Simulation of IoT Based Tree Localization Application in Contiki-OS

Shivam Trivedi[1], Naren Tada[2], and Dr. Tejas Patalia[3]

[1]M.Tech student, Department of Computer Engineering, V.V.P Engineering College, Gujarat Technological University, Gujarat, India.
shivtrivedi197@gmail.com

[2]Assistant Professor, Department of Computer Engineering, V.V.P Engineering College, Gujarat Technological University, Gujarat, India.
naren.tada@gmail.com

[3]Head of Department, Department of Computer Engineering, V.V.P Engineering College, Gujarat Technological University, Gujarat, India.
pataliatejas@rediffmail.com

Abstract— **Internet of thing is widely used term as progression in innovation. It is a system of physical devices embedded with electronic devices, sensors, and software which empower these devices to exchange data through the internet. Moreover, to monitor environmental conditions on the real-time basis we utilize GPS framework but it expends more energy. Here, the issue is to get real-time data by utilizing low energy in a deployed network. In this research paper, we have concentrated to improve GPS independent localization algorithm for real-time tree monitoring. This algorithm comprises of two sub-algorithms which are RSSI and trilateration. For monitoring trees in real time then localization plays a key role in the system. Localization utilizes routing of data and routing also consumes more energy. Therefore, we require energy efficient routing protocol and system which stays last long. For energy efficient routing we are utilizing RPL routing protocol which is developed specially for low power and lossy networks. Simulation of the whole system is carried out in Contiki-OS with the help of built-in COOJA simulator.**

Keywords— ***IoT*, Environment monitoring, Localization, Contiki OS, COOJA, Wireless sensor network, Sensor network.**

I. Introduction

Trees are playing a crucial role in the ecosystem. Urban street trees have many proven benefits including, providing oxygen, storing carbon dioxide, stabilizing soil, providing shelter and shades and food and raising property value and many more. These benefits are crucial for communities but trees are more fragile then we think and the urban landscape is a harsh place to thrive and survive. City foresters are trying hard to tracking and managing urban forests but almost 60% urban trees are grown on private property. This makes difficult to track down and managing each and every tree.

Keeping in mind the importance of tree in our life, Indian judiciary system have decided, once in five years, mandatory carrying out a tree census program which is also called as tree act 1975. In April 2016, Pune Municipal Corporation started a high-tech tree census program [1] to track the location, type and other details of every tree in the city area. In their high-tech project they are using GPS & GIS, as the system is equipped with GPS then there must be a sensor present. When we start talking about sensors, the question that many people ask is: “Why don’t we just use GPS, surely that’s enough?” GPS is everywhere it’s on our phones, it’s in our cars, and it tells us where we are.

As GPS is using 1.5 GHz high frequency; those radio waves could not penetrate through a dense medium. When we talk about the urban area, there are many high rise buildings so the sky becomes obscured and sensor barely sees the satellites, then it may be possible of inaccurate reading. In another case, where it is placed underwater or underground; Radio waves might penetrate few millimeters not more than that and again we get inaccurate readings. When we are talking about monitoring trees, which is quite similar to the previous scenario where sensor might not receive radio signals because of a high tree canopy. Water molecules of leaves absorb radio waves from the satellites, and they won’t allow signals to penetrate. Therefore, the whole idea is to create a smart tree monitoring system using localization algorithm which is GPS independent.

The goal of this paper is to localize blind nodes using low power routing. For the routing, we have used RPL protocol which was developed by IETF (Internet Engineering Tasks Force) for low power and lossy network [2]. For localization three anchor nodes are used, anchor nodes are the nodes whose positions are known. They perform RSSI based trilateration to estimate the position of blind nodes in the deployed network. This work was performed in Contiki-OS [3] with the help of COOJA simulator.

II. PROPOSED SYSTEM

As trees are the static entity we don’t need to measure mobile node. Therefore, it reduces the complexity a bit. We need a system which gives us real-time data and precise location of the tree. The proposed solution works in four phases: i) RSSI calculation, ii) Distance estimation iii) Capturing position coordinates, and iv) position estimation, After completion of first phase sensors starts monitoring tree. Sensor stores data and send it to the base station based on the information we get, we need to take action to regulated expected values. In the below figure we’ve depicted the flow of my proposed system

.

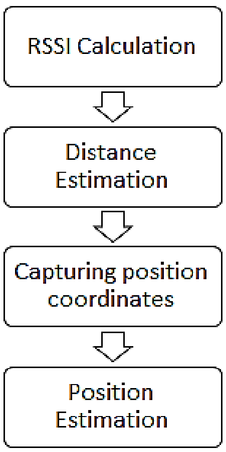

Figure 2.1 Proposed System

A. *Received Signal Strength Indicator*

The distance estimation includes calculation of RSSI values from the anchor nodes and based on that value, the corresponding distance estimate is figured by the blind node. The RSSI values are obtained by using CC2420 radio transceiver that is fitted in all the sensor nodes. The mathematical formulation is given as follows:

$$RSS = RSS\ VAL + RSS\ OFFSET\ [dBm]$$

Where RSS OFFSET is roughly -45 [4] and RSS VAL is the power received by the CC2420 radio transceiver. Utilizing RSSI value, the distance estimation is done as follows. Expecting that the transmission power P_{tx}, the path loss model, and the path loss coefficient are known, the receiver can use the received signal strength P_{rcvd} to solve for the distance d in a path loss equation like;

$$d = ((C * P_{tx}) / P_{rcvd})^{\alpha}$$

B. *Trilateration*

In geometry, trilateration is the way toward deciding outright or relative areas focuses on estimation of separations, utilizing the geometry of circles, spheres or triangles. Trilateration involves measuring distance and three anchor nodes. Using those three anchor nodes it locates blind nodes within their range. If we use more than three anchor nodes then it is called as multilateration which have more accuracy than trilateration. See the below image to understand more about this concept.

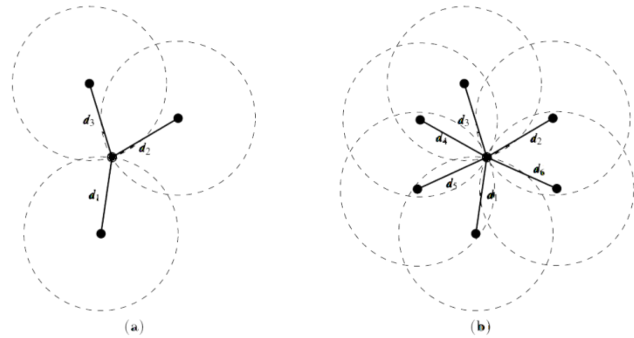

Figure 2.2: Distance measurements and multilateration (Ranging circles). (a) Trilateration. (b) Multilateration.

C. *RPL for Low Power and Lossy Network*

RPL [2] is a routing protocol for low power and lossy network. To support large-scale network IETF has defined this routing protocol. RPL is distance vector routing protocol, it does not contain pre define topology but it will generate through the construction of DODAG (Distance Oriented Directed Acyclic Graph). It is a tree-like structure in which one node allows to have multiple parent nodes. The DODAG topology maintains with control messages such as DODAG Information Object (DIO), DODAG Information Solicitation (DIS) and Destination Advertisement Object (DAO) messages these all messages belongs to Internet Control Message Protocol (ICMP).

D. *Small Scale RPL Simulation in Contiki*

For the small scale RPL simulation, we have chosen COOJA simulator. A brief introduction regarding Contiki and Cooja is given in the next chapter. COOJA takes more time for processing and this reason to choose it for small-scale simulation. Network topology has been given in fig 2.3.

Figure 2.3 Small scale implementation of RPL in COOJA

In this simulation node number 1 root node which is responsible for creating DODAG root. Other nodes are neighbor nodes. Node number 4 is directly in the transmission range of root node. Other nodes are not in direct range of root. Therefore, if they want to communicate with a root node, they need to make a connection with node number 4.

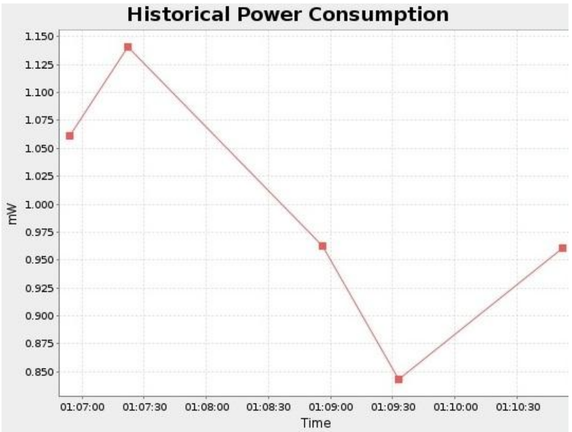

Figure 2.4 Historical Power Consumption

Fig 2.4 describes how power utilizes during routing in RPL. Initially, the power consumption is high due to all nodes are participating in the formation of stable DODAG. Once the DODAG is formed, there is no need to process data. Therefore, chronologically power consumption of entire topology will decrease. RPL topology is stable topology until any new node is added or any present node is removed from the topology.

Therefore, we are concluding that RPL is a best suitable protocol for our proposed system. RPL uses multi-hop environment which is beneficial for environmental monitoring because this is huge system and thousands of nodes are deployed for mounting the environment. One more thing RPL promises, even working in a huge network it always chooses the best path that is shorter routing path. RPL also promises network stability and fast network set up. Therefore, RPL is best suitable for our proposed system.

III. Implementation and Simulation

All the simulations were performed using Contiki-OS v-2.7 and emulated sky motes. The proposed algorithm was simulated in Contiki's built-in simulator COOJA. The environment is set to unit disk graph medium (UDGM): distance loss for the deployment of sky motes in the network. Up to four nodes were deployed for the purpose of simulation that includes a blind node (also called as sink node) and three anchor nodes. The transmission range and interference range is set to 50 and 100 meters respectively. Enable viewing of actual positions of deployed nodes in the network in order to compare with the estimated position obtained in the mote output area. The deployed anchor nodes should be within the transmission range of the blind node in order to enable transmission of packets.

A. *RSSI Contiki Implementation*

To estimate precise location using Trilateration method, we first have to calculate RSS value of each node. To implement this procedure in Contiki, we need to make simulation. Here we have taken a simulation of nodes. The distance estimation involves calculation of RSSI values from the anchor nodes and based on that value, the corresponding distance estimate is calculated by the blind node. The RSSI values are obtained by using CC2420 radio transceiver that is fitted in all the sensor nodes.

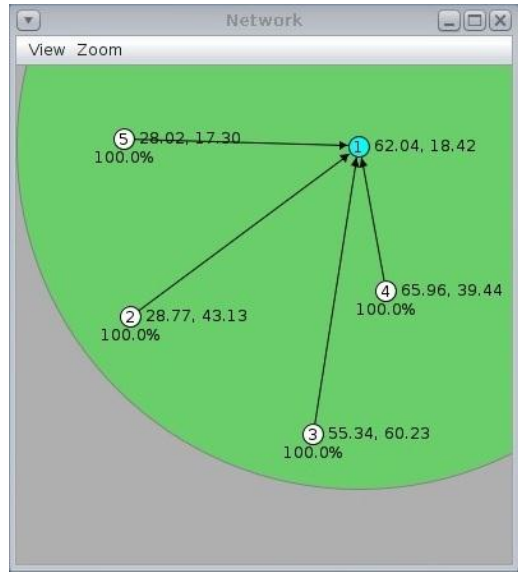

Figure 3.1 Simulation of RSSI

Fig 3.1 shows the network of RSSI simulation. Node number 1 is a root node and other nodes are neighbor nodes. RSSI is a line of sight communication; therefore, if any node is outside of the transmission range of root node then we cannot get the RSSI value of that particular node. Here, in fig 3.1, all nodes are in the transmission range of root node so that we are able to get RSSI of each node.

RSSI refers to the transmitter power output as received by a reference antenna (receiver). Transmitter continuously sends packets to the receiver and receiver nodes continuously receive those packets. To calculate RSSI form those received packets we need proper function. In Contiki, we have packetbuf attr() which is used to create an outbound packet to store an inbound packet. When the driver reads out received frame, the RSSI observed during the reception will get copied to the packetbuf.

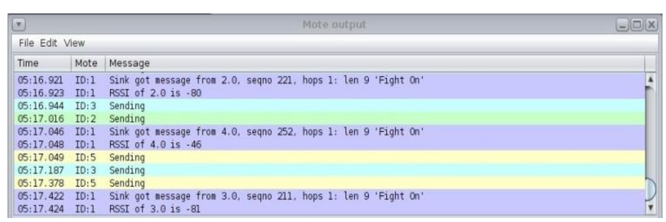

Figure 3.2 Mote output of RSSI

Fig 7.2 shows the output of the RSSI value of the created network in fig 7.1. Root node which id node number 1 is getting output from the other nodes. Node number 2 and 3 are located far from the root node, therefore, we are getting high RSSI value for them which is -80 dbm and -81 dbm respectively. Node number 4 is located near to the root node, therefore, its RSSI value is high as compared to node number 2 and 3 which is -46 dbm.

B. Getting Position of Nodes

There is no built-in mechanism to get a position of network nodes in Contiki. Another thing is that source of Contiki is written in C language and COOJA simulator is using JAVA language for simulation. We can easily get nodes position using COOJA motes but using different motes types we can’t utilize this functionality. For the implementation of our system, we are utilizing TMote Sky which often called sky motes, if we select this mote environment for node position we can’t get any results. For solving this problem we need to create some variables and interfaces which can communicate between Contiki and COOJA. Therefore, we require creating a new interface in the COOJA simulator.

After creating a new interface in COOJA we have to create a new simulation in COOJA simulator. After adding mote interface selected C source file which calculates the position of nodes. Fig 7.6 shows the network topology of 2 nodes whose position is going to estimate. For the simplicity, we have chosen only 2 nodes. Node number 1 is a root node and node number 2 is a neighbor node. The simulation environment is 2D hence only X and Y coordinate will be computed.

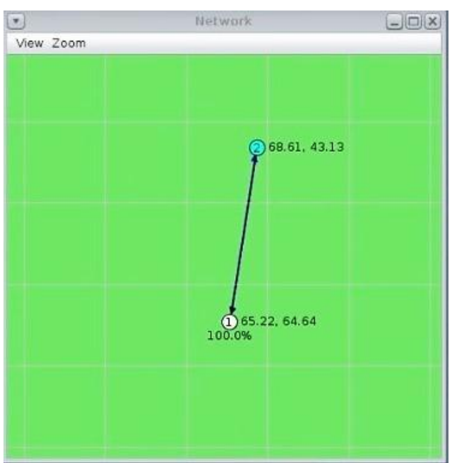

Figure 3.3 Network topology of 2 nodes

After hitting start in the simulator the script will run in the background and started getting the position of nodes. But as the compiler used in Contiki does not support floating point value we are unable to get exact floating value. The next thing is we are running a simulation in a 2D environment but the code is able to work in a 3D environment. That means the source code is designed to work in real time environment which captures all X, Y & Z coordinates. Due to simulation environment restrictions, we are only getting a value of X and Y coordinates.

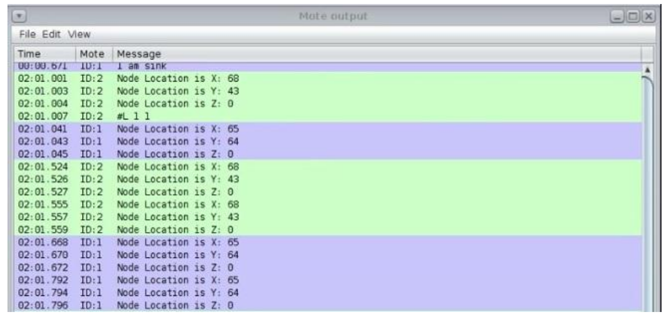

Figure 3.4 Mote output for node position

B. Trilateration Algorithm

Before, moving towards algorithm there are few terms which are required to be clear in mind. The first terms among them are Anchor node, this is the node whose location is already known which will help us to find that blind node. The second term is a blind node, a node whose location is not decided yet, which we are going to calculate. The third term is a coordinate matrix, as we are using 2D environment for the simulation this coordinate matrix will store the X & Y axis in an array. The fourth and last term is estimated distance matrix, this term will store the X & Y coordinates of a blind node after calculation.

The simulation was conducted at a specific 8 positions coordinates of the anchor nodes and resulting in actual and estimated coordinates of blind nodes. After estimating all position coordinates, the localization calculated as follows.

$$\text{Localization Error} = ((x_2-x_1)^2 - (y_2-y_1)^2)^{0.5}$$

In the above equation (x_1, y_1) is actual position coordinates of a blind node and (x_2, y_2) is the estimated position coordinates of the blind node. The simulation was repeated for more than 50 times and obtained values are finalized. The overall localization error was obtained by finding the average of all the obtained localization error during the simulation.

For the simulation of trilateration algorithm, we have created a simulation using 8 sky motes in COOJA simulator. In figure 3.5, 8 sky motes are shown using 10-meter boundary grid. In the deployed network we require to define 3 anchor nodes to start the trilateration process, hence node 1 (85.31, 41.94), node 6 (64.24, 37.49), and node 8 (89.54, 15.23) are anchor node and will try to estimate the position of remaining blind nodes.

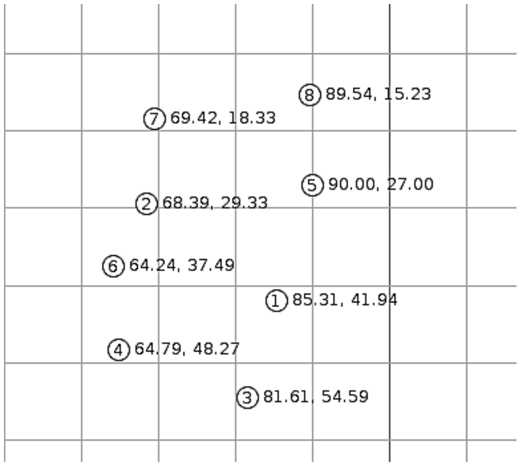

Figure 3.5: Network topology for proposed system

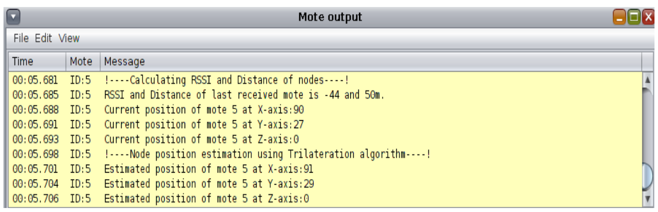

Figure 3.6: Mote output of proposed algorithm

IV. Result and Analysis

The steps involved in the proposed system are shown in figure 2.1. The very first step is to calculate the RSSI value of every node and from it algorithm calculate the distance of the deployed nodes. For that 3 pre-defined anchor nodes send the packets to the blind nodes, and based on the received power RSSI value will be calculated. Based on the calculated RSSI value blind node estimates its distance from the anchor node. In fig. 3.6, entire output is shown for the mote number 5.

Mote No.	Actual Coordinates	Estimated Coordinates	Localization Error (m)
1	(85.31, 41.94)	(85.31, 41.94)	0.00
2	(68.39, 29.33)	(67.39, 28.33)	1.4142
3	(81.61, 54.59)	(80.61, 55.59)	1.4142
4	(64.79, 48.27)	(66.79, 46.27)	2.2828
5	(90.00, 27.00)	(91.00, 29.00)	2.236
6	(64.24,37.49)	(64.24,37.49)	0.00
7	(69.42, 18.33)	(70.42, 19.33)	3.6055
8	(89.54, 15.23)	(89.54, 15.23)	0.00

Table 4.1: Simulation Results and Localization Error

After calculating and estimating, RSSI and distance from anchor nodes, for position estimation of blind nodes, we need to get the actual coordinates of anchor nodes. The position coordinates of anchor nodes will be sent to the blind nodes. Utilizing the anchor nodes coordinates blind nodes position estimation will be performing using trilateration algorithm. For the position estimation of blind nodes, we must require estimated distance and position coordinates of anchor nodes.

Here, for the distance estimation, we are utilizing RSSI and which are prone to error because of multipath propagation in the real environment. There must be some error present in the final estimated position coordinates of the blind nodes. Actual position coordinates of the blind nodes and estimated position coordinates are shown in the table below with localization error.

The simulation was repeated more than 10 times with fixed coordinates and the overall localization error was obtained by finding the average of all the obtained localization error during the simulation. The overall localization error obtained after performing all the simulation rounds is 1.37 meters. In the real-time environment, multipath shading and multipath fading must be present and RSSI is prone to error. Yet, the obtained localization error is acceptable in the real-time environment. Simulation of last round is shown in table 4.1 for all the 8 deployed nodes.

V. Conclusion and Future Work

This proposed system was built for the purpose of an evolution of GPS independent localization scheme. Utilizing this scheme we can determine position coordinates of a blind node on the based on the distance estimation and position coordinates of the anchor nodes deployed in the network. For the simulation, we have utilized Contiki-OS with the help of COOJA simulator. After getting final simulation result the overall localization error obtained is about 1.37 meters. In the localization field, this margin is not accurate enough. Yet, the environment in which this system will be going to use is experiencing varying channel characteristic, multipath fading, and multipath shadowing, then the obtained localization error is acceptable.

This work can be additionally reached out later on by enhancing the positional accuracy of a blind node by considering distance estimations from more than 3 nodes; this is about utilizing multilateration strategy. The positional accuracy of a sensor node is vigorously relying upon the precision of the distance estimation based on the RSSI value. Since RSSI value does not stay consistent notwithstanding when the sender and recipient node does not move, henceforth to improve the precision of distance estimation, RSSI values are taken ceaselessly for stretched out period and arrived at the midpoint of out to acquire a decent separation evaluate bringing about enhancing the positional exactness of the blind node.

References

[1] P. M. Corporation, "Tree census: Pune municipal corporation." https://pmc.gov.in/en/tree-census-1. Last accessed on 15 March 2018.

[2] O. Gaddour and A. Koubˆaa, "Rpl in a nutshell: A survey," Computer Networks, vol. 56, no. 14, pp. 3163–3178, 2012.

[3] "Contiki: The open source operating system for the internet of things." http://www.contiki-os.org/. Last accessed on 2 February 2018.

[4] "Chipcon products from texas instruments." http://web.stanford.edu/class/cs244e/papers/cc2420.pdf. Last accessed on 10 February 2018.