



Server-Based Indoor Location Detection System

Osman Kerem Perente^(✉) and Tacha Serif

Yeditepe University, Atasehir, 34755 Istanbul, Turkey
{kperente, tserif}@cse.yeditepe.edu.tr

Abstract. With the advancement of technology and telecommunication services, data consumption rates are increasing ever since. People for long have started using applications with the help of contextual information to improve their user experience. Thus, providing a cross-platform location service to further enrich such applications has become a necessity. For this purpose, numerous client-based indoor location systems on mobile devices are developed to perform this task. Nevertheless, most of the time these systems suffer from elimination of features from operating systems for security purposes. Indeed, with the current security trends, to ensure the privacy of mobile users, mobile operating system designers are progressively eliminating certain low-level features such as reading RSSI and introducing randomized MAC addresses. Thus, in this study, the authors propose, design and implement a server-based indoor positioning system to eliminate platform dependency and to provide the location detection in wide range of devices. The designed server-based system is scalable and platform independent; hence can run on virtually any family of smart device. Furthermore, the evaluation findings indicate that the proposed system performs in acceptable accuracy to client-based systems compared to more complex and costly implementations.

Keywords: Server-Based · Location detection · RSSI · Indoor positioning

1 Introduction

Mobile devices are used in every field in our lives. With the advancing technologies and telecommunication services, data consumption rates have been increasing with an exponential pace. People use applications with the help of contextual information to improve the user experience. Thus, providing an accurate location service to further enrich such applications has become a must.

Although Global Positioning System (GPS) is widely adopted as the standard positioning system for outdoor environments, it lacks the required accuracy in indoor environments where a clear view of sky is not available. Therefore, it is necessary to use different techniques for indoor location detection, such as Wi-Fi IEEE 802.11, Bluetooth 4.0 radio signals. Many of these systems use client-based positioning approach where location estimation is done on the user's device. However, in recent releases of mobile operating systems (OS), many low-level functionalities are gradually being disabled or becoming harder to access due to security and privacy concerns. As an example, such feature restrictions include the elimination of RSSI reading of wireless access points and randomized MAC addresses to disguise the device in iOS 7 and later [1]. Therefore,

the implementation of some low-level functionalities in some OSs can be more cumbersome and harder to achieve compared to other OSs.

In this paper, a server-based location estimation system is proposed and evaluated for accuracy and performance. The proposed system uses sniffer devices to gather signal information from mobile devices. The position calculation is done on the server which enables platform independency. Accordingly, this paper is structured as follows; Sect. 2 reviews and details similar location-aware systems. Section 3 elaborates on the location estimation techniques and methodologies and Sect. 4 details the requirements and the proposed design of the system. Section 5 depicts the prototype implementation and Sect. 6 details the approach used to conduct the evaluations and discusses the findings. Finally, Sect. 7 draws conclusions and highlights the possible future development venues.

2 Background

The lack of consistent and reliable location detection mechanism for indoor environments attracted a lot of interest from both academia and industry. As a result, over the last decades many localization techniques were developed and tested with various properties. Among the proposed techniques, the most popular ones utilized radio signals as their key component.

In 2001, the Bat location sensor system [2] is developed by AT&T Laboratories in Cambridge using ultrasonic personalized badges known as Bats. Trilateration technique works by emitting an ultrasonic signal from the badge and using the receivers on the ceiling to calculate time-of-flight values of the signal. Trilateration technique is then utilized to perform location detection. The test results indicated that the Bat location sensor system estimated the location of badge wearing personnel with 3 cm error rate 95% of the time. Bergamo and Mazzini [3], which used triangulation on Wireless Sensor Networks by the received signal powers from two wireless beacons that were placed to two known locations. Their work focused on improving accuracy by modeling wireless fading when movement is present. Following this trend, Bahl and Padmanabhan [4] designed an RF based localization system based on empirical and signal propagation models in location estimation and compared several methods in each model. Empirical method based on fingerprinting achieved distance error of just less than three meters.

Generally, the positioning systems are designed and tested in relatively small or controlled environments where systems behave optimally. Hence, the test results do not always reflect the real-life performance of these proposed systems. Accordingly, Wirola et al. [5] has comparatively highlighted the requirement and methodology differences between implementing indoor positioning system in a large and small scale indoor environments. Li et al. [6], developed a positioning system at Queen Mary University in London which aims to identify room the user is currently in. The system is deployed on the second and third floors of Electronic Engineering building. In this study, they incorporated Wi-Fi RSSI positioning system with GSM-based approach and their results indicate that the proposed system achieved room level positioning with 72% accuracy. Similarly, Mathisen et al. [7] evaluated several positioning techniques utilizing the

existing Wi-Fi infrastructure in a university hospital floor of 160,000 m². Their proposed system, for various algorithms, achieved a mean error-rates from 8 to 15 m relative to ground positions.

3 Methodology

Accordingly, this section details the methodologies and algorithms that are used to implement location detection and estimation systems.

3.1 Distance Determination

Determining the distance between two points is the first step of location detection. One of these points has a fixed location with known coordinates and, generally the other one is a mobile device whose location will be determined. There are several methods that can be used to determine the distance. Such as time of arrival (ToA), time difference of arrival (TDoA), angle of arrival (AoA) and received signal strength indicator (RSSI). These methods utilize radio signals and their propagation characteristics in free space.

3.2 Received Signal Strength Indicator

Most of the techniques discussed above require line-of-sight (LOS) operation. However, indoor environments seldom have direct LOS, which degrades the value of the time and the angle information extracted from the received signal due to multipath effect [8]. On the other hand, RSSI method performs well both in LOS and non-LOS environments, changing with respect to the distance from the transmitter. RSSI distance determination mainly uses IEEE 802.11 wireless local networking standard, with wide availability in many consumer handsets and extensive deployment in many urban areas.

RSSI is a good option for determining the distance between devices because of the attenuation property of radio signals when the distance between the receiver and the transmitter increase. Also, it is an easier method to implement and it does not have the complexities of the other methods. Using the RSSI method, the distances between two points of measurements in terms of signal propagation can be calculated using the Euclidean distance theorem (Eq. 1).

$$d = \sqrt{\sum_{i=0}^n (q_i - p_i)^2} \quad (1)$$

3.3 Fingerprinting

Fingerprinting is a technique that is evolved to increase the positioning accuracy in non-LOS indoor environments by constructing a radio map at pre-determined target locations. It is performed by sampling more than one RSS measurement over time in each

target reference point to better map the environment. Once the signal map of the environment is constructed, the location detection can be achieved by comparing the real-time signal information with the signal map and calculating the position using a subset of signal information contained in the map.

To utilize this technique, location detection procedure is broken down into two distinct phases – offline and online phase.

Offline Phase. This phase can also be called learning phase, implying that the necessary signal characteristics of the environment is constructed for the real localization.

Firstly, the indoor environment is evaluated and the locations for signal sampling are determined. These locations are called reference points (RP), with each point having a distinct x and y coordinate and an array of average signal values. As the RSSI value collection and signal map generation can be a time-consuming process, an offline phase tool is needed to help the collection of acquirable signal strengths of access points (AP) at RPs.

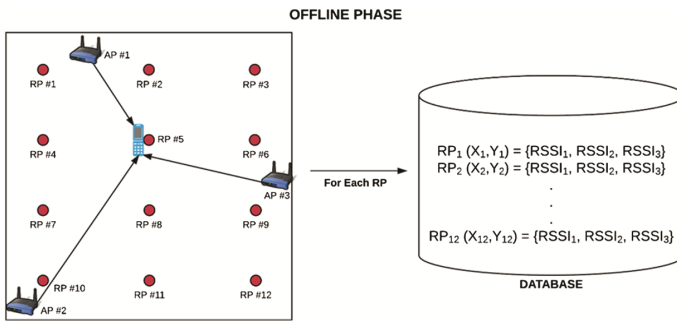


Fig. 1. Offline phase of fingerprinting technique

As depicted in Fig. 1, the idea is to move the mobile device through the RPs one by one, collecting RSSI samples at each RP. For a single RP, the mobile device collects a pre-determined number of RSSI values for each AP received from each accessible AP and stores the average of these AP-specific RSSI values in signal map database. The average of RSSI samples measured at every RP is inserted into the database, as RP Data (RPD) vector in the form $\{ID, S_1, S_2, S_3\}$ where ID identifies the reference point, whereas S_i denotes the signal strength received from the AP_i at a specific RP.

Online Phase. In this phase, the real-time localization operation is performed. Accordingly, the mobile device initiates the collection of RSSI values from each available AP at the point where the location is to be determined. Euclidean distances are calculated by using the collected RSSI of the available APs and signal map retrieved from the database. Since the signal map contains signal measurements done at each RP in the environment, the Euclidean distance calculation is performed for each RP present in the signal map. Thus, for every RP in the signal map, the distances between MU and each RP is obtained (Fig. 2). After this step, K-Nearest Neighbor (KNN) algorithm is used

to calculate location of the MU by selecting the closest RPs. The steps of this operation are given in detail below.

- For a given test point (TP), make at most 10 observations per each available AP using mobile device and store them in the observation set (OS). Here OS_i denotes observation stack of i^{th} AP. $OS_i = \{O_1, O_2, \dots, O_{10}\}$
- For each AP, calculate the average of all observations in OS, and generate the average signal strength (SS) vector of MU. $SS = \{s_1, s_2, \dots, s_n\}$ (Here n represents the number of available APs at the location of mobile device)
- For each RP calculate the Euclidean distance between the RP and TP using the RPD and SS vector.
- Perform KNN algorithm for the closest k neighbors (RPs). The weighted average of the x and y coordinates of the candidate RPs are the estimated MU location.

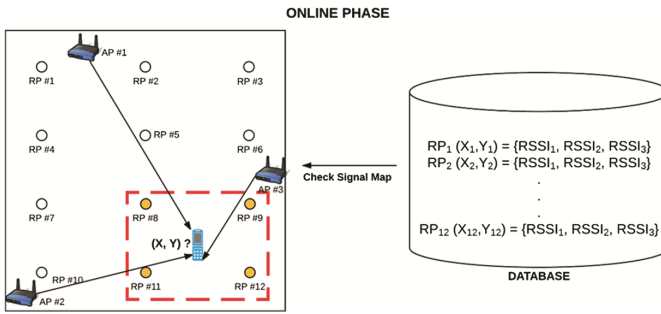


Fig. 2. Online phase of fingerprinting technique

3.4 K-Nearest Neighbor (KNN) Algorithm

K-nearest neighbor algorithm is a very simple classification and regression machine learning algorithm. It involves k-nearest training data to classify or determine value of an object by measuring its distance to that of the training data. K is a pre-determined number, which can be selected to optimally represent the class or value of an object.

To determine the closest training data, distance must be measured. One way to measure the distance between the object and training data is to use Euclidean distances. This distance value can also be used to improve regression based KNN, in which case the algorithm evolves in to weighted k-nearest neighbor (WKNN) algorithm. The weight can be taken as the inverse of the distance $1/d$, to improve the weight of the closer training data during calculation. The localization accuracy can be further improved by using WKNN, in such case that the weights are taken using the complementary neighbors.

$$x = \frac{\sum_{i=1}^4 (x_i * d_{4-i})}{\sum_{i=1}^4 d_i} \quad y = \frac{\sum_{i=1}^4 (y_i * d_{4-i})}{\sum_{i=1}^4 d_i} \quad (2)$$

In Eq. 2, x_i denotes the x -coordinate of RP_i , y_i denotes the y -coordinate of RP_i and d_i denotes the measured distance between the user position and RP_i . By multiplying x_i and y_i with d_{4-i} , the weight can be incorporated in to the algorithm (Fig. 3).

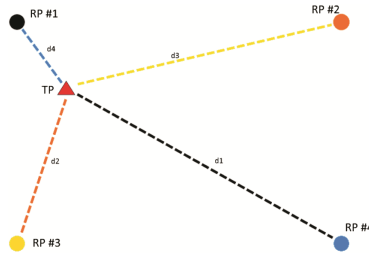


Fig. 3. WKNN in fingerprinting

4 Analysis and Design

The system architecture of the proposed system can be seen in Fig. 4. This location detection solution is composed of a server and at least three wireless sniffer devices, where each device is connected to the server via Ethernet connection. The server hosts database where the signal map of the environment and sniffer device information is stored. By making use of this architecture adding additional sniffer devices for increased coverage becomes very easy and affordable.

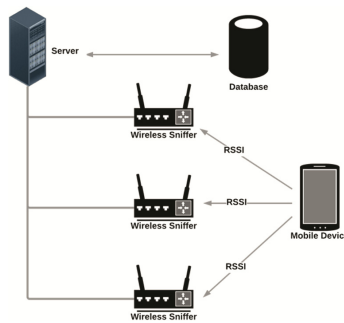


Fig. 4. System architecture

In this scenario, sniffer devices are required to capture 802.11 frames over the air and filter them according to the MAC addresses of the mobile device that needs its position estimated. For this scenario to work, the mobile device is required to be connected to a wireless infrastructure to initiate server registration for its MAC address, and also to generate wireless traffic for the sniffers to capture. This wireless traffic between the MU and the wireless network is essential because the system exploits the packets captured from this traffic to determine the position of the user. Therefore, to

make this system work, an additional access point is needed for the mobile device to connect and communicate.

Accordingly, the infrastructure requirements of the proposed system is as follows; the server platform should be capable of performing the necessary calculations in a short amount of time and handle large number of consecutive requests of position estimations. Also, since the mobile devices runs on battery and have limited resources, the mobile application must be relatively simple and straightforward to minimize battery consumption. Therefore, the system should follow the thin client and thick server model. Considering, the size of the location detection area cloud systems can be used. However, in our proof of concept, the server should be able to handle requests sent from a single device within medium-sized room.

As depicted in Fig. 5, the sequence diagram of the system, once the client is connected to the AP serving internet access, the client can send the system startup command to start the positioning process. Then the server contacts to wireless sniffers to start Airodump processes for capturing 802.11 frames. Once the general system startup is accomplished, the client can send positioning request periodically, the server requests new station data from the sniffers and performs positioning algorithm to calculate user position. The server then sends the calculated user position back to the client.

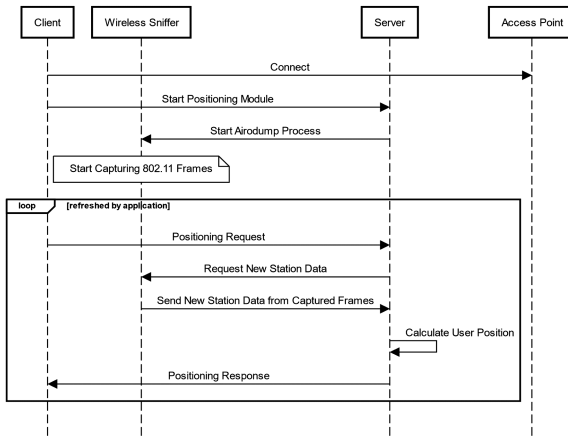


Fig. 5. Sequence diagram of the system

5 Implementation

5.1 Wireless Sniffer Modules

As wireless sniffer devices, three Linux PCs with Ubuntu operating system are employed. These PCs are regular desktop stations with Mid ATX chassis, but in a real-life scenario these PCs can be substituted with mini-PCs.

The sniffing process is performed using a special wireless application suite called Aircrack-ng [9]. From the suit member application, our study makes use of Airodump-ng application, which enables the wireless network interface adapter to sniff wireless packets in the environment. The sniffer function requires special wireless adapters which can be put into the monitoring mode. Therefore, TP-Link 802.11n wireless adapter with model number WL722 N is selected. This device utilizes an Atheros chipset, which can easily put into monitoring mode and coupled with Airodump-ng. After the monitoring interface is created, Airodump-ng process is executed. Then, Airodump-ng simply uses this newly created monitoring interface and listens to all available wireless packets across all Wi-Fi channels. Collected wireless data is then used to create device lists on the terminal. The data then can easily be dumped to a CSV that contains the following information: MAC address of the station, first time seen, last time seen, signal strength of the station, and information whether a station is connected to an AP, and if connected, the MAC address of this AP.

Bearing in mind the MAC address of the device that requested position estimation, the data from CSV file can be parsed to get its signal strength. Consequently, its MAC address along with its signal strength values are send to the server every time a sniffer captures 802.11 frames containing this MAC address.

5.2 Server-Side Implementation

Server-side application is implemented in Java programming language along with a JDBC database. The server has separate components for configuring and running the positioning system. These components (Fig. 6) are; offline phase calibration, online phase localization, sniffer registration, and database management. Sniffer registration module accepts the addition of new sniffer modules for later expandability, in our case three sniffer modules are generated. Offline phase calibration is responsible for creating the signal map by the help of sniffer modules. Online phase module is responsible for real-time localization and mobile device connection management. Database module is responsible for database connection and management.

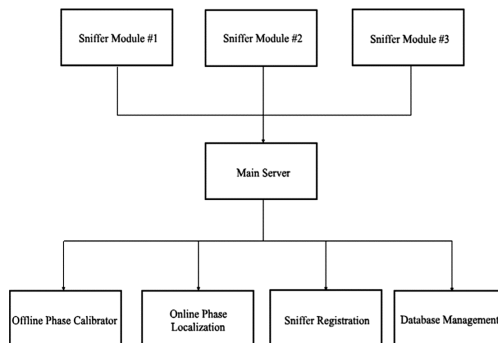


Fig. 6. Server modules

In the case of a positioning request from the mobile device, the server contacts the sniffer module and requests the most recent signal data for the device in question. Since sniffing and the signal strength collection is a continuous process, the online phase module extracts the required data from the database via the help of the database management module and performs necessary positioning calculation. The result of this calculation is written to the database continuously and updated in every few seconds when necessary data is gathered for another location update. For the online phase localization WKNN algorithm is chosen and deployed in the server. The k value in the WKNN algorithm is chosen to be four, which gives the best results in the test environment the system is implemented.

6 Tests and Results

6.1 Test Environment

The proposed system is implemented and evaluated in the Pervasive System Laboratory in the Engineering Faculty. The test environment was the whole of the laboratory, which is approximately 47 m². In order to cover the entire test environment, the sniffer PCs are sporadically spread around the laboratory, so they were as far away from each other as possible. On the other hand, 15 reference points (RP) are selected and each one of them is distributed with 1.4-m intervals. Furthermore, 10 test points (TP) are selected at random locations to estimate the location and calculate the error-rate. The coordinates of each sniffer PC, RP and TP is calculated in reference to the test bed and implemented on the database to be used in location determination.

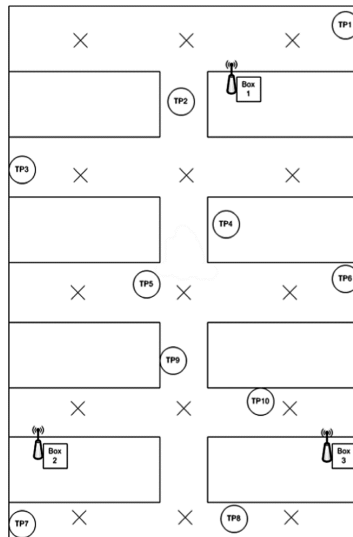


Fig. 7. Test environment

A detailed depiction of the test environment is provided in Fig. 7, where the RPs are marked with “X”s, sniffer PCs with an antenna, and the TPs with their TP numbers. In order to measure the error-rate of the proposed solution, a number of measurements are conducted on 10 test points by utilizing the fingerprinting and finding how they compared with the actual coordinates within the test environment.

6.2 Evaluation Results

In order to evaluate the proposed system for its performance, 10 pre-determined test points were set. At each of these test locations, the system tried to capture 20 airborne packets that were transmitted from the mobile device and by making use of the collected packet information to perform the location estimation. According to the CDF for location estimation error-rate in Fig. 8, the systems performs within 3 m positioning error-rate 70% of the time.

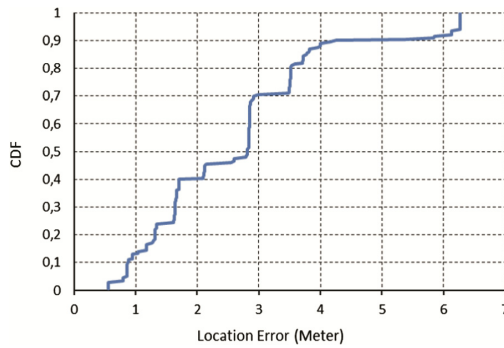


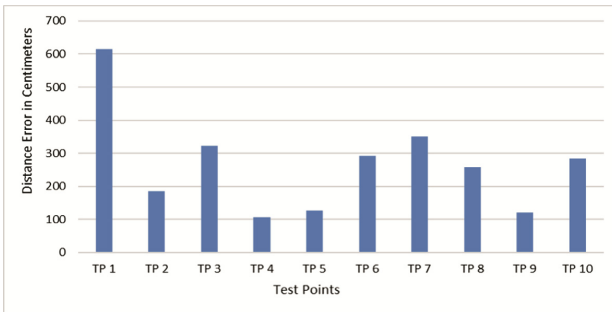
Fig. 8. CDF location estimation

As it can be seen from Table 1, where the results are given in centimeters, the mean system location estimation error-rate ranged between 105.85 cm and 615.15 cm. Overall mean of location estimation error-rate of this solution was 266.15 cm and standard deviation was 38.77 cm. To elaborate more on the single test results, it can be seen that 5th test point has the lowest error with 55.44 cm. Whereas 1st test point has the greatest error with 626.99 cm.

It can be noted that test points (2, 4, 5, 9, 10) that are covered within reference points has lower location estimation error, whereas test points (1, 3, 6, 7, 8) which are not covered within the reference points and reside on the outer radius and extremities of the room has greater error-rate (Fig. 9). This phenomenon can be the result of the WKNN algorithm itself, where the candidate location estimate is calculated using the nearest four RPs. In the case of test points that reside in the locations that are not directly covered within RPs, the WKNN algorithm cannot perform optimally, thus it increases the error-rate.

Table 1. Detailed test results

TP	N	Mean	Std. deviation	Std. error	Minimum	Maximum
1	20	615.51	22.47	5.03	538.52	626.99
2	20	184.66	67.57	15.11	117.00	372.65
3	20	321.96	57.61	12.88	255.16	425.03
4	20	105.85	21.94	4.91	85.56	131.31
5	20	126.97	51.38	11.49	55.44	170.66
6	20	293.09	23.55	5.27	285.35	371.32
7	20	350.99	0.99	0.22	348.62	353.62
8	20	257.62	102.94	23.02	133.27	399.53
9	20	121.46	36.75	8.22	79.88	167.17
10	20	283.41	2.47	0.55	278.98	291.27
Total	200	266.15	38.77	8.67	55.44	626.99

**Fig. 9.** Test results per TP

7 Conclusion

In this paper, a study is undertaken to evaluate a server-based indoor positioning system that is platform independent. The server-based system proposed to be scalable mainly for large area deployment and mobile OS independent. Therefore, it does not suffer from client-based systems where they need OS API compatibility. The server-based positioning system aimed to perform in similar performance accuracy to client-based systems albeit a more complex design process. The findings show that it performs within these expectations. The average positioning error for server-based system is 266 cm which is comparable to the tested systems when they are evaluated.

For the future work, the server-based system can be improved by adding hybrid functionality with sensor-based methods. Such system with iBeacon and PDR support, can choose the most accurate method on-the-go and cut down the positioning error even in large real-world environments. Furthermore, the algorithm can be improved such as adding probabilistic approaches to plain Wi-Fi and Bluetooth techniques. Consequently, the accuracy and performance of the server-based system can be tested in a large indoor environment such as the supermarket branch used in Wi-Fi positioning system.

References

1. iOS 7 Eliminates MAC Address as Tracking Option, Signaling Final Push Towards Apple's Own Ad Identifier Technology. <https://techcrunch.com/2013/06/14/ios-7-eliminates-mac-address-as-tracking-option-signaling-final-push-towards-apples-own-ad-identifier-technology/>. Accessed 1 Apr 2018
2. Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A., et al.: Implementing a sentient computing system. *IEEE Comput.* **34**(8), 50–56 (2001)
3. Bergamo, P., Mazzini, G.: Localization in sensor networks with fading and mobility. In: 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (2002)
4. Bahl, P.N., Padmanabhan, V.: RADAR: An in-building RF-based user location and tracking system. In: 19th Annual Joint Conference of the IEEE Computer and Communications Societies (2000)
5. Wirola, L., Laine, T.A., Syrjarinne, J.I.: Mass-market requirements for indoor positioning and indoor navigation. In: International Conference on Indoor Positioning and Indoor Navigation (2010)
6. Li, K.J., Bigham, J., Bodanese, E.L., Tokarchuk, L.: Location estimation in large indoor multi-floor buildings using hybrid networks. In: IEEE Wireless Communications and Networking Conference (2013)
7. Mathisen, A., Sorensen, S.K., Stisen, A., Blunck, H., Gronbaek, K.: A comparative analysis of indoor WiFi positioning at a large building complex. In: 2016 International Conference on Indoor Positioning and Indoor Navigation (2016)
8. Hightower, J., Borriello, G.: Location Sensing Techniques, University of Washington, Paper# UW-CSE-01-07-0 (2001). <ftp://ftp.cs.washington.edu/tr/2001/07/UW-CSE-01-07-01.pdf>. Accessed 2 Apr 2018
9. Aircrack-ng. <https://www.aircrack-ng.org>. Accessed 2 Apr 2018