Tom Van Haute, Eli De Poorter, Thiemo Voigt, and Adam Wolisz. Web-based platform for evaluation of rf-based indoor localization algorithms. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 834–840. IEEE, 2015.

[138] Vlado Handziski, Andreas Köpke, Andreas Willig, and Adam Wolisz. TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor network. In *Proc. of the 2nd Intl. Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality, (RealMAN 2006)*, 2006.

[139] Tom Van Haute, Eli De Poorter, Jen Rossey, Ingrid Moerman, Vlado Handziski, Arash Behboodi, Filip Lemic, Adam Wolisz, Niklas Wiström, Thiemo Voigt, Pieter Crombez, Piet Verhoeve, and Jose Javier De Las Heras. The evarilos benchmarking handbook: evaluation of rf-based indoor localization solutions. In *2e International Workshop on Measurement-based Experimental Research, Methodology and Tools (MERMAT 2013)*, pages 1–6, 2013.

[140] Filip Lemic, Vlado Handziski, Adam Wolisz, Timotheos Constambeys, Christos Laoudias, Demetrios Zeinalipour-Yazti, Christos Panayiotou, Stephan Adler, Simon Schmitt, Heiko Will, Thomas Hillebrandt, Marcel Kyas, Yuan Yang, and Yubin Zhao. Evarilos open challenge: track 3. Technical Report TKN-14-005, Telecommunication Networks Group, Technische Universität Berlin, 2014.

[141] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000.

[142] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2926–2931. IEEE, 2001.

[143] Eiman Elnahrawy, Xiaoyan Li, and Richard P Martin. The limits of localization using signal strength: A comparative study. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 406–414. IEEE, 2004.

[144] Filip Lemic, Arash Behboodi, Vlado Handziski, and Adam Wolisz. Experimental decomposition of the performance of fingerprinting-based localization algorithms. In *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, pages 355–364. IEEE, 2014.

[145] Stephan Adler, Simon Schmitt, and Marcel Kyas. Experimental evaluation of the spatial error distribution of indoor localization algorithms. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2014*, pages 44–53. IEEE, 2014.

[146] Jun-geun Park, Dorothy Curtis, Seth Teller, and Jonathan Ledlie. Implications of device diversity for organic localization. In *INFOCOM, 2011 Proceedings IEEE*, pages 3182–3190. IEEE, 2011.

[147] Tom Haute, Eli Poorter, Pieter Crombez, Filip Lemic, Vlado Handziski, Niklas Wirström, Adam Wolisz, Thiemo Voigt, and Ingrid Moerman. Performance analysis of multiple indoor positioning systems in a healthcare environment. *International journal of health geographics*, 15(1):1, 2016.

[148] Tom Van Haute, Eli De Poorter, Filip Lemic, Vlado Handziski, Adam Wolisz, Niklas Wiström, and Thiemo Voigt. Comparability of rf-based indoor localization solutions in heterogeneous environments: an experimental study. *International Journal of Ad Hoc and Ubiquitous Computing, Special Issue on Localization and Positioning for Healthcare Applications*, 2015.

[149] Giuseppe Caso, Luca De Nardis, Filip Lemic, Vlado Handziski, Adam Wolisz, and Maria-Gabriella Di Benedetto. Vifi: Virtual fingerprinting wifi-based indoor positioning via multi-wall multi-floor indoor propagation model. *IEEE Transactions on Mobile Computing*, 2017.

[150] Arash Behboodi, Filip Lemic, and Adam Wolisz. A mathematical model for fingerprinting-based localization algorithms. *IEEE Transactions on Communication*, 2017. Submitted for publication.

[151] Niklas Wirström, Arash Behboodi, Filip Lemic, Thiemo Voigt, and Adam Wolisz. Localization using anonymous measurements. In *2015 International Conference on Distributed Computing in Sensor Systems*, pages 137–146. IEEE, 2015.

[152] Zhuoling Xiao, Hongkai Wen, Andrew Markham, and Niki Trigoni. Robust indoor positioning with lifelong learning. *IEEE Journal on Selected Areas in Communications*, 33(11):2287–2301, 2015.

[153] Zhuoling Xiao, Hongkai Wen, Andrew Markham, and Niki Trigoni. Robust pedestrian dead reckoning (r-pdr) for arbitrary mobile device placement. In *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, pages 187–196. IEEE, 2014.

[154] Theodore S Rappaport, Shu Sun, Rimma Mayzus, Hang Zhao, Yaniv Azar, Kevin Wang, George N Wong, Jocelyn K Schulz, Mathew Samimi, and Felix Gutierrez. Millimeter wave mobile communications for 5g cellular: It will work! *IEEE access*, 1:335–349, 2013.

[155] Ekram Hossain and Monowar Hasan. 5g cellular: key enabling technologies and research challenges. *IEEE Instrumentation & Measurement Magazine*, 18(3):11–21, 2015.

[156] Eldad Perahia, Carlos Cordeiro, Minyoung Park, and L Lily Yang. Ieee 802.11 ad: Defining the next generation multi-gbps wi-fi. In *Consumer Communications and Networking Conference (CCNC),*

2010 7th IEEE, pages 1–5. IEEE, 2010.

[157] Esmond Mok and Günther Retscher. Location determination using wifi fingerprinting versus wifi trilateration. *Journal of Location Based Services*, 1(2):145–159, 2007.

[158] Tom Van Haute, Eli De Poorter, Filip Lemic, Vlado Handziski, Niklas Wirström, Thiemo Voigt, Adam Wolisz, and Ingrid Moerman. Platform for benchmarking of rf-based indoor localization solutions. *IEEE Communications Magazine*, 53(9):126–133, 2015.

[159] Ahmed Abdelhadi, Felipe Rechia, Arvind Narayanan, Thiago Teixeira, Ricardo Lent, Driss Benhaddou, Hyunwoo Lee, and T Charles Clancy. Position estimation of robotic mobile nodes in wireless testbed using geni. *arXiv preprint arXiv:1511.08936*, 2015.

[160] Xiaohua Tian, Ruofei Shen, Duowen Liu, Yutian Wen, and Xinbing Wang. Performance analysis of rss fingerprinting based indoor localization. *IEEE Transactions on Mobile Computing*, 2016.

[161] N. Moayeri, M. O. Ergin, F. Lemic, V. Handziski, and A. Wolisz. Perfloc (part 1): An extensive data repository for development of smartphone indoor localization apps. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7, 2016.

[162] Nader Moayeri, Mustafa Onur Ergin, Filip Lemic, Vlado Handziski, and Adam Wolisz. PerfLoc: A Comprehensive Repository of Experimental Data for Evaluation of Smartphone Indoor Localization Apps, 2016. 2nd KuVS Expert Talk on Localization.

[163] Filip Lemic, Arash Behboodi, Vlado Handziski, and Adam Wolisz. Increasing interference robustness of wifi fingerprinting by leveraging spectrum information. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 1200–1208. IEEE, 2015.

[164] Filip Lemic, Vlado Handziski, Giuseppe Caso, Luca De Nardis, and Adam Wolisz. Enriched training database for improving the wifi rssi-based indoor fingerprinting performance. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 875–881. IEEE, 2016.

[165] Filip Lemic, Vlado Handziski, Giuseppe Caso, Pieter Crombez, Luca De Nardis, Adam Wolisz, Tom Van Haute, and Eli De Poorter. Poster: Toward extrapolation of wifi fingerprinting performance across environments. In *17th International Workshop on Mobile Computing Systems and Applications*. ACM, 2016.

[166] Binghao Li, James Salter, Andrew G Dempster, and Chris Rizos. Indoor positioning techniques based on wireless lan. In *LAN, first IEEE international conference on wireless broadband and ultra wideband communications*. Citeseer, 2006.

[167] Insoo Sohn. Localization performance analysis of knn in ieee 802.11 tgn channel. In *International Conference on ICT Convergence (ICTC)*, pages 219–220. IEEE, 2011.

[168] Franz Aurenhammer. Voronoi diagrams: a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[169] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.

[170] E Damosso and LM Correia. Digital mobile radio towards future generation systems-cost action 231 final report. *European Commission, Luxembourg, Belgium*, (0.2):0–4, 1999.

[171] Giuseppe Caso and Luca De Nardis. On the applicability of multi-wall multi-floor propagation models to wifi fingerprinting indoor positioning. In *Future Access Enablers of Ubiquitous and Intelligent Infrastructures*, pages 166–172. Springer, 2015.

[172] Upkar Varshney and Ron Vetter. Emerging mobile and wireless networks. *Communications of the ACM*, 43(6):73–81, 2000.

[173] SG Mallat and Gabriel Peyré. A wavelet tour of signal processing. 2009.

[174] Tevfik Yucek and Huseyin Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE communications surveys & tutorials*, 11(1):116–130, 2009.

[175] Jan Rabaey, Adam Wolisz, Ali Özer Ercan, Alvaro Araujo, Fred Burghardt, Samah Mustafa, Arash Parsa, Sofie Pollin, I-Hsiang Wang, and Pedro Malagon. Connectivity brokerageâĂŤenabling seamless cooperation in wireless networks: a white paper. *Berkeley Wireless Research Center*, 2010.

[176] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review*, 41 (1):53–53, 2011.

[177] Andy Bavier, Nick Feamster, Mark Huang, Larry Peterson, and Jennifer Rexford. In vini veritas: realistic and controlled network experimentation. *ACM SIGCOMM Computer Communication Review*, 36(4):3–14, 2006.

[178] José Hernández-Orallo, David L Dowe, Sergio España-Cubillo, M Victoria Hernández-Lloreda, and Javier Insa-Cabrera. On more realistic environment distributions for defining, evaluating and developing intelligence. In *International Conference on Artificial General Intelligence*, pages 82–91. Springer, 2011.

[179] Shuai Che and Kevin Skadron. Benchfriend: correlating the performance of gpu benchmarks. *International Journal of High Performance Computing Applications*, 28(2):238–250, 2014.

[180] Aashish Shreedhar Phansalkar. *Measuring program similarity for efficient benchmarking and performance analysis of computer systems*. ProQuest, 2007.

[181] Robin L Plackett. Karl pearson and the chi-squared test. *International Statistical Review/Revue Internationale de Statistique*, pages 59–72, 1983.

[182] Jacob Cohen. A power primer. *Psychological bulletin*, 112(1):155, 1992.

[183] VS Abhayawardhana, IJ Wassell, D Crosby, MP Sellars, and MG Brown. Comparison of empirical propagation path loss models for fixed wireless access systems. In *2005 IEEE 61st Vehicular Technology Conference*, volume 1, pages 73–77. IEEE, 2005.

[184] Ólafur R Helgason, Emre A Yavuz, Sylvia T Kouyoumdjieva, Ljubica Pajevic, and Gunnar Karlsson. A mobile peer-to-peer system for opportunistic content-centric networking. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, pages 21–26. ACM, 2010.

[185] Sacha Trifunovic, Bernhard Distl, Dominik Schatzmann, and Franck Legendre. Wifi-opp: ad-hocless opportunistic networking. In *Proceedings of the 6th ACM workshop on Challenged networks*, pages 37–42. ACM, 2011.

[186] Joseph Mitola and Gerald Q Maguire. Cognitive radio: making software radios more personal. *IEEE personal communications*, 6(4):13–18, 1999.

[187] Ian F Akyildiz, Won-Yeol Lee, Mehmet C Vuran, and Shantidev Mohanty. Next generation/dynamic

spectrum access/cognitive radio wireless networks: A survey. *Computer networks*, 50(13):2127–2159, 2006.

[188] Shravan Rayanchu, Ashish Patro, and Suman Banerjee. Airshark: detecting non-wifi rf devices using commodity wifi hardware. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 137–154. ACM, 2011.

[189] Tan Zhang, Ashish Patro, Ning Leng, and Suman Banerjee. A wireless spectrum analyzer in your pocket. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 69–74. ACM, 2015.

[190] Filip Lemic, Vlado Handziski, Giuseppe Caso, Pieter Crombez, Luca De Nardis, Adam Wolisz, Tom Van Haute, and Eli De Poorter. Toward extrapolation of wifi fingerprinting performance across environments. In *17th International Workshop on Mobile Computing Systems and Applications*, pages 69–74. ACM, 2016.

[191] Mahesh K Marina, Valentin Radu, and Konstantinos Balampekos. Impact of indoor-outdoor context on crowdsourcing based mobile coverage analysis. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, pages 45–50. ACM, 2015.

[192] Carlos Pereira, Ludimar Guenda, and Nuno Borges Carvalho. A smart-phone indoor/outdoor localization system. In *International conference on indoor positioning and indoor navigation (IPIN)*, pages 21–23, 2011.

[193] Masakatsu Kourogi, Nobuchika Sakata, Takashi Okuma, and Takeshi Kurata. Indoor/outdoor pedestrian navigation with an embedded gps/rfid/self-contained sensor system. *Advances in Artificial Reality and Tele-Existence*, pages 1310–1321, 2006.

[194] Kai Lingemann, Hartmut Surmann, Andreas Nuchter, and Joachim Hertzberg. Indoor and outdoor localization for fast mobile robots. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2185–2190. IEEE, 2004.

[195] Savvas Papaioannou, Hongkai Wen, Andrew Markham, and Niki Trigoni. Fusion of radio and camera sensor data for accurate indoor positioning. In *Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on*, pages 109–117. IEEE, 2014.

[196] Thomas Kleine-Ostmann and Amy E Bell. A data fusion architecture for enhanced position estimation in wireless networks. *IEEE communications letters*, 5(8):343–345, 2001.

[197] Filip Lemic, Vlado Handziski, Nitesh Mor, Jan Rabaey, John Wawrzynek, and Adam Wolisz. Toward standardized localization service. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2016.

[198] Filip Lemic, Vlado Handziski, Mladen Miksa, Jan Rabaey, John Wawrzynek, and Adam Wolisz. Selection and aggregation of location information provisioning services. *IEEE International Conference on Communications (ICC'17)*, 2017.

[199] Filip Lemic, Vlado Handziski, Ivan Arturo Azcarate Rodriguez, John Wawrzynek, Jan Rabaey, and Adam Wolisz. Design, implementation, and evaluation of the standardized localization service. *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'17)*, 2017.

[200] Jose A Castellanos and Juan D Tardos. *Mobile robot localization and map building: A multisensor fusion approach*. Springer Science & Business Media, 2012.

[201] André Coelho, Filipe Meneses, and Rui José. Delay-tolerant positioning for location-based log-

ging with mobile devices. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 129–136. Springer, 2012.

[202] Christoph Fuchs, Nils Aschenbruck, Peter Martini, and Monika Wieneke. Indoor tracking for mission critical scenarios: A survey. *Pervasive and Mobile Computing*, 7(1):1–15, 2011.

[203] Jeffrey Hightower and Gaetano Borriello. A Survey and Taxonomy of Location Systems for Ubiquitous Computing. *IEEE Computer*, 34(8):57–66, 2001.

[204] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.

[205] Quentin Jones, Sukeshini A Grandhi, Steve Whittaker, Keerti Chivakula, and Loren Terveen. Putting systems into place: a qualitative study of design requirements for location-aware community systems. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 202–211. ACM, 2004.

[206] Rainer Mautz. Indoor positioning technologies. 2012. PhD Dissertation.

[207] MFS van der Ham, S Zlatanova, E Verbree, and R Voûte. *Real Time Localization of Assets in Hospitals using Quuppa Indoor Positioning Technology*. MasterâĂŹs thesis, TU Delft, 2015.

[208] Jan M Rabaey. The swarm at the edge of the cloud-a new perspective on wireless. In *VLSI Circuits (VLSIC), 2011 Symposium on*, pages 6–8. IEEE, 2011.

[209] Alastair R Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55, 2003.

[210] Nitesh Mor, Ben Zhang, John Kolb, Douglas S Chan, Nikhil Goyal, Nicholas Sun, Ken Lutz, Eric Allman, John Wawrzynek, Edward A Lee, et al. Toward a global data infrastructure. *IEEE Internet Computing*, 20(3):54–62, 2016.

[211] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[212] Richard Harper. *Inside the smart home*. Springer Science & Business Media, 2006.

[213] Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. The smart thermostat: using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 211–224. ACM, 2010.

[214] Ehab Mahmoud Mohamed, Kei Sakaguchi, and Seiichi Sampei. Millimeter wave beamforming based on wifi fingerprinting in indoor environment. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1155–1160. IEEE, 2015.

[215] Jimmy Jessen Nielsen. *Location based network optimizations for mobile wireless networks-a study of the impact of mobility and inaccurate information*. PhD thesis, Aalborg university, 2011.

[216] Marvin M Theimer, Michael J Spreitzer, Mark D Weiser, Richard J Goldstein, Daniel C Swinehart, William N Schilit, and Roy Want. Specifying and establishing communication data paths between particular media devices in multiple media device computing systems based on context of a user or users, September 22 1998. US Patent 5,812,865.

[217] Roy Friedman, Alex Kogan, and Yevgeny Krivolapov. On power and throughput tradeoffs of wifi

and bluetooth in smartphones. *IEEE Transactions on Mobile Computing*, 12(7):1363–1376, 2013.

[218] Filip Lemic, Arash Behboodi, Vlado Handziski, Anatolij Zubow, and Adam Wolisz. Location-based decision-making mechanism for device-to-device link establishment. *86th Vehicular Technology Conference (VTC2017-Fall)*, 2017.

[219] Filip Lemic, Arash Behboodi, Vlado Handziski, Anatolij Zubow, and Adam Wolisz. Location-based mechanism for positioning of a mobile relay. *13th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (ACM Q2SWinet)*, 2017.

[220] Cisco. Cisco visual networking index: global mobile data traffic forecast update, 2014–2019. *Tech. Rep*, 2015.

[221] Arash Asadi, Qing Wang, and Vincenzo Mancuso. A survey on device-to-device communication in cellular networks. *IEEE Communications Surveys & Tutorials*, 16(4):1801–1819, 2014.

[222] Vinita Daiya, Jemimah Ebenezer, SAV Satya Murty, and Baldev Raj. Experimental analysis of rssi for distance and position estimation. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pages 1093–1098. IEEE, 2011.

[223] Giovanni Zanca, Francesco Zorzi, Andrea Zanella, and Michele Zorzi. Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks. In *Proceedings of the workshop on Real-world wireless sensor networks*, pages 1–5. ACM, 2008.

[224] Mohammud Z Bocus, Carl P Dettmann, and Justin P Coon. An approximation of the first order marcum q-function with application to network connectivity analysis. *IEEE Communications Letters*, 17(3):499–502, 2013.

[225] Andrea Borrelli, Cristiano Monti, Marco Vari, and Franco Mazzenga. Channel models for ieee 802.11 b indoor system design. In *Communications, 2004 IEEE International Conference on*, volume 6, pages 3701–3705. IEEE, 2004.

[226] Yasamin Mostofi, Alejandro Gonzalez-Ruiz, Alireza Gaffarkhah, and Ding Li. Characterization and modeling of wireless channels for networked robotic and control systems-a comprehensive overview. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4849–4854. IEEE, 2009.

[227] Chenshu Wu, Zheng Yang, Chaowei Xiao, Chaofan Yang, Yunhao Liu, and Mingyan Liu. Static power of mobile devices: Self-updating radio maps for wireless indoor localization. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 2497–2505. IEEE, 2015.

[228] Ralf Pabst, Bernhard H Walke, Daniel C Schultz, Patrick Herhold, Halim Yanikomeroglu, Sayandev Mukherjee, Harish Viswanathan, Matthias Lott, Wolfgang Zirwas, Mischa Dohler, et al. Relay-based deployment concepts for wireless and mobile broadband radio. *IEEE Communications Magazine*, 42(9):80–89, 2004.

[229] Mahanth Gowda, Ashutosh Dhekne, and Romit Roy Choudhury. The case for robotic wireless networks. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1317–1327. International World Wide Web Conferences Steering Committee, 2016.

[230] Aggelos Bletsas, Ashish Khisti, David P Reed, and Andrew Lippman. A simple cooperative diversity method based on network path selection. *IEEE Journal on selected areas in communications*, 24(3):659–672, 2006.

[231] Magnus Lindhé and Karl Henrik Johansson. Using robot mobility to exploit multipath fading. *IEEE Wireless Communications*, 16(1):30–37, 2009.

[232] Marc S. Paolella. *Intermediate probability: a computational approach*. John Wiley, Chichester, England ; Hoboken, NJ, 2007. ISBN 978-0-470-02637-3.

[233] Masahiro Hiyama, Elis Kulla, Tetsuya Oda, Makoto Ikeda, and Leonard Barolli. Application of a manet testbed for horizontal and vertical scenarios: performance evaluation using delay and jitter metrics. *Human-centric Computing and Information Sciences*, 1(1):3, 2011.

[234] Hongyi Wu, Chunming Qiao, Swades De, and Ozan Tonguz. Integrated cellular and ad hoc relaying systems: icar. *IEEE Journal on Selected areas in Communications*, 19(10):2105–2115, 2001.

[235] Min Zhou, Qimei Cui, Riku Jantti, and Xiaofeng Tao. Energy-efficient relay selection and power allocation for two-way relay channel with analog network coding. *IEEE Communications Letters*, 16(6):816–819, 2012.

[236] Ritesh Madan, Neelesh B Mehta, Andreas F Molisch, and Jin Zhang. Energy-efficient cooperative relaying over fading channels with simple relay selection. *IEEE Transactions on Wireless Communications*, 7(8), 2008.

[237] Amitav Mukherjee, S Ali A Fakoorian, Jing Huang, and A Lee Swindlehurst. Principles of physical layer security in multiuser wireless networks: A survey. *IEEE Communications Surveys & Tutorials*, 16(3):1550–1573, 2014.

[238] Hongkai Wen, Zhuoling Xiao, Niki Trigoni, and Phil Blunsom. On assessing the accuracy of positioning systems in indoor environments. In *International Conference on Embedded Wireless Systems and Networks (EWSN)*, pages 1–17. Springer, 2013.

[239] Hongkai Wen, Zhuoling Xiao, Andrew Markham, and Niki Trigoni. Accuracy estimation for sensor systems. *IEEE Transactions on Mobile Computing*, 14(7):1330–1343, 2015.

[240] Ashutosh Dhekne, Mahanth Gowda, and Romit Roy Choudhury. Extending cell tower coverage through drones. In *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, pages 7–12. ACM, 2017.

[241] Christos Laoudias, Georgios Larkou, Demetrios Zeinalipour-Yazti, and Christos G Panayiotou. Airplace: Indoor geolocation on smartphones through wifi fingerprinting. *Mobile Computing*, page 37.

[242] Heiko Will, Thomas Hillebrandt, and Marcel Kyas. The geo-n localization algorithm. In *Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1âĂŞ–10, 2012.

[243] Yuan Yang, Yubin Zhao, and Marcel Kyas. Rbgf: recursively bounded grid-based filter for indoor position tracking using wireless networks. *IEEE Communications Letters*, 18(7):1234–1237, 2014.

[244] S Adler, S Schmitt, and M Kyas. 3com (3 centers of mass) indoor localization algorithm. *EVARILOS Open Challenge, Short paper*, 2014.

[245] Larry C Andrews and Ronald L Phillips. *Mathematical techniques for engineers and scientists*, volume 118. Spie Press, 2003.

[246] S. M. Moser. Expectations of a noncentral chi-square distribution with application to IID MIMO Gaussian fading. In *2008 International Symposium on Information Theory and Its Applications*, pages 1–6, December 2008.

Publications

Parts of this dissertation have been published in the following publications.

Journal, Conference, and Workshop Proceedings

- Filip Lemic, Jasper Büsch, Mikolaj Chwalisz, Vlado Handziski, and Adam Wolisz. “Infrastructure for benchmarking rf-based indoor localization under controlled interference”. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2014, pages 26-35. IEEE, 2014
- Filip Lemic, Vlado Handziski, Niklas Wirström, Tom Van Haute, Eli De Poorter, Thiemo Voigt, and Adam Wolisz. “Web-based platform for evaluation of rf-based indoor localization algorithms”. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 834-840. IEEE, 2015
- Filip Lemic, Vlado Handziski, Adam Wolisz, Timotheos Constambeys, Christos Laoudias, Stephan Adler, Simon Schmitt, and Yuan Yang. “Experimental evaluation of rf-based indoor localization algorithms under rf interference”. In *2015 International Conference on Location and GNSS (ICL-GNSS)*, pages 1-8. IEEE, 2015
- Filip Lemic, Arash Behboodi, Vlado Handziski, and Adam Wolisz. “Experimental decomposition of the performance of fingerprinting-based localization algorithms”. In *5th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 355-364. IEEE, 2014
- Arash Behboodi, Niklas Wiström, Filip Lemic, Thiemo Voigt, and Adam Wolisz. “Interference effect on localization solutions: signal feature perspective”. In *2015 IEEE 81st Vehicular Technology Conference (VTC-Spring)*, pages 1-7. IEEE, 2015
- Filip Lemic, Arash Behboodi, Vlado Handziski, and Adam Wolisz. “Increasing interference robustness of wifi fingerprinting by leveraging spectrum information”. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 1200-1208. IEEE, 2015
- Filip Lemic, Vlado Handziski, Giuseppe Caso, Luca De Nardis, and Adam Wolisz. “Enriched training database for improving the wifi rssi-based indoor fingerprinting performance”. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 875-881. IEEE, 2016
- Filip Lemic, Vlado Handziski, Giuseppe Caso, Pieter Crombez, Luca De Nardis, Adam Wolisz, Tom Van Haute, and Eli De Poorter. “Toward extrapolation of wifi fingerprinting performance across environments”. In *17th International Workshop on Mobile Computing Systems and Applications (ACM HotMobile)*, pages 69-74. ACM, 2016
- Filip Lemic, Vlado Handziski, Nitesh Mor, Jan Rabaey, John Wawrzynek, and Adam Wolisz. “Toward standardized localization service”. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1-8, 2016

- Filip Lemic, Vlado Handziski, Mladen Miksa, Jan Rabaey, John Wawrzynek, and Adam Wolisz. "Selection and aggregation of location information provisioning services". In *26th International Conference on Computer Communications and Networks (ICCCN)*, 2017.
- Filip Lemic, Vlado Handziski, Ivan Azcarate, John Wawrzynek, Jan Rabaey, and Adam Wolisz. "Design, implementation, and evaluation of the standardized localization service". In *8th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017.
- Filip Lemic, Arash Behboodi, Vlado Handziski, Anatolij Zubow, and Adam Wolisz. "Location-based decision-making mechanism for device-to-device link establishment". In *86th Vehicular Technology Conference (VTC2017-Fall)*, 2017.
- Filip Lemic, Arash Behboodi, Vlado Handziski, Anatolij Zubow, and Adam Wolisz. "Location-based Decision-making Mechanism for Relay Usage and Selection". In *13th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (ACM Q2SWinet)*, 2017. Accepted for publication

Technical Reports

- Filip Lemic. "Benchmarking of quantile-based indoor fingerprinting algorithm". TKN Technical Report Series TKN-14-001, Telecommunication Networks Group, Technische Universität Berlin, 2014
- Filip Lemic and Vlado Handziski. "Data management services for evaluation of rf-based indoor localization". TKN Technical Report Series TKN-14-002, Telecommunication Networks Group, Technische Universität Berlin, 2014
- Filip Lemic. "Service for calculation of performance metrics of indoor localization benchmarking experiments". Technical Report TKN-14-003, Telecommunication Networks Group, Technische Universität Berlin, 2014
- Filip Lemic, Vlado Handziski, Adam Wolisz,Timotheos Constambeys, Christos Laoudias, Demetrios Zeinalipour-Yazti, Christos Panayiotou, Stephan Adler, Simon Schmitt, Heiko Will, Thomas Hillebrandt, Marcel Kyas, Yuan Yang, and Yubin Zhao. "Evarilos open challenge: track 3". Technical Report TKN-14-005, Telecommunication Networks Group, Technische Universität Berlin, 2014

Posters and Demonstration Abstracts

- Filip Lemic, Vlado Handziski, Anatolij Zubow, Adam Wolisz. "Poster Abstract: Standardized Localization Service". In *15th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2017
- Filip Lemic, Vlado Handziski, Giuseppe Caso, Pieter Crombez, Luca De Nardis, Adam Wolisz, Tom Van Haute, and Eli De Poorter. "Poster: Toward extrapolation of wifi fingerprinting performance across environments". In *17th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. ACM, 2016
- Filip Lemic, Vlado Handziski, Niklas Wirström, Tom Van Haute, Eli De Poorter, Thiemo Voigt, and Adam Wolisz. "Demo abstract: virtual experimental evaluation of rf-based indoor localization algorithms". In *European Conference on Wireless sensor networks (EWSN2015)*, pages 1-3, 2015
- Filip Lemic, Jasper Büsch, Mikolaj Chwalisz, Vlado Handziski, and Adam Wolisz. "Demo abstract: testbed infrastructure for benchmarking rf-based indoor localization solutions under controlled interference". In *11th European Conference on Wireless Sensor Networks (EWSN2014)*, 2014

Additional Publications

During the course of my doctorate study the following contributions have also been made, in addition to the ones that directly contributed to the dissertation.

- Filip Lemic, James Martin, Christopher Yarp, Douglas Chan, Vlado Handziski, Robert Brodersen, Gerhard Fettweis, Adam Wolisz, and John Wawrzynek. "Localization as a feature of mmwave communication". In *Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1033-1038. IEEE, 2016
- Filip Lemic, Ivan Azcarate, Vlado Handziski, Anatolij Zubow, Adam Wolisz. "Talk Proposal: Standardized Localization Interface". 2017 GeoIT WhereCamp Conference, 2017
- Filip Lemic. "Poster: Localization as a feature of mmWave communication". Berkeley Wireless Research Center (BWRC) Summer 2016 Retreat, 2016
- Arash Behboodi, Filip Lemic, and Adam Wolisz. "A mathematical model for fingerprinting-based localization algorithms". In *IEEE Transactions on Mobile Computing* 2017. Submitted for publication
- Arash Behboodi, Filip Lemic, and Adam Wolisz. "Hypothesis testing based model for fingerprinting localization algorithms". In *2017 IEEE 85th Vehicular Technology Conference (VTC2017-Spring)*, 2017.
- Arash Behboodi, Filip Lemic, and Adam Wolisz. "Interference effect on the performance of fingerprinting localization". In *8th International Conference on Indoor Positioning and Indoor Navigation (IPIN)* 2017.
- Giuseppe Caso, Luca De Nardis, Filip Lemic, Vlado Handziski, Adam Wolisz, and Maria-Gabriella Di Benedetto. "Vifi: Virtual fingerprinting wifi-based indoor positioning via multi-wall multi-floor indoor propagation model". In *IEEE Transactions on Mobile Computing*, 2017. Submitted for publication
- Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski, Souvik Sen, Jasper Büsch, Filip Lemic et al. "A realistic evaluation and comparison of indoor location technologies: experiences and lessons learned". In *14th international conference on information processing in sensor networks*, pages 178-189. ACM, 2015
- Nader Moayeri, Mustafa Onur Ergin, Filip Lemic, Vlado Handziski, and Adam Wolisz. "Perfloc (part 1): An extensive data repository for development of smartphone indoor localization apps". In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1-7, 2016
- Tom Van Haute, Eli De Poorter, Filip Lemic, Vlado Handziski, Niklas Wirström, Thiemo Voigt, Adam Wolisz, and Ingrid Moerman. "Platform for benchmarking of rf-based indoor localization solutions". In *IEEE Communications Magazine*, 53(9):126-133, 2015
- Tom Haute, Eli Poorter, Pieter Crombez, Filip Lemic, Vlado Handziski, Niklas Wirström, Adam Wolisz, Thiemo Voigt, and Ingrid Moerman. "Performance analysis of multiple indoor positioning systems in a healthcare environment". In *International journal of health geographics*, 15(1):1, 2016
- Tom Van Haute, Eli De Poorter, Filip Lemic, Vlado Handziski, Adam Wolisz, Niklas Wiström, and Thiemo Voigt. "Comparability of rf-based indoor localization solutions in heterogeneous environments: an experimental study". In *International Journal of Ad Hoc and Ubiquitous Computing, Special Issue on Localization and Positioning for Healthcare Applications*, 2015

- Nader Moayeri, Mustafa Onur Ergin, Filip Lemic, Vlado Handziski, and Adam Wolisz. "PerfLoc: A Comprehensive Repository of Experimental Data for Evaluation of Smartphone Indoor Localization Apps", 2nd KuVS Expert Talk on Localization, 2016
- Niklas Wirström, Arash Behboodi, Filip Lemic, Thiemo Voigt, and Adam Wolisz. "Localization using anonymous measurements". In *2015 International Conference on Distributed Computing in Sensor Systems*, pages 137-146. IEEE, 2015
- Tom Van Haute, Eli De Poorter, Jen Rossey, Ingrid Moerman, Vlado Handziski, Arash Behboodi, Filip Lemic, Adam Wolisz, Niklas Wiström, Thiemo Voigt, Pieter Crombez, Piet Verhoeve, and Jose Javier De Las Heras. "The evarilos benchmarking handbook: evaluation of rf-based indoor localization solutions". In *2e International Workshop on Measurement-based Experimental Research, Methodology and Tools (MERMAT 2013)*, pages 1-6, 2013
- Tom Van Haute, Eli De Poorter, Filip Lemic, Vlado Handziski, Niklas Wiström, Adam Wolisz, and Ingrid Moerman. "Demonstration abstract: platform for benchmarking rf-based indoor localization solutions". In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1-2. IEEE, 2016
- Arash Behboodi, Pieter Crombez, Jose Javier De Las Heras, Eli De Poorter, Vlado Handziski, Filip Lemic, Ingrid Moerman, Tom Van Haute, Piet Verhoeve, Thiemo Voigt, Niklas Wirström, and Adam Wolisz. Poster: "Evaluation of rf-based indoor localization solutions for the future internet". In *The Future Network and Mobile Summit (FuNeMS'13)*, 2013