Telecommunication Networks Group

Technical University Berlin
Telecommunication Networks Group

Systematic Objective Evaluation of RF-based Indoor Localization Algorithms

EVARILOS Open Challenge: Track 3

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, Yubin Zhao

{lemic, handziski, wolisz}@tkn.tu-berlin.de

Berlin, October 2014

TKN Technical Report TKN-14-05

TKN Technical Reports Series
Editor: Prof. Dr.-Ing. Adam Wolisz

Abstract

This document presents systematic evaluation of Radio Frequency (RF)-based indoor localization algorithms performed in the scope of the Track 3 of the EVARILOS Open Challenge. The document begins with describing the environment in which the evaluation of indoor localization algorithms was performed. It continues with description of hardware components that were used for various functions necessary for objective benchmarking of competitors' algorithms. Namely, the document presents hardware components for autonomous mobility, interference generation and monitoring and nodes that were used as part of the indoor localization System Under Test (SUT). The document further describes the evaluation scenarios in which different types of interference were generated in order to evaluate the influence of RF interference on competitors' algorithms. Moreover, the document describes the evaluation procedure and metrics that were used to characterize the performance of different SUTs. The document also presents results of the competition, firstly for each competitor separately, and finally in the summarized way.

Keywords:

TKN testbed infrastructure, TKN environment, autonomous mobility, interference generation, interference monitoring, indoor localization, benchmarking, EVARILOS Open Challenge, experiments with RF interference, fingerprinting, low-power sensor nodes

Acknowledgments:

This work has been partially funded by the European Commission (FP7-ICT-FIRE) within the project EVARILOS (grant No. 317989). The author Filip Lemic was partially supported by DAAD (German Academic Exchange Service).

Contents

List of Acronyms

AP Access Point

CS Carrier Sensing

CSMA Carrier Sense Multiple Access

EBH EVARILOS Benchmarking Handbook

EBP EVARILOS Benchmarking Platform

kNN k-Nearest Neighbours

LTE Long Term Evolution

OMF cOntrol and Management Framework

OS Operating System

PC Personal Computer

RF Radio Frequency

ROS Robot Operating System

RSS Received Signal Strength

RSSI Received Signal Strength Indicator

SSH Secure Shell

SUT System Under Test

TCP Transmission Control Protocol

TKN Telecommunication Network Group

TUB Technische Universität Berlin

TWIST TKN Wireless Indoor Sensor Network Testbed

UDP User Datagram Protocol

URI Uniform Resource Identifier

VPN Virtual Private Network

WiFi Wireless Fidelity

WMP Wireless Mac Processor

WLAN Wireless Local Area Network

Chapter 1

Introduction

This document describes the Track 3 of the EVARILOS Open Challenge, which aimed on objective and comparable evaluation of RF-based indoor localization algorithms. Track 3 of the challenge encouraged researchers to deploy and evaluate their localization algorithms on the hardware that was provided in the TKN testbed environment.

This document is structured as follows. Chapter 2 describes the TKN testbed environment, while Chapter 3 gives the short generic description of different components that are part of the testbed infrastructure. Chapter 4 describes how those generic components were instantiated for specific purpose of experimental evaluation and benchmarking of RF based indoor localization algorithms. Chapter 5 gives detailed instructions for competitors of Track 3 of the challenge, ranging from requirements for the SUTs algorithms to the guidelines for using different infrastructural nodes. Chapter 6 describes four evaluation scenarios, namely the reference scenario and three interference scenarios. Chapter 7 gives the description of the evaluation procedure, together with all metrics that were used for evaluation, how these metrics were calculated and how the final scores were produced. Chapter 8 shortly describes the evaluated algorithms and results per algorithm, while Chapter 9 presents summarized results of Track 3 of the challenge. Chapter 10 concludes the document and summarizes the lessons learned.

Chapter 2

TKN Testbed Environment

Telecommunication Network Group (TKN) testbed is located at the 2^{nd}, 3^{rd} and 4^{th} floor of the Telecommunication Network Group building in Berlin. According to the EVARILOS Benchmarking Handbook (EBH) [1], TKN testbed environment can be characterized as “Big” with “’Brick walls”, i.e. more than 1500 m^2 area with more than 50 rooms. TKN testbed is an office environment mostly comprising of three types of rooms, namely small offices ($14m^2$), big offices ($28m^2$) and laboratories ($42m^2$), as shown in Figure 2.2. Furthermore, TKN testbed is a dynamic environment, meaning that there was a number of people moving in the premises and constant opening of doors or slight movements of infrastructure (chairs, tables, etc.) were expected. Furthermore, uncontrolled wireless interference typical for office environments was expected during the evaluation, as presented in Figure 2.1.

TYPES OF INTERFERENCE SOURCE	
Microwave	✓
WiFi	✓
DECT	✓
Bluetooth	✓
Zigbee	✓

Figure 2.1: Sources of uncontrolled interference

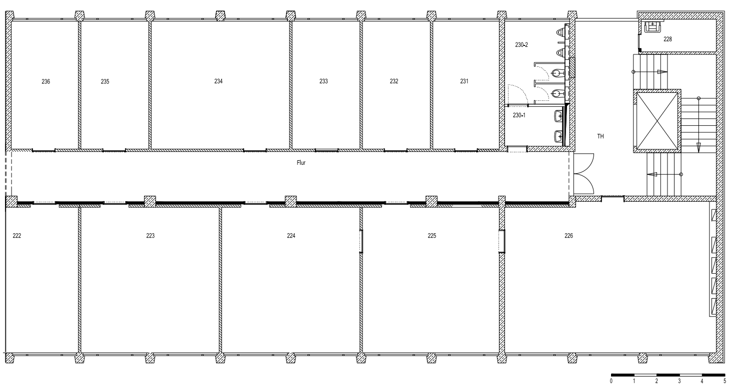

Figure 2.2: Footprints of the 2^{nd} floor of the TKN testbed environment

Chapter 3

TKN Hardware Overview

This chapter gives a short generic description of different types of hardware that are part of the TKN infrastructure and are currently available for experimentation in TKN testbed, namely:

- TKN Wireless Indoor Sensor Network Testbed;
- Turtlebot II Robotic Platform;
- WLAN Access Points;
- WMP on Alix2D2 Embedded PCs;
- Low-Cost Spectrum Analysers (WiSpys);
- R&S FSV7 Spectrum Analyser;
- R&S SMBV100A Signal Generator.

3.1 TKN Wireless Indoor Sensor Network Testbed

The TKN Wireless Indoor Sensor Network Testbed (TWIST) 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 (Figure 3.2). The TWIST instance at the TKN office building is one of the largest remotely accessible testbeds with 204 SUT sockets, currently populated with 102 eyesIFX and 102 Tmote Sky nodes (Figure 3.1). The nodes are deployed in a 3D grid spanning 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 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 3.3. Within the rooms the sensor nodes are attached to the ceiling.

Figure 3.1: Tmote Sky, eyes IFXv2, NLSU2 supernode / USB Hub

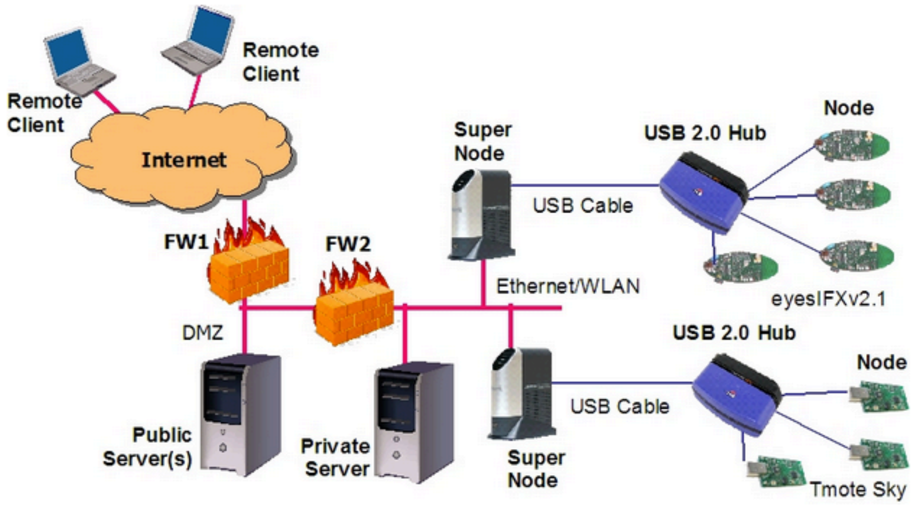

Figure 3.2: Hardware components of TWIST testbed

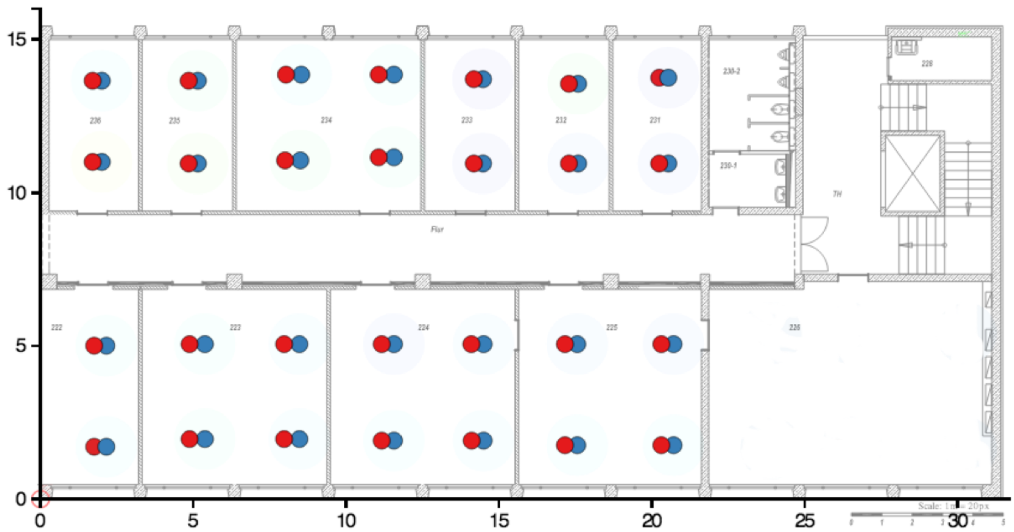

Figure 3.3: Locations of sensor nodes in the 2^{nd} floor of TWIST testbed

3.2 Turtlebot II Robotic Platform

Turtlebot II robotic platform [2] comprises of a mobile base called Kobuki, a laptop, a router and a Microsoft Kinect 3D camera sensor (Figure 3.4). On the software side we use Robot Operating System (ROS) [3], an open source approach for robots. ROS comes with numerous drivers and libraries that cover everything from low-level communication with hardware to the higher layer tasks, such as mapping, navigation and obstacle avoidance. Besides that, ROS is also a communication middleware that transports information between components in ROS. The dominating scheme is topic oriented asynchronous message exchange in the fashion of publish-subscribe, but it also has means for synchronous communication. It is easily extensible through either publishing or subscribing to existing topics or through creating new ones. By doing so, ROS can also be used to transport arbitrary data. This allows to use ROS for controlling robots and extends the system by adding components on top of that.

We have set up an autonomous testbed environment in which we use the Turtlebot to position the SUT at different locations (Figure 3.5). To do that we leverage the navigational capabilities of ROS that also includes obstacle avoidance. ROS uses a a-priori given map and localizes itself by matching the depth information of the Kinect 3D camera with the outline of the known map. ROS provides a simple interface to request the robot to drive autonomously to a given coordinate, so called goal, and a path planer is calculating the best path towards it. We have embedded these calls to move the robot to the next location into a higher schedule. First we define a set of way-points that have to be covered in the survey, then the robot iterates autonomously over each one of them. The whole procedure is followed in an unstructured office environment with dynamic obstacles, like humans, opening / closing the doors, etc.

Figure 3.4: Turtlebot Robotic Platform

For communicating with the rest of the infrastructure the mobile platform is equipped with a Wireless Local Area Network (WLAN) Access Point (AP) that operates in client mode and connects to one of the six APs deployed on every floor in our building. We are controlling the robot's AP by a ROS component that is location aware and selects the most appropriate AP in the different parts of the floor.

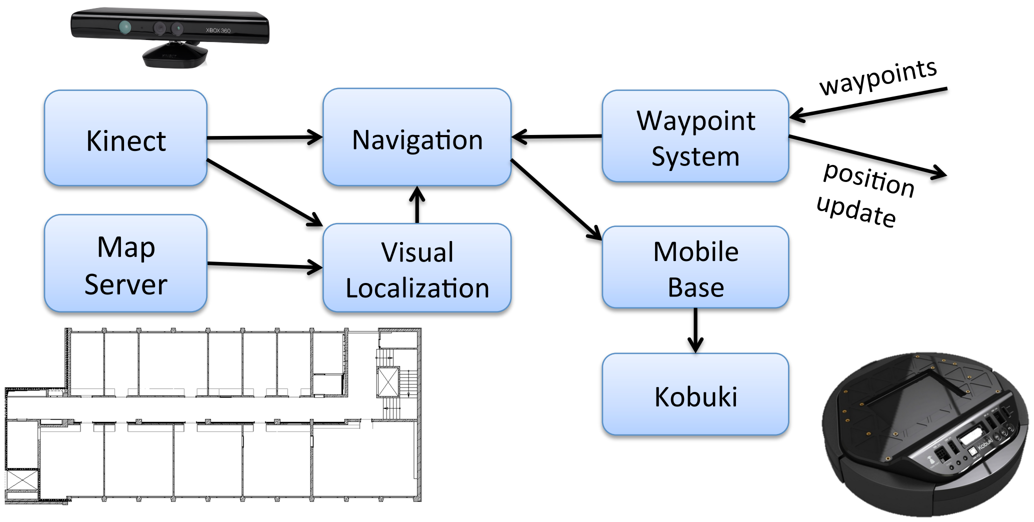

Figure 3.5: Robotic Platform Design

3.3 WLAN Access Points

TKN testbed is equipped with 18 dualband TP-link N750 APs (model TL-WDR4300) [4] (Figure 3.6). They run OpenWRT as Operating System (OS) and cOntrol and Management Framework (OMF) as control and measurement plane. Positions of the WLAN APs in the 2^{nd} floor of TKN testbed are given in Figure 3.7.

Figure 3.6: TL-WDR4300 WLAN Access Point

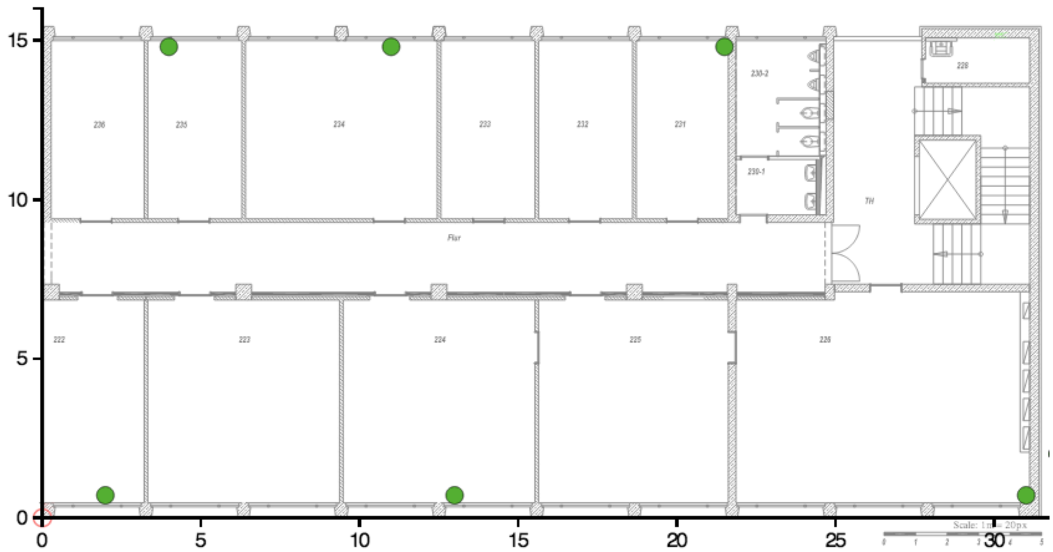

Figure 3.7: Locations of WLAN routers in the 2^{nd} floor of TKN testbed

3.4 WMP on Alix2D2 Embedded PCs

Wireless Mac Processor (WMP) is a customizable WLAN 802.11b/g MAC [5]. It is running on ALIX2D2 embedded PCs [6] equipped with Broadcom WL5011S 802.11b/g cards and shown in Figure 3.8. In our infrastructure three ALIX2D2 exist. The customization of the MAC protocol is done via JAVA GUI. At the ALIX2D2 the firmware is loaded by the so called "bytecode-manager" on the Broadcom cards.

Figure 3.8: Alix2D2 Embedded PC

3.5 Low-Cost Spectrum Analyzers

The TKN infrastructure also comprises several WiSpy sensing devices (Figure 3.9). These are low-cost spectrum scanners that monitor activity in the 868 MHz, 2.4 and 5 GHz spectrum, and output the measured RF energy and the quality of the received signals.

Figure 3.9: WiSpy USB dongle

3.6 R&S FSV7 Spectrum Analyzer

Rhode&Schwarz FSV7 signal and spectrum analyzer [7] (Figure 3.10) is a very flexible and fast signal and spectrum analyzer covering the frequency range between 9 kHz and 7 GHz. It is simple extensible by several measurement applications and toolboxes. Furthermore, it is possible, by buying appropriate license, to add complete receiver chains like Bluetooth, LTE, WiMAX or WLAN.

Figure 3.10: R&S FSV7 spectrum analyzer

3.7 R&S SMBV100A Signal Generator

Rhode&Schwarz SMBV100A is a very flexible signal generator [8] (Figure 3.11). 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 to generate different standards conform signals like e.g., WiMAX, WLAN or LTE. Together with the R&S FSV7 spectrum analyzer complete transmissions chains can be set up.

Figure 3.11: R&S SMBV100A Signal Generator

Chapter 4

Instantiation of TKN Infrastructure for Indoor Localization Experiments

We have instantiated our testbed infrastructure for specific purpose of evaluation and benchmarking of different indoor localization solutions and algorithms, and as such we have used it in the Track 3 of the EVARILOS Open Challenge. The overview of different capabilities of TKN tested for this specific purpose is given in Figure 4.1 [9, 10].

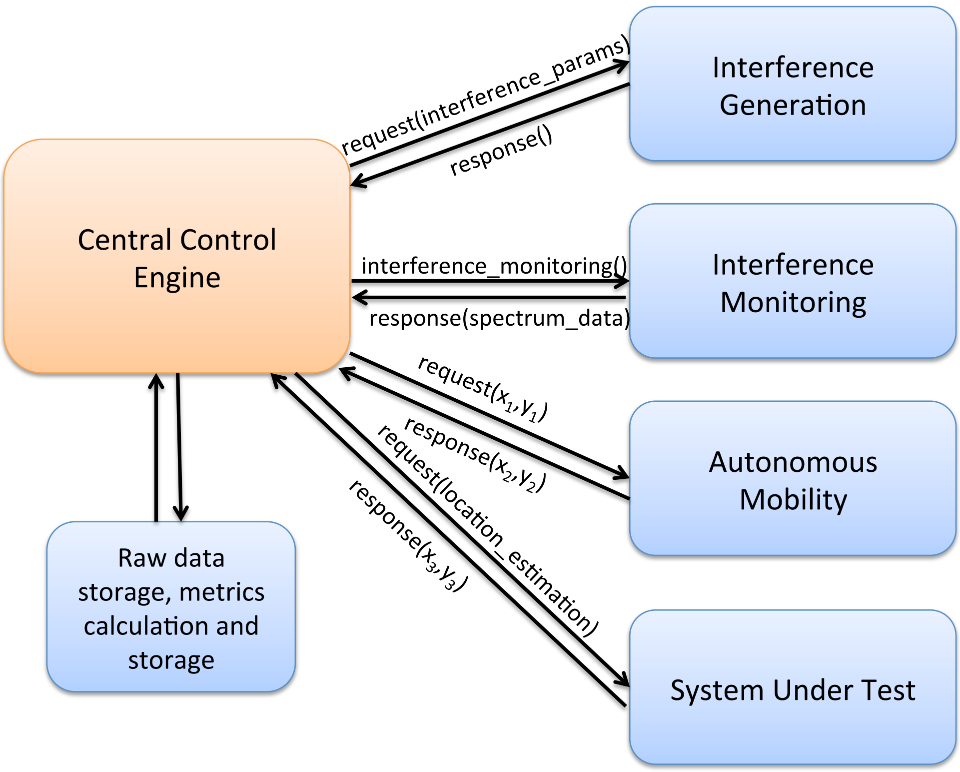

Figure 4.1: Overview of TKN testbed for evaluation and benchmarking of different indoor localization SUTs

The TKN infrastructure leverages a robotic mobility platform which enables autonomous, accurate and repeatable positioning of localization devices in the environment. Furthermore, it integrates devices for generating controlled RF interference, which can be used to evaluate the influence of RF interference on the performance of the localization SUT. Using the TKN infrastructure it is possible to create different types of wireless interference, such as IEEE 802.11 (WiFi), IEEE 802.15.4 (ZigBee) or IEEE 802.15 (Bluetooth). For validation of the resulting RF context, the infrastructure features devices that monitor the RF spectrum at different measurement points in order to guarantee equal conditions for all SUTs. Finally, testbed enables deployment of different indoor localization algorithms or solutions to be tested on the already existing infrastructural components.

4.1 Autonomous Mobility Platform

For automatic transportation of the localized device of the SUT to different evaluation points, without the presence of a human test-person and in a repeatable way, we use the Turtlebot II robotic platform. Turtlebot provides an interface to request the robot to drive autonomously to a given coordinate. For an experiment one can define a set of measurement points and the robot iterates over each of them. When a measurement point is reached, this event and the current location and orientation of the robot is reported back to the experimenter. The location estimations provided by the robot are highly accurate, achieving the mean localization errors of around 20 cm, so these location estimations can be considered as ground truths for indoor localization experiments.

4.2 Interference Generation

During an evaluation of the proposed localization solution the impact of external interference is mostly not considered. However, it can have an influence on the performance of the indoor localization SUT. For this reason we have developed means to generate various types of interference scenarios, as presented in Figure 4.2. The most common type of wireless activity in the 2.4 GHz is the Wireless Fidelity (WiFi) traffic. We have adapted the interference scenarios from [11, 12] and using the OMF [13] we are able to create in our testbed the interference context of typical home or office environments. This type of interference can be created using TKN WLAN routers and Alix PC devices (Figure 4.3), but also additional devices such as laptops can be used to extend the infrastructure. Furthermore, it is possible to use R&S signal generator for creating different interference patterns, such as jamming on the WiFi channel, WiFi traffic, Bluetooth traffic, etc. For generating IEEE 802.15.4 interference patterns, TWIST sensor nodes can be used, where the locations of these nodes are given in Figure 3.3.

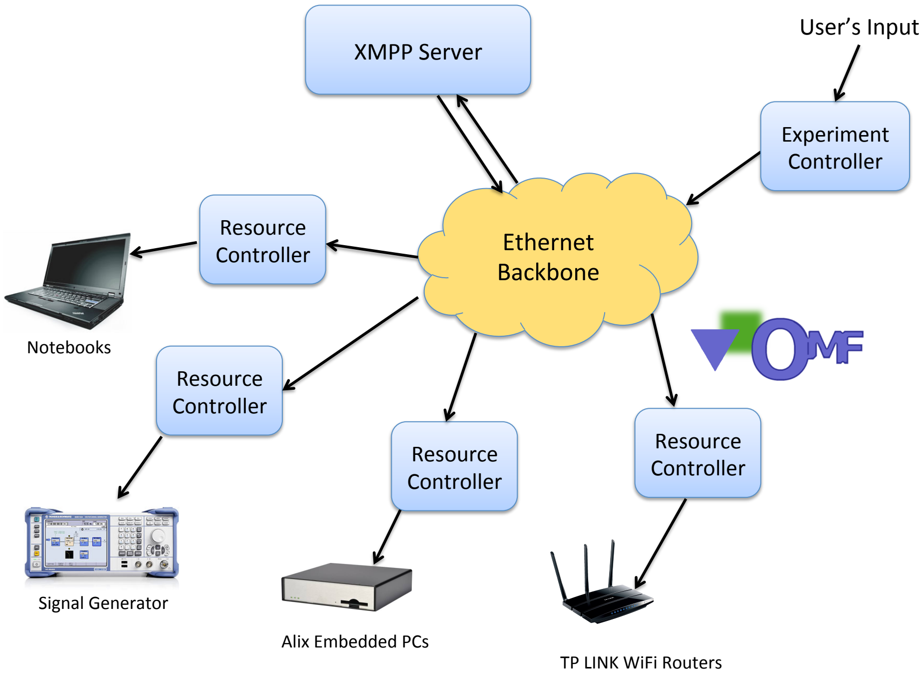

Figure 4.2: Interference generation

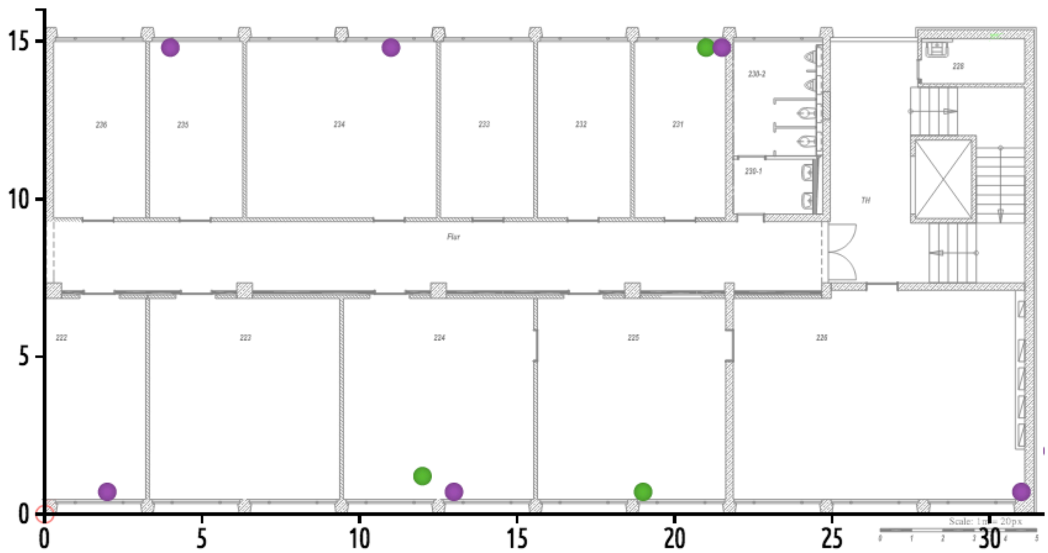

Figure 4.3: Locations of Alix Embedded PCs (green dots) and TPLINK WLAN routers (purple dots) that can be used for interference generation in the 2^{nd} floor of TKN testbed

4.3 Interference Monitoring

In the previous section we have described that we can generate different interference scenarios based on the needs of an experiment. Still, the spectrum in the ISM 2.4 GHz band is free for use and we do not have full control over all devices operating in those frequencies. We have disabled the university infrastructure WiFi network in the 2.4 GHz band in the building but the signal from the surroundings can still be received, as summarized in Figure 4.4. That is why it is necessary to monitor the spectral environment to tell if it is looking as expected. We can use OMF to orchestrate WiSpy [14] devices to perform spectrum sensing. We are using one of them connected to the robot to make sure that the measured interference is not exceeding planned one. Furthermore, for monitoring the wireless spectrum with much finer granularity R&S spectrum analyzer can be used. Same as WiSpy devices, it is controlled using OMF. Finally, some nodes of the TWIST testbed can be used as the distributed spectrum analyzer, where the locations of nodes are given in Figure 3.3.

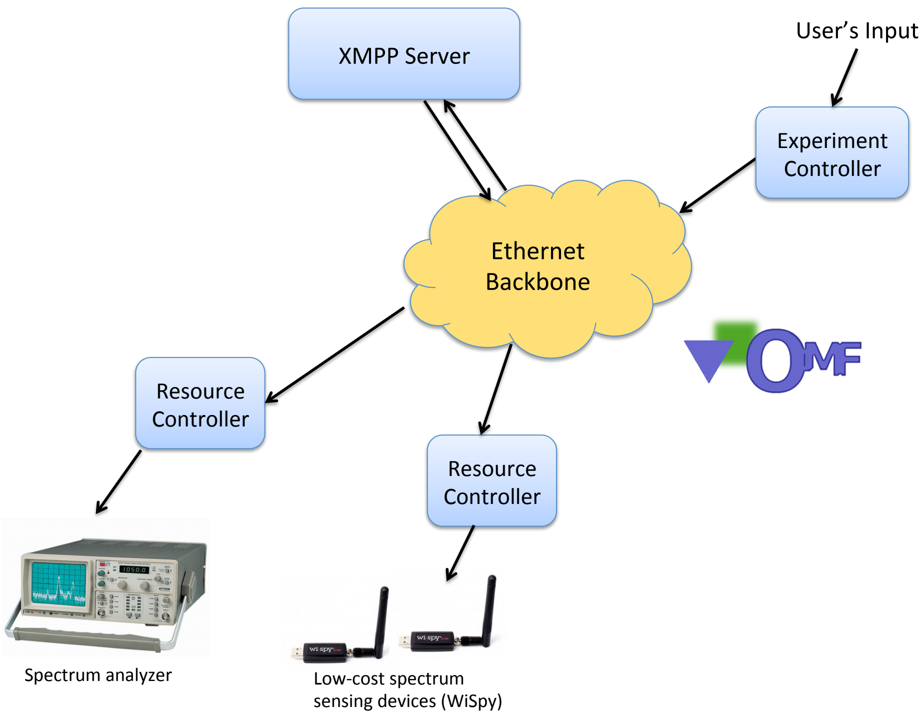

Figure 4.4: Interference monitoring

4.4 System Under Test

Parts of TKN testbed infrastructure can be used for deploying different SUTs devices to be evaluated. Namely, mobile part of the SUT can be mounted on the Turtlebot robotic platform and the devices that we can provide as a mobile part of the SUT are given as follows:

- TmoteSky low power sensor node;
- EyesIFX low power sensor node;
- Shimmer low power sensor node;
- Smartphone Nexus S (GT-I9023);
- Tablet Nexus 7, 2013 version;
- Notebook Lenovo ThinkPad;
- Notebook Asus 1215n.

Furthermore, for static parts (anchor nodes) of SUTs TP LINK WLAN access points can be used, and their locations are given in Figure 4.5. Finally, some TWIST sensor nodes (TmoteSky nodes in 2.4 GHz ISM band or EyesIFX nodes in 868 MHz ISM band) can be used as anchor points for deployment of different SUTs, with locations given in Figure 4.6.

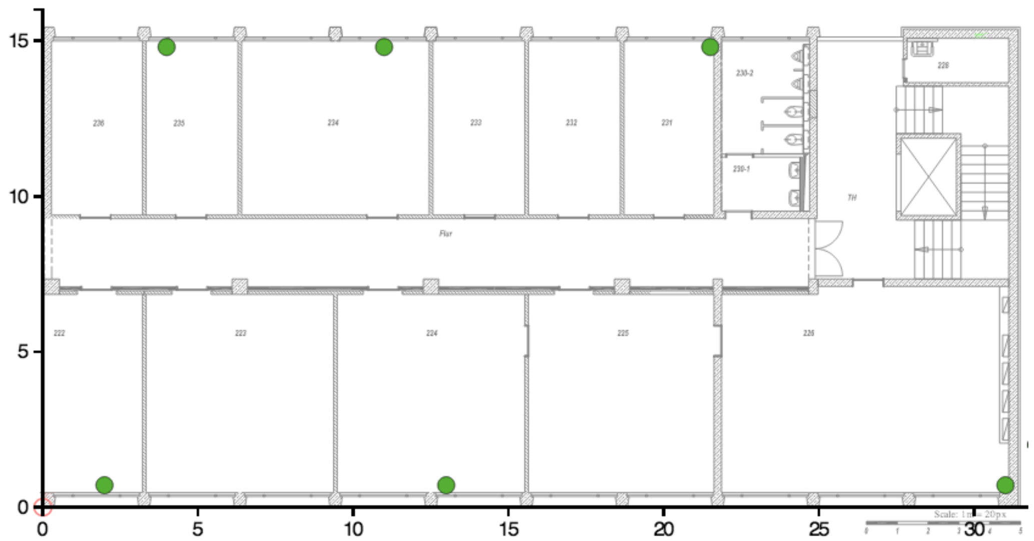

Figure 4.5: Locations of TP LINK WLAN routers that can be used as a part of SUTs in the 2^{nd} floor of TKN testbed

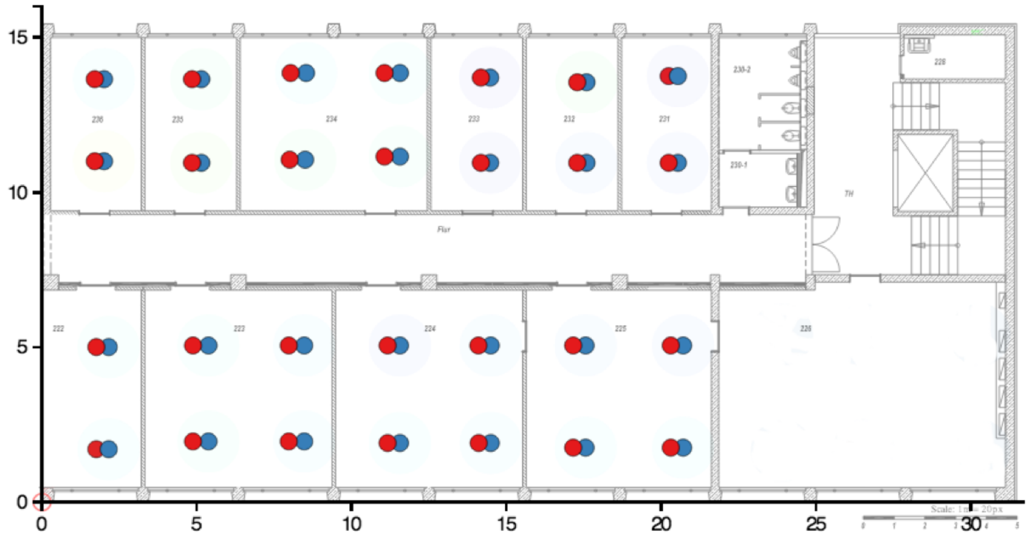

Figure 4.6: Locations of sensor nodes that can be used as a part of SUTs in the 2^{nd} floor of TKN testbed

Chapter 5

Instructions for Competitors

This chapter gives instructions for competitors for adapting their algorithms and using the hardware provided in the TKN testbed.

5.1 Interfacing with the SUT

All competitors had to deploy their algorithm on the devices intended for deploying SUTs and described previously in the document. Further, competitors had to provide an HTTP Uniform Resource Identifier (URI) on which their algorithm listens for requests for location estimation. Upon request, the algorithms had to provide the location estimate as a JSON response in the following format:

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

JSON parameters *coordinate_x* and *coordinate_y* were required parameters and as such they had to be reported upon request. Parameter *coordinate_z* was an optional parameter, due to the 2D evaluation environment. If this parameter was provided from a SUT, evaluation team also calculated 3D localization error, although this information was not be used in final scoring. Finally, parameter *room_label* was an optional parameter and if it was not provided the EVARILOS Benchmarking Platform automatically mapped the room estimate from the estimated coordinates x and y.

Coordinates (x,y) or (x,y,z) of the location estimates had to be calculated according to the predefined *zero-point* coordinate $(x,y,z) = (0,0,0)$, given in Figure 5.1. Furthermore, room labels had to be the same as ones indicated in figure. The technical team provided a footprint of the environment in vector format to all competitors.

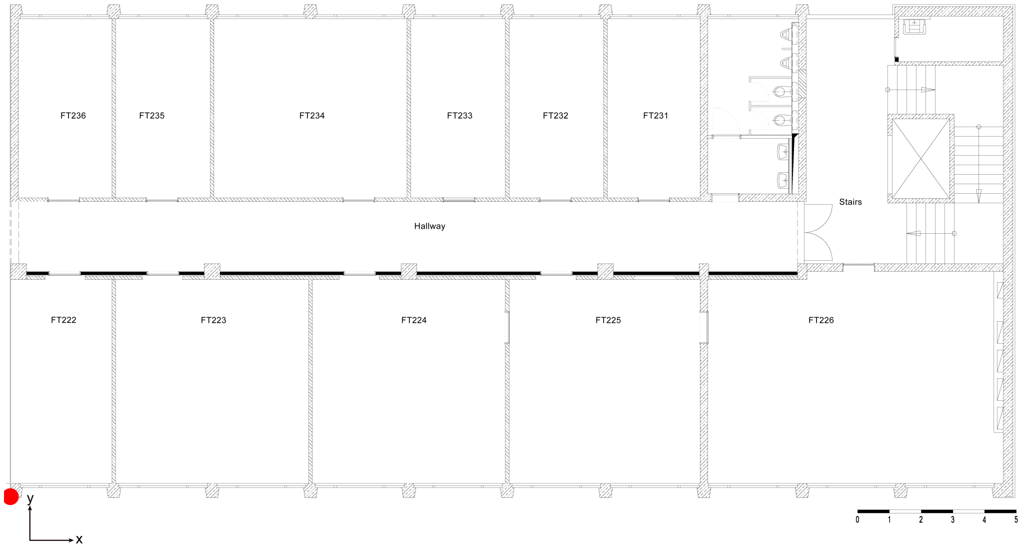

Figure 5.1: Zero-point location and room labels

The technical team supported the competitors in deploying their algorithms on desired hardware, interfacing SUT with the EVARILOS Benchmarking Platform, controlling the robotic mobility platform, generating different interference scenarios and interference monitoring. Furthermore, each competing team was given 4 hours in order to train or parametrize their algorithms in the TKN testbed environment, before the evaluation process started. During that time competitors were also supported by the technical team.

5.2 Usage of Different Hardware Components

This section shortly describes how competing teams were able to use the hardware for their experiments and fine-tuning of their indoor localization algorithms. All competitors were provided with the Virtual Private Network (VPN) access to the testbed network. For the deployment and training part all competitors were able to generate the desired interference scenarios, navigate the robotic platform, etc. as described below. On the other side, for the evaluation part the evaluation committee decided on the locations of interference sources and evaluation points.

5.2.1 SUT Mobile Nodes

Different devices could have been used as mobile parts of SUTs. Users were able to use Secure Shell (SSH) tunnels to the desired devices in order to deploy their algorithms on those devices. Furthermore, for some devices competitors were able to completely change the firmware of a device (mostly for low-power sensor nodes). All additional requests had to be mentioned in advance and the technical committee decided to allow them or not.

5.2.2 SUT Infrastructure Nodes

As mentioned before, as infrastructural parts of SUTs the wireless sensor network of WLAN routers could have been used, depending on the requirements of algorithms. Locations of infrastructural nodes were known to the competitors in advance. The TKN Wireless Indoor Sensor Network Testbed (TWIST) was accessible through the web interface. More details and instructions on how to use testbed can be found on following URL address:

`http://www.twist.tu-berlin.de/wiki`

Different possibilities existed for using the WLAN routers as SUT infrastructural nodes. All routers were running OpenWRT, a Linux distribution for embedded devices [15]. First possibility of usage of WLAN APs was using them in a access point mode, where parameters on an AP, such as transmission power or wireless channel, could have been set using a script that was provided. Furthermore, competitors were given an SSH access to all WLAN routers where they were able to deploy their scripts or anything else necessary for running their SUT. Finally, competitors were able to change the entire firmware of WLAN routers if needed.

5.2.3 Autonomous Mobility

Autonomous mobility platform was accessible over the web interface where competitors were able to click on the location where they want to position the robotic platform and their SUT. As second way of usage, competitors were able to send the robot to a location by setting the coordinates of desired location. Finally, it was also possible to provide a set of way-points to the robotic platform in order to fully automate even the training phase for different indoor localization algorithms. Robot provided its current location or adequate messages if the desired location was currently not reachable.

5.2.4 Interference Generation

Competitors were, to the certain degree, able to generate the interference scenarios using the above described devices. Namely, the code for generating three interference scenarios described below was provided by the technical team, and the users were able to select the nodes on which the code should run.

5.2.5 Interference Monitoring

Competitors were able to use different devices for monitoring interference levels. They were able to obtain the dumps of wireless spectrum from the WiSpy device on the robot or WiSpy devices on multiple other locations. Furthermore, competitors were able to use TWIST nodes for distributed spectrum sensing. Finally, competitors were also able to use the spectrum analyzer for collecting the spectrum traces with much finer granularity than the one achieved with low-cost WiSpy devices. The code for spectrum monitoring was also provided by the technical team.

Chapter 6

Evaluation Scenarios

This chapter presents interference scenarios that were artificially generated in TWIST testbed in order to evaluate different indoor localization algorithms. The goal was to determine if and to which extend different types and amounts of RF interference can influence the indoor localization performance. The text below presents the reference scenario and describes three interference scenarios that were used for evaluation in TKN testbed.

6.1 Reference Scenario

This reference scenario was instantiated on the 2^{nd} floor of the TKN testbed in Berlin. It was called Reference scenario, since no artificial interference was generated and the presence of uncontrolled interference was minimized. According to the EVARILOS Benchmarking Handbook (EBH), this scenario was an instance of the "Small office" type of scenarios. In this scenario 20 evaluation points were defined (the locations are given in Figure 6.1).

At each evaluation point, the indoor localization SUT was requested to estimate the location. The SUT device was carried to each evaluation point using the robotic platform. The navigation stack of the robotic platform gave an order of magnitude more accurate location estimation than the SUTs and the location obtained from the robotic platform was considered as the ground truth for the evaluation.

The experiments were performed in the afternoons, so the influence of uncontrolled interference was minimized. Furthermore, the wireless spectrum was monitored using the WiSpy device attached to the robotic platform and all measurements with the interference threshold above certain level were repeated. Finally, during each experiment a measurement of the wireless spectrum was taken with the spectrum analyser at a predefined location.

Figure 6.1: Locations of evaluation points

6.2 Interference Scenario 1

In this interference scenario instantiated in the TKN testbed 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, as shown in Figure 6.2. Summary of this interference scenario is given in Table 6.1.

Table 6.1: Interference scenario 1 summary

Types of interference		**Parameters of interference sources**	
WiFi	✖	Number of sources	5
Microwave	✖	Power	0 dBm
Bluetooth	✖	Waveform	Carrier jamming
ZigBee	✔	Start & stop time	Beginning & end of experiment
Synthetic	✖	Traffic model	IEEE 802.15.4 radio

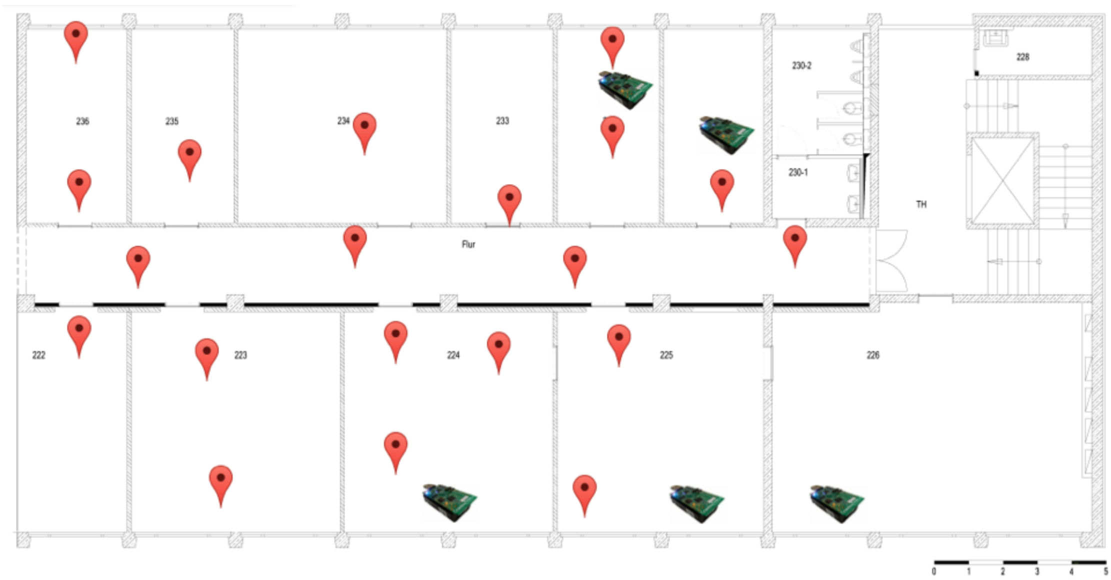

Figure 6.2: Locations of interference sources in the interference scenario 1

6.3 Interference Scenario 2

Second interference scenario instantiated in the TKN testbed defined interference types that is usual for the office or home environments. Namely, interference was emulated using 4 WiFi embedded Personal Computers (PCs): server, access point, data client, and video client. The server acted as a gateway for the emulated services. The data client was emulated as 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 with the bandwidth of 50 Mbps. The AP was working on a WiFi channel overlapping with the SUT's channel and with the transmission power set to 20 dBm (100 mW). Summary of a described interference scenario is given in Table 6.2, while the locations of interference transmission streams are given in Figure 6.3.

Table 6.2: Interference scenario 2 summary

Types of interference	
WiFi	✔
Microwave	✖
Bluetooth	✖
ZigBee	✖
Synthetic	✖

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

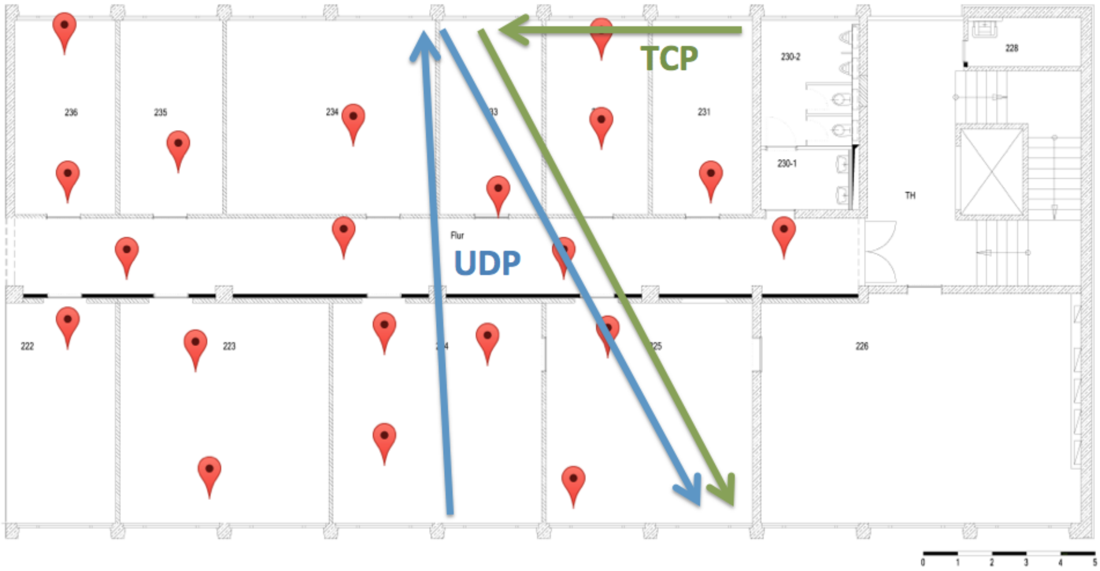

Figure 6.3: Locations of interference sources in the interference scenario 2

6.4 Interference Scenario 3

For the third interference scenario instantiated in the TKN testbed signal generator was used for generation of synthetic interference, with location given in Figure 6.4. The generated synthetic interference had an envelope of the characteristic WiFi signals, but without any Carrier Sensing (CS). The summary of interference scenario 3 is given in Table 6.3.

Table 6.3: Interference scenario 3 summary

Types of interference		**Parameters of interference sources**	
WiFi	✖	Number of sources	1
Microwave	✖	Power	20 dBm
Bluetooth	✖	Waveform	Power envelope
ZigBee	✖	Start & stop time	Beginning & end of experiment
Synthetic	✔	Traffic model	Synthetic IEEE 802.11 traffic

Figure 6.4: Locations of interference sources in the interference scenario 3

Chapter 7

Evaluation Procedure

Track 3 of the challenge evaluates RF-based indoor localization algorithms deployed on top of the existing hardware in TKN testbed. This chapter describes the evaluation procedure followed in the Track 3 of the challenge.

7.1 Evaluation Process

Indoor localization algorithms were evaluated in 20 different evaluation points under four interference scenarios. The evaluation points were selected by the evaluation team and were the same for all evaluated algorithms. In the first run, all algorithms were evaluated in the environment without controlled interference and the metrics were calculated. The following three runs of evaluation were performed in the environment with three different interference scenarios described before. The locations of interference sources were selected by the evaluation team and were the same for all evaluated algorithms. For each location at each measurement point, the EVARILOS Benchmarking Platform requested a location estimate from the SUT. The evaluated data at each location was automatically stored and metrics were calculated and presented in real-time.

7.2 Evaluation Metrics

For the Track 3 of the challenge the following metrics were calculated:

- Performance metrics - obtained from the experiment:
  - Geometric or point level accuracy of location estimation;
  - Room level accuracy of location estimation;
  - Latency or delay of location estimation.
- Derived metric - calculated from the performance metrics:
  - Interference sensitivity of indoor localization algorithm.

7.2.1 Point Level Accuracy

Point level accuracy at one evaluation point is defined as the Euclidean distance between the ground truth provided by the robotic platform (x_{GT}, y_{GT}) and the location estimated by indoor localization algorithm (x_{EST}, y_{EST}), given with the following equation:

$$PointAccuracy = \sqrt{(x_{GT} - x_{EST})^2 + (y_{GT} - y_{EST})^2}\,[m] \tag{7.1}$$

7.2.2 Room Level Accuracy

Room level accuracy of location estimation is a binary metrics stating the correctness of the estimated room, given with the following equation:

$$RoomAccuracy = \begin{cases} 1 & \text{if the estimated room is correct;} \\ 0 & \text{if the estimated room is not correct or not estimated;} \end{cases} \tag{7.2}$$

7.2.3 Latency of Location Estimation

Latency or delay of location estimation is the time that the SUT needs to report the location estimate when requested. The time that will be measured in the evaluation is the difference between the moment when the request for indoor localization has been sent to the SUT ($t_{request}$) and the moment when the response arrives ($t_{response}$), given with the equation:

$$Latency = t_{response} - t_{request}\,[s] \tag{7.3}$$

7.2.4 Interference Sensitivity

Interference sensitivity of indoor localization algorithm is a metric that reflects the influence of different interference types to the performance of the indoor localization algorithm. In this evaluation, interference sensitivity was expressed as the percentage of change in other metrics in the scenarios with interference in comparison to the performance in the scenario without interference (reference scenario). For the case of generalized metric (M), the interference sensitivity is given with the following equation:

$$InterferenceSensitivity = \frac{M_{reference} - M_{interference}}{M_{reference}} \cdot 100\,[\%] \tag{7.4}$$

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

7.3 Capturing the Evaluation Metrics

The evaluation procedure was performed in four steps, namely in four interference scenarios. In each step, for each of the 20 evaluation points, the set of metrics (point accuracy, room accuracy, latency) were obtained. For each set, the 75^{th} percentile of point level accuracy and latency was calculated, together with percentage of correctly estimated rooms, as shown in Figure 7.1.

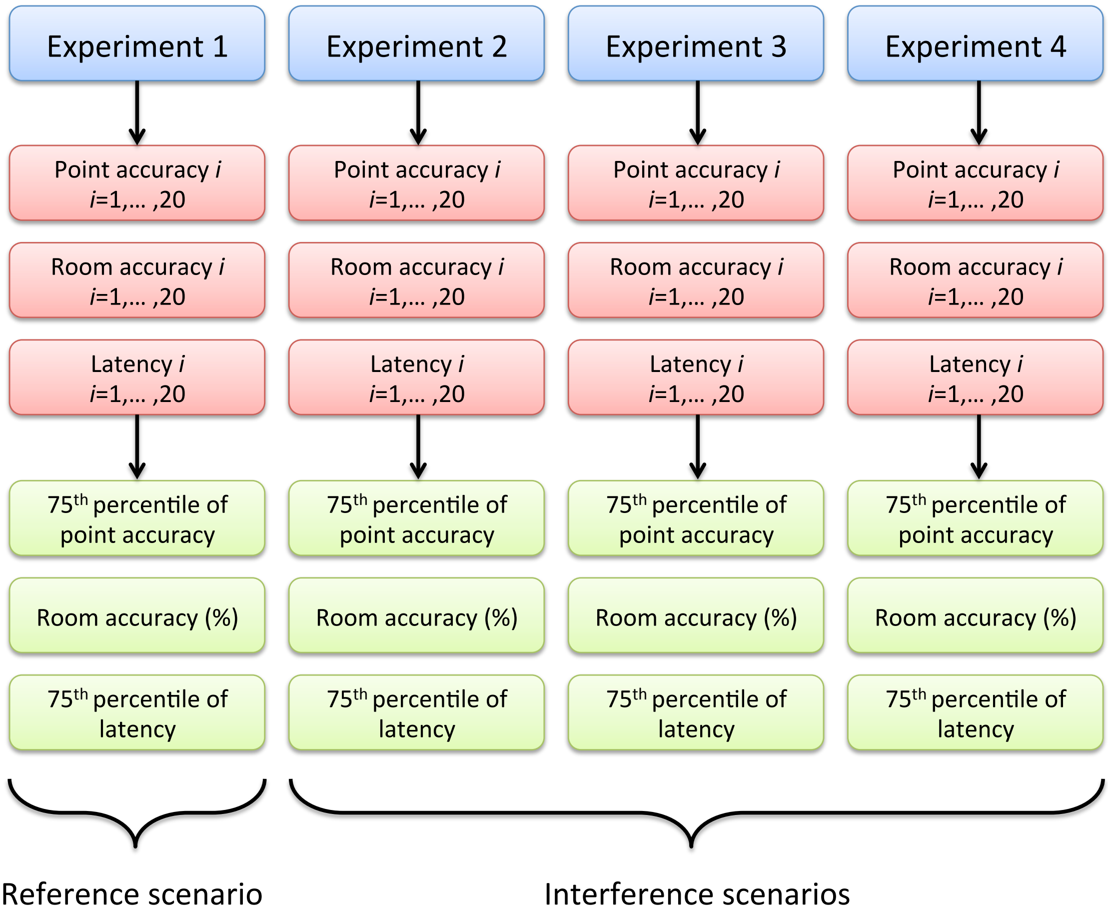

Figure 7.1: Procedure of capturing performance metrics

Interference sensitivity was calculated using the principle given in Figure 7.2. For each interference scenario the interference sensitivity is calculated for each performance metric, using the Equation 7.4. The overall interference sensitivity is the averaged interference sensitivity over all interference scenarios and all performance metrics, given with following equation:

$$\overline{M} = \frac{1}{9}\sum_{i=1}^{3}(M_1(i) + M_2(i) + M_3(i)) \tag{7.5}$$

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 75^{th} percentile of point accuracy, interference sensitivity of percentage of room level accuracy and interference sensitivity of 75^{th} percentile of latency for scenario i, respectively.

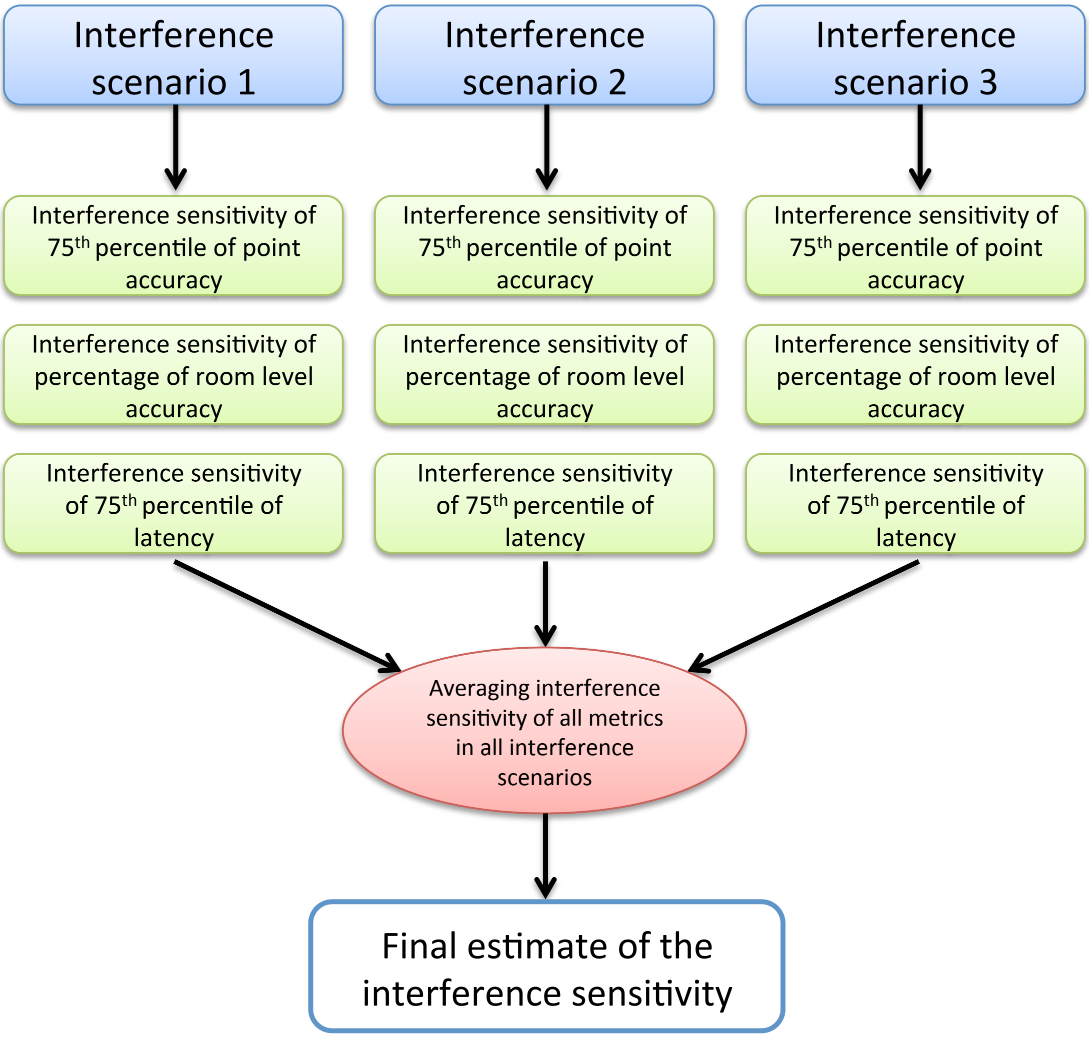

Figure 7.2: Procedure of capturing derived metric - interference sensitivity

7.4 Calculation of Final Score

Final scores were calculated according to the approach described in the EVARILOS Benchmarking Handbook and presented in Figure 7.3. EBH proposes the calculation of the score for each metric according to a linear function that is defined by specifying acceptable and desired acceptable value for the metric. Furthermore, EBH proposes the use of weighting factors for defining the importance of each metric for a given use-case.

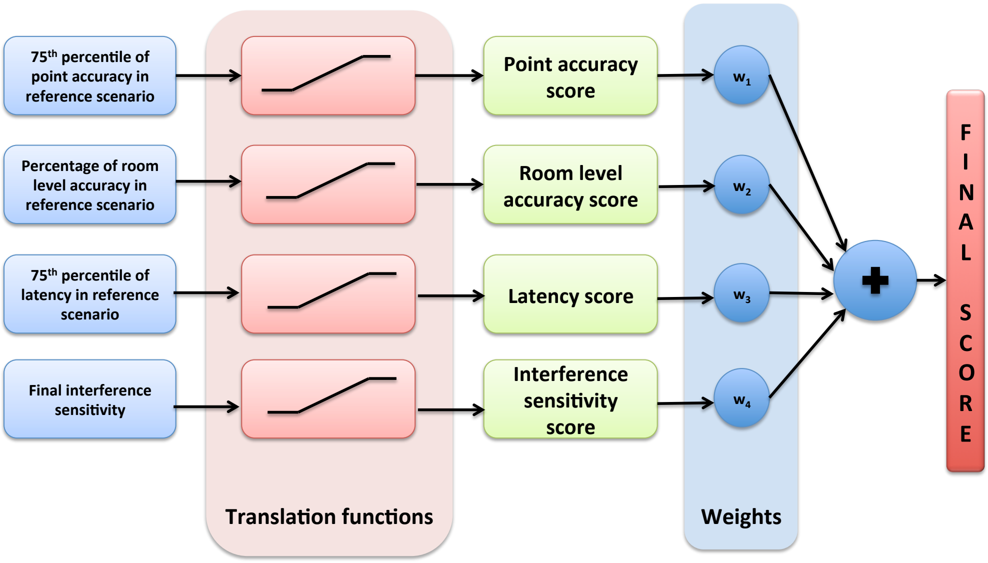

Figure 7.3: Calculation of the final score

In general, the linear translation function for calculating the score of each particular metric is given in Figure 7.6, where scores can range from 0 to 10. 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 error, while $M_{desired}$ is the desired average localization error.

$$Score = \max\left(0, \min\left(10, 10\frac{m - M_{acceptable}}{M_{desired} - M_{acceptable}}\right)\right) \tag{7.6}$$

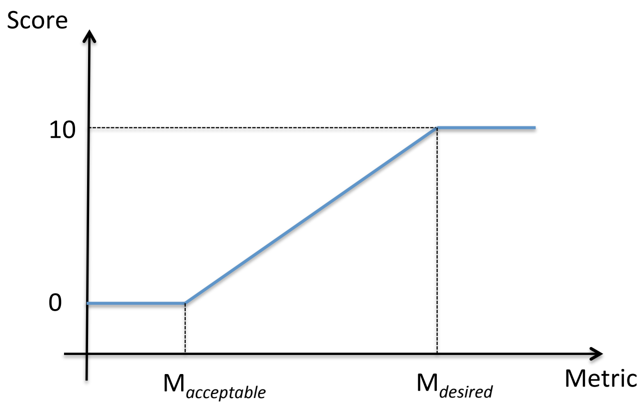

Figure 7.4: Linear translation function for each metric in case when $M_{acceptable} < M_{desired}$

In the Track 3 of the Open Challenge three winners were declared, based on the final scores for three different sets of marginal values and weights. For the calculation of final scores in the Track 3 of the challenge, the sets of marginal values and weights that were used for different metrics in each category are given in Table 7.1, Table 7.2 and Table 7.3.

Table 7.1: Marginal values and weights for the first category

Metric	**Acceptable value** ($M_{acceptable}$)	**Desired value** ($M_{desired}$)	**Weight**
Point accuracy	10 m	1 m	0.40
Room level accuracy	50 %	90 %	0.40
Latency	20 sec	1 sec	0.10
Interference sensitivity	50 %	10 %	0.10

Table 7.2: Marginal values and weights for the second category

Metric	**Acceptable value** ($M_{acceptable}$)	**Desired value** ($M_{desired}$)	**Weight**
Point accuracy	10 m	1 m	0.20
Room level accuracy	50 %	90 %	0.20
Latency	20 sec	1 sec	0.50
Interference sensitivity	50 %	10 %	0.10

Table 7.3: Marginal values and weights for the third category

Metric	**Acceptable value** ($M_{acceptable}$)	**Desired value** ($M_{desired}$)	**Weight**
Point accuracy	10 m	1 m	0.20
Room level accuracy	50 %	90 %	0.20
Latency	20 sec	1 sec	0.10
Interference sensitivity	50 %	10 %	0.50

Chapter 8

Description and Results of Evaluated Systems Under Test

This chapter gives short descriptions of the indoor localization algorithms evaluated in the Track 3 of the EVARILOS Open Challenge. Due to the black-box nature of evaluating different SUTs in this competition, we don't discuss in details the results achieved by each SUT. We present the localization errors at each evaluation point for each algorithm and shows how the errors per point change due to additional interference in the interference scenarios. Additionally, due to its size, it the appendix we show the confusion matrices for each algorithm in each scenario, where each confusion matrix shows the correlation between a room where the measurement was taken and the room estimated by the SUT. Summary results per competitor are also presented in the following sections of this chapter. Note that, due to better visibility, localization errors per evaluation points are drawn as circles whose sizes are reduced and are proportional to the size of the environment, but the relative ratios between errors are fixed. For the more accurate representations of absolute localization errors per points interested reader is directed to:

`http://ebp.evarilos.eu:5011`

8.1 Quantile-based Indoor Fingerprinting using Dedicated WiFi APs

Authors: F. Lemic
Affiliation: Telecommunication Networks Group (TKN), Technical University of Berlin (TUB)
E-mail: lemic@tkn.tu-berlin.de
Country: Germany

8.1.1 Short Description

A growing demand for the information about location of numerous devices in indoor and urban environments raises the need for indoor localization. Indoor localization is needed for various applications and services, and it is considered as one of the key enablers of the Future Internet concepts. One of the most promising approaches in indoor localization is fingerprinting using information from the WiFi infrastructure. We propose a new fingerprinting-based indoor localization algorithm [16, 17] that makes use of the Received Signal Strength Indicator (RSSI) values from WiFi beacon packets for estimating the location. Namely, for generating fingerprints our algorithm uses the quantiles of RSSI values from beacon packets transmitted from various WiFi APs in the premises. Furthermore, the proposed algorithm uses Pompeiu-Hausdorff distance for calculating the difference between training fingerprints and ones generated by user to be localized. Finally, the proposed algorithm uses the k-Nearest Neighbours (kNN) procedure with the parameter k set to 4.

8.1.2 Evaluation Results

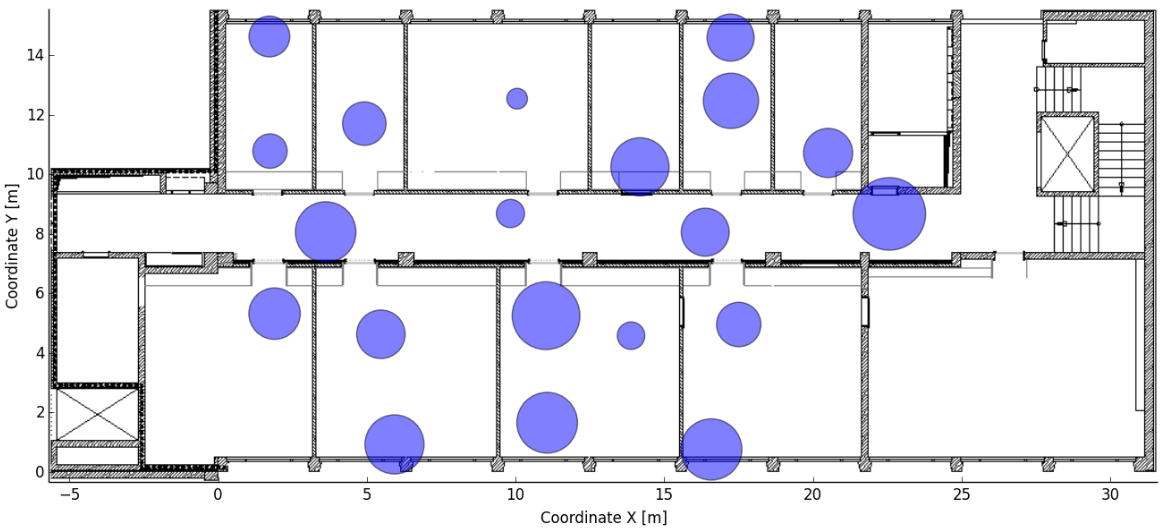

Figure 8.1: Reference scenario

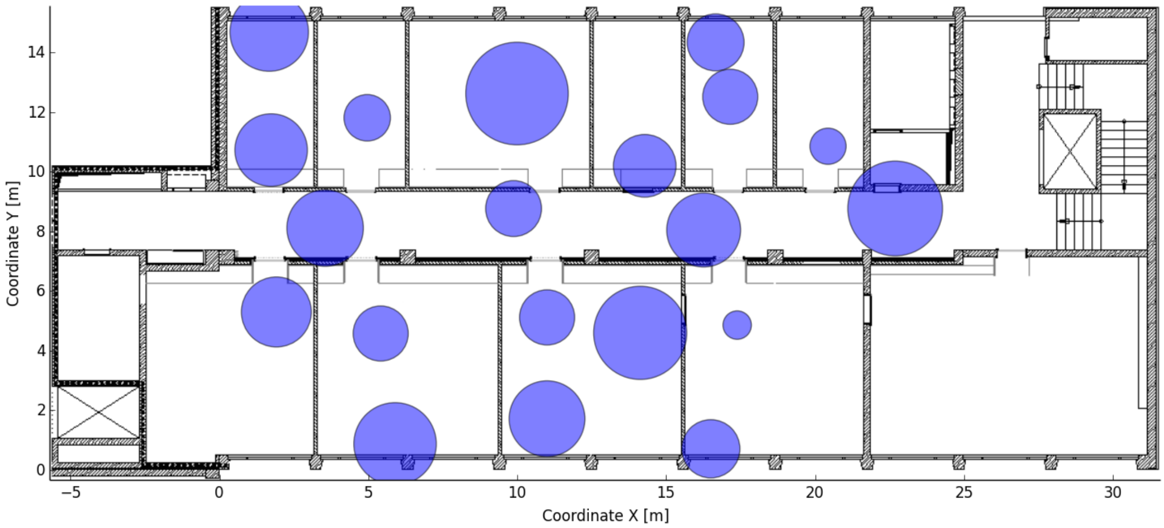

Figure 8.2: Interference scenario 1

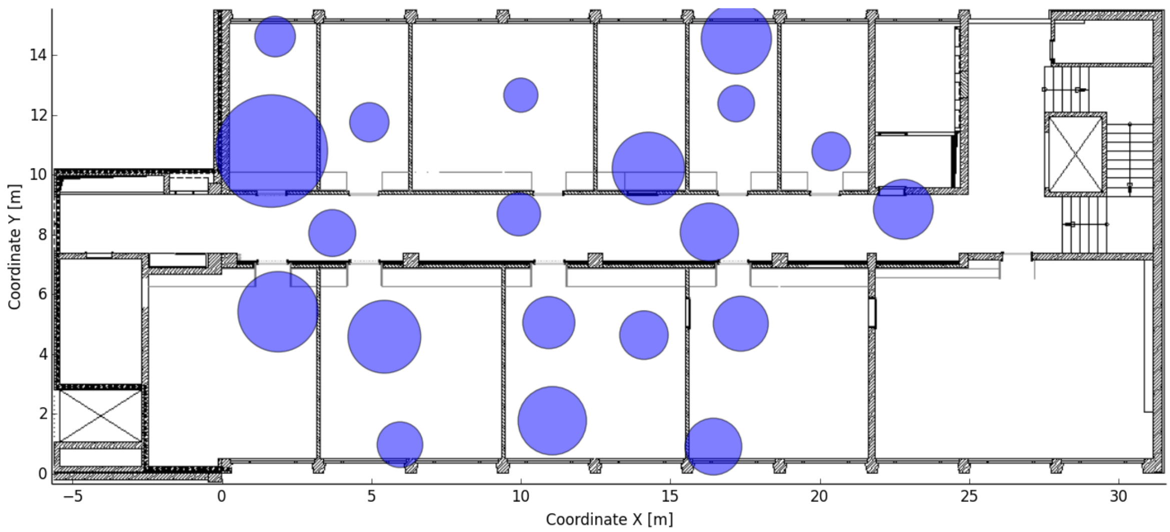

Figure 8.3: Interference scenario 2

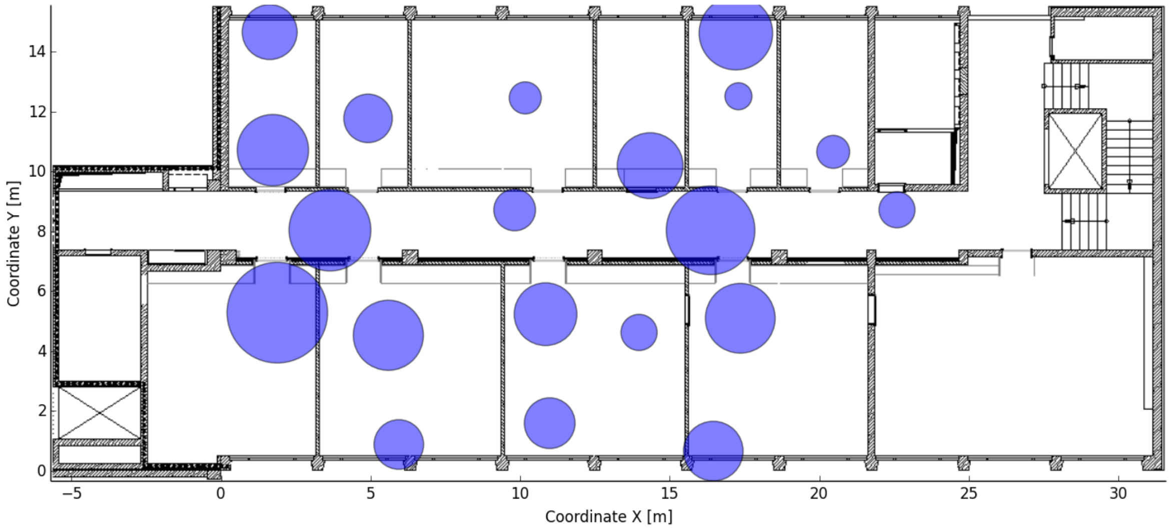

Figure 8.4: Interference scenario 3

Table 8.1: Primary metrics summary

Metric	Reference scenario	Interference scenario 1	Interference scenario 2	Interference scenario 3
Mean localization error [m]	2.82	5.20	3.84	4.02
Median localization error [m]	2.61	4.81	3.11	3.56
RMS localization error [m]	3.14	5.88	4.75	4.80
75 percentile localization error [m]	3.87	6.46	5.14	5.36
90 percentile localization error [m]	4.19	9.49	5.87	7.37
Min. localization error [m]	0.47	0.88	1.26	0.79
Max. localization error [m]	5.76	11.46	13.70	10.93
Room level accuracy [%]	70.0	45.0	40.0	45.0
Mean latency [s]	20.08	20.58	19.88	20.02
Median latency [s]	20.05	20.45	19.83	20.05
RMS latency [s]	20.09	20.58	19.88	20.02
75 percentile latency [s]	20.11	20.71	19.95	20.11
90 percentile latency [s]	20.25	21.64	20.03	20.16
Min. latency [s]	19.66	19.92	19.71	19.67
Max. latency [s]	21.02	21.94	20.11	20.26

8.2 Indoor Geolocation for Android Smartphones with Airplace

Authors: T. Constambeys[1], C. Laoudias[2], D. Zeinalipour-Yazti[1], and C. G. Panayiotou[2]

Affiliation: [1] Department of Computer Science, University of Cyprus, [2] KIOS Research Center for Intelligent Systems and Networks, University of Cyprus

E-mail: timo0190@gmail.com, {laoudias, christosp}@ucy.ac.cy, dzeina@cs.ucy.ac.cy

Country: Cyprus

8.2.1 Short Description

We present the Airplace indoor geolocation platform developed for Android tablets and smartphones [18]. Airplace exploits the available WLAN infrastructure and monitors Received Signal Strength (RSS) values from the surrounding APs to determine the unknown user location. Our system utilizes a number of RSS 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. n the open-source version of Airplace3 we have implemented several fingerprinting algorithms, including the K-Nearest Neighbour (KNN) and Weighted K-Nearest Neighbour (WKNN) deterministic algorithms, as well as the Maximum A Posteriori (MAP) and Minimum Mean Square Error (MMSE) probabilistic algorithms. For the purposes of the EVARILOS Open Challenge remote competition we will rely on the proprietary Radial Basis Function Network (RBFN) algorithm developed in-house (see [19] for the basic algorithm functionality). We will demonstrate the real-time localization capabilities of Airplace and the robustness to interference during the remote competition for RF-based indoor localization algorithms. We firstly present the benchmarking results of our algorithms using the Android smartphone Google Nexus S as the mobile node for the deployment of the algorithm. The fingerprinting procedure in this case uses only a set of dedicated APs. Similarly, the second set of results uses only dedicated APs, but here the mobile node for deploying the algorithm is an Android tablet Google Nexus 7. Finally, we present the results for our algorithm using the same mobile node, Android tablet Google Nexus 7, but here fingerprinting procedure is done using all APs visible in the environment.

8.2.2 Evaluation Results

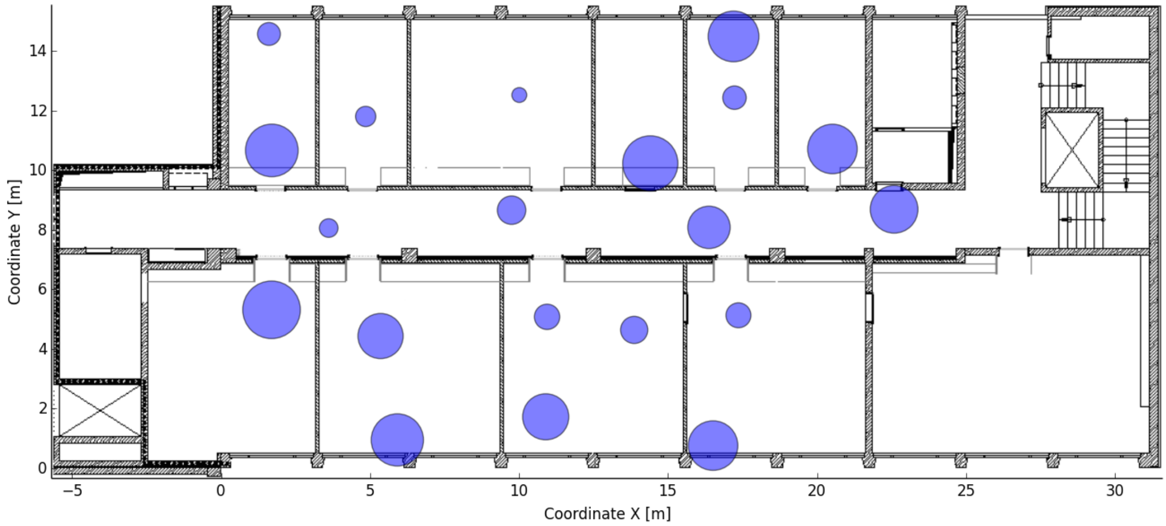

Figure 8.5: Reference scenario - Android smartphone Google Nexus S

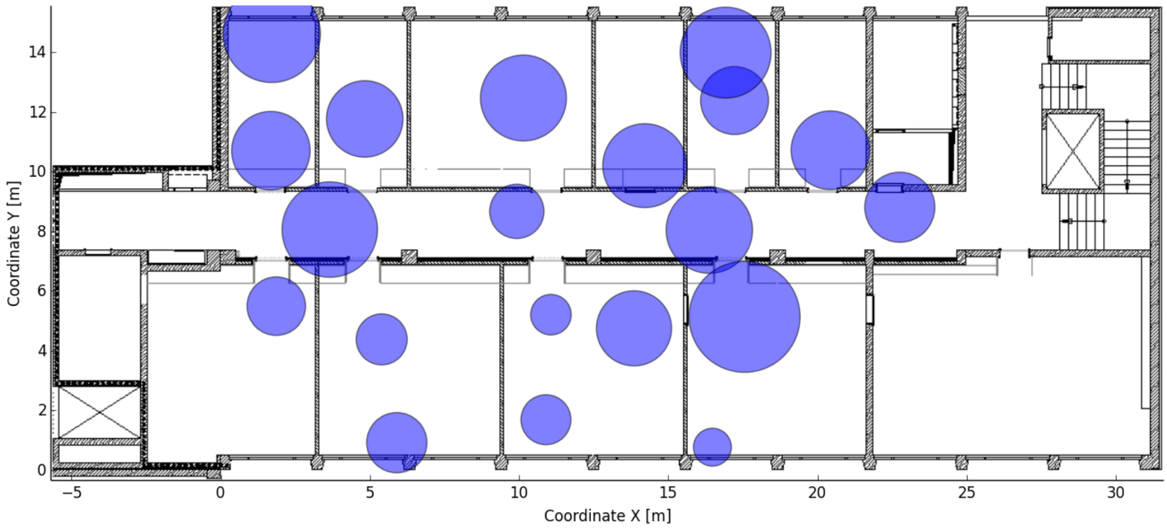

Figure 8.6: Interference scenario 1 - Android Smartphone Google Nexus S

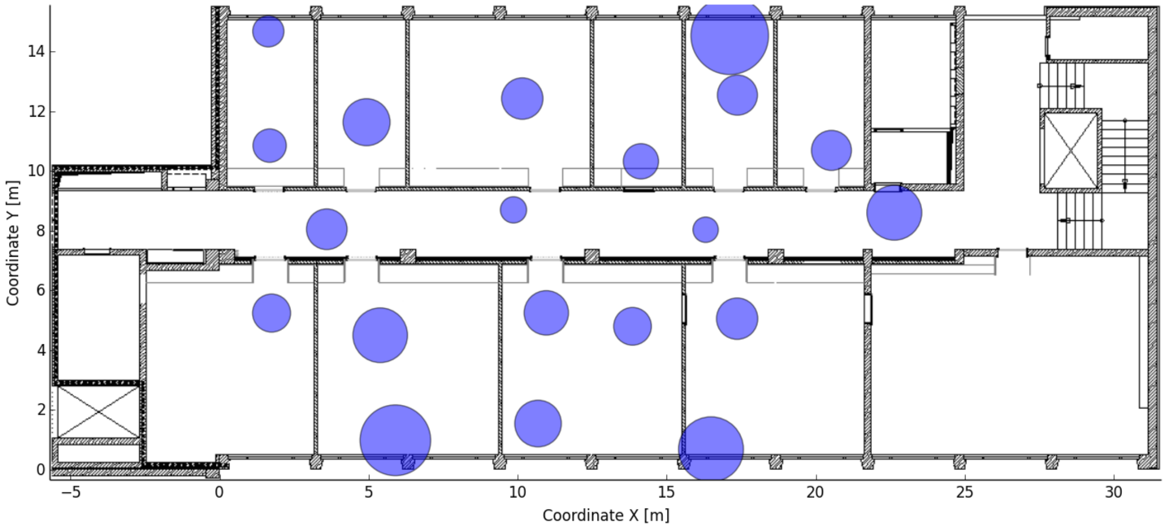

Figure 8.7: Interference scenario 2 - Android Smartphone Google Nexus S

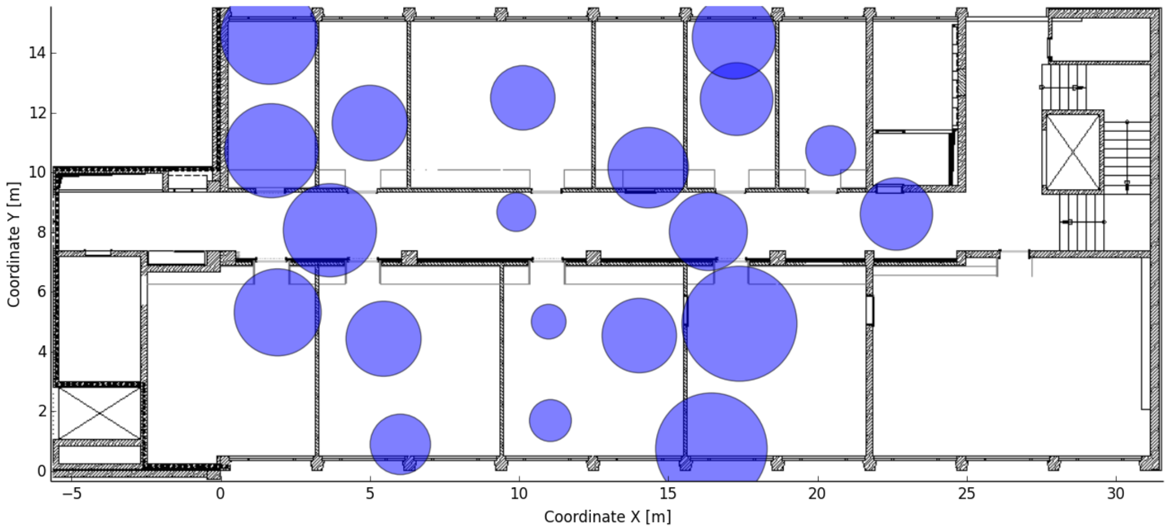

Figure 8.8: Interference scenario 3 - Android Smartphone Google Nexus S

Table 8.2: Primary metrics summary - Android Smartphone Google Nexus S

Metric	Reference scenario	Interference scenario 1	Interference scenario 2	Interference scenario 3
Mean localization error [m]	1.77	6.12	2.36	6.63
Median localization error [m]	2.09	6.25	1.83	6.16
RMS localization error [m]	2.10	6.84	2.80	7.50
75 percentile localization error [m]	2.71	8.04	2.61	8.53
90 percentile localization error [m]	3.05	9.93	4.67	10.70
Min. localization error [m]	0.24	1.56	0.70	1.31
Max. localization error [m]	3.62	13.39	6.54	14.36
Room level accuracy [%]	80.0	30.0	85.0	20.0
Mean latency [s]	2.85	2.83	5.07	2.80
Median latency [s]	2.82	2.65	4.10	2.78
RMS latency [s]	2.86	2.94	6.29	2.83
75 percentile latency [s]	3.07	2.93	4.74	3.06
90 percentile latency [s]	3.13	3.14	7.56	3.23
Min. latency [s]	2.25	2.03	2.44	2.14
Max. latency [s]	3.29	5.94	18.08	3.61

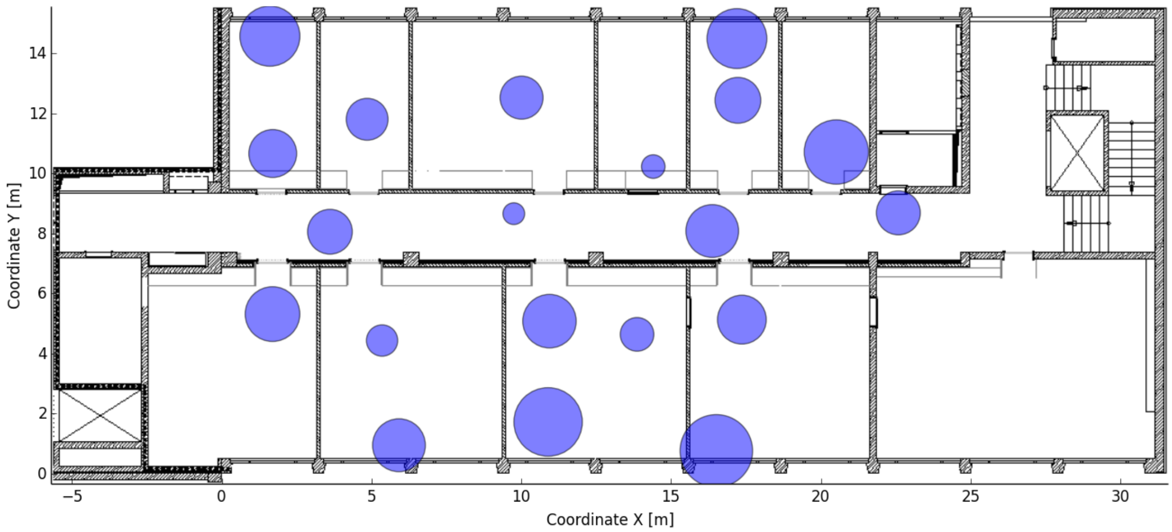

Figure 8.9: Reference scenario - Android Tablet Google Nexus 7

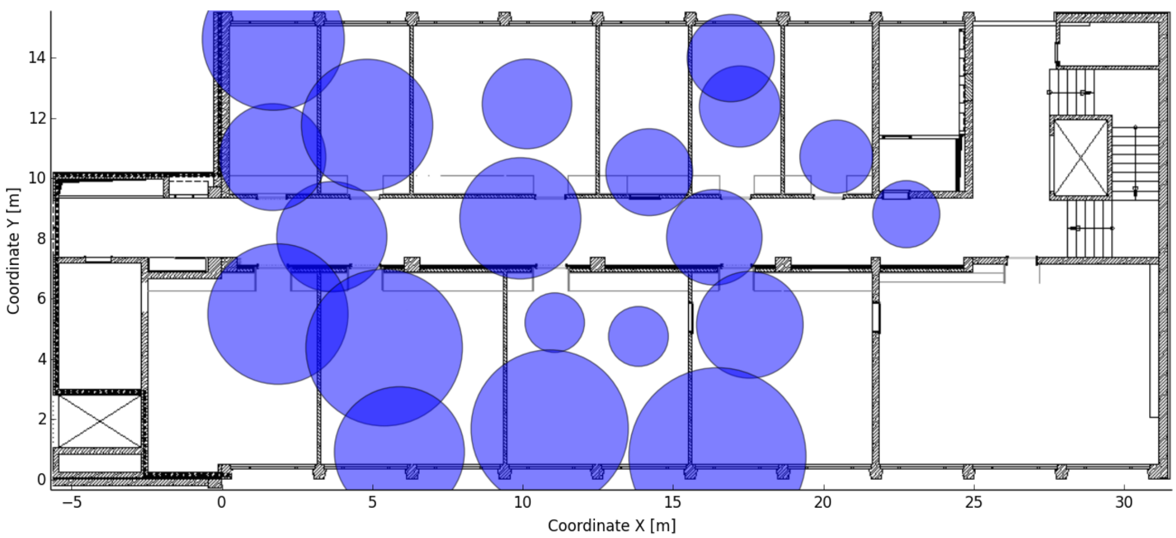

Figure 8.10: Interference scenario 1 - Android Tablet Google Nexus 7

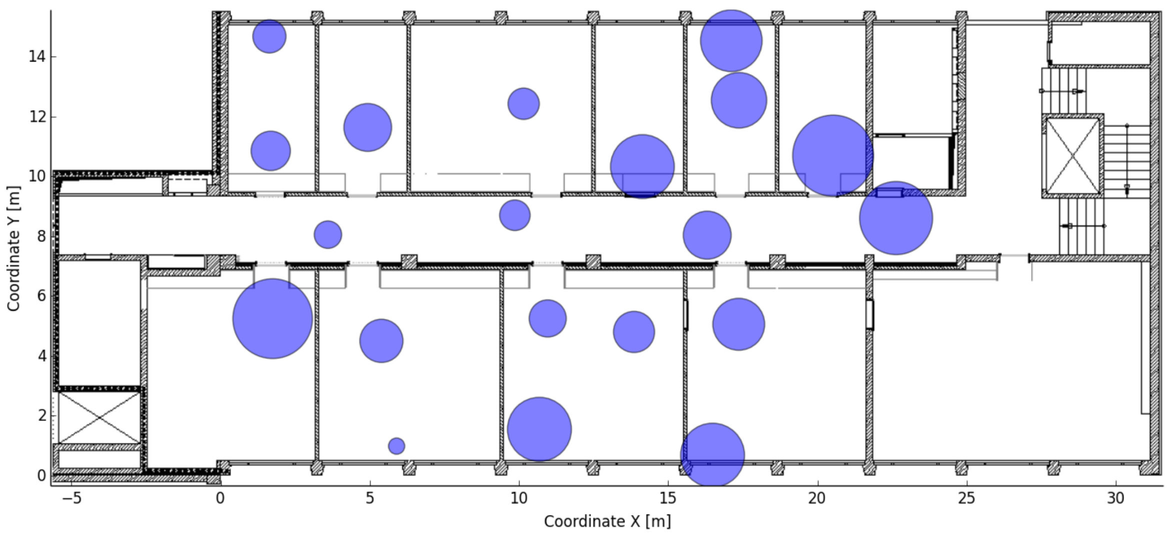

Figure 8.11: Interference scenario 2 - Android Tablet Google Nexus 7

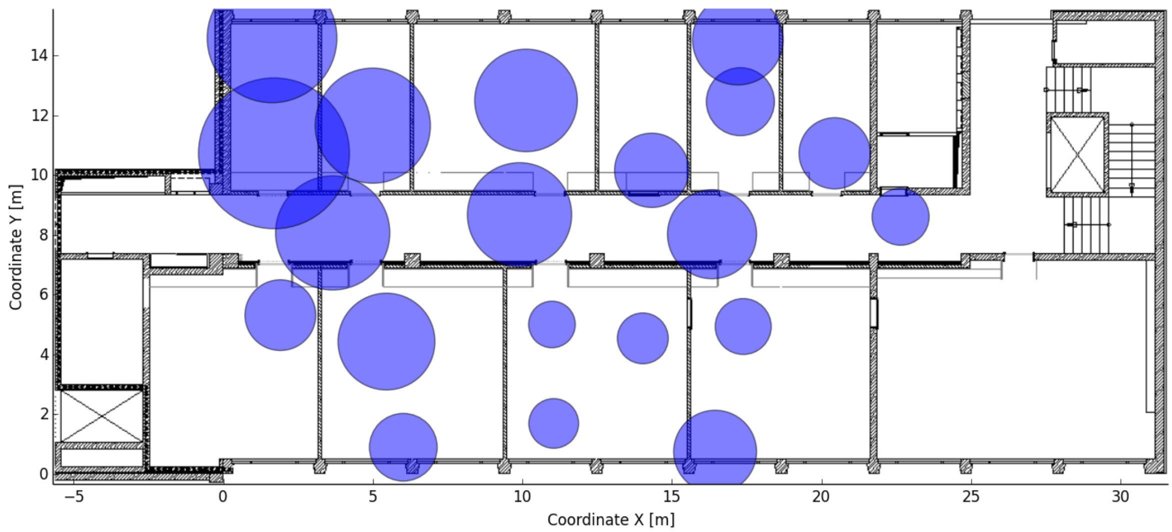

Figure 8.12: Interference scenario 3 - Android Tablet Google Nexus 7

Table 8.3: Primary metrics summary - Android Tablet Google Nexus 7

Metric	Reference scenario	Interference scenario 1	Interference scenario 2	Interference scenario 3
Mean localization error [m]	2.70	13.91	2.99	8.56
Median localization error [m]	2.52	12.17	2.48	6.71
RMS localization error [m]	3.02	16.18	3.58	10.28
75 percentile localization error [m]	3.38	19.11	4.42	11.48
90 percentile localization error [m]	4.51	26.23	5.89	14.74
Min. localization error [m]	0.51	3.79	0.28	2.36
Max. localization error [m]	5.67	33.55	7.12	24.64
Room level accuracy [%]	50.0	5.0	50.0	15.0
Mean latency [s]	3.20	3.61	3.42	5.33
Median latency [s]	3.18	3.32	3.31	3.77
RMS latency [s]	3.22	3.70	3.51	6.62
75 percentile latency [s]	3.51	3.64	3.65	5.10
90 percentile latency [s]	3.69	4.90	4.26	7.93
Min. latency [s]	2.34	2.83	2.16	2.43
Max. latency [s]	3.91	5.81	5.67	18.52

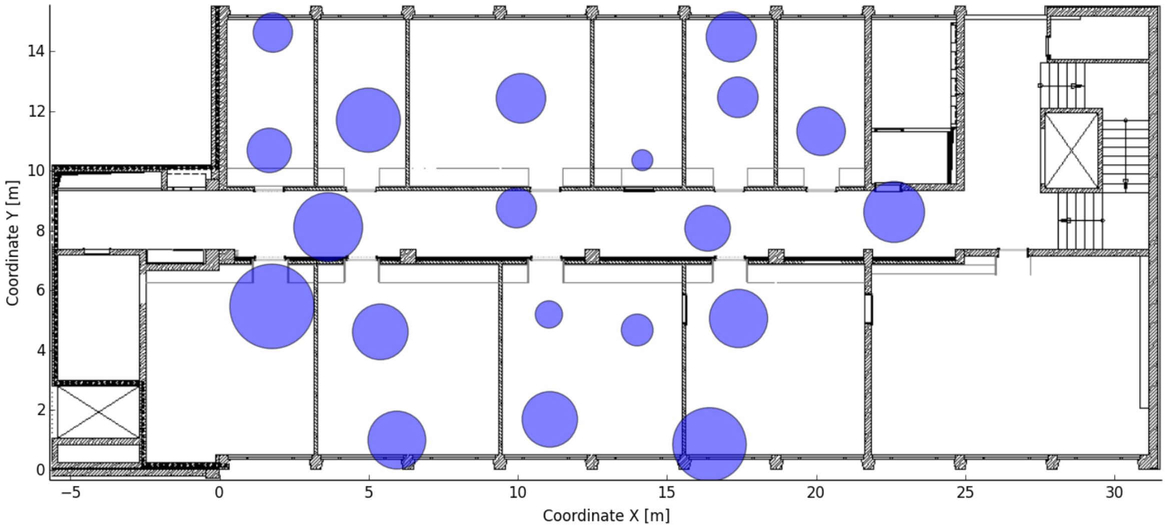

Figure 8.13: Reference scenario - Android Tablet Google Nexus 7, All APs

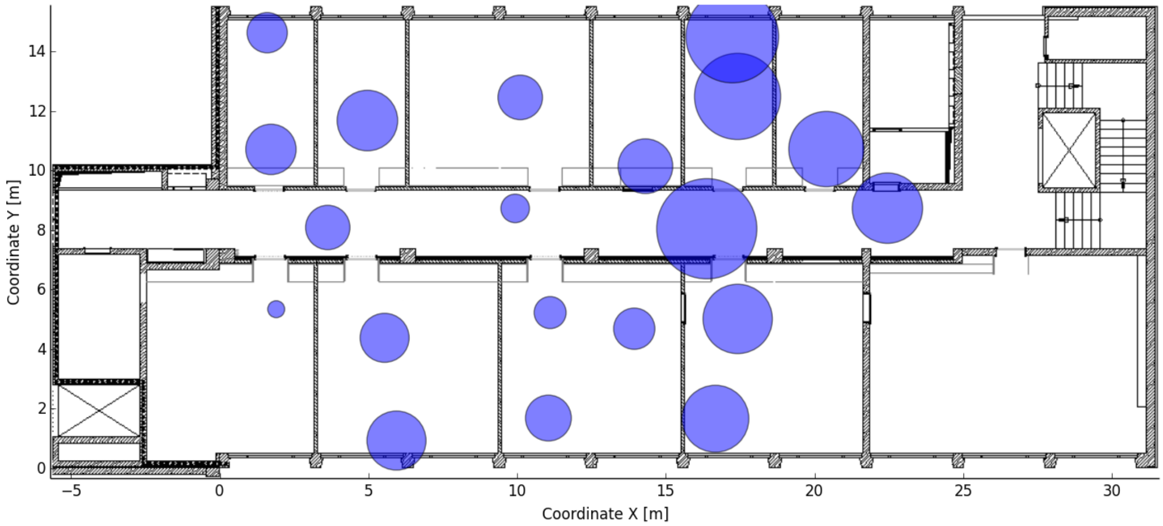

Figure 8.14: Interference scenario 1 - Android Tablet Google Nexus 7, All APs

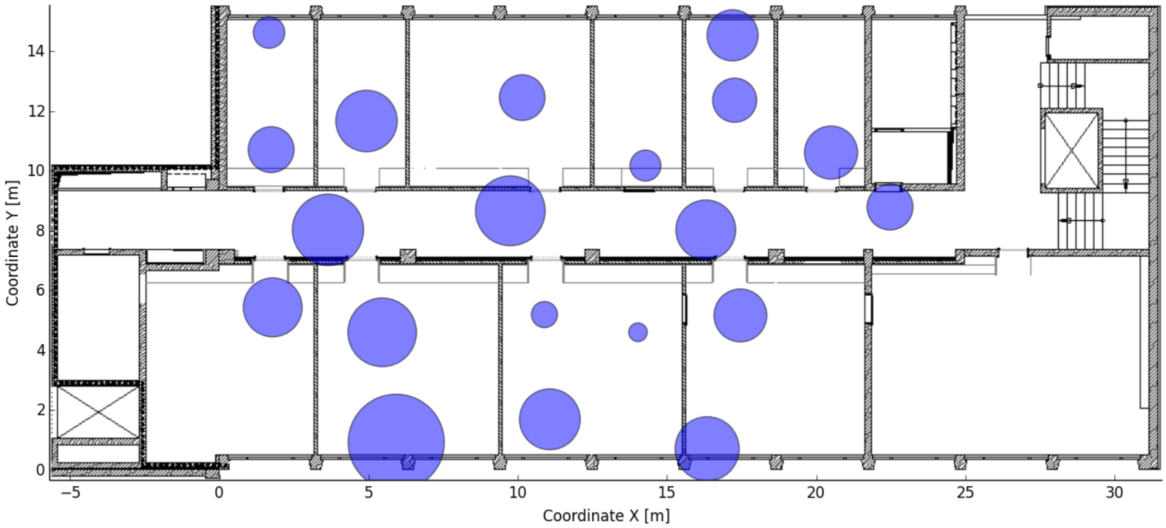

Figure 8.15: Interference scenario 2 - Android Tablet Google Nexus 7, All APs

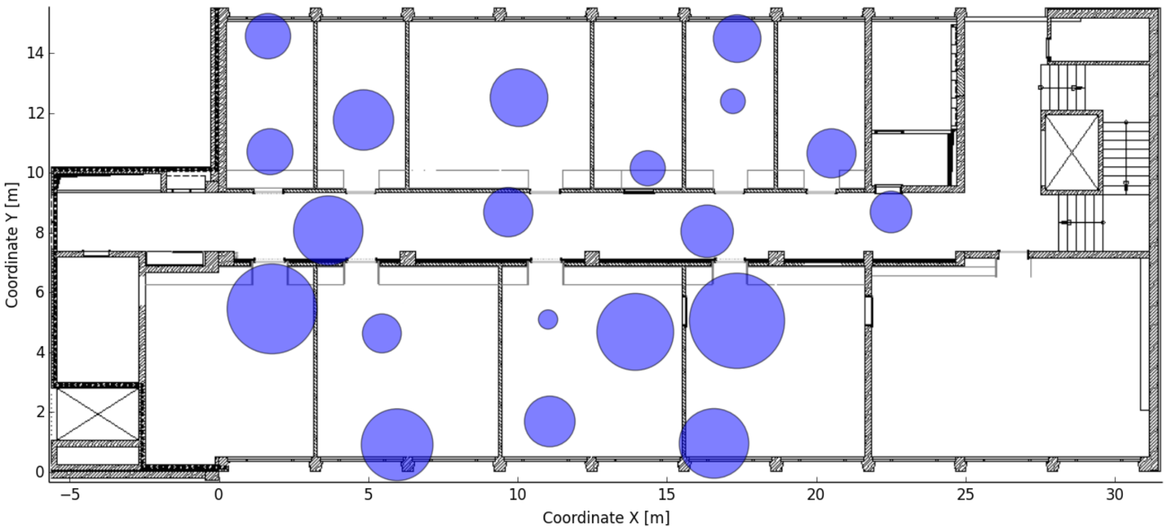

Figure 8.16: Interference scenario - Android Tablet Google Nexus 7, All APs

Table 8.4: Primary metrics summary - Android Tablet Google Nexus 7, All APs

Metric	Reference scenario	Interference scenario 1	Interference scenario 2	Interference scenario 3
Mean localization error [m]	3.06	3.97	3.37	3.62
Median localization error [m]	2.71	3.02	3.05	2.70
RMS localization error [m]	3.52	4.86	3.98	4.38
75 percentile localization error [m]	3.77	5.29	4.20	5.22
90 percentile localization error [m]	5.23	8.24	5.30	6.62
Min. localization error [m]	0.48	0.31	0.38	0.40
Max. localization error [m]	7.73	10.96	10.04	9.82
Room level accuracy [%]	50.0	40.0	50.0	45.0
Mean latency [s]	3.62	3.43	3.59	3.35
Median latency [s]	3.67	3.59	3.45	3.23
RMS latency [s]	3.66	3.46	3.63	3.41
75 percentile latency [s]	3.85	3.76	3.92	3.67
90 percentile latency [s]	4.13	3.87	4.31	4.28
Min. latency [s]	2.47	2.30	2.38	2.21
Max. latency [s]	4.74	4.06	4.96	4.39

8.3 Geo-n Localization Algorithm

Authors: S. Adler, S. Schmitt, H. Will, T. Hillebrandt, M. Kyas
Affiliation: Department of Mathematics and Computer Science, Freie Universität Berlin
E-mail: {adler, schmitt, will, hillebrandt, kyas}@inf.fu-berlin.de
Country: Germany

8.3.1 Short Description

The Geo-n algorithm was presented in 2012 by Will, Hillebrandt, and Kyas [20]. The algorithm 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 to obtain the most representative intersection points between every pair of circles induced by anchor coordinates and distance measurements. Geo-n then uses these to estimate the position of unlocalized nodes.

8.3.2 Evaluation Results

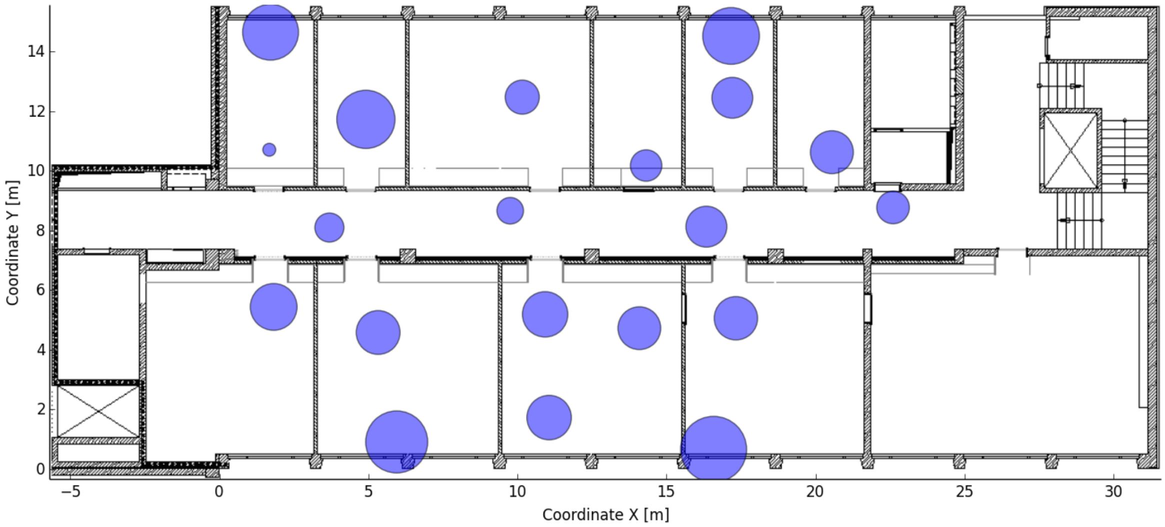

Figure 8.17: Reference scenario

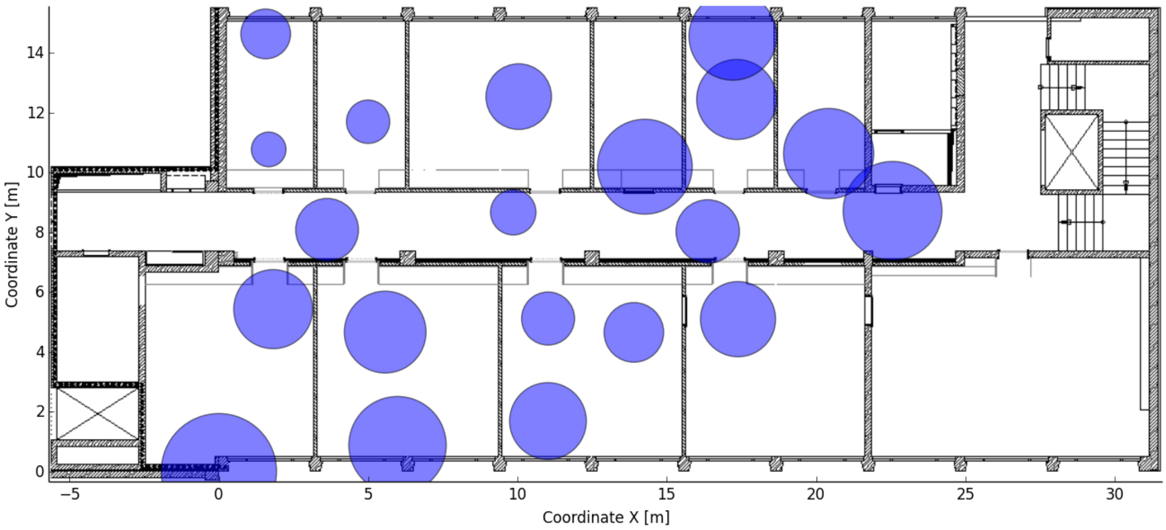

Figure 8.18: Interference scenario 1

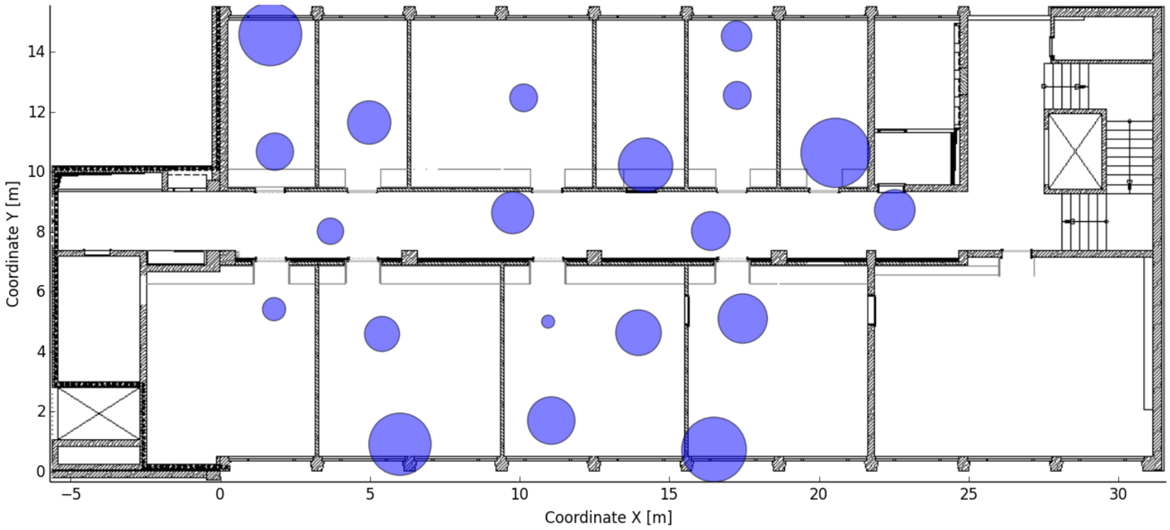

Figure 8.19: Interference scenario 2

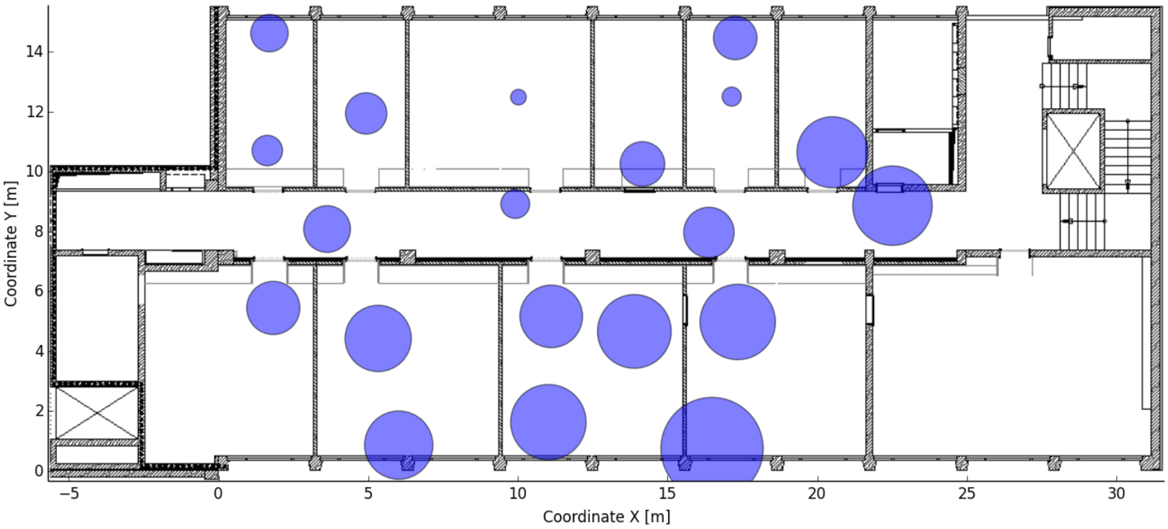

Figure 8.20: Interference scenario 3

Table 8.5: Primary metrics summary

Metric	Reference scenario	Interference scenario 1	Interference scenario 2	Interference scenario 3
Mean localization error [m]	2.18	6.21	2.15	3.70
Median localization error [m]	2.03	6.26	1.84	2.90
RMS localization error [m]	2.47	7.06	2.57	4.58
75 percentile localization error [m]	2.64	8.49	2.77	5.55
90 percentile localization error [m]	3.74	10.33	4.29	6.22
Min. localization error [m]	0.18	1.31	0.18	0.26
Max. localization error [m]	4.75	14.42	5.18	11.30
Room level accuracy [%]	60.0	25.0	65.0	35.0
Mean latency [s]	2.64	1.23	2.06	3.26
Median latency [s]	0.01	0.01	0.01	0.01
RMS latency [s]	6.36	4.05	5.19	8.30
75 percentile latency [s]	0.48	0.08	0.28	1.19
90 percentile latency [s]	7.91	1.49	7.81	13.38
Min. latency [s]	0.01	0.01	0.01	0.01
Max. latency [s]	21.45	17.28	16.06	31.06

8.4 RSS Range-based Positioning Using a Grid-based Likelihood Estimation

Authors: Y. Yang, Y. Zhao, M. Kyas
Affiliation: Department of Mathematics and Computer Science, Freie Universität Berlin
E-mail: {yuan.yang, yubin.zhao, marcel.kyas}@fu-berlin.de
Country: Germany

8.4.1 Short Description

The Received Signal Strength (RSS) ranging from indoor networks is the most convenient technique to extract for positioning purposes. Thus, our system explores RSS ranging from all reachable reference sensor nodes in indoor scenarios. For the nonlinear and non-Gaussian positioning problem under highly imprecise RSS ranging measurements, we propose a grid-based likelihood estimation (Grid-likelihood) as analytic solutions are difficult. 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; 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 as we set 36 grid cells for each positioning trial in the contest implementation, causing a low computation and memory complexity.

8.4.2 Evaluation Results

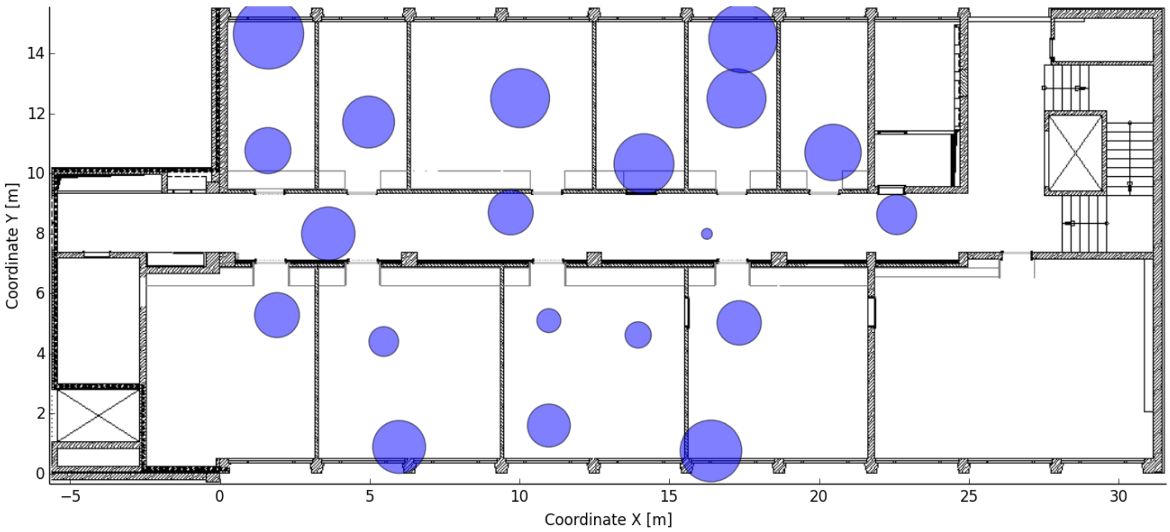

Figure 8.21: Reference scenario

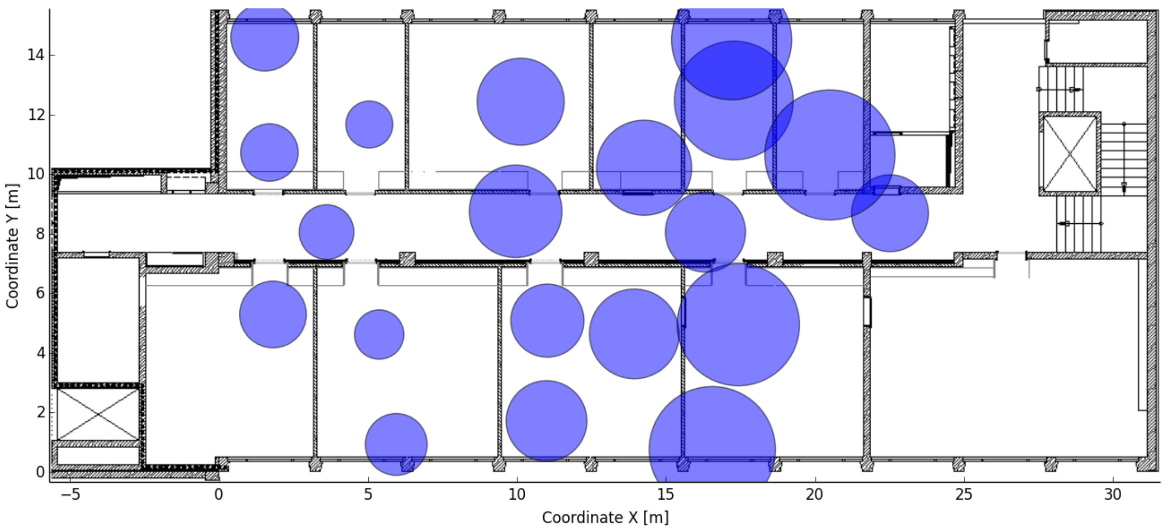

Figure 8.22: Interference scenario 1

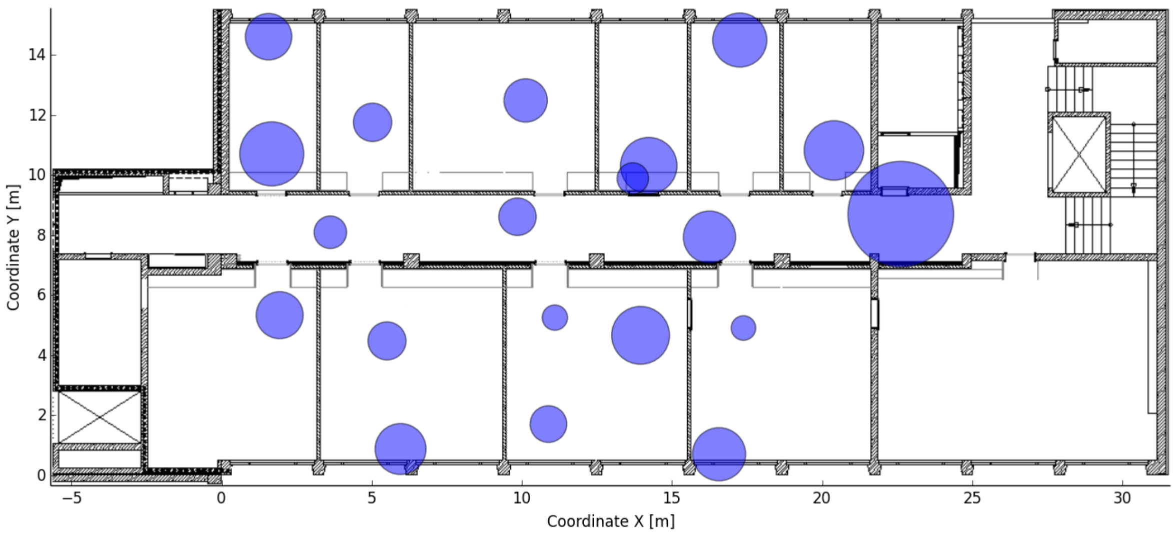

Figure 8.23: Interference scenario 2

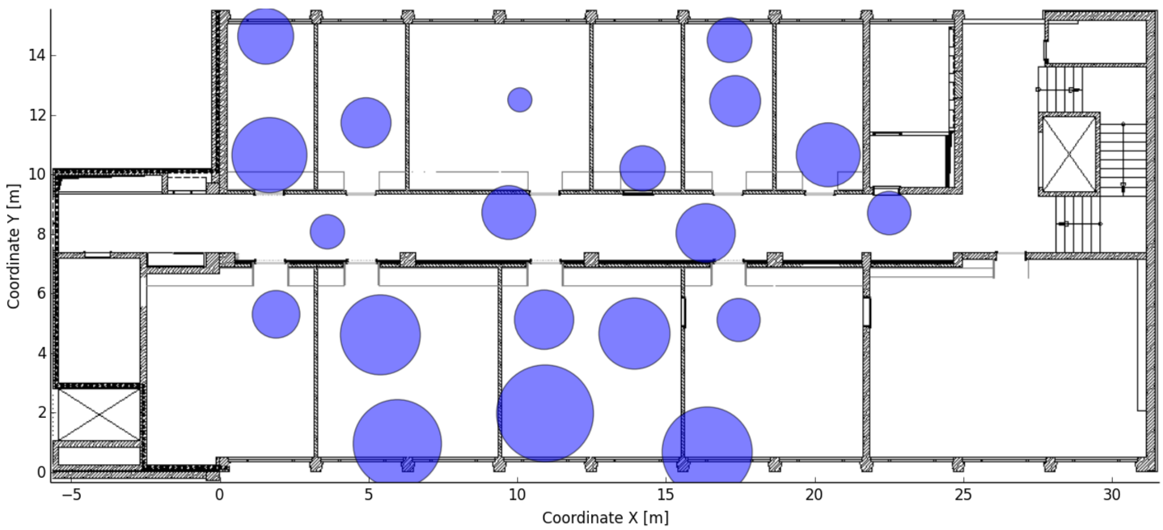

Figure 8.24: Interference scenario 3

Table 8.6: Primary metrics summary

Metric	Reference scenario	Interference scenario 1	Interference scenario 2	Interference scenario 3
Mean localization error [m]	2.66	8.60	2.77	4.18
Median localization error [m]	2.59	7.06	2.35	3.29
RMS localization error [m]	3.01	10.01	3.65	4.93
75 percentile localization error [m]	3.75	11.26	3.22	5.68
90 percentile localization error [m]	4.19	16.41	3.87	8.56
Min. localization error [m]	0.13	2.42	0.65	0.63
Max. localization error [m]	5.33	18.48	12.00	10.25
Room level accuracy [%]	45.0	15.0	55.00	15.00
Mean latency [s]	2.20	1.41	2.68	3.83
Median latency [s]	0.01	0.01	0.01	0.01
RMS latency [s]	5.49	4.00	5.34	9.00
75 percentile latency [s]	0.02	0.12	2.42	0.40
90 percentile latency [s]	11.62	3.00	8.72	15.24
Min. latency [s]	0.01	0.01	0.01	0.01
Max. latency [s]	17.03	13.79	15.12	31.05

8.5 3CoM (3 Centers of Mass) Indoor Localization Algorithm

Authors: S. Adler, S. Schmitt, M. Kyas
Affiliation: Department of Mathematics and Computer Science, Freie Universität Berlin
E-mail: {adler, schmitt, kyas}@inf.fu-berlin.de
Country: Germany

8.5.1 Short Description

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 we want to localize. This approach works well in environments with a very dense anchor placement. In our submission to the EVARILOS Open Challenge [21] we extend this approach and use the center of mass of the positions of the three strongest anchor nodes we receive as our own position. While we are aware that this approach is far from optimal we want to show that this simple method will result in remarkably good results for a lot of testing positions.

8.5.2 Evaluation Results

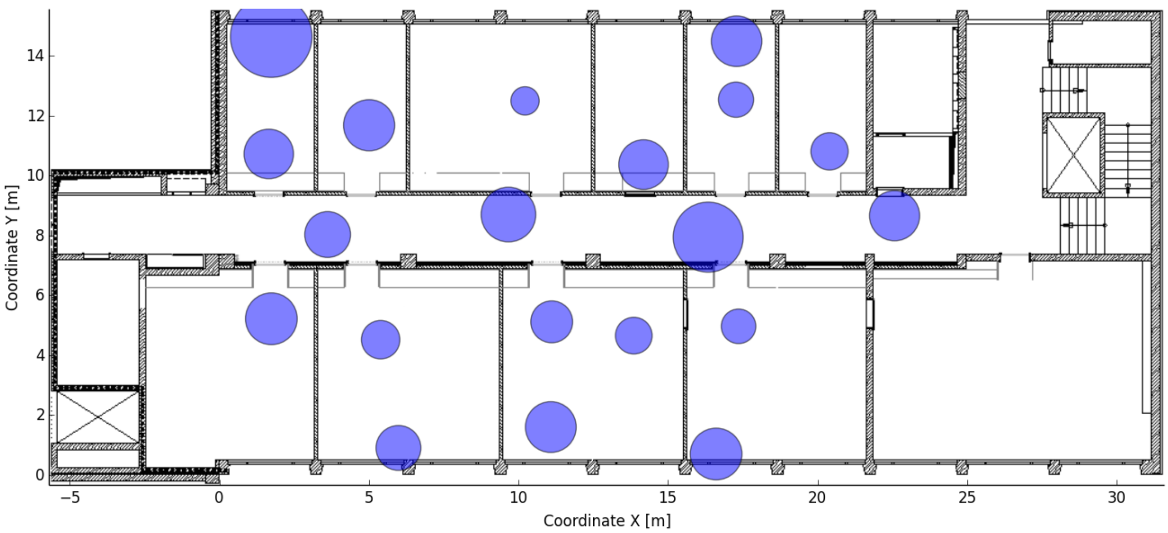

Figure 8.25: Reference scenario

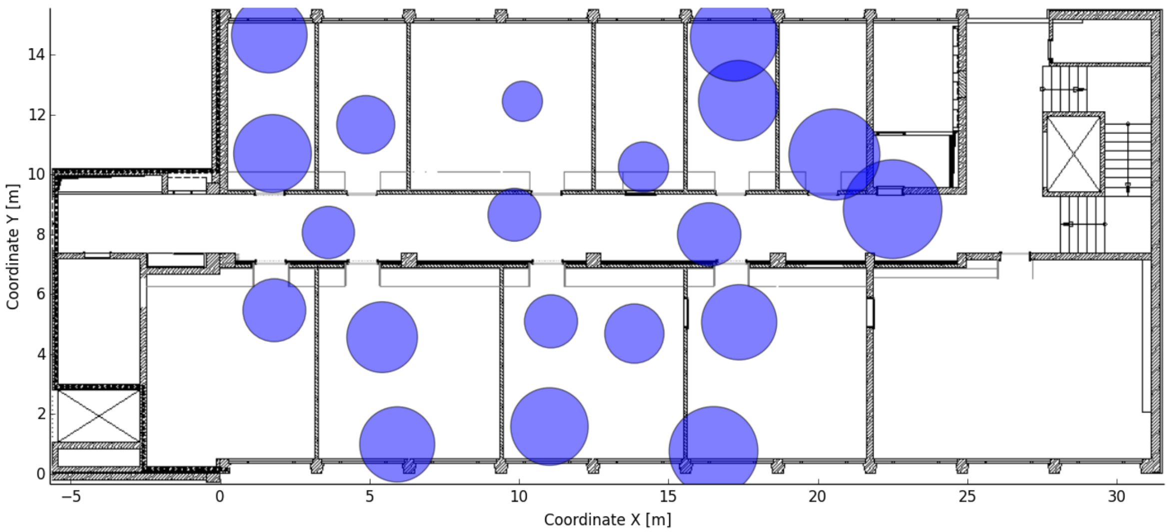

Figure 8.26: Interference scenario 1

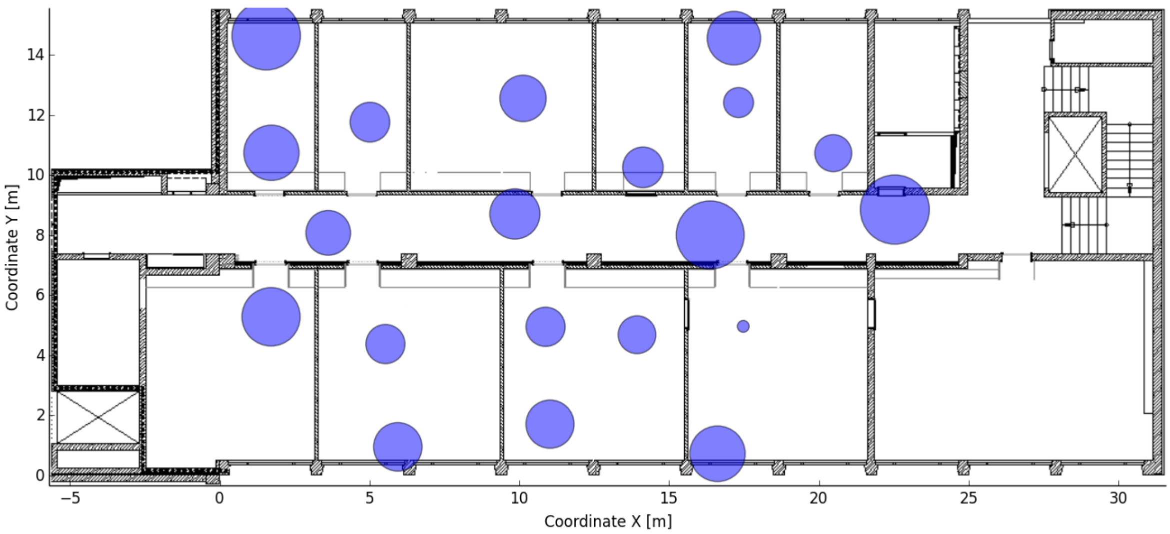

Figure 8.27: Interference scenario 2

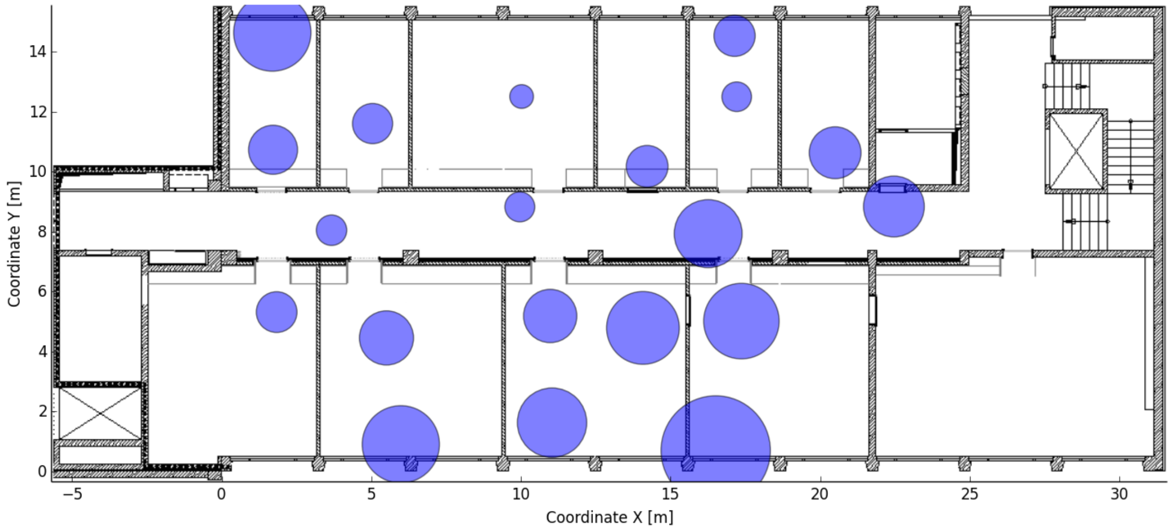

Figure 8.28: Interference scenario 3

Table 8.7: Primary metrics summary

Metric	Reference scenario	Interference scenario 1	Interference scenario 2	Interference scenario 3
Mean localization error [m]	2.62	5.50	2.59	3.72
Median localization error [m]	2.65	5.76	2.41	2.98
RMS localization error [m]	2.97	5.97	2.91	4.68
75 percentile localization error [m]	2.85	6.66	3.30	5.32
90 percentile localization error [m]	3.44	8.58	5.00	6.44
Min. localization error [m]	0.87	1.75	0.15	0.61
Max. localization error [m]	7.20	10.54	5.15	12.91
Room level accuracy [%]	55.0	20.0	65.0	35.0
Mean latency [s]	0.18	0.14	2.84	1.29
Median latency [s]	0.00	0.01	0.01	0.00
RMS latency [s]	0.68	0.45	6.75	3.59
75 percentile latency [s]	0.01	0.01	0.91	0.05
90 percentile latency [s]	0.22	0.11	11.37	4.42
Min. latency [s]	0.00	0.00	0.00	0.00
Max. latency [s]	3.01	1.76	23.17	13.72

Chapter 9

Summarized Results of Track 3 of EVARILOS Open Challenge

This chapter presents summarized final results of the Track 3 of EVARILOS Open Challenge. Since the focus of the competition was the evaluation in the scenario with interference, we firstly present the localization errors achieved by different SUTs at each evaluation point for each scenario. Further, we present errors averaged over all competitors at each evaluation point, to show the general spatial variability of localization errors due to position in the environment and interference scenarios. Finally, we present the ranking of evaluated solutions for three different evaluation scenarios, each emphasizing different indoor localization performance metrics according to the Tables 7.1, 7.2, 7.3.

9.1 Point Accuracies per Evaluation Points

As described before, we evaluated the performance of each SUT in 20 evaluation points, labeled according to Figure 9.2. The legend of evaluated systems is given in Figure 9.1. Localization errors per competitors in each evaluation point in the reference scenario are given in Figure 9.3, while the averaged errors per evaluation points are given in Figure 9.4. As visible in the figures, in general higher localization errors are achieved close to the borders, i.e. outside walls, of the environments, for example evaluation points 6, 10, 17 and 20. Consequently, in the center of the environment smaller localization errors are achieved, e.g. in evaluation points 2, 3, 9 and 14.

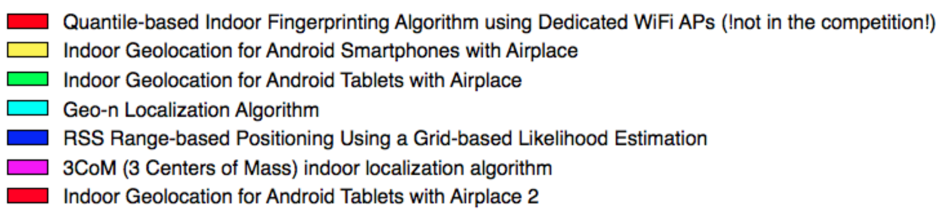

Figure 9.1: Legend: Evaluated SUTs

Figure 9.2: Labelled evaluation points

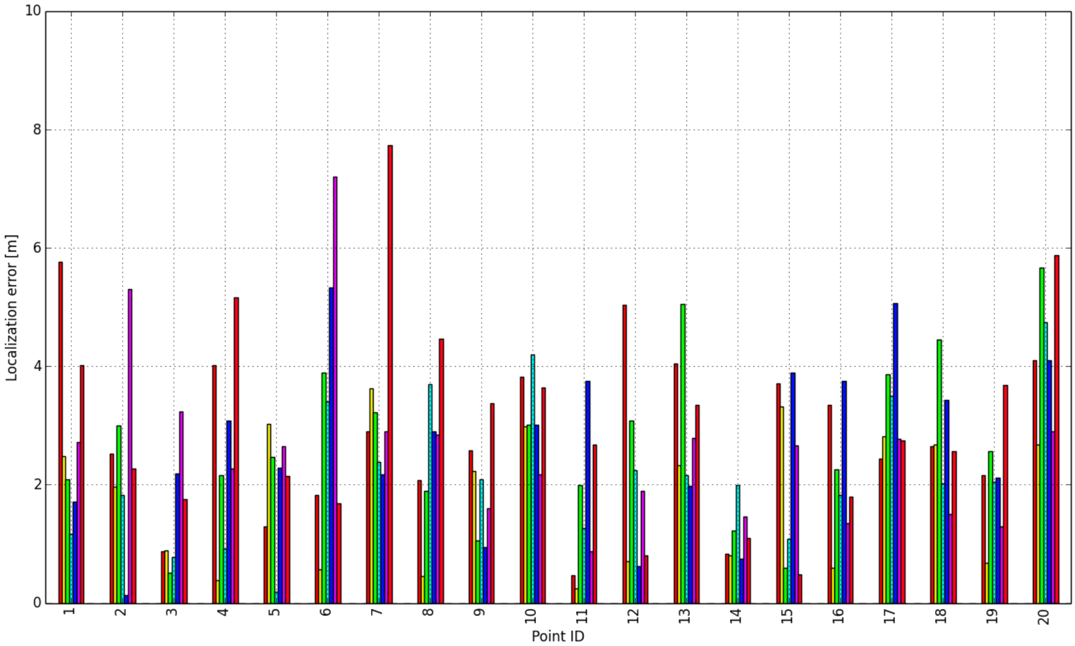

Figure 9.3: Reference scenario: average localization errors per evaluation points

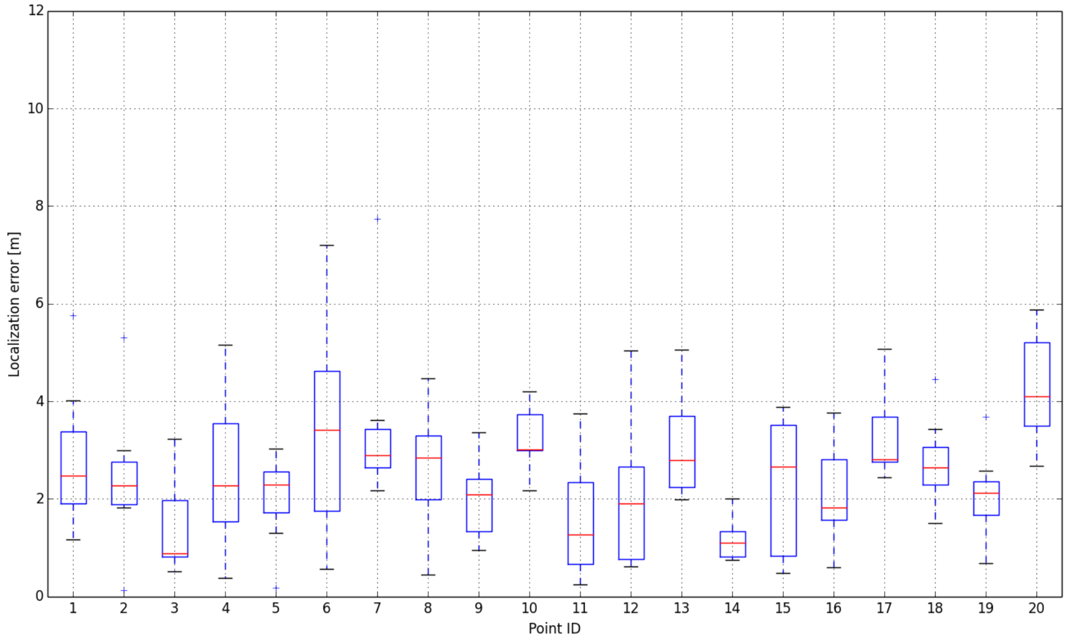

Figure 9.4: Reference scenario summary

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 solutions, the higher localization errors are generally obtained. In the interference scenario 1, jamming on one IEEE 802.15.4 channel, the achieved localization errors per competitor at each evaluation point are given in Figure 9.6, while Figure 9.5 depicts the averaged localization errors for all competitor at each evaluation point. It is clear from the figures that this interference scenario significantly degrades the point accuracy of all evaluated SUTs. Particularly interesting is to observe the tendency of achieving high localization errors in the evaluation points close to the sources of interference, i.e. evaluation points 1, 16, 17, 18, 19 and 20.

Significantly smaller influence on the accuracy of indoor localization is achieved in the interference scenario 2. However, on some solutions the influence is still visible, as shown in Figures 9.7 and 9.8. One reason for the lower impact of this interference scenario on the accuracy of indoor localization is the fact that in this scenario usual WiFi traffic was generated, i.e. Carrier Sense Multiple Access (CSMA) was enabled. In other words, the results achieved in the interference scenario 2 confirm the value of frequency agility approaches 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 using signal generator, the performance of all evaluated SUTs is again significantly degraded compared to the reference scenario, as shown in Figures 9.9 and 9.10. Similarly to the interference scenario 1, the localization errors are higher in the evaluation points close to the interference source, e.g. points 19 and 20.

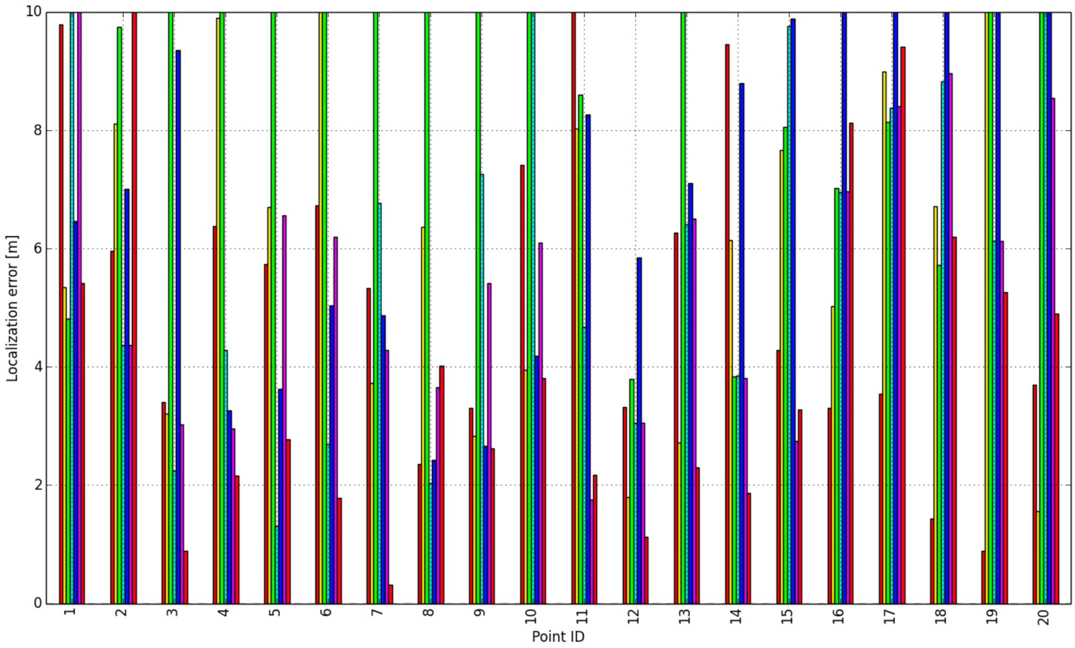

Figure 9.5: Interference scenario 1: average localization errors per evaluation points

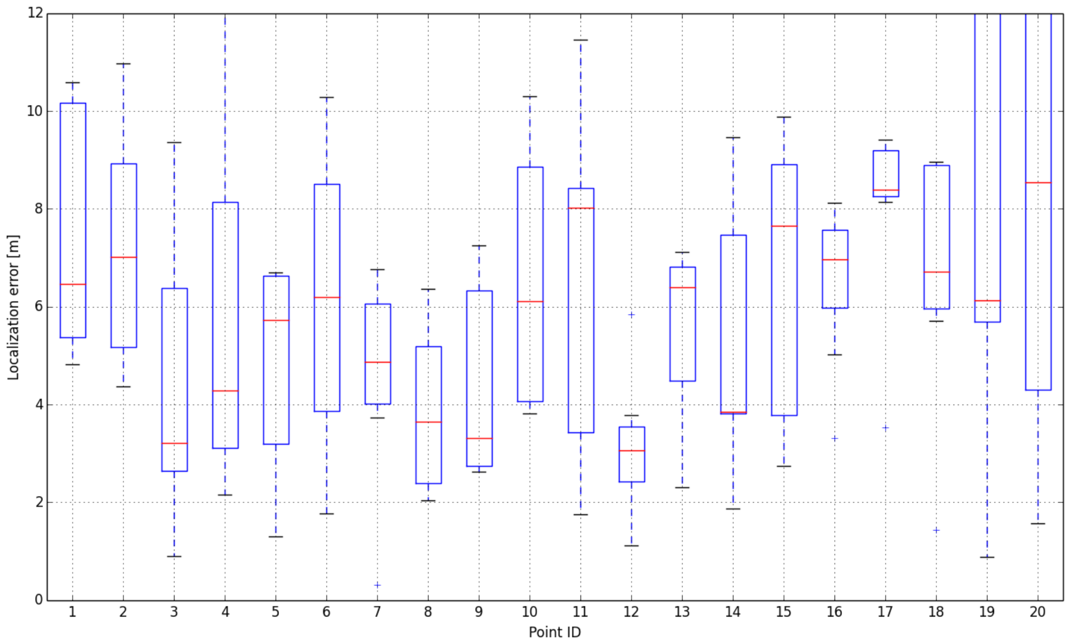

Figure 9.6: Interference scenario 1 summary

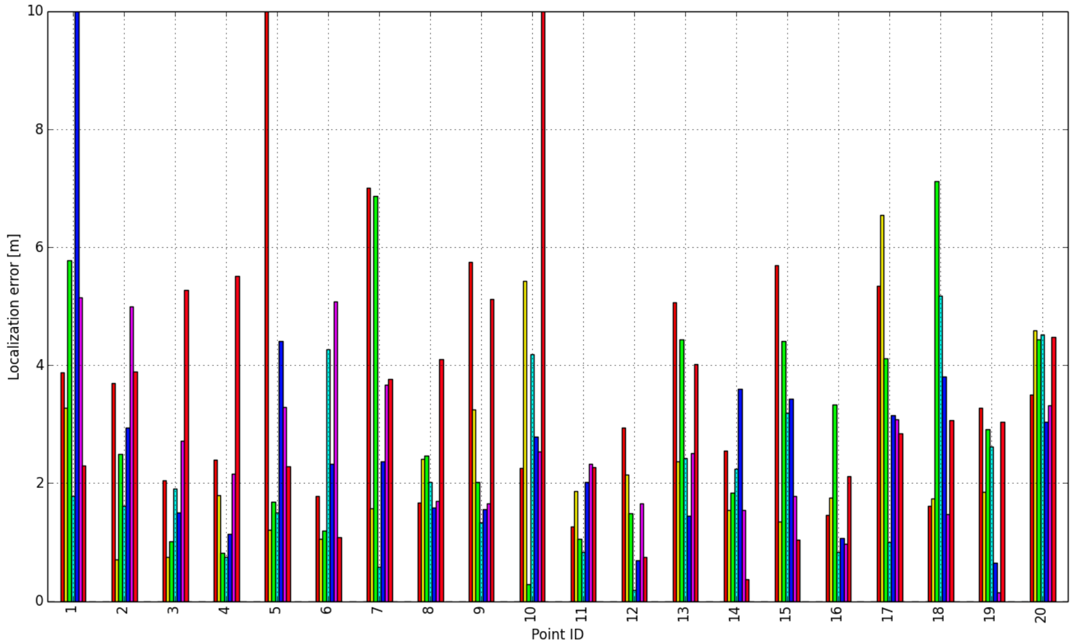

Figure 9.7: Interference scenario 2: average localization errors per evaluation points

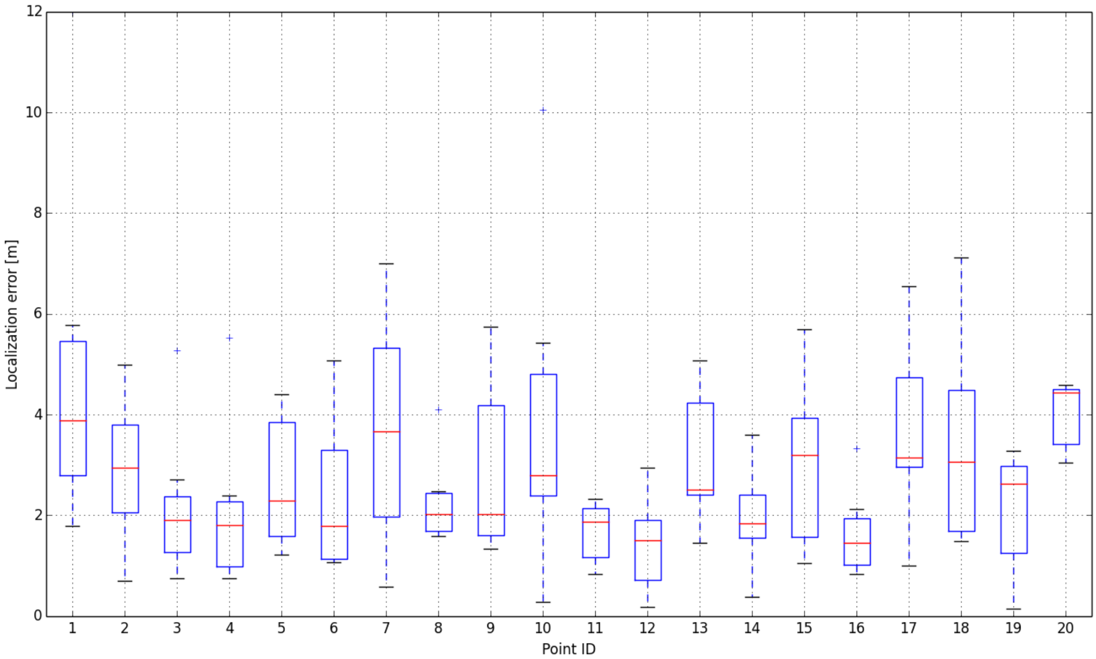

Figure 9.8: Interference scenario 2 summary

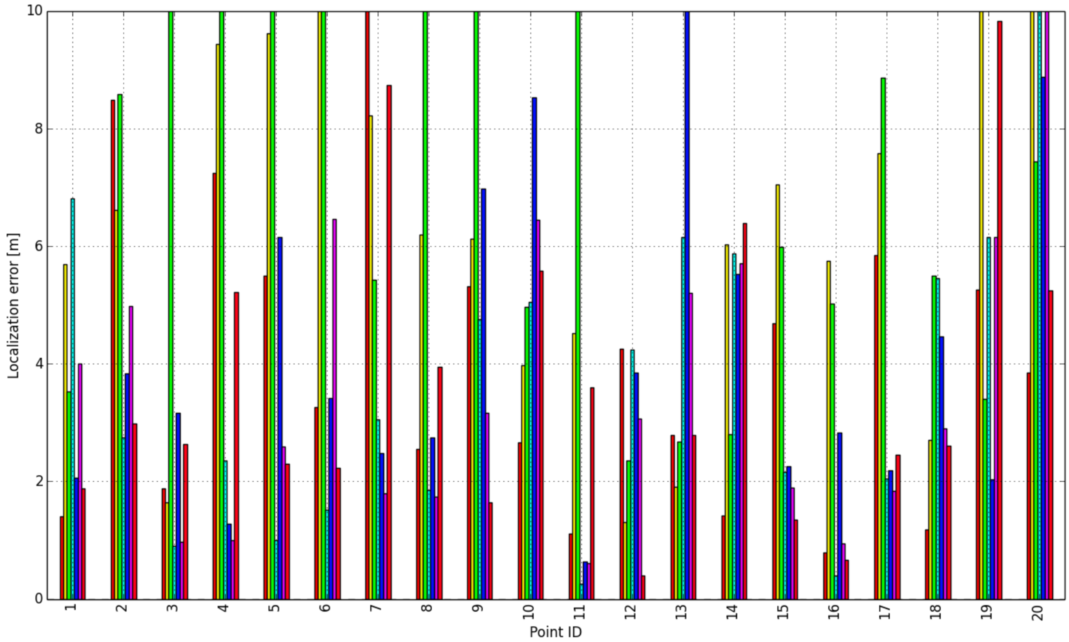

Figure 9.9: Interference scenario 3: average localization errors per evaluation points

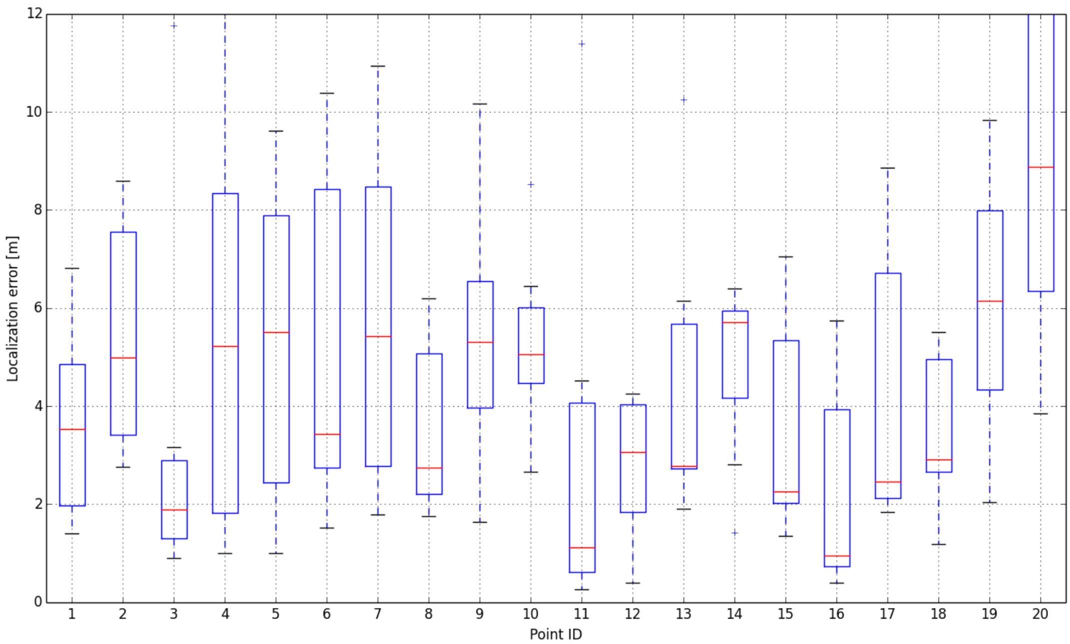

Figure 9.10: Interference scenario 3 summary

9.2 Final results of EVARILOS Open Challenge - Track 3

The ranking of competitors in Track 3 of the EVARILOS Open Challenge for three evaluation scenarios is given in Tables 9.1, 9.2, 9.3. First evaluation scenario focuses on the point and room level accuracy of indoor localization by giving the highest score to these metrics. The best performance is achieved by the fingerprinting-based algorithm named "Indoor Geolocation for Android Smartphones with Airplace", achieving the average localization error of only 1.77 m and room accuracy of 80% in the reference scenario, and the final score 7.13/10. Note that the point accuracy in the tables is the 75 percentile localization error, which was the metric used for calculating final scores.

Table 9.1: Summary results of the evaluation scenario 1

SUT	Point acc. [m]	Room acc. [%]	Latency [s]	Interference sensitivity [%]	Point acc. score	Room acc. score	Latency score	Interference sens. score	Final score
Indoor Geo. for Android Smartphones with Airplace	2.71	80.00	3.07	61.43	8.10	7.50	8.91	0.00	7.13
Quantile-based Indoor Fingerprinting Algorithm	3.87	70.00	20.11	28.09	6.81	5.00	0.00	5.48	5.27
Geo-n Localization Algorithm	2.64	60.00	0.48	62.50	8.18	2.50	10.00	0.00	5.27
3CoM (3 Centers of Mass) Indoor Localization	2.85	55.00	0.01	37.30	7.94	1.25	10.00	3.17	4.99
Indoor Geo. for Android Tablets with Airplace 2	3.77	50.00	3.85	14.17	6.92	0.00	8.50	8.96	4.51
RSS Range-based Positioning	3.75	45.00	0.02	42.75	6.94	0.00	10.00	1.81	3.96
Indoor Geo. for Android Tablets with Airplace	3.38	50.00	3.51	99.48	7.35	0.00	8.68	0.00	3.81
Indoor Geo. for Android Smartphones with Airplace 2	10.65	10.00	2.82	5.12	0.00	0.00	9.04	10.00	1.90

In the evaluation scenario 2, emphasizing the response time or latency of evaluated algorithms, the winner is again the fingerprinting-based algorithm named "Indoor Geolocation for Android Smartphones with Airplace", as shown in Table 9.2. 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 the 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 based on low-power sensor nodes, named "Geo-n Localization Algorithm".

Table 9.2: Summary results of the evaluation scenario 2

SUT	Point acc. [m]	Room acc. [%]	Latency [s]	Interference sensitivity [%]	Point acc. score	Room acc. score	Latency score	Interference sens. score	Final score
Indoor Geo. for Android Smartphones with Airplace	2.71	80.00	3.07	61.43	8.10	7.50	8.91	0.00	7.58
3CoM (3 Centers of Mass) Indoor Localization	2.85	55.00	0.01	37.30	7.94	1.25	10.00	3.17	7.16
Geo-n Localization Algorithm	2.64	60.00	0.48	62.50	8.18	2.50	10.00	0.00	7.14
RSS Range-based Positioning	3.75	45.00	0.02	42.75	6.94	0.00	10.00	1.81	6.57
Indoor Geo. for Android Tablets with Airplace 2	3.77	50.00	3.85	14.17	6.92	0.00	8.50	8.96	6.53
Indoor Geo. for Android Tablets with Airplace	3.38	50.00	3.51	99.48	7.35	0.00	8.68	0.00	5.81
Indoor Geo. for Android Smartphones with Airplace 2	10.65	10.00	2.82	5.12	0.00	0.00	9.04	10.00	5.52
Quantile-based Indoor Fingerprinting Algorithm	3.87	70.00	20.11	28.09	6.81	5.00	0.00	5.48	2.91

Finally, the ranking of competitors in the evaluation scenario 3, emphasizing the interference sensitivity in different interference scenarios, is given in Table 9.3. The winning algorithm is the fingerprinting-based algorithm named "Indoor Geo. for Android Tablets with Airplace 2". Although the other performance metrics achieved by this algorithm in the reference scenario are average, compared to some 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 evaluation scenario. Except for the winning algorithm, the algorithm "Indoor Geo. for Android Smartphones with Airplace 2" achieved a perfect interference sensitivity score, reason for that being extremely poor accuracy of location estimation, probably due to the errors in the deployment phase. The errors caused similarly bad performance of this algorithm in the interference scenarios, resulting in a high interference sensitivity score. However, it is clear from the final scores of all other competitors that RF interference causes the degradation in the performance of RF-based indoor localization.

Table 9.3: Summary results of the evaluation scenario 3

SUT	Point acc. [m]	Room acc. [%]	Latency [s]	Interference sensitivity [%]	Point acc. score	Room acc. score	Latency score	Interference sens. score	Final score
Indoor Geo. for Android Tablets with Airplace 2	3.77	50.00	3.85	14.17	6.92	0.00	8.50	8.96	6.71
Indoor Geo. for Android Smartphones with Airplace 2	10.65	10.00	2.82	5.12	0.00	0.00	9.04	10.00	5.90
Quantile-based Indoor Fingerprinting Algorithm	3.87	70.00	20.11	28.09	6.81	5.00	0.00	5.48	5.10
3CoM (3 Centers of Mass) Indoor Localization	2.85	55.00	0.01	37.30	7.94	1.25	10.00	3.17	4.43
Indoor Geo. for Android Smartphones with Airplace	2.71	80.00	3.07	61.43	8.10	7.50	8.91	0.00	4.01
RSS Range-based Positioning	3.75	45.00	0.02	42.75	6.94	0.00	10.00	1.81	3.29
Geo-n Localization Algorithm	2.64	60.00	0.48	62.50	8.18	2.50	10.00	0.00	3.14
Indoor Geo. for Android Tablets with Airplace	3.38	50.00	3.51	99.48	7.35	0.00	8.68	0.00	2.34

Chapter 10

Conclusion

This document presented the EVARILOS Open Challenge - Track 3 in a following manner. It shortly described the TKN testbed environment and infrastructure, and how it can be locally or remotely used in a centralized and repeatable way for evaluation and benchmarking of indoor localization algorithms. Moreover, the document gave directions and guidelines for the competitors on how and on which devices to deploy their systems and what are the requirements for interfacing the testbed's infrastructure. Furthermore, it presented the scenarios in which different indoor localization algorithms will be evaluated, each scenario described with a set of evaluation points in the TKN testbed environment and the interference patterns generated in order to evaluate the impact of interference on the evaluated algorithms. The document also presented the procedure to be followed in the evaluation of the performance, describing which metrics are used, how they are calculated and how the final scores are obtained in three different evaluation scenarios. Finally, the document presented the results of the evaluation and ranking of the evaluated systems in three different scenarios. The evaluation results how the good performance of the evaluated algorithms, with best performance being 1.77 m in the average localization error, 80% in the room level accuracy and less then 1 sec in the response time. However, results also show that the generated interference pattern highly influence the performance of the evaluated systems, which motivates for a further in-depth analysis on how to mitigate observed negative impact.

Chapter 11

Appendix

11.1 Quantile-based Indoor Fingerprinting using Dedicated WiFi APs

Reference scenario											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222	•	•									
FT223		•									
FT224			•••								
FT225				••							
FT226											
FT230											
FT231											
FT232					•	•					
FT233											
FT234						•	•	•			
FT235											
FT236									•	••	•
hallway											•••
stairs											
no_room											
Interference scenario 1											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222	•										
FT223											
FT224			••								
FT225				••							
FT226											
FT230											
FT231					•						
FT232						•					
FT233						•					
FT234							•	•		•	••
FT235			•								
FT236									•		
hallway		••								•	••
stairs											
no_room											
Interference scenario 2											

	FT222	FT223	FT224	FT225	FT231	FT232	FT233	FT234	FT235	FT236	hallway
FT222											
FT223		•									
FT224			••							•	
FT225			•	••							
FT226											
FT230											
FT231						•					•
FT232					•						
FT233						•					
FT234		•						•			•
FT235									•		
FT236	•									•	•
hallway							•				•
stairs											
no_room											
Interference scenario 3											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway_2nd**
FT222											
FT223		•									
FT224			••								
FT225			•	•							
FT226											
FT230											
FT231											
FT232						•					
FT233											
FT234								•			•
FT235											
FT236									•	•	
hallway	•	•		•	•	•	•			•	••
stairs											•
no_room											

11.2 Indoor Geolocation for Android Smartphones with Airplace

Reference scenario											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	•	••									
FT224			•••	•							
FT225				•							
FT226											
FT230											
FT231					•						
FT232						••	•				
FT233											
FT234								•			
FT235									•	•	

FT236										•	
hallway											••••
stairs											
no_room											
Interference scenario 1											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222		•									
FT223	•	•	•								
FT224			••								
FT225				•							
FT226											
FT230											•
FT231					•	•	•				
FT232								•			
FT233											
FT234									•	•	
FT235											
FT236											
hallway										•	••
stairs											
no_room				•							•
Interference scenario 2											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222	•	•									
FT223		•									
FT224			•••	•							
FT225				•							
FT226											
FT230											
FT231					•	•					
FT232						•					
FT233							•				
FT234								•			
FT235									•		
FT236										••	
hallway											••••
stairs											
no_room											
Interference scenario 3											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223											
FT224	•	••	••								
FT225											
FT226											
FT230					•	•					•
FT231											
FT232											
FT233								•			
FT234									•	•	

FT235											
FT236											
hallway						•	•			•	••
stairs											
no_room			•	•							•

11.3 Indoor Geolocation for Android Tablets with Airplace

Reference scenario											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	•	••									
FT224			••	••							
FT225											
FT226											
FT230											•
FT231					•	•					
FT232											
FT233							•				
FT234								•	•		
FT235										•	
FT236											
hallway			•							•	•••
stairs											
no_room						•					
Interference scenario 1											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223											
FT224			•								
FT225			•								
FT226				•							
FT230					•	••					
FT231											
FT232											
FT233											
FT234											
FT235											
FT236											
hallway											•
stairs											
no_room	•	••	•	•			•	•	•	••	•••
Interference scenario 2											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	•	••	•								
FT224			••	••							
FT225											
FT226											

FT230											•
FT231						•					
FT232							•				
FT233											
FT234								•			•
FT235									•		
FT236										••	
hallway											••
stairs											
no_room					•	•					
Interference scenario 3											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	•	•									
FT224		•	••					•			
FT225			•								
FT226											••
FT230						•					
FT231					•		•				
FT232											
FT233											
FT234											
FT235											
FT236											
hallway				•		•					
stairs											
no_room				•					•	••	••

11.4 Indoor Geolocation for Android Tablets with Airplace, All APs

Reference scenario											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222		•									
FT223		•									
FT224			•••								
FT225				•							
FT226				•							
FT230											•
FT231					•	•					
FT232						•					
FT233							•				•
FT234								•			•
FT235											
FT236	•								•	••	•
hallway											
stairs											
no_room											
Interference scenario 1											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**

FT222	•	•									
FT223		•									
FT224			•••								
FT225											
FT226				••							
FT230											
FT231											
FT232											
FT233											
FT234							•	•			•
FT235											
FT236									•	••	•
hallway											••
stairs											
no_room					•	••					
Interference scenario 2											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223											
FT224			•••								
FT225				••							
FT226											
FT230											
FT231					•	••					•
FT232											
FT233							•				•
FT234								•			
FT235											•
FT236	•	•							•	••	•
hallway		•									
stairs											
no_room											
Interference scenario 3											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222		•									
FT223		•									
FT224			••								
FT225			•								
FT226				•							
FT230											
FT231					•						•
FT232						••					
FT233							•	•			•
FT234											•
FT235											
FT236	•								•	••	•
hallway											
stairs											
no_room											

11.5 Geo-n Localization Algorithm

Reference scenario											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	•	••									
FT224			•••	••							
FT225											
FT226											
FT230											•
FT231											
FT232					•	••					•
FT233							•				
FT234								•			
FT235										•	
FT236										•	
hallway									•		••
stairs											
no_room											
Interference scenario 1											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223											
FT224											
FT225											
FT226											
FT230											
FT231											
FT232											
FT233											
FT234											
FT235		•					•	•	•		
FT236										•	•
hallway	•	•	•••	•	•	••					•••
stairs											
no_room				•						•	
Interference scenario 2											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222	•										
FT223		••									
FT224			•••	••							
FT225											
FT226											
FT230											•
FT231											
FT232						••					•
FT233					•						
FT234							•	•			
FT235									•		
FT236										•	
hallway											••

stairs											
no_room										●	
Interference scenario 3											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222		●									
FT223			●								
FT224											
FT225											
FT226											
FT230											
FT231											
FT232				●		●●					●
FT233					●		●				
FT234								●	●		
FT235											
FT236										●●	●
hallway	●	●	●●	●				●			●●
stairs											
no_room											

11.6 RSS Range-based Positioning Using a Grid-based Likelihood Estimation

Reference scenario											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	●	●●									●
FT224			●●●	●●							
FT225											
FT226											
FT230											
FT231					●	●●					
FT232							●				
FT233											
FT234									●		
FT235										●	
FT236											
hallway											●●●
stairs											
no_room								●		●	
Interference scenario 1											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223		●	●●●								
FT224											
FT225											●●
FT226											
FT230											
FT231											

FT232											
FT233											
FT234											
FT235							•				
FT236	•			••	•	••		•	•	•	•
hallway		•									•
stairs											
no_room										•	
Interference scenario 2											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	•	••									
FT224			•••	•							
FT225				•							
FT226											
FT230											
FT231					•	•					
FT232											•
FT233							•				
FT234							•	•	•		
FT235										••	
FT236											
hallway											••
stairs											
no_room											•
Interference scenario 3											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223											
FT224											
FT225											
FT226											
FT230											
FT231						•					
FT232					•	•					•
FT233				•							•
FT234							•	•	•	•	
FT235										•	
FT236											•
hallway	•	•	••	•							•
stairs											
no_room		•	•								

11.7 3CoM (3 Centers of Mass) Indoor Localization Algorithm

Reference scenario											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	•	••									•

FT224			●●●								
FT225				●●							
FT226											
FT230											
FT231					●						●
FT232						●●	●				●
FT233											
FT234								●		●	
FT235										●	●
FT236											
hallway									●		
stairs											
no_room											
Interference scenario 1											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223		●									
FT224											
FT225											
FT226											
FT230											
FT231											
FT232											
FT233											
FT234							●	●	●	●●	●
FT235	●	●									●
FT236											
hallway			●●●	●●	●	●●					●●
stairs											
no_room											
Interference scenario 2											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											
FT223	●	●●									●
FT224			●●●								
FT225				●●							
FT226											
FT230											
FT231					●						
FT232						●●					●●
FT233							●				
FT234								●			
FT235									●	●	●
FT236											
hallway										●	
stairs											
no_room											
Interference scenario 3											
	FT222	**FT223**	**FT224**	**FT225**	**FT231**	**FT232**	**FT233**	**FT234**	**FT235**	**FT236**	**hallway**
FT222											

FT223	•										
FT224											
FT225											
FT226											
FT230											
FT231											•
FT232				•	•	••					•
FT233							•				
FT234			•					•			
FT235									•	•	
FT236											
hallway		••	••	•						•	••
stairs											
no_room											

References

[1] T. Van Haute, E. De Poorter, J. Rossey, I. Moerman, V. Handziski, A. Behboodi, F. Lemic, A. Wolisz, N. Wiström, T. Voigt, P. Crombez, P. Verhoeve, and J. J. De Las Heras, "The EVARILOS Benchmarking Handbook: Evaluation of RF-based Indoor Localization Solutions," in *Proceedings of 2nd International Workshop on Measurement-based Experimental Research, Methodology and Tools (MERMAT'13)*, May 2013.

[2] "Turtlebot II Documentation," 2013. Available at `http://wiki.ros.org/Robots/TurtleBot`.

[3] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop*, 2009.

[4] "N750 Dualband WLAN Router," 2013. Available at `http://www.tp-link.com.de/products/?categoryid=2166`.

[5] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Wireless mac processors: programming mac protocols on commodity hardware," in *INFOCOM, 2012 Proceedings IEEE*, pp. 1269–1277, IEEE, 2012.

[6] "Alix2D2," 2013. Available at `http://www.pcengines.ch/alix2d2.htm`.

[7] "R&S FSV7 Signal Analyzer," 2013. Available at `http://www.rohde-schwarz.de/de/Produkte/messtechnik-testsysteme/signal-und-spektrumanalysatoren/FSV-%7C-%C3%9Cberblick-%7C-100-%7C-6392.html`.

[8] "R&S SMBV100A Vector Signal Generator," 2013. Available at `http://www.rohde-schwarz.de/de/Produkte/messtechnik-testsysteme/signalgeneratoren/SMBV100A.html`.

[9] F. Lemic, J. Büsch, M. Chwalisz, V. Handziski, and A. Wolisz, "Demo Abstract: Testbed Infrastructure for Benchmarking RF-based Indoor Localization Solutions under Controlled Interference," in *EWSN'14, University of Oxford, Oxford, UK*, 2014.

[10] F. Lemic, J. Büsch, M. Chwalisz, V. Handziski, and A. Wolisz, "Infrastructure for Benchmarking RF-based Indoor Localization under Controlled Interference," in *Proceedings of 3rd International Conference on Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS'14)*, November 2014.

[11] W. Liu, M. Mehari, S. Bouckaert, L. Tytgat, I. Moerman, and P. Demeester, "Demo Abstract: A Proof of Concept Implementation for Cognitive Wireless Sensor Network on a Large-scale Wireless Testbed," in *EWSN'13*, 2013.

[12] "CREW Repository," 2013. Available at `http://www.crew-project.eu/repository`.

[13] "OMF 6 Documentation," 2013. Available at `http://omf.mytestbed.net/projects/omf6/wiki/Wiki`.

[14] "Metageek Wi-Spy," 2013. Available at `http://www.metageek.net/products/wi-spy`.

[15] "OpenWrt - Wireless Freedom," 2014. Available at `https://www.openwrt.org/`.

[16] F. Lemic, "Benchmarking of Quantile-based Indoor Fingerprinting Algorithm," TKN Technical Report Series TKN-14-001, Telecommunication Networks Group, Technische Universität Berlin, July 2014.

[17] F. Lemic, A. Behboodi, V. Handziski, and A. Wolisz, "Experimental Decomposition of the Performance of Fingerprinting-based Localization Algorithms," in *Proceedings of 5th International Conference on Indoor Positioning and Indoor Navigation (IPIN'14)*, October 2014.

[18] T. Constambeys, C. Laoudias, D. Zeinalipour-Yazti, and C. G. Panayiotou, "Indoor Geolocation for Android Smartphones with Airplace," Short paper, EVARILOS Open Challenge - Track 3, May 2014.

[19] C. Laoudias, P. Kemppi, and C. G. Panayiotou, "Localization using radial basis function networks and signal strength fingerprints in wlan," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6, IEEE, 2009.

[20] H. Will, T. Hillebrandt, and M. Kyas, "The Geo-n localization algorithm," in *Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–10, 2012.

[21] S. Adler, S. Schmitt, and M. Kyas, "3CoM (3 Centers of Mass) Indoor Localization Algorithm," Short paper, EVARILOS Open Challenge - Track 3, May 2014.