Wi-Fi RSSI Localization based on ESP32

1.Introduction

Much research has focused on developing services architectures for location-aware systems. Using the angle of arrival, determining the angle of arrival of an RF signal requires using multiple antennas. Although this technique yields a high accuracy up to 40 cm on average as reported in [1], they would require special devices - mobile phones that use multiple antennas. The use of time of arrival faces synchronization issues as well as multipath. This worsens its accuracy and reliability [2]. Pretty research in RSSI Localization has been paid to the fundamental and challenging problem of locating and tracking mobile users in in-building environments, which include: trilateration (which as a poor accuracy), Particle filtering (which is good), Kalman filtering (which is also good), and the RSSI Fingerprinting method [3]-[6] . The goal of the project is to investigate the RSSI Finger printing method, implement and test an RSSI Fingerprinting Algorithm based on a new device--ESP32.

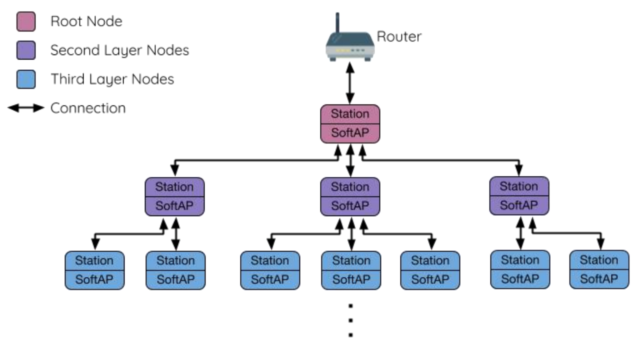
ESP32 is a chip with stable performance, high integration, ultra-low power consumption, Wi-Fi & Bluetooth solution, and more importantly, it has ESP-MESH network [7]. Its own Wi-Fi mesh network is strong, self-organizing and self-healing. Based on the ESP-MESH network, we can build the physical model easily.

The project uses a series of methods to build and verify the correctness of models and parameters to achieve the value of such networks with accurate user location and tracking capabilities. The experimental results are encouraging, indicating that a large number of location-aware services can be established on the ESP-MESH network.

2. Hardware Environment

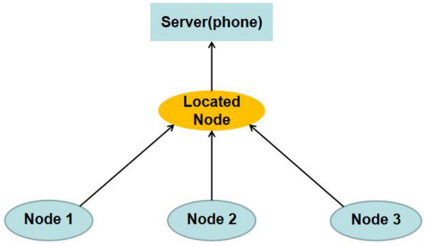
2.1 Structure

ESP-MESH is built atop the infrastructure Wi-Fi protocol and can be thought of as a networking protocol that combines many individual Wi-Fi networks into a single WLAN [8]. In Wi-Fi, stations are limited to a single connection with an AP (upstream connection) at any time, whilst an AP can be simultaneously connected to multiple stations (downstream connections). However ESP-MESH allows nodes to simultaneously act as a station and an AP. Therefore a node in ESP-MESH can have multiple downstream connections using its softAP interface, whilst simultaneously having a single upstream connection using its station interface. This naturally results in a tree network topology with a parent-child hierarchy consisting of multiple layers.



ESP-MESH Tree Topology

Based on the structure, we can build it as the localization structure. The root node, which connect with several STAs and the router(server), will be the localized node in the model. The leaf node, which detect the RSSI from the root node, will send the RSSI to Located node. The root node collect the information and computed the localization by certain algorithm. Then send the result to server, which maybe a phone.



Real Model

2.2 Network Formation

The root node can be dynamically elected based on the signal strength between each node and the router. Once selected, the node will connect with the router and begin allowing downstream connections to form.

Once the root node was selected, idle nodes in range of the root node will begin connecting with the root node thereby forming the second layer of the network. Once connected, the second layer nodes become intermediate parent nodes (assuming maximum permitted layers > 2) hence the next layer to form.

The preferred parent node is determined based on the following criteria:

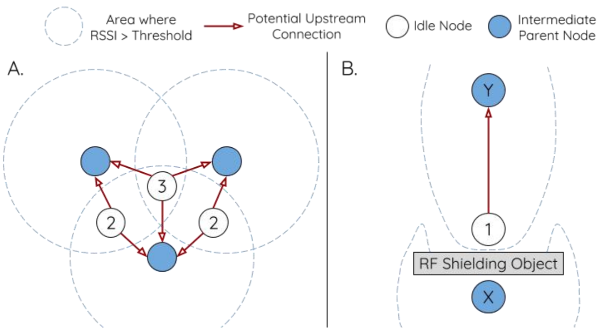
1.Which layer the parent node candidate is situated on;

2.The number of downstream connections (child nodes) the parent node candidate currently has;

That means the idle nodes will choose the higher layer nodes and less number of downstream connection nodes as theirs parents nodes.

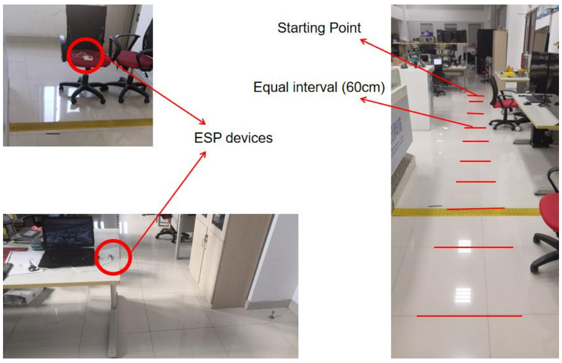
2.3 RSSI Thresholding

Once the ESP-MESH network has formed, the management of the network is contiune. The signal strength of a connection is represented by RSSI of the beacon frames of the parent node. To prevent nodes from forming a weak upstream connection, ESP-MESH implements an RSSI threshold mechanism. If a node detects a beacon frame with an RSSI below a preconfigured threshold, the transmitting node will be disregarded when forming an upstream connection.



Effects of RSSI Thresholding

In this project, the RSSI threshold will set as low as possible. Since the RSSI need to provide long enough distance to make the experiment success. To set the parameter reasonable, we did the experiment: record the relationship between RSSI and distance.

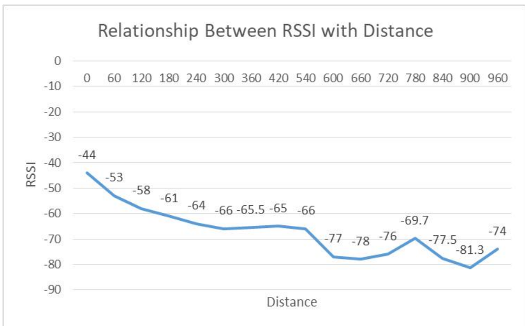


Experiment Environment

The Radio signal propagation loss model [9]:

$$P = -10\eta \log_{10}\left(\frac{d}{d_0}\right) + \chi$$

P is the received signal strength, ***d_0*** is a reference distance, χ is the Gaussian distribution factor, ***d*** is the distance between transimitter and receiver, η is the loss factor. From the figure, the experiment results is simlar to the loss model, while the experimental result is not a smooth curve. The reason why the curve is not smooth is the mutipath effect. From the result, we can set the RSSI Thresholding as -85dB. The localized node will not disconnect with the STAs to make the localization success.

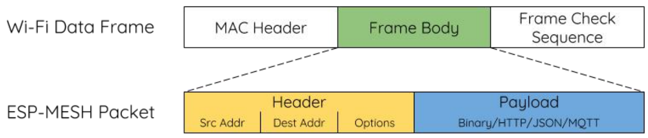


Experiment Result

2.4 Data Transmission

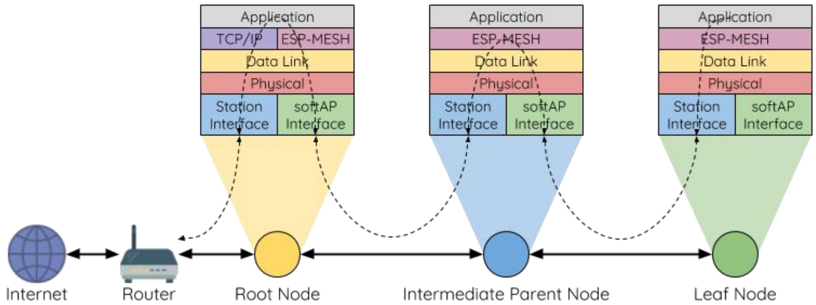
ESP-MESH network data transmissions use ESP-MESH packets. ESP-MESH packets are entirely contained within the frame body of a Wi-Fi data frame. A multi-hop data transmission in an ESP-MESH network will involve a single ESP-MESH packet being carried over each wireless hop by a different Wi-Fi data frame.

The following diagram shows the structure of an ESP-MESH packet and its relation with a Wi-Fi data frame.



ESP-MESH Packet

ESP-MESH is able to handle pack forwarding entirely on the mesh layer. A TCP/IP layer is only required on the root node when it transmits/receives a packet to/from an external IP network. The following diagram illustrates the various network layers involved in an ESP-MESH bidirectional data stream. Using the stream, we can transffer the computed position to our phone or other systems.



ESP-MESH Bidirectional Data Stream

3. Algorithm

At offline, consider there are $\boldsymbol{n}$ STAs in experiment environment, which can be expressed as a set $\boldsymbol{A = \{a_1, a_2, \dots, a_n\}}$. Suppose a total of sample points in offline locations is $\boldsymbol{m}$, which can be expressed as set $\boldsymbol{L = \{l_1, l_2, \dots, l_n\}}$, each samples corresponds to one coordinate $\boldsymbol{(x_i, y_i)}$, that is $\boldsymbol{l_i \leftrightarrow (x_i, y_i)}$. The RSSI fingerprint of each point is an n-dimensional vector, which expressed as a set $\boldsymbol{S = \{s_1, s_2, \dots, s_n\}}$. In this way, the offline RSSI fingerprint database can be expressed as a matrix of $\boldsymbol{m*n}$ dimensions.

$$L = \begin{matrix} l_1(x_1, y_1) \\ l_2(x_2, y_2) \\ \vdots \\ l_m(x_m, y_m) \end{matrix} \leftrightarrow \begin{Bmatrix} rssi_1^1 & rssi_1^2 & \cdots & rssi_1^n \\ rssi_2^1 & rssi_2^2 & \cdots & rssi_2^n \\ \vdots & \vdots & \ddots & \vdots \\ rssi_n^1 & rssi_n^2 & \cdots & rssi_m^n \end{Bmatrix}$$

At online state, every $\boldsymbol{\Delta t}$ from $0\sim t$, we detected RSSIs and compute the localization , we will get the localizations.

3.1 Nearest Neighbor(s) in Signal Space

Consider that the RSSI measurement value measured at time $\boldsymbol{t_k}$ is $\boldsymbol{s(t_k)}$, then the euclidean distance $\boldsymbol{D_i(t_k)}=| \boldsymbol{s(t_k)} - \boldsymbol{s_i} |^2$ at the i-th location fingerprint in the fingerprint library at this time is measured [6][10].

Finally, we find the nearest distance $\boldsymbol{D_i(t_k)}=| \boldsymbol{s(t_k)} - \boldsymbol{s_i} |^2$ and output its position $\boldsymbol{(x_i, y_i)}$.

3.2 Multiple Nearest Neighbor(s) (KNN)

Consider that the RSSI measurement value measured at time $\boldsymbol{t_k}$ is $\boldsymbol{s(t_k)}$, then the euclidean distance $\boldsymbol{D_i(t_k)}=| \boldsymbol{s(t_k)} - \boldsymbol{s_i} |^2$ at the i-th location fingerprint in the fingerprint library at this time is measured [6][10].

Then, we order the distance $D_i(t_k)$ ($i=\{1,...,n\}$) from small to large, and select first k distance $D_j(t_k)$ ($j=\{1,...,k\}$). Given $(x'(t_k),y'(t_k))$ as the output of KNN, the $(x'(t_k),y'(t_k))$ is as following:

$$x'(t_k)=\frac{1}{K}\sum_{j=1}^{K}x_j(t_k)$$

$$y'(t_k)=\frac{1}{K}\sum_{j=1}^{K}y_j(t_k)$$

3.3 Weighted KNN (WKNN)

Consider that the RSSI measurement value measured at time t_k is $s(t_k)$, then the euclidean distance $D_i(t_k)=|\, s(t_k) - s_i\,|^2$ at the i-th location fingerprint in the fingerprint library at this time is measured [6][10].

Then, we order the distance $D_i(t_k)$ ($i=\{1,...,n\}$) from small to large, and select first k distance $D_j(t_k)$ ($j=\{1,...,k\}$). Given $(x'(t_k),y'(t_k))$ as the output of WKNN, the $(x'(t_k),y'(t_k))$ is as following:

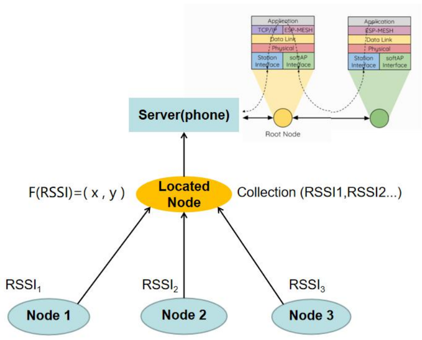
$$x'(t_k)=\sum_{j=1}^{K}\omega_j x_j(t_k)$$

$$y'(t_k)=\sum_{j=1}^{K}\omega_j y_j(t_k)$$

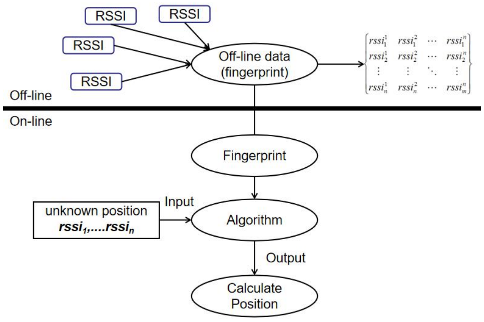
4. Experiment

Based on ESP-MESH, we build the localization structure. The root node, which connect with several STAs and the router(server), will be the localized node in the model. The leaf node (node1, node2, node3), which detect the RSSI from the root node, will send the RSSI to Located node. The root node collect the information and computed the localization by certain algorithm. Then send the result to server, which maybe a phone.

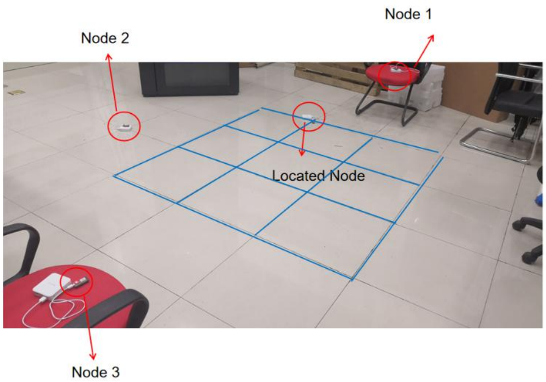
We use TCP/IP transmit information from the root node when it transmits/receives a packet to/from an external IP network. The following diagram illustrates the various network layers involved in an ESP-MESH bidirectional data stream.

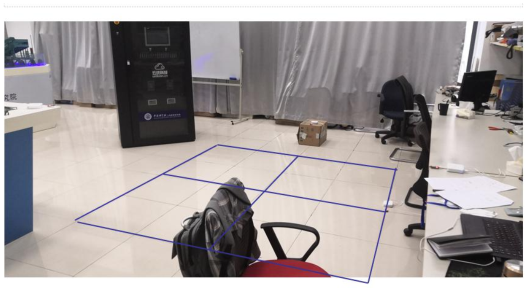


In this project, we have two steps, offline and online. At offline step, we collected enough nodes' RSSI datas serveral times, and store the average data in fingerprint database. At online step, our root node computed the location by the RSSIs. The experiment uses 3 algorithm NNSS, KNN, and WKNN.



We choose three STAs in three different place to localize a grid position. For 4*4 square, each grid is 60*60cm. For 3*3 square, the grid is 1.2*1.2 m. The experiment is like following:





5. Experiment Result

We choose three STAs in three different place to localize a grid position. Each points record 20 locations to calculate correct rate.

5.1 NNSS

The 4*4 square result was as following:

x\y	0	1	2	3
0	100%	90%	100%	100%
1	85%	**10%**	**0%**	70%
2	100%	85%	**0%**	**0%**
3	100%	**5%**	20%	100%

As the figure show, the center points has lower accurancy, the border points has higher accurancy. The reason why it's hard to distinguish the center points is that the resolution(0.6m) is high. The resolution restrict the center points distinguish.

So we enlarge the grid. The 3*3 square result was as following:

x\y	0	1	2
0	100%	100%	100%
1	100%	100%	**55%**
2	100%	100%	**100%**

x\y	0	1	2
0	100%	100%	100%
1	100%	100%	95%
2	100%	100%	**0%**

As the figure show, the results were improved a lot. However, the corner points get worse results. The impact of multipath has influence to the points RSSI receive.

5.2 KNN

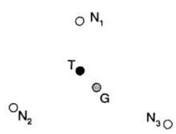
The results were shown as follow:

x\y	0	1	2
0	0%	0%	0%
1	0%	0%	100%
2	0%	0%	0%

The accurary is totally bad. This is that the entire database is in a fixed position on the grid. When a point has been correctly located in the database, KNN needs two other to calculate the average, which causes the originally correct point to be shifted. So we change the mind to calculate the error of the positions, the results as follow:

x\y	0	0.5	1	1.5	2
0	0.33,1.33		1.00,0.33		0.67,0.33
0.5		0.17,0.17		1.00,0.17	
1	0.33,0.33		0.67,0.67		0.00,0,00
1.5		0.17,0.5		0.5,0.17	
2	0.33,0.67		1.00,0.33		0.00,1.00

The error was different: The border points, fixed on the original grid point, have average error (*x=0.48,y=0.56*); The center, not fixed the grid points such as (0.5, 0.5), have average error (*x=0.46,y=0.25*). The center points have higher precision. The reason is that the KNN algorithm is suit for calculate the center poins instead of border points.



5.3 WKNN

After several test, we set the w_1=0.6, w_2=0.3, w_3=0.1 as parameters, the results is shown as follow:

x\y	0	0.5	1	1.5	2
0	0.10,0.80		0.40,0.30		0.40,0.10
0.5		0.15,0.40		0.9,0.25	
1	0.30,0.10		0.40,0.30		0.40,0,50
1.5		0.15,0.15		0.35,0.15	
2	0.10,0.40		0.40,0.00		0.00,0.50

The error was improved a lot. The border poins have average error (*x=0.28,y=0.30*) while KNN's results is (*x=0.48,y=0.56*); The center points have average error (*x=0.39,y=0.24*) while the KNN's result is (*x=0.46,y=0.25*).

The WKNN algorithm improve the results. This is beacause that the entire database is in a fixed position on the grid. WKNN allocates the shortest distance high weight, which make the position more rely on one location.

6. Conclusion

A ESP32 based RSSI Fingerprinting algorithm was implemented and tested. We use 3 different algorithms, WKNN shows the best results (0.28m, 0.30m). The measured RSSI fingerprints database was not optimal due to insufficient data and non-uniformity. The measured RSSI fingerprints have a potential to perform better if the RSSIs are very large and smartly sorted to reject outliers.

Reference

[1] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, “Spotfi: Decimeter level localization using wifi”, in *SIGCOMM 15*, 2015.

[2] Y. Chen and H. Kobayashi, “Signal strength based indoor geolocation”, in *Communications, 2002. ICC 2002. IEEE International Conference*, 2002, vol.1:436-439.

[3] P. Bahl and V. N. Padmanabhan, “Radar: an in-building rf-based user location and tracking system”, in *INFOCOM 2000*, 2000, vol.2: 775-784.

[4] T. Roos, “A probabilistic approach to wlan user location estimation”, in *International Journal of Wireless Information Networks*, 2002.

[5] M. Youssef and A. Agrawala, “The horus location determination system”, *Wireless Networks*, 2008, vol.14, no.3: 357-374.

[6] Paramvir Bahl and Venkata N. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System”, *Proceedings IEEE Infocom*, 2000.

[7] https://www.espressif.com/zh-hans/products/socs/esp32/overview

[8] https://docs.espressif.com/projects/esp-idf/zh_CN/stable/api-guides/mesh.html

[9] Wang Gai-yun, Lu Jia-zhuo, Jiao Ao， Guo Zhi-chao and ZHANG Qi, “RSSI Centroid Location Algorithm Optimized by Chaos ParticleSwarm Chicken Swarm Fusion Algorithm”, in *Computer Engineering*, 2020.

[10]高威,王可东. 基于 WiFi 的 RSSI 指纹定位方法[C]. 第十一届中国卫星导航年会论文集.中国卫星导航系统管理办公室学术交流中心:中科北斗汇(北京)科技有限公司,2020:104-108.