Ad-hoc mesh network working as a tracking system with the use of ESP32, Mosquitto broker and Paho MQTT client

Levy G. da S. Galvão[1] e Victor A. Ferraz[2]

***Abstract*— This paper implements an indoor tracking technique based in the bidirectional measurement of the received signal strength (RSSI) of each node present in an *ad-hoc* mesh network composed by ESP32 microcontrollers, so the distance between each node with known position and the unknown node can be estimated and allowing to track the coordinates of the latter device.**

I. INTRODUCTION

This paper describes an indoor tracking method to track a movable node position in real time by a *ad-hoc* mesh network composed by others three fixed nodes and, of course, the movable node. Every node is implemented by ESP32 microcontrollers capable of wireless connection. The distance is found using the Received Signal Strength (RSSI) value of each node 2.4GHz Wi-Fi signal.

Another method to compete with the RSSI is the Time-of-flight (ToF). The ToF method involves to measure the time that the signal takes to travel at the given wave speed (in the case of electromagnetic waves, this speed is the speed of light itself) from one node to other [1]. Since the wave speed is known and the time can be measured, the distance is found by a simple product. But since the devices used in this project have a maximum clock of 240 MHz, the time resolution would have be very large, resulting in large errors. A solution is to use a device with higher clock rate or track a device in an outdoor environment that involves high distances whose the error due to the clock is negligible.

Considering all complexities involving the propagation of 2.4GHz electromagnetic waves, such as obstacles that causes reflection, diffraction in dispersion, high attenuation and environmental changes causes by movement of people and objects, the application must to be restricted to indoor environments with low movement rate.

To improve the results, this paper will use a bidirectional RSSI measurement so the distance can be computed with higher efficiency than an unidirectional measurement, since a mean can be applied in the two distances to find a most accurate that minimize error [2]. The bidirectional reading is justified since each node can exchange data packets [3].

Once the distances of each fixed node to the movable node is found, a tracking algorithm is applied based in the trilateration, which is any technique that uses the distance between an unknown node and three other nodes with known position to track the coordinates of the unknown node. Coverage areas are defined and the intersection between them gives the unknown node position [2]. In practice the found position is an approximation of the real coordinates with more than one point. It is up to the experimental apparatus to reduce this error.

So in the following sections will it be explained:

- The trilateration equations;
- The *ad-hoc* mesh network topology;
- The bidirectional RSSI measurement tracking algorithm implemented by the network;
- Details on the experimental apparatus;
- Experimental results;
- Final comments on the results.

II. MATERIAL AND METHODS

A. Trilateration

Considering that the fixed nodes are arranged in a way that each one occupies a vertex of an isosceles triangle of side d, therefore height $h = d\sqrt{3}/2$, we can define the spatial arrangement of the trilateration nodes as seen in the figure 1. Each of of the three fixed nodes (yellow, blue and red) and the movable node (black) are correctly labeled and divided by color. Also the variables were set to denote the radius between each fixed node (FXD_ND_1, FXD_ND_2 and FXD_ND_3) and the movable node (MVB_ND_1) which is positioned in the ordered pair (r_x, r_y).

So the objective of the tracking algorithm is to execute a trilateration where each radius $r_k = \{r_1, r_2, r_3\}$ is given by the relation between received signal strength of each node to the respective distance in meters, so the ordered pair (r_x, r_y) for the movable node is found.

We can define the circle C_k centered in the coordinates of the fixed node $f_k = (r_k^x, r_k^y)$ with $k = \{1, 2, 3\}$, so the radius r_k of each circle is nonetheless than the Euclidian distance of the coordinates from the circle center to the point $m = (r_x, r_y)$ that represents the movable node m. So the distance from m to f_i is defined as:

$$d(m, f_k) = r_k = \sqrt{(r_x - r_k^x)^2 + (r_y - r_k^y)^2} \qquad (1)$$

However the points f_k are well defined for $k = \{1, 2, 3\}$ as pointed in figure 1, resulting in: $f_1 = (0, 0)$, $f_2 = (d, 0)$ and $f_3 = (d/2, d\sqrt{3}/2)$. Substituting each f_k coordinates in equation 1 the equation system of equation 2 is assembled (with $r_z = r_x^2 + r_y^2$):

*This work was not supported by any organization

[1] Electrical engineering undergraduate, Universidade Federal do Rio Grande do Norte, Brazil.

[2] Electrical engineering Ph.D, Universidade Federal do Rio Grande do Norte, Brazil.

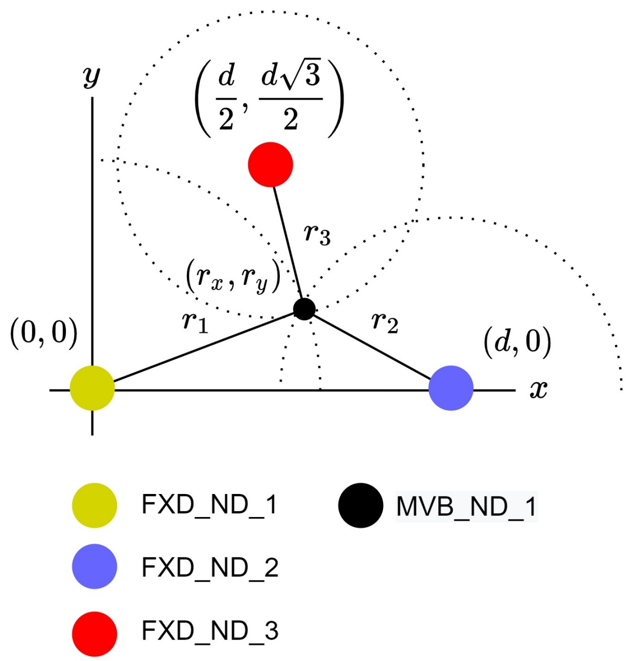

Fig. 1. Spatial arrangement of the trilateration nodes. Source: own.

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & -2d & 0 \\ 1 & -d & -d\sqrt{3} \end{bmatrix} \begin{bmatrix} r_z \\ r_x \\ r_y \end{bmatrix} = \begin{bmatrix} r_1^2 \\ r_2^2 - d^2 \\ r_3^2 - d^2 \end{bmatrix} \tag{2}$$

Solving the system of equation in 2, we obtain the movable point x-coordinate in equation 3 and the y-coordinate in equation 4.

$$r_x = \frac{1}{2d}(r_1^2 - r_2^2 + d^2) \tag{3}$$

$$r_y = \frac{1}{2\sqrt{3}d}(r_1^2 + r_2^2 - 2r_3^2 + d^2) \tag{4}$$

B. Network topology

The *ad-hoc* network is composed of four clients ESP32-WROOM-32U of which one is connected with an omnidirectional 3dBi antenna for higher power resolution; a PC running the Mosquitto MQTT broker locally; and a Paho MQTT Python client running as server that will present the results. The MQTT IoT architecture of this application is shown in the figure 2.

All ESP32 clients, from the point of view of the Wi-Fi, are configured in the mode access point and station. The reason for this choice is that each ESP32 must scan the Wi-Fi networks to retrieve the desired RSSI, in the case of node MVB_ND_1 it must read the RSSI value from all FXD_ND_Xs nodes and FXD_ND_Xs nodes must read the RSSI value of node MVB_ND_1. A ESP32 can only scan the Wi-Fi networks in the station (STA) mode, but to produce a network it has to be in the access point (AP) mode. It is important to note that all communication is done via the MQTT protocol and at no time will any ESP32 in station mode connect to another that is in access point mode.

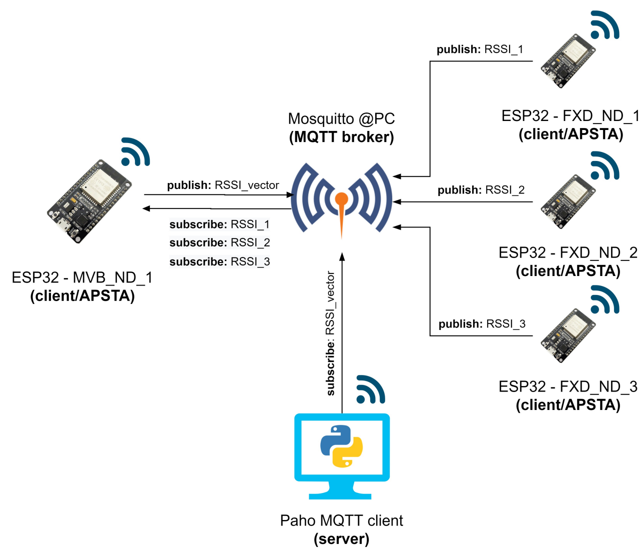

Fig. 2. Application MQTT IoT architecture. Source: own.

Once the MVB_ND_1 must perform a mean of the RSSI bidirectional readings, this client must to subscribe to the topics published by the nodes FXD_ND_Xs that contains the RSSI readings, also the latter nodes must publish correctly those readings. Once all RSSI values are together, which are the three read at node MVB_ND_1 and the others three read each one in the node FXD_ND_X, a simple mean is applied in the pairs and a vector of size three with the final RSSI values is created and published, so the Python server can receive this data via the subscription.

In the following section the details of the tracking algorithm running in loop are explained.

C. Tracking algorithm

The algorithm runs as follows:

1) All ESP32 clients use the Wi-Fi station mode capability of scan to apply in the nearer Wi-Fi networks and extract the specific RSSI values in dBm;
2) The fixed nodes publish the read values of RSSI;
3) The movable node subscribe to the topic of read values of RSSI by the fixed nodes and if the values sent by them reach the movable node, the latter apply a simple mean in the pairs and publish a vector of 3 values of RSSI, each corresponding to one distance value from the movable node to the corresponding fixed node (if the RSSI values from the other sources do not reach the movable node, the latter apply the mean in previous values until new values reaches it);
4) While the ESP32 clients prepare to another round of scan, the Python server receives the vector of 3 RSSI values and convert each one in distance and apply the trilateration to find the movable node position and outputs a numerical and graphical response to the user;
5) Repeat the first step;

III. EXPERIMENTAL CONFIGURATION

First of all, there is a need to establish a formulation to convert the RSSI value in dBm to a respective distance in meters. Kurose [3] proposes the equation 5 which establishes a relation between the RSSI value and a distance. Despite existing more complex formulas that takes into account other factors that are not present in this work [2], the equation 5 present itself as a simple alternative that embraces all the variables that we have in hands.

$$RSSI = REF - 10 \times PL \times log_{10}(d) \tag{5}$$

Where $RSSI$ is the received signal strength given in dBm; REF is the reference value of RSSI at 1 meter; PL is the path loss, which in case of free space is 2 (this value will be used in this work); and d that is the distance between the nodes.

The values of PL is the consequence of experimental measures for the given experimental apparatus. So to the determine this value for each device, the ESP32 were configured in a manner that its antennas where in line of sight, distancing 1 meter and without any moving obstacles in the vicinity.

The RSSI values where measured in a 30 minutes span and the results are shown in the figure 3. A moving mean of size 50 samples and normalized where passed through each of the measures to give a better looking plot. Also the mean of all samples of each curve where taken to serve as the reference value at 1 meter, a.k.a REF, for each device ($REF1 = -43.5225dBm$, $REF2 = -26.3252dBm$, $REF3 = -29.7935dBm$).

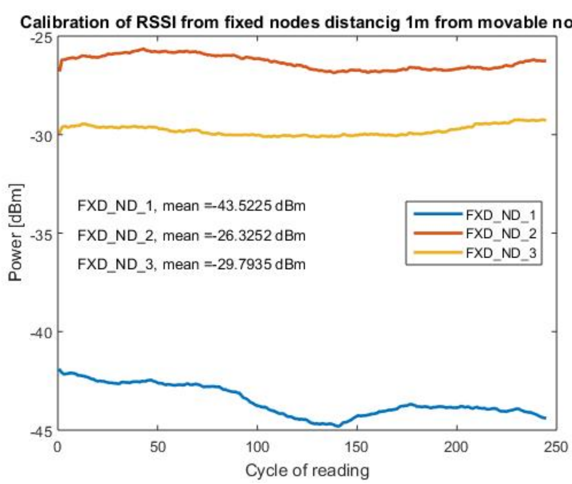

Fig. 3. Calibration experimental results for the movable node reading the RSSI of each fixed node. Source: own.

The reference values of RSSI in some cases differs between device and must be set regularly depending on the environmental configuration and in the state of the device.

In the following sub sections there is some experiments that were made to validate the model.

A. Experiments

TO validate the system there were made 4 experiments. Each experiment had the position of the fixed nodes maintained fixed all the time and in line of sight with each one and with the movable node. The distance d between each fixed node was fixed to $d = 150cm$ and the resulting height of the isosceles triangle is $h = 130cm$. At the end, the coordinates of the fixed nodes became: $f_1 = (0, 0)\ cm$, $f_2 = (150, 0)\ cm$ and $f_3 = (75, 130)\ cm$.

The movable node had its position changed in each experiment, in a way that:

- Experiment 1: $m = (75, 65)\ cm$;
- Experiment 2: $m = (0, 0)\ cm$;
- Experiment 3: $m = (150, 0)\ cm$;
- Experiment 4: $m = (75, 130)\ cm$;

So the first experiment had the movable node in the center of the triangle and the other threes experiments had the movable node next to each fixed node, from node 1 to 3 according to the experiments 2 to 4.

The results of each experiment are commented in the results section.

IV. RESULTS

The graphical results of each experiment are presented in the figure 4, 5, 6 and 7.

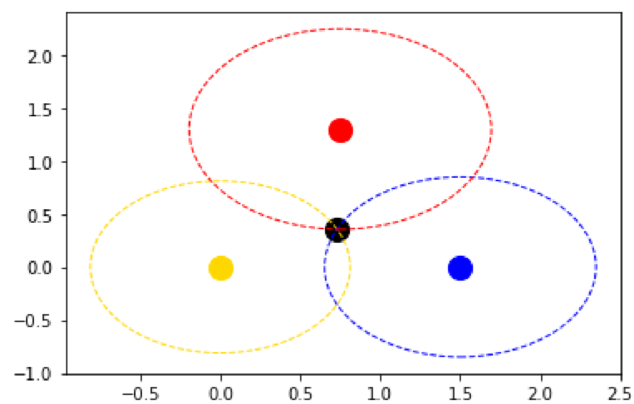

Fig. 4. First experiment graphical results. Source: own.

The first experiment showed good results, estimating the theoretical point $m = (0.75, 0.65)$ meters to point $M = (0.73, 0.48)$ meters with the RSSI vector been $\{-28.5, -16.0, -24.0\}$ dBm. This result might be explained by good calibration results that were performed in this range.

Although the other experiments do not show close results in the way that:

- Experiment 2 tracked the point $m = (0, 0)$ meters to point $M = (-18.76, 12)$ meters with the RSSI vector been $\{-18.0, -44.0, -33.0\}$ dBm;
- Experiment 3 tracked the point $m = (1.5, 0)$ meters to point $M = (1.2, 0.75)$ meters with the RSSI vector been $\{-46.0, -22.5, -26.5\}$ dBm;

- Experiment 4 tracked the point $m = (0.75, 1.3)$ meters to point $M = (-0.54, 1.2)$ meters with the RSSI vector been $\{-37.5, -32.5, -12.0\}$ dBm;

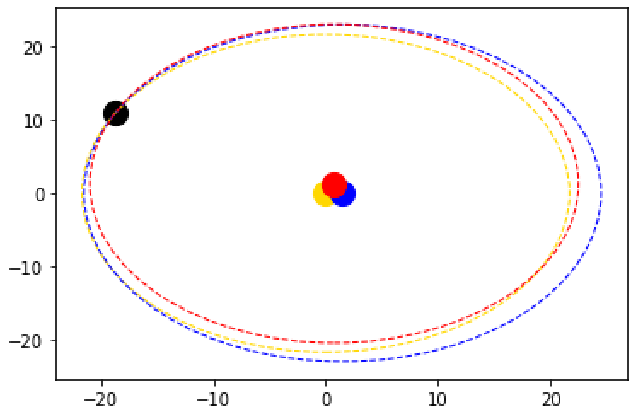

Fig. 5. Second experiment graphical results. Source: own.

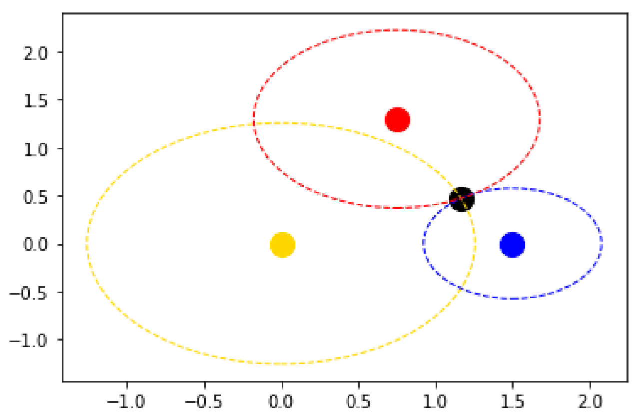

Fig. 6. Third experiment graphical results. Source: own.

Despite the results do not match with the real values of distance, the tracking algorithm indicates a good estimation for the direction that the movable node is located.

The divergent values might be a direct result of the 1 point calibration, since the calibration did not take into account the proximity of the movable node to the other nodes.

V. CONCLUSIONS

Finally the results did not showed good estimations when the movable node is next to the fixed nodes, but when in a far field, it presented good approximations.

A suggestion for future works is to realize a multi point calibration routine that takes into account different scenarios and different position.

Despite all, the tracking system pointed the movable node in the direction that it actually is, so in applications that require to locate a specific node for e.g. security purposes or to find a lost device, the users should have a good starting point to do the research.

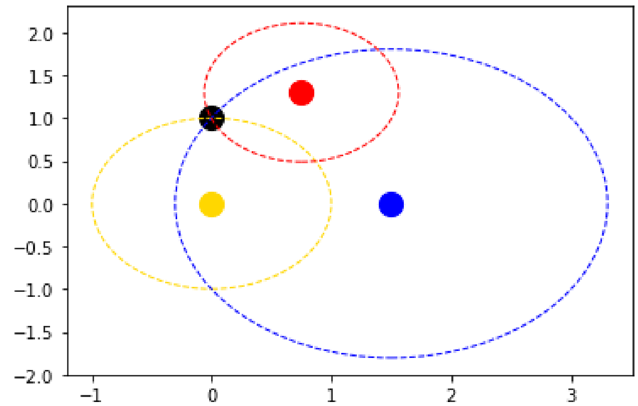

Fig. 7. Fourth experiment graphical results. Source: own.

REFERENCES

[1] Ramírez, Alejandro. Time-of-flight in wireless networks as information source for positioning. Diss. Technische Universität München, 2011.

[2] Bellecieri, Ysla, Filippe Jabour, and Eugenia Jabour. "Localização indoor baseada na leitura bidirecional do rssi." Seminários de Trabalho de Conclusão de Curso do Bacharelado em Sistemas de Informação 1.1 (2016).

[3] Kurose, James, and Keith Ross. "Computer networks: A top down approach featuring the internet." Peorsoim Addison Wesley (2010).