Determining Physical Location of Wireless Access Point using Smart Devices

Vasin Suttichaya[1]
Department of Computer Engineering
Faculty of Engineering, Mahidol University
25/25 Salaya, Phuttamonthon,
Nakhon Pathom, 73170 Thailand
vasin.sut@mahidol.ac.th[1]

Ting-Wei Liang [2], Nannapat Meemongkolkiat [3]
Department of Computer Engineering
Mahidol University International College
999 Salaya, Phuttamonthon,
Nakhon Pathom, 73170 Thailand
talachanun.lia@student.mahidol.edu [2], nannapat.mee@student.mahidol.edu[3]

***Abstract*—Indoor Position System (IPS) is a method for locating objects in an enclosed space. This paper applies the concept of IPS to locate the physical location of wireless access point. The proposed system can be used for locating the suspected access point in the building. The system uses at least three smartphones to form the points in two-dimensional geometry. These smartphones will gather Wi-Fi's information and send them to the server. Then, trilateration is applied to calculate the distance and the location of the targeted access point. The physical location will then send back to each smartphone. The smartphone can use this position to navigate the user to the targeted access point. The result shows that, in the enclosed space, the error of the proposed method is around 3.5 meters.**

***Index Terms*—Indoor Position System, Received Signal Strength Indication, Trilateration**

I. INTRODUCTION

Presently, the Internet is one of the most important source of information. Information that are searched using the Internet can be used in many ways, such as education, investigation, and entertainment. We also use the Internet to communicate as it is faster and more efficient medium comparing to other ways of communications. As a result the Internet became one of the most popular way to exchange information with people all around the world. One of the efficient methods that people use to connect to the Internet is through wireless local area networking, also known as Wi-Fi.

Wi-Fi has two advantages, which are simplicity and mobility. An example for its simplicity could be seen in this modern world, all smart devices such as mobile phones and laptops can connect to a nearby access point with sometimes just one simple tap. Furthermore, its mobility can be shown that a device can be anywhere near the access point area without needing further cabling like Local Area Network (LAN).

On the other hand, advantages come with disadvantages. The efficiency of the Internet leads to a new kind of threat, often called cybercrime. They were able to gain access to other devices easily using network connection. Using network for this kind of practice will result in private information being at risk.

To prevent the cybercrimes, developing a device that can locate suspicious access points could be highly beneficial in preventing information from being taken illegally. This paper proposes the method for determining the physical location of the targeted Wi-Fi access point. The proposed method focuses on the design of the prototype system and the experimental results obtained from our prototype. The design of the prototype considers 3 properties as follows:

1) The system must be easy to implement. It also requires less additional hardware in order to calculate the distance.
2) The system can be used on regular Wi-Fi infrastructure without additional firmware. Furthermore, it can process with wide range of information that available in general market Wi-Fi access points.
3) The localization system can calculate the distance accurately.

This paper is organized as follows. Section II presents the related works. The proposed method and its implementation issues are presents in Section III. We presents the experimental in Section IV and discussion in Section V. Section VI presents the conclusion.

II. RELATED WORKS

Some related works are presented in this section.

A. Indoor Localization System using Received Signal Strength Indicator

The received signal strength indicator (RSSI) is a measurement for indicating the power of radio signal that received by the acceptor. RSSI can be used to approximate the distance between the access point and the mobile device. There are many papers proposed the indoor localization system using RSSI [1]–[6]. The distance can be estimated using Equation (1)

$$d = 10^{\frac{A-RSSI}{10n}}, \quad (1)$$

where d is the distance, A is RSSI at 1 meter, $RSSI$ is the received signal strangth indicator, and n is the path loss exponent.

B. Rogue Access Point Detection

Many researchers attempt to identify the rogue access point by analyzing the flow of network. However, it is difficult to locate the physical location of the rogue access point.

Watkins et al. [7] proposed the method for detecting rogue access point. This method uses the magnitude of round trip time (RTT) network traffic to as an anomaly indicator. The RTT are used to determine connection type whether is wired or wireless, where the RTT for wired are lesser in duration than the wireless. Probability of error on differentiation can be reduced by using the standard deviation of RTT. After determining the connection type, the wireless access point are then compared to lists of authorized access point to determine the rogue access point.

K.Tejaswi et al. [8] analyze the connection type between normal access point and rogue access point using TCP ACK-pairs. The differentiating can be calculate by using algorithm with sequential hypothesis test and log-likelihood function, with or without training data. Higher accuracy results are obtained with training data. The online detection system, detecting on IP address basis will collects packets for identifying ACK-pairs and uses the algorithm as described above to determine connection type. To detect rogue access point, the information of wireless access point is compared to authorized access point lists. The results obtained can also use to locate the access router based in the IP address and subnet addressing structure from the ARP table, which stored mapping between IP address and the corresponding MAC address. The method is strong against MAC address spoofing, but if the rogue access point changes, the inter-ACK time the algorithm will then not be able to detect it.

C. Physically Locating Wi-Fi Intruders

Adelstein et al. [9] proposes the method for combining Wireless Intrusion Detection Systems (WIDS) and directional antennas to locate the location of the intruders. They uses 2 type of Access Point (AP) which are omni-directional AP and WIDS AP. The WIDS AP is located on the perimeter of the area to show the remote station of the intruders. The intruder will connect to the WIDS before coming inside, the detection will use the attack signature and behavior base to show its external location. This can be used to trigger the alarm. The directional antennas is used to show the bearing of the intruders by using the antenna's signature.

III. Proposed Architecture

The proposed system is presented in this section. The mobile client will gather RSSIs and the location of the mobile client in real-time. These information is sent to the server. The server uses trilateration method for calculating the distance and the direction to the targeted access point. After obtaining the physical location, the server sends the location of the targeted access point to all mobile devices. The detail of client and server will be elaborated in the following subsections.

A. Mobile Client Procedure

The process of the mobile client is describe in Algorithm 1. The algorithm aims to provide a brief explanation of how the mobile application works during the process. The application takes address and port of server as inputs. Firstly, the client attempts to connect to the server. The client will wait until the server accept the client. After the connection to server is established, the client will obtain its identification number from the server, as shown in line 6. In line 7, the application will allow users to send their location, which can be edited to improve accuracy of the current location. At the Wi-Fi selection page, the application will displays a list of available Wi-Fi as in line 9. After the targeted Wi-Fi is selected, the application will collect the Wi-Fi's SSID and BSSID and send to the server, as shown in line 10 to 11. Then, the application will continue to collect RSSI from the targeted Wi-Fi, as shown in line 13. Then, RSSI value is send to the server. The Wi-Fi's physical location, in form of Latitude and Latitude, is calculated at the server side. In line 15, if the destination Latitude and Latitude is not NULL, then the bearing will be calculated and kept in the variable $azimuth$. This value is used to adjust the arrow to display the direction for the user.

Algorithm 1 Mobile Client Procedure

```
1:  procedure ClientSide(svport, svaddress)
2:      socket ←CreateSocket()
3:      repeat
4:          socket.Connect(svport, svaddress)
5:      until Server accepts client
6:      id ← socket.Receive()
7:      (lat, lon) ←GetCurrentLocation()
8:      wifiList ← ScanWiFi()
9:      targetedWifi ← UserSelect(WiFiList)
10:     (ssid, bssid) ← targetedWiFi.InFo()
11:     socket.send(ssid, bssid, lat, lon, id)
12:     while True do
13:         rssi ← GetRSSI(targetedWiFi)
14:         socket.send(rssi)
15:         APlat, APlon ← socket.Receive()
16:         if APlat ≠ NULL and APlon ≠ NULL then
17:             break
18:     repeat
19:         (lat, lon) ←GetCurrentLocation()
20:         azimuth ←Bearing(lat, lon, APlat, APlon)
21:         AdjustArrow(azimuth)
22:     until Distance(lat, long, APlat, APlon) > 0
```

B. Server Procedure

The process of the server is described in Algorithm 2, 3, 4, and 5. Algorithm 2 provides the overview of how the server works. Algorithm 3 shows the process for handling the connection of each mobile client. Algorithm 4 and 5 elaborates the process of calculating Latitude and Latitude of the targeted access point.

The server process starts in Algorithm 2. The global variable num_client records the number of clients. There are two global dictionaries, $clientInfo$ and $rssiDict$. The dictionary $clientInfo$ uses the identification number of the client as a key for accessing the data. This dictionary records the information of mobile client, which consists of the current location of the mobile device and the information of the targeted access point. Similarly, the dictionary $rssiDict$ also uses the identification number of the mobile client as a key. The $rssiDict$ keeps the list of RSSI values that are gathered from each client. After the connection has been established, the server will send the identification number to the client, as shown in line 10. Then, the server creates a new thread for handling the connection for each client.

Algorithm 2 Server Procedure

1: **procedure** SERVERSIDE($port, address$)
2:     $socket \leftarrow$CREATESOCKET()
3:     $socket$.BIND($port, address$)
4:     $num\_client \leftarrow 0$ ▷ Global variable
5:     $clientInfo \leftarrow \{\}$ ▷ Global dictionary
6:     $rssiDict \leftarrow \{\}$ ▷ Global dictionary
7:     **while** True **do**
8:         $client, address \leftarrow socket$.ACCEPT()
9:         $num\_client \leftarrow num\_client + 1$
10:         $client$.SEND(num_client)
11:         STARTTHREAD(HANDLECONNECTION, $client$)

The connection process of each client is elaborated in Algorithm 3. The server receives the client information and records them in $clientInfo$, as shown in line 2 to 3. Then, the server waits until it receives the RSSI value from the client. The RSSI value is recorded in $rssiDict$. If the number of connected clients is more than three, then the process for calculating the position of the targeted access point is called, as shown in line 7 to 8. Finally, after obtaining the location, in the form of Latitude and Latitude, the server sends the Wi-FI access point location to the mobile device.

Algorithm 3 Handle Connection

1: **procedure** HANDLECONNECTION($client$)
2:     $(ssid, bssid, lat, lon, id) \leftarrow client$.RECEIVE()
3:     $clientInfo\,[id] \leftarrow (ssid, bssid, lat, lon)$
4:     **while** $True$ **do**
5:         $rssi \leftarrow client$.RECEIVE()
6:         $rssiDict\,[id]$ .APPEND($rssi$)
7:         **if** $num\_client \geq 3$ **then**
8:             $(APlat, APlon) \leftarrow$GETLOCATION()
9:             $client.send\,(APlat, APlon)$

The process for calculating the location of the targeted access point is described in Algorithm 4. The algorithm consists of two procedures, GETLOCATION and GENLOCATION. The procedure GENLOCATION is recursively called for enumerating all possible combinations of clients to form three coordinate points. The identification number of selected 3 clients is records in the list $mobiles$. Then, CALLOCATION is used to calculate the distance and the destination Latitude and Latitude of the access point. All possible locations is recorded in the list $locations$. Finally, FINDCENTER is used for calculating the center of geolocation from all possible locations.

Algorithm 4 Get Location

1: **procedure** GETLOCATION
2:     $mobiles \leftarrow [0, 0, 0]$
3:     $locations \leftarrow []$
4:     GENLOCATIONS($locations, mobiles, 0, 0$)
5:     $(APlat, APlon) \leftarrow$FINDCENTER($locations$)
6:     **return** $(APlat, APlon)$
7: **procedure** GENLOCATIONS($locations, mobiles, index, i$)
8:     **if** $index = 3$ **then**
9:         $(APlat, APlon) \leftarrow$CALLOCATION($mobiles$)
10:         $locations$.APPEND($(APlat, APlon)$)
11:         **return**
12:     **if** $num\_client \leq i$ **then**
13:         **return**
14:     $mobiles\,[index] = i + 1$
15:     GENLOCATIONS($locations, mobiles, index + 1, i + 1$)
16:     GENLOCATIONS($locations, mobiles, index, i + 1$)

The process for calculating Latitude and Longitude is elaborated in Algorithm 5. This algorithm takes the list of three identification numbers. The five most frequent RSSI values are selected, as shown in line 5. These RSSI values will be converted to distance of the targeted access point, as shown in line 8. Then, the algorithm calculates all possible locations and finds the geolocation center, as shown in line 12 to 16.

Algorithm 5 Calculate Location

1: **procedure** CALLOCATION($mobiles$)
2:     $dist \leftarrow \{\}$
3:     **for** id in $mobiles$ **do**
4:         $rssi \leftarrow []$
5:         $rssi$.APPEND(SELECTTOPFIVE $(rssiDict\,[id])$)
6:         $dist\,[id] \leftarrow []$
7:         **for** $value$ in $rssi$ **do**
8:             $distance \leftarrow 10^{(rssi_1 - value)/(10n)}$
9:             $dist\,[id]$ .APPEND($distance$)
10:     $(id_1, id_2, id_3) \leftarrow mobiles$
11:     $Locs \leftarrow []$
12:     **for** d_1 in $dist\,[id_1]$ **do**
13:         **for** d_2 in $dist\,[id_2]$ **do**
14:             **for** d_3 in $dist\,[id_3]$ **do**
15:                 $Locs$.APPEND(TRILAT $(mobiles, d_1, d_2, d_3)$)
16:     **return** FINDCENTER($Locs$)

C. Implementation Issues

There are some issues that the developer should consider during the implementation.

1) There are three main global variables, $num\_client, clientInfo$, and $rssiDict$, that are separately accessed by each thread. Mutual exclusion technique must be implemented to avoid race condition.
2) There is a possibility that Latitude and Longitude, calculated by GETLOCATION and FINDCENTER in Al-

gorithm 4, is $NULL$. This implies that the current information is inaccurate.

3) BSSID and SSID are used for verifying that all clients detect the same Wi-Fi access point.
4) The RSSI value at one meter, $rssi_1$, and the path loss exponent, n, need to calibrate before using. In the real implementation, mobile devices should allow the user to select the value of path loss exponent since this value depends on the environment.
5) The server is implemented on PC with Intel Core i5 2.3 GHz with 8 GB RAM. It is impossible to use one of the mobile device in the system as a server since Algorithm 4 requires a lot of computational power.

IV. Experimental

A. Environment Setup

For testing the proposed system, we set up 3 different testing locations. The floor plans of all testing locations are illustrated in Figure 1 to 3.

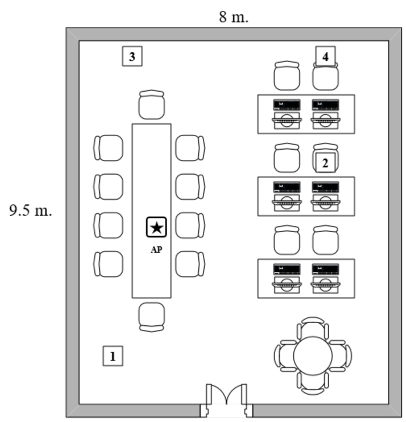

Fig. 1. Enclosed Space.

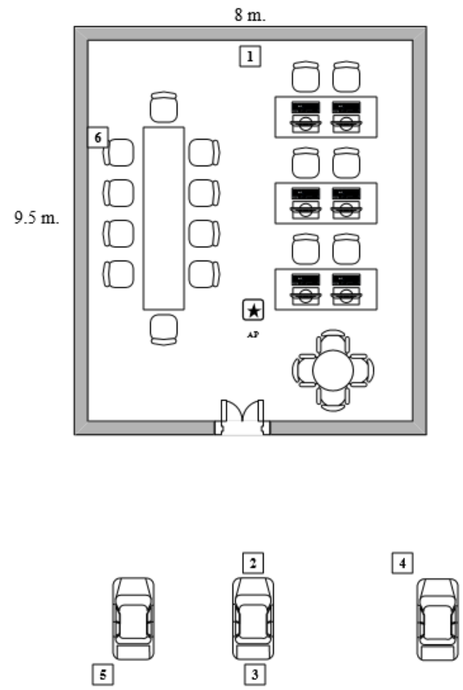

Fig. 2. Semi-Open Space.

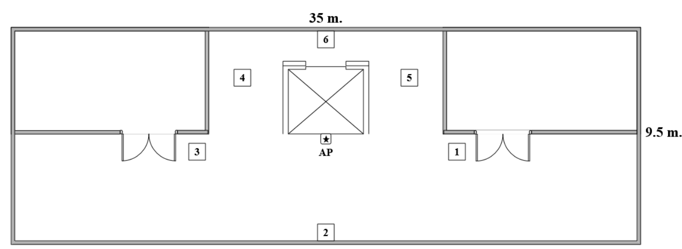

Fig. 3. Open Space.

B. Testing Procedure

- **Equipment:** The equipment used for the experiment are Android mobile devices with operating system higher than Nougat, one laptop as the server, and one access point. The mobile devices need to be check for magnetic field and accelerometer in order for the compass function to be available.
- **Position Setting:** We use laser distance measurer to measure the distance. The position of the access point and mobile devices are shown in Figure 1 to 3. The number in the pictures represents the position of each mobile device.
- **Calibrating:** Both RSSI value at one meter and path loss exponent value need to calibrate before starting the experiment. The path loss exponent value mostly depends on the environment. Thus, each mobile device user needs to select the suitable path loss exponent value before calculating the distance. The location of each mobile device is also needed to manually calibrate since GPS is not possible to determine the exact location of the mobile device.
- **Starting the Experiment:** In this experiment, mobile devices connect to the server via 4G LTE network. All mobile devices starts collecting the RSSI values from the targeted access point. Then, these RSSI values are sent to the server to calculate Latitude and Longitude of the access point. If the calculation succeed, then Latitude and Longitude will be sent back to each mobile device.
- **Verifying:** We verify the accuracy of the proposed method by computing the distance between the computed location and the real location of the access point.

C. Experimental Results

1) Enclosed Space: The purpose of this scenario is to verify the concept of the proposed algorithm. The floor plan and the positions of all devices is illustrated in Figure 1. Instead of using Latitude and Longitude, we use the Cartesian coordinates system to represent the position in this scenario since the room is too small. The coordinate $(0, 0)$ is set at the lower left of the room. The position of all mobile devices, the most frequent RSSI values, and the calculated distances are shown in Table I.

TABLE I
EXPERIMENTAL RESULTS FOR ENCLOSED SPACE

Device 1: (2.049, 1.275) Actual Distance: 2.00		Device 2: (6.480, 5.530) Actual Distance: 4.00	
RSSI Values	**Distances**	**RSSI Values**	**Distances**
-54	1.8200	-54	3.6236
-55	1.9100	-55	4.0000
		-57	4.8772
Device 3: (2.528, 9.500) Actual Distance: 4.29		**Device 4: (7.001, 9.500) Actual Distance: 5.20**	
RSSI Values	**Distances**	**RSSI Values**	**Distances**
-64	6.1741	-64	5.6468
-61	4.6988	-63	5.2000
-60	4.2900	-65	6.1320

2) Semi-Open Space: The purpose of the test is to verify the accuracy of the proposed system when there are a few interference. Geographic coordinate system is used to represent the position of the access point and all mobile devices. This scenario uses six mobile devices to measure the RSSI values. The floor plan and the positions of all devices is illustrated in Figure 2. The access point is located in the room with two mobile devices. Other mobile devices are located outside the room. The environment outside the room is an open space outside the building. There are some objects that can interfere RSSI such as cars, walls, and trees. The position of all mobile devices, the most frequent RSSI values, and the calculated distances are shown in Table II.

TABLE II
EXPERIMENTAL RESULTS FOR SEMI-OPEN SPACE

Device 1: (13.79595, 100.32608) Actual Distance: 8.00		Device 2: (13.79596, 100.32617) Actual Distance: 7.00	
RSSI Values	**Distances**	**RSSI Values**	**Distances**
-58	7.1699	-61	6.4220
-39	0.8963	-62	7.0000
-40	1.0000	-64	8.3700
-57	6.4267		
-56	5.7600		
Device 3: (13.79590, 100.32631) Actual Distance: 12.00		**Device 4: (13.79602, 100.32617) Actual Distance: 8.60**	
RSSI Values	**Distances**	**RSSI Values**	**Distances**
-77	10.0000	-69	11.0450
-84	15.4600	-72	14.1610
-86	17.5000	-41	1.0860
-85	16.4500		
Device 5: (13.79588, 100.32620) Actual Distance: 19.65		**Device 6: (13.79591, 100.32614) Actual Distance: 5.00**	
RSSI Values	**Distances**	**RSSI Values**	**Distances**
-78	18.4000	-69	5.6000
-80	21.5440	-66	4.7000
-77	17.1130	-70	5.9000
-76	15.8940	-67	5.0000
		-63	3.9000

3) Open Space: The purpose of the test is to verify the accuracy of the proposed system in the real environment. Geographic coordinate system is used to represent the position of the access point and all mobile devices. This scenario uses six mobile devices to measure the RSSI values. The floor plan and the positions of all devices is illustrated in Figure 3. The access point is located in front of the operated elevator. The position of all mobile devices, the most frequent RSSI values, and the calculated distances are shown in Table III.

TABLE III
EXPERIMENTAL RESULTS FOR OPEN SPACE

Device 1: (13.79585, 100.32612) Actual Distance: 10.00		Device 2: (13.79576, 100.32621) Actual Distance: 10.00	
RSSI Values	**Distances**	**RSSI Values**	**Distances**
-68	9.1200	-72	13.1826
-67	8.3176	-70	10.9648
-69	10.0000	-77	20.8930
-70	10.9648		
Device 3: (13.79567, 100.32611) Actual Distance: 10.00		**Device 4: (13.79574, 100.32614) Actual Distance: 10.60**	
RSSI Values	**Distances**	**RSSI Values**	**Distances**
-71	10.0000	-77	10.0000
-69	8.4310	-75	8.6975
-73	11.8597		
-70	9.1825		
Device 5: (13.79583, 100.32614) Actual Distance: 10.65		**Device 6: (13.79548, 100.32612) Actual Distance: 10.00**	
RSSI Values	**Distances**	**RSSI Values**	**Distances**
-78	8.7992	-67	11.1034
-79	9.3804	-70	15.1991
-80	10.0000	-68	12.3285

The calculated position, along with the real physical location of the access point, of each scenario is shown in Table IV. The error of the proposed is measured by computing the distance between the computed location and the real location of the access point. The error of each scenario is shown in Table V.

TABLE IV
COMPARING THE PHYSICAL LOCATION AND CALCULATED LOCATION

Test Location	Physical Location	Calculated Location
Enclosed Space	(2.603, 4.452)*	(4.799, 7.182)*
Semi Open Space	(13.795875, 100.326100)†	(13.795950, 100.326060)†
Open Space	(13.795772, 100.326129)†	(13.795977, 100.326094)†

∗ Catesian coordinate system
† Geographic coordinate system

TABLE V
ERROR OF EACH SCENARIO

Test Location	**Error** (m.)
Enclosed Space	3.5039
Semi Open Space	9.3923
Open Space	23.1063

V. DISCUSSION

In the enclosed space, the location that calculated from the proposed system slightly deviates from the physical location by 3.5039 meters. Practically, an error of 3.5 meters is acceptable since it is close enough to locate the access point.

The experiment in the open space environment shows that, if the environment contains many interference, the error will dramatically increase since RSSI is sensitive to the environment. We attempt to reduce the fluctuate value of RSSI by using the mode of RSSI values. Note that the average of RSSI values might not be suitable for reducing the error since the formula for calculating the distance is very sensitive.

The condition of the mobile phone is also one of the factor that causes RSSI fluctuation. We suggest using mobile phone that is in good condition for more accurate results. There are other available methods that could be more suitable in obtaining the distance using the RSSI values. However, these methods require additional hardware and more complex calculations.

Using GPS in indoor environment is not as accurate as in outdoor environment. Therefore, the mobile application allows the user to directly input the position of the mobile device. The mobile application also allows the user to select and change the path loss exponent value. This value is very important since it directly affects the correctness of the calculated distance.

VI. Conclusion

This paper proposed the prototype system for calculating the physical location of the targeted access point. The traditional IPS attempts to locate the mobile device by using three access points. However, the proposed method differ from the traditional IPS in such a way that the proposed method uses at least three mobile devices to locate the position of the access point. The proposed system can help the investigators when they want to locate the suspected access point in the building.

The proposed system is divided into 2 parts: the mobile client side and the server side. The mobile client attempts to collect RSSI values and send to the server. The server uses RSSI values from clients to calculate the distance between each mobile devices to the access point. Then, trilateration algorithm is applied to calculate the physical location of the access point.

We test the prototype system in 3 different locations, enclosed space, semi-open space, and open space. The enclosed space is the environment that we can control the interference. The semi-open space is the environment that contains a lot of interference. The open space is the real environment inside the building. The experimental result shows that the error of the proposed method is low in the enclosed space. However, in the environment that contains a lot of interference, the result is not as accurate as expected due to many factors that affect the calculation.

For future works, we will enhance the correctness of the proposed method by using least-square trilateration algorithm.

References

[1] Z. Yang, Z. Zhou, and Y. Liu, "From rssi to csi: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 25:1–25:32, Dec. 2013. [Online]. Available: http://doi.acm.org/10.1145/2543581.2543592

[2] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz, "A probabilistic room location service for wireless networked environments," in *Proceedings of the 3rd International Conference on Ubiquitous Computing*, ser. UbiComp '01. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 18–34. [Online]. Available: http://dl.acm.org/citation.cfm?id=647987.741335

[3] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '04. New York, NY, USA: ACM, 2004, pp. 70–84. [Online]. Available: http://doi.acm.org/10.1145/1023720.1023728

[4] A. Goswami, L. E. Ortiz, and S. R. Das, "Wigem: A learning-based approach for indoor localization," in *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '11. New York, NY, USA: ACM, 2011, pp. 3:1–3:12. [Online]. Available: http://doi.acm.org/10.1145/2079296.2079299

[5] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach, "Robotics-based location sensing using wireless ethernet," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '02. New York, NY, USA: ACM, 2002, pp. 227–238. [Online]. Available: http://doi.acm.org/10.1145/570645.570674

[6] G. V. Zàruba, M. Huber, F. A. Kamangar, and I. Chlamtac, "Indoor location tracking using rssi readings from a single wi-fi access point," *Wirel. Netw.*, vol. 13, no. 2, pp. 221–235, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1007/s11276-006-5064-1

[7] L. Watkins, R. A. Beyah, and C. L. Corbett, "A passive approach to rogue access point detection," in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, ser. GLOBECOM '07, 2007, pp. 355 – 360.

[8] K.Tejaswi, D. Umar, and K.Bhavana, "A review of online rogue access point detection," *International Journal of Computer Science and Engineering Technology*, vol. 3, no. 5, pp. 177–182, May 2013.

[9] F. Adelstein, P. Alla, R. Joyce, and G. G. R. III, "Physically locating wireless intruders," *Journal of Universal Computer Science*, vol. 11, no. 1, pp. 4–19, jan 2005.