Technische Universität München

Lehrstuhl für Kommunikationsnetze

Prof. Dr.-Ing. Wolfgang Kellerer

Project ASTRA – Asset Tracking for hospitals

Wireless Sensor Networks Lab

Supervisor: Samuele Zoppi

Members:

Abhishek Sunkum Rammurthy

Pramod Tikare Muralidhara

Yadhunandana Rajathadripura Kumaraiah

Abstract:

This project was undertaken as part of Wireless Sensor Networks Laboratory. We introduce a solution for tracking of assets in the big enterprises in an efficient way. Assets are considered as moving wireless nodes and so visible to tracking wireless nodes which form a network. We use ad hoc distance vector protocol for dynamic –energy aware routing between nodes of the network. There are two types of nodes gateway nodes and asset tracking nodes. Asset tracking nodes forward asset information to gateway through shortest path calculated based on distance and energy awareness. The project is implemented using networking operating system Contiki which is based on protothreads, event driven programming model [1], hence ideally suited for low power/resource constrained embedded systems.

Table of Contents

Table of Figures

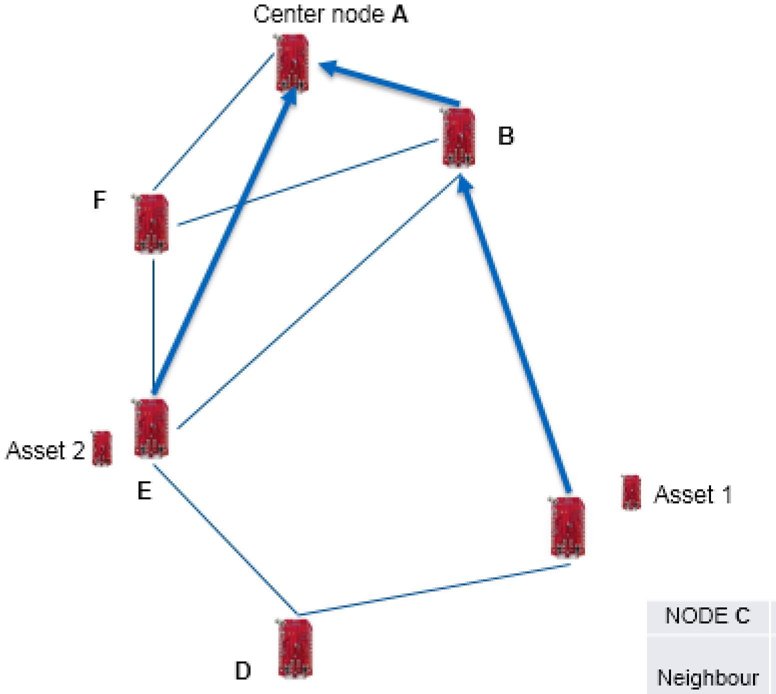
1. Introduction

In large enterprises, the main problem is to track the assets. Asset tracking solution allows users to view enterprise assets located across multiple locations. The solution was designed to provide real-time visibility and control of all high-value and/or mission-critical assets to help increase asset and labor optimization in many different enterprises.

Our project is specifically targeted towards hospitals asset management, where people tend misplace high valued/Mission critical assets such as Sphygmomanometer, Electrocardiogram etc. Due to increased patient volume, these equipments (assets) tend to get misplaced more frequently. Misplacement of assets reduces the labor productivity and assets are not available when there is an emergency. [2] An average hospital can lose $4,000 per day in lost wages due to time spent searching for mobile medical equipment, not to mention the over-procurement of assets to ensure availability. Through efficient asset tracking enterprises can automatically collect information about how frequently an asset is utilized then can decide whether enterprises need to procure more assets or not.

At present, Asset Tracking solutions use different technologies such as ZigBee, BLE, RFID and combinations of these, hence there is a lack of standardization and heterogeneous implementation. This exponentially increases the complexity of the system. Using wireless sensor nodes provides homogeneous, standard implementation which is scalable with literally zero increase of complexity.

2. Architecture/Working – Minimum hop energy aware routing algorithm.



NODE C	
Neighbour	Hops to A
B	2
D	3

Figure 1: Architecture Graph

As can be seen from the above diagram, there is one gateway node/Center node and remaining nodes are asset tracking nodes. Each asset tracking node forwards the asset tracking information to gateway node using its routing table, which is built on the objective of minimizing energy usage and hop count. Every asset continuously broadcasts its presence. The tracking motes forward this asset packet towards central mote. In the above diagram, Node C has two routes to reach the gateway node one through node b with cost 2 and through node D with a cost of 3. This cost is estimated based battery level of the nodes on the route and number of hops. Highlighted lines in the figure show the path taken to reach gateway node. C->B->A.

Dynamic rerouting:

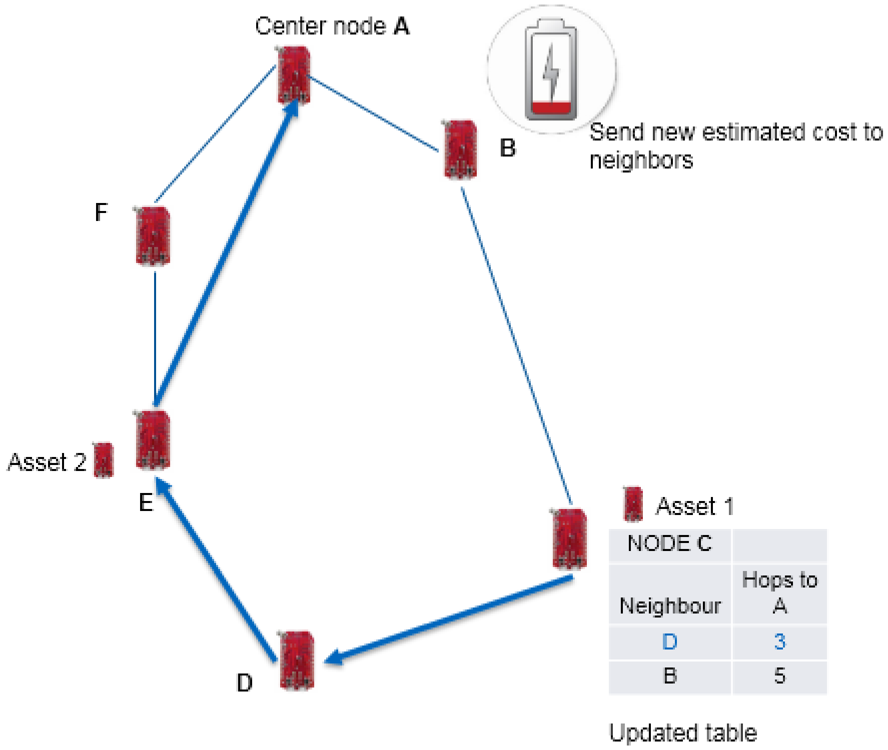


Figure 2: Dynamic Routing

Now let's look at what happens when the battery level gets reduced in the node B. As the battery level gets changed, B estimates the new cost to reach gateway. B informs its neighbors about the new cost. Neighbors update this table with new cost. New path to gateway is estimated based on the new cost.

In the above figure, updated routing table shows that node C. We can see that cost to reach gateway through B is increased from 2 to 5. So, the shortest path is through D with a cost of 3. This demonstrates the energy aware dynamic rerouting of the packets based on the new battery level.

Approach:

In this project, totally 8 motes are used. two are used as asset motes, five tracking motes and one gateway mote. Asset motes are kept on assets which need to be tracked. They continuously broadcast the asset information. This information is forwarded by asset tracking motes towards the gateway mote.

3. Implementation:

Application block diagram:

Following diagram represents brief overview of the project implemented.

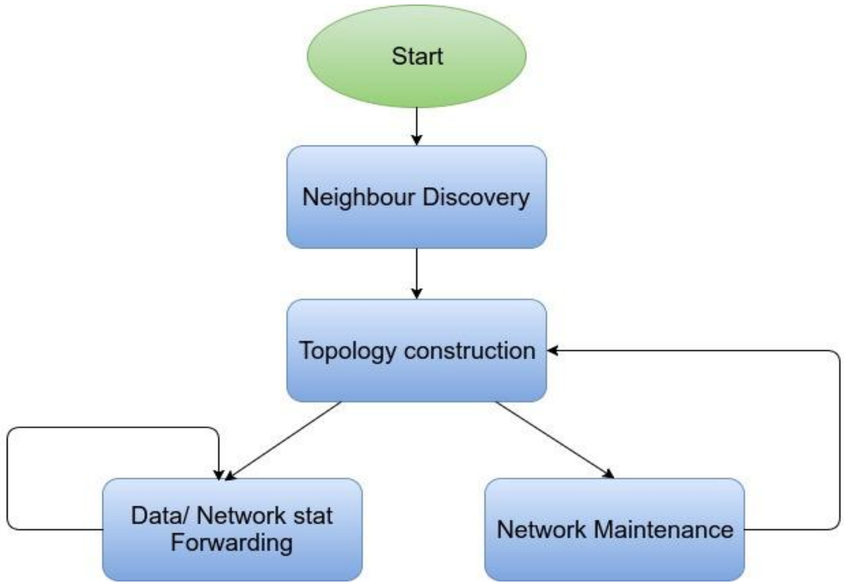


Figure 3: Application Block Diagram

As soon as the system boots up, all the motes enter neighbor discovery mode. In this mode, motes broadcast neighbor discovery packets with cost infinity (Cost field is five bits wide, hence cost greater than or equal to 31 is considered as infinity), In this process, when the broadcast packet is received motes update the routing table and corresponding cost with infinity.

Once neighbors are found, motes enter topology construction mode. Nodes inform their neighbors about gateway presence and cost to reach gateway to their neighbors. After this shortest path is estimated to reach the gateway.

After topology construction phase, two processes are running concurrently, Data/Network stat forwarding process and Network maintenance mode. Data/Network statistics process as the name suggests forwards the asset tracking and network statistics packets such as battery level and routing table to gateway node.

Let's go into each of the modes shown in the figure in a detailed way.

Neighbor discovery mode:

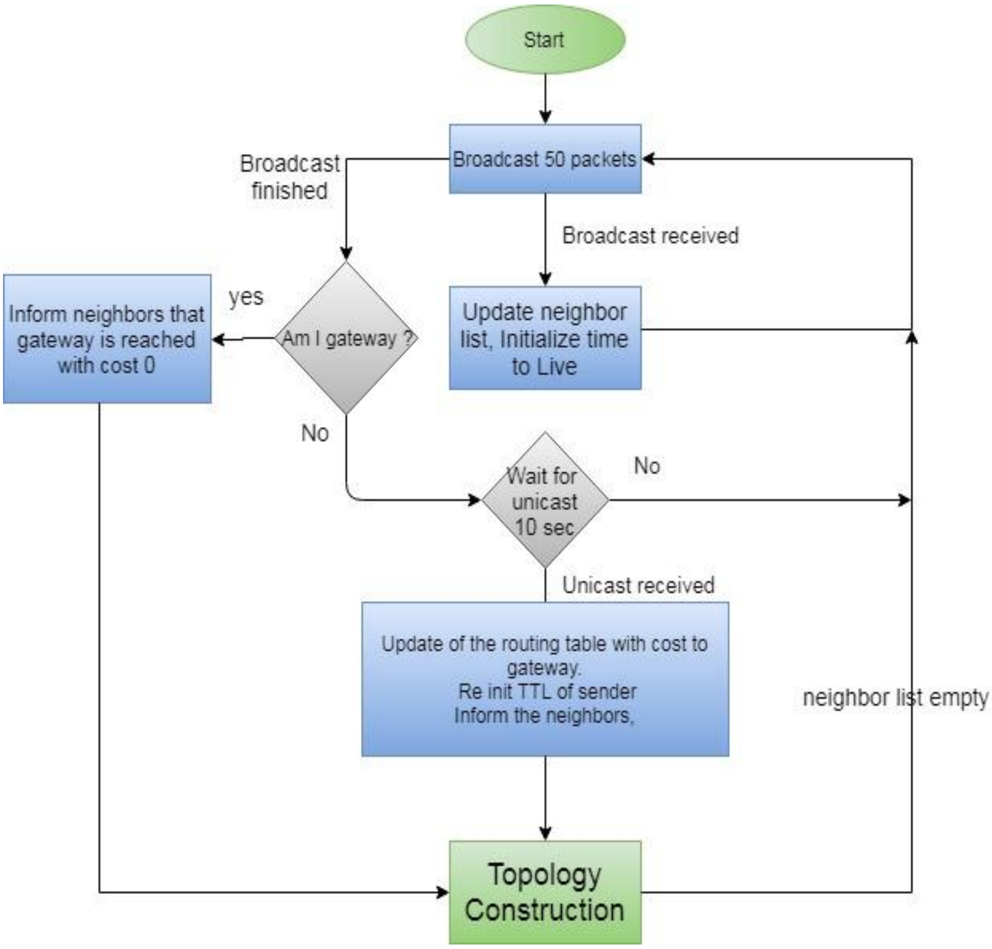


Figure 4 : Neighbourhood Discovery

As shown in the above diagram. When the motes boot up for the first time, they enter neighbor discovery mode. Every node broadcasts 10 packets by uniformly choosing a random time between one and five (Five is chosen due to five motes in action can in be changed depending on the size of the network). During this process if broadcast packet is received, the sender of the packet is considered as the neighbor so it will be added to the neighbor list with cost infinity and TTL of the neighbor is initialized. Once the broadcast of ten packets is finished, all the motes enter topology construction mode. Gateway node informs its neighbor that it can be reached with cost zero. Other nodes wait for this information. Once information about gateway is received, Nodes forward this information to nearby nodes after updating their cost. After the gateway is found by every node. Nodes estimates shortest route to gateway and will share only this route to neighbors. Through this method topology of the whole network is constructed, then network enters Data/Network stat forwarding and Network maintenance mode.

Routing and network maintenance:

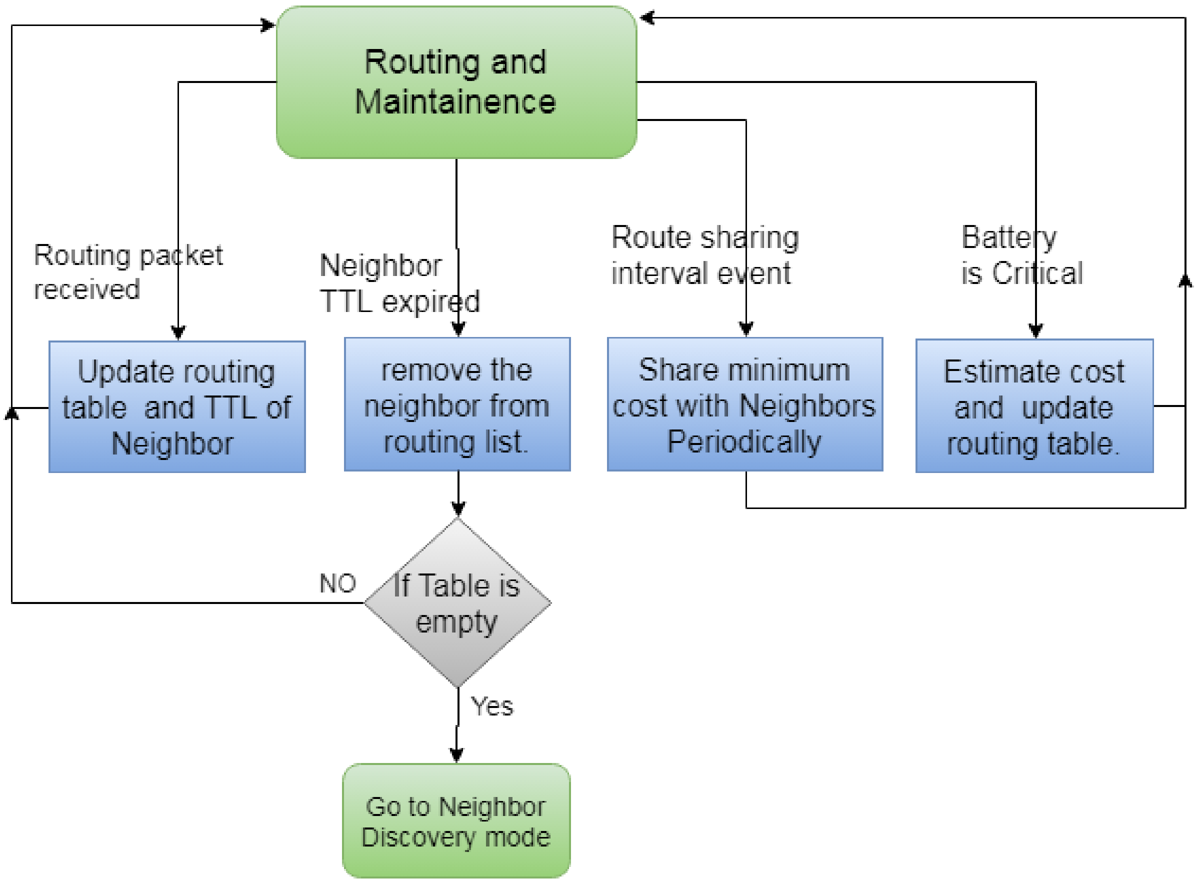


Figure 5: Routing and Maintenance

Once the Neighbor Discovery is completed, nodes maintain their neighbors and send routing packets to them periodically. Whenever Routing packet is received, node will update the cost of the neighbor in the routing table and reinitialize the TTL of the neighbor. If TTL of the neighbor is expired then, nodes will remove the entry pertaining to that neighbor from the routing table. If routing table (same as neighbor table) becomes empty node will enter neighbor discovery mode. If the battery level of the node is critical, cost to reach gateway through that node increases this information is propagated to the neighbors so that neighbors can update their tables.

Data/Networking stat forwarding:

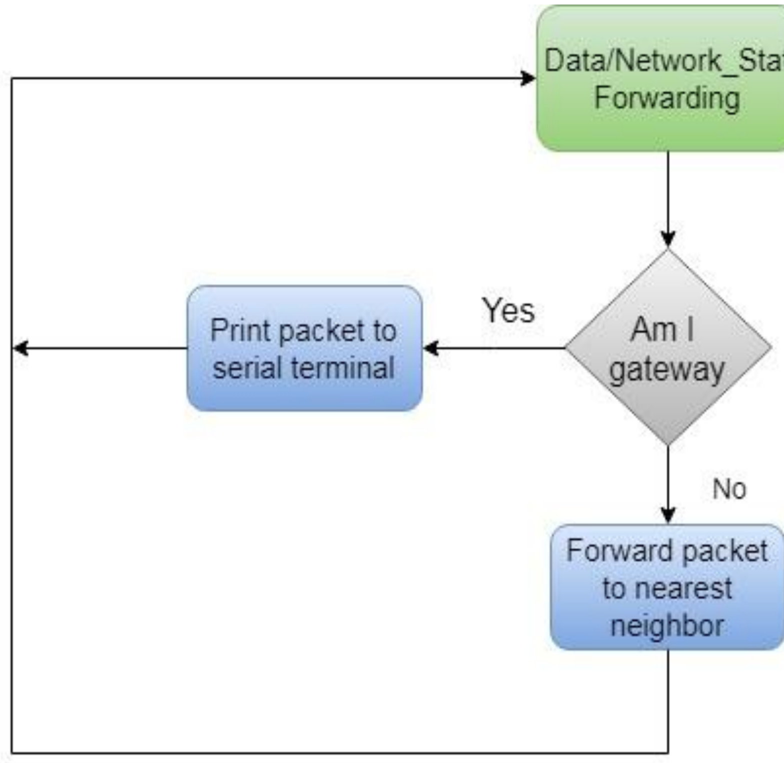


Figure 6 : Data Forwarding

Nodes send battery level and routing table packets periodically to Gateway for statistics and User Interface purpose.

Whenever network stat or a data packet is received, packet will be forwarded towards gateway without processing them. Gateway finally processes these packets and extracts required information to show on user interface.

If the routing table packets are not received from any node within a specified amount of time, then that node is considered as failed therefore marked with red color to indicate Node Failure.

Cost estimation based on the battery level:

In the present implementation, New cost is estimated as soon as the battery level gets changed and this new cost is informed to all neighbors. Currently the press of the button simulates battery level change. When a button is pressed battery level is reduced by 50% and cost is increased by 3.

4. Packet Types and Structure:

Neighbor Discovery Packet:

Broadcast	COST_TO_GATEWAY

Routing/TTL Packet

UNICAST	NEAREST_NEIGHBOR	COST_TO_GATEWAY

Asset Tracking packet:

ASSET	RSSI	ASSET_ADDR	NODE_ADDR1	NODE_ADDR2

Table Statistics Packet:

TABLE	NODE_ADDR	NEIGHBOR1	COST1	NEIGHBOR2	COST2

Battery Level Packet:

BATTERY	NODE_ADDR	BATTERY_LEVEL

5. Salient features of the implemented routing protocol

Failure recovery:

If any of the node in the route fails network doesn't break. Nodes will select the different routing protocol based on the routes available.

Self-Organization:

When a new node gets added to the network, it can automatically join to the network without manual intervention(plug-and-play).

Dynamic topology formation:

As discussed earlier, routing is performed based on the objective of minimizing energy. Since the energy level of the nodes change dynamically, Topology of the network needs to adapted. Implemented protocol exactly supports this feature, in other words if the energy of the tracking nodes gets low, Topology is adapted in such a way that less number of packets are forwarded through that node.

6. Graphical User Interface:

The Graphical User interface was developed using QT Creator. During the network operation, Gateway is connected to a computer through USB serial cable. The data collected by the gateway is printed out on the serial COM/ tty port in the relevant format required for the interpretation at the GUI.

QT components used:

Qtextserialport – An open source library which handles the low-level procedures for serial communication.

QPainter – To draw the nodes and the path of the packet.

Qtext for printing the received messages.

Combo Box and buttons for serial port communication

Below is an image of the GUI during the network operation.

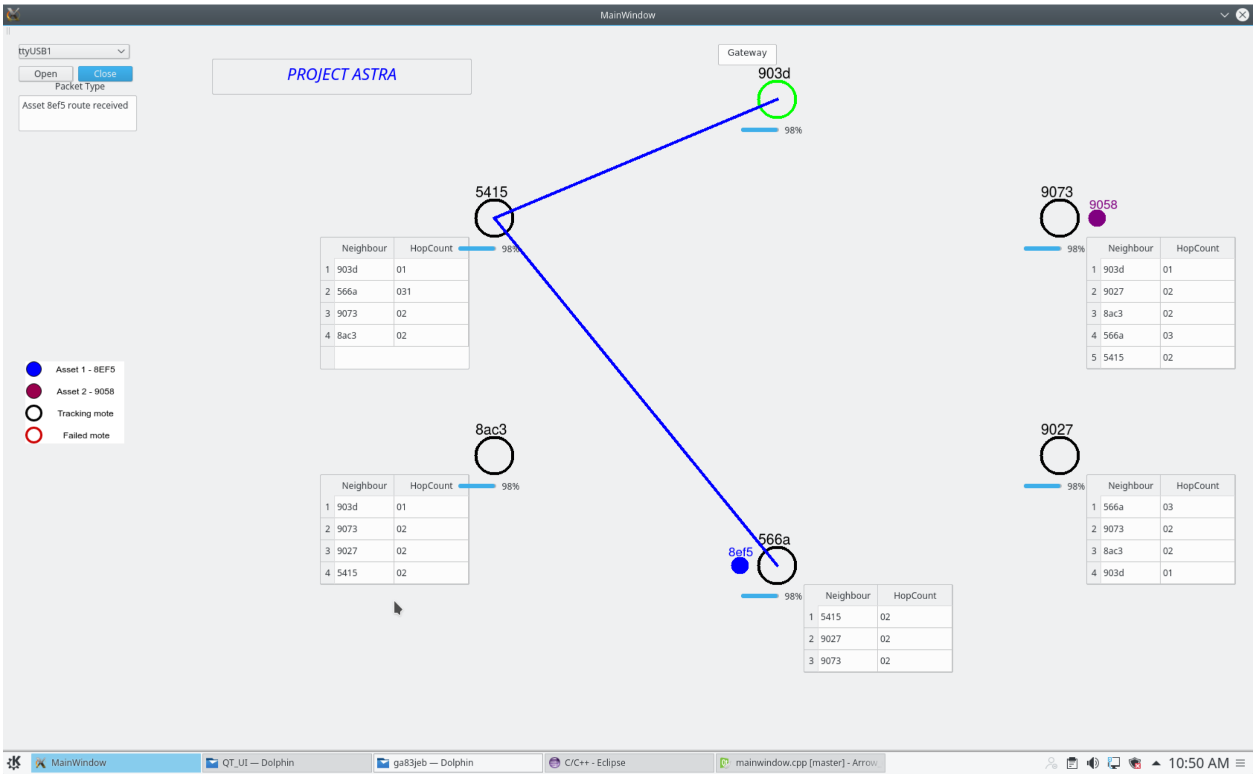


Figure 7 : GUI screenshot displaying the components in the project

Components in the GUI:

As can be seen in the figure, we have multiple nodes and the color highlighted identifies the functionality of each of the circles in the figure.

The Gateway is named and is represented by Green outlined circle. Rest Tracking nodes are in Black outlined circles. All the nodes have a fixed RIME address displayed on top of them.

The Routing tables and the Battery level corresponding to each of the nodes except the asset nodes can be seen besides each node.

2 assets that are being monitored are in BLUE and MAGENTA color. The 1st asset ASSET1 – 8EF5 is in blue solid circle and the path taken by this asset is in blue color.

The second asset ASSET2 – 9058 is in solid circle in Magenta color and the path from this asset's parent node to the Gateway is highlighted in Magenta color.

Apart from these, there are options to select the COM/ tty port for serial communication and a Text Editor which displays the type of the packet received by the Gateway.

Working Demo Explanation:

In the figure, we see that all the nodes have healthy battery life and are sending their Alive messages to Gateway, which also encompasses their table information. This Table packet is being displayed at the corresponding nodes. Asset1 – 8EF5 is located at the node 566A. Since 566A's shortest path to Gateway is through 5415, we see the relevant hops being taken by the Asset Tracking packet.

Likewise, the asset tracking packet from the node 9073 which receives the asset2 – 9058's broadcast reaches the Gateway in 1 hop and this path is also displayed alternatively in the GUI.

Avoiding Looping problem:

Also, note that in the routing table of 5415, the hop count corresponding to 566A is 31 which signifies that 566A goes through 5415 and hence 5415 should not again have a route though 566A which would result in the formation of loops.

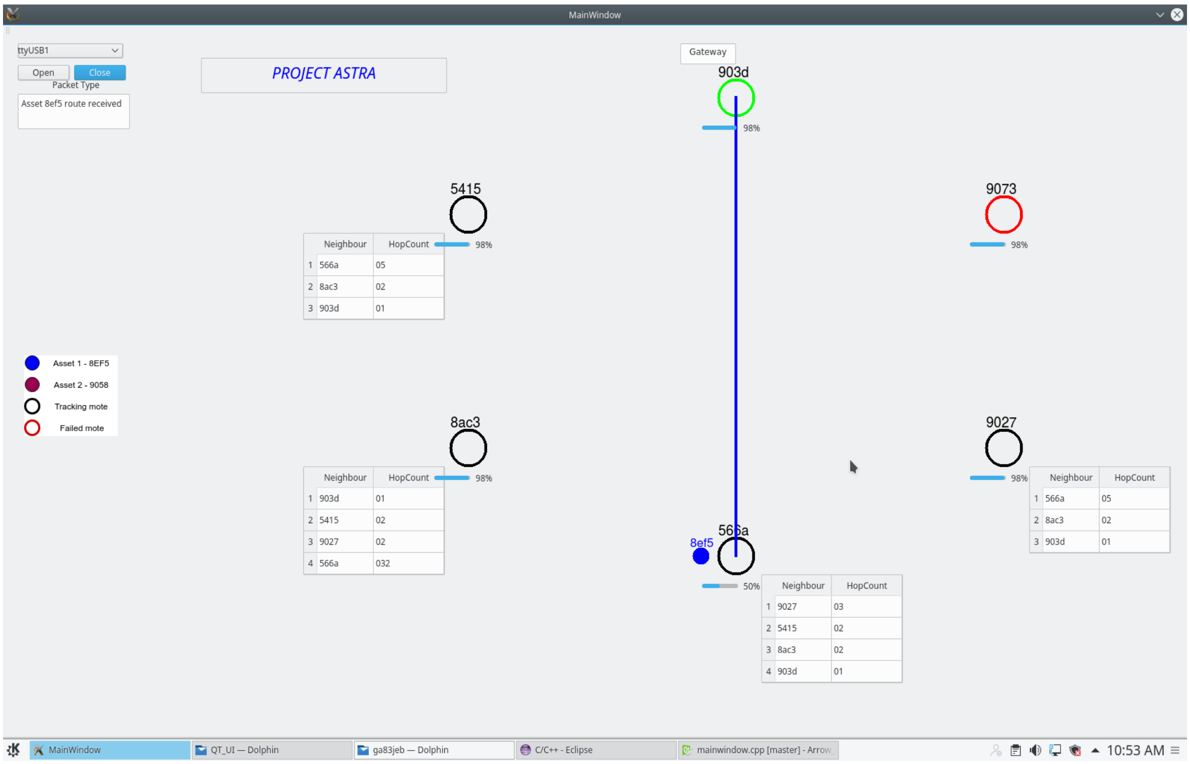


Figure 8 : GUI screenshot showing the Battery drain at 566A and node 9073 failure

In the above figure, we have the following scenario,

- Asset 1 is located near the node 566A. Now 566A has a direct hop to the Gateway, 903D and hence the packet is directly sent to the Gateway without any hops.
- Also, since we emulate button press for Battery drain, the user button was pressed which simulated the battery level going below 1.6V. This drain in the battery is also visible at the Battery level of the node 566A.

Dynamic Routing

As per our Routing strategy, wherein whenever there is a drain in the battery, we increase the cost of reaching through the node by 3 hop counts. This change in the hop counts is visible at the Routing tables of the other nodes. The tables at the node 5415 and node 9027 have an updated cost of 5 through the node 566A and hence they avoid the route through 566A.

Node failure and self-organization on recovery:

At this point, the node 9073 was powered down deliberately to simulate a node failure. When there is no TTL messages from 9073, its neighbors will remove node 9073 from their Routing table.

The Routing tables of all the nodes are updated and can be visualized at the GUI.

On recovery of the node, since the node wakes up in broadcast recovery mode, its neighbors when receive messages of this type, will share their routing table with the recovered node and the node updates its routing table with all the entries. It finds the shortest path to Gateway and starts working normally like any other node in the network.

Change in Asset location:

Whenever the asset is moved from one node to the other node. The real time updating of the asset location was demonstrated.

7. Lab setup for the Final presentation:

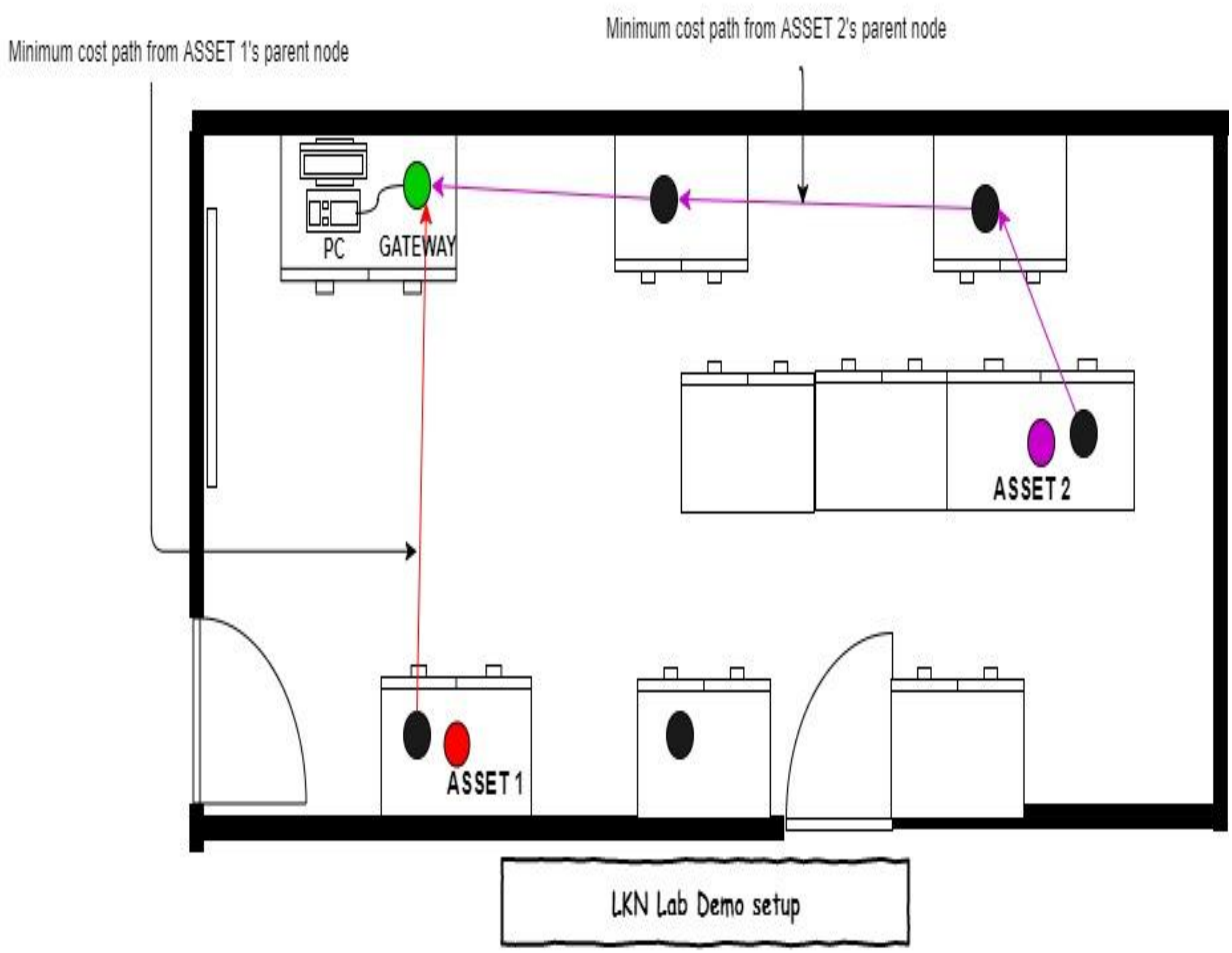


Figure 9 : Lab setup for the final presentation

The figure shows the lab setup during the final presentation. The red filled circle is the Asset1 and the Magenta circle is Asset2 and all Tracking nodes are in black circles. Gateway is in Green. We can see the path of asset tracking information from tracking nodes towards gateway.

8. Conclusion and Room for improvement:

As a part of the Wireless Sensor Networks laboratory, a robust, scalable minimum hop, energy aware Routing algorithm using Ad-Hoc Distance Vector Routing was developed and an application of tracking the assets in hospitals was demonstrated successfully.

As discussed, the algorithm increases the hop count by 3 counts depending on the available battery life. A potential research needs to be conducted to find a correlated estimation of the cost in relation to the available battery life as done in the paper[5]. The other improvement could be implementing an efficient RSSI filtering technique for filtering out the noise present in the RSSI values. Such improvements could take the project towards a more reliable and stable application for localization applications.

8. References:

[1] Dunkels et al, Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems, SenSys'06 Link to paper.

[2] Survey on Asset Tracking: http://www.versustech.com/rtls-solutions/asset-tracking-healthcare-hospital/

[3] Asset Management - Manage and protect valuable mobile medical equipment with the ultimate location platform by Centrak.

[4] J.-E. Garcia et all - A Novel DSR-based Energy-efficient Routing Algorithm for Mobile Ad-hoc Networks.

[5] Stojmenovic - Dept. of Comput. Sci., Ottawa Univ., Ont., Canada. Power-aware localized routing in wireless networks.