CS4222 Project Report

Group 33

National University of Singapore
School of Computing

Lian Jiade	A0187884E
Chen Kaibin	A0187250E
Chan Jun Ren	A0182514J
Ren Hao	A0188950N

Overview

This project aims to design a system that is able to detect the duration(s) in which two sensors are "in-proximity" and not "in-proximity". "In-proximity" here refers to within a distance of 3m. In the scope of this project, the most suitable way to estimate the distance between the two sensor nodes is using the RSSI value. We first conducted multiple experiments under different settings to derive a suitable RSSI threshold. Subsequently, we used the Quorum protocol for neighbor discovery to detect whether two nodes are "in-proximity" or not "in-proximity".

Protocol Implemented

The neighbor discovery protocol we implemented is the Quorum protocol. We have chosen this protocol due to its deterministic approach and low power transmission during a packet exchange.

The Quorum-based protocol makes use of an *n x n* grid where each slot in the grid signifies a particular time slot. A node begins by randomly choosing a particular time slot in the grid. The grid is then iterated and the node will start its discovery phase when it is in the same row and column as that of the chosen time slot. All other nodes in the discovery phase will be able to detect each other.

As contact tracing devices have to be available throughout the day, the N_SIZE chosen for the grid has to be large enough such that the power consumption is minimized. However, contact tracing is also a highly time-sensitive operation as there is only a short time frame where two nodes are within proximity of each other (the situation where two people walk past each other). Thus, the N_SIZE also has to be small enough to achieve a sufficiently high frequency of data transfer.

Based on our experiment results, we selected the N_SIZE of 16 for the Quorum-based protocol as this was determined to be an optimal value when we are considering latency and power.

We also considered an alternative protocol which was the birthday protocol. In the birthday protocol, a node randomly wakes up, broadcasts its presence, listens for incoming packets, and it goes to sleep. Although we found that the birthday protocol has slightly lesser power consumption, the birthday protocol is non-deterministic bound, unlike the Quorum-based protocol.

Logic for Proximity Detection

Variables

```
ABSENCE_LIMIT: limit for ABSENT to be printed
MIN_CONTACT: minimum duration for DETECT to be printed
RSSI_THRESHOLD: threshold for determining whether node is in proximity
device_node head: head of linked list
```

Functions

```
void push_rssi(device_node node, int rssi)
   -  adds the RSSI value into the device_node

int get_avg_rssi(device_node node)
   -  returns the average of the RSSI values stored in the device_node

void add_node(int id, unsigned longtime stamp, signed short rssi)
   -  adds the node with the relevant fields to the head

void remove_node(devie_node prev, device_node to_remove)
   -  removes node from linked list

void process_node(int id, unsigned long curr_timestamp, signed short rssi)
   -  entry point for node

void check_for_absence(unsigned long curr_timestamp)
   -  removes nodes that have not sent any packets in the last 30 seconds
```

Device_node fields

```
int id: id of node
unsigned long timestamp: timestamp of the first packet in/out of proximity
unsigned long last_pkt_recv_timestamp: timestamp of last packet received
int rssi_1, rssi_2 rssi_3: 3 most recent RSSI values
bool in_proximity: flag for whether sensor tag is within 3m
bool is_printed: flag for whether DETECT has been printed
```

Algorithm

1. When a packet from a neighboring node is received, it is first passed into the *process_node()* function
2. If the node has not been processed before, it is added to the linked list (*head*) with the relevant attributes set
3. If the node has been previously processed (present in the linked list)
    a. The value of the *last_pkt_received* is updated
    b. The RSSI value of the packet is pushed into the corresponding *device_node*
    c. The average of the last 3 RSSI values is checked against the *RSSI_THRESHOLD*. The *RSSI_THRESHOLD* is used as an indicator for whether the sensor tag is within 3 meters
        i. If the average RSSI does not exceed the *RSSI_THRESHOLD*
            1. If the node was previously already in proximity and at least *MIN_CONTACT* time has elapsed since the value in the *timestamp* field, *DETECT* is printed.
                a. The *is_printed* flag of the device_node is then set to true to prevent additional print statements
            2. If it is the first packet that causes the average RSSI to dip below the threshold, the *in_proximity* flag is set to true and the timestamp of this packet is recorded in the *timestamp* field
        ii. If the average RSSI exceeds the *RSSI_THRESHOLD*
            1. If the node was previously out of proximity and at least *ABSENT_LIMIT* time has elapsed since the value in the *timestamp* field, *ABSENT* is printed
            2. If it is the first packet that causes the average RSSI to dip above the threshold, the *in_proximity* flag is set to false and the timestamp of this packet is recorded in the *timestamp* field
4. During the active time slots of the quorum-based protocol, the *check_for_absence(long curr_timestamp)* method is called
    a. Any node that has not sent any packets for *ABSENT_LIMIT* seconds is removed

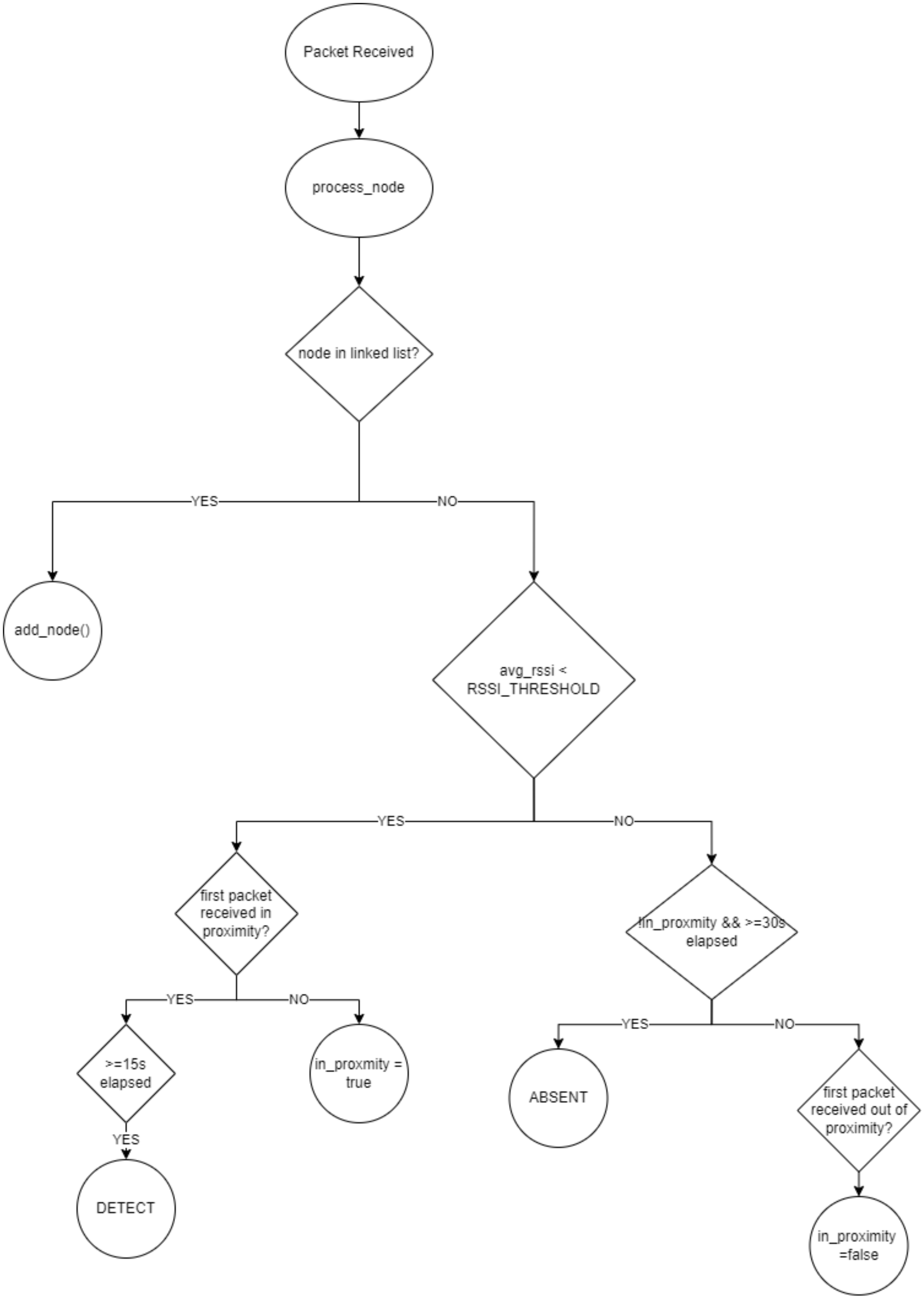

Fig. Control Flow Graph of logic for graph detection

System Evaluation

RSSI Threshold

The RSSI value is mainly used to estimate the distance between two nodes. However, there are several factors that can affect the RSSI value collected between two sensor nodes, such as the orientation, the number of obstacles between the nodes, and the radio environment. Therefore, to figure out a better, more accurate RSSI threshold value at a distance of 3m between two nodes, we repeated the experiments multiple times at different locations under different conditions.

Experiment Location	Experiment Condition	Highest RSSI Observed at a distance of 3m	Lowest RSSI Observed at a distance of 3m	Average RSSI Observed
SOC Classroom	Sensortag Orientation (Face away from each other)	65	61	63
SOC Classroom	Sensortag Orientation (Face towards each other)	63	59	61
SOC Classroom	Sensortag in pocket	68	63	65.5
CS4222 Lab	Sensortag Orientation (Face away from each other)	66	61	63.5
CS4222 Lab	Sensortag Orientation (Face towards each other)	61	58	59.5
CS4222 Lab	Sensortag in pocket	72	65	68.5

The average value for each experiment was then obtained by recording the highest and lowest RSSI observed over 5 readings and then taking the average of the two. Eventually, we set the RSSI Threshold to estimate the distance of 3m to be **64**. This value was the total average value obtained after running the RSSI calculation function for two sensors under the different locations and conditions as shown in the table above.

Power Consumption

We test two different protocols: The Birthday protocol and the Quorum-based protocol. For the Quorum-based protocol, we need to decide the N Size, Time Slot, and thus the Duty Cycle. We think that the Maximum Discovery Latency should be around 7.5 seconds. The reason for that value can be explained by the following diagram:

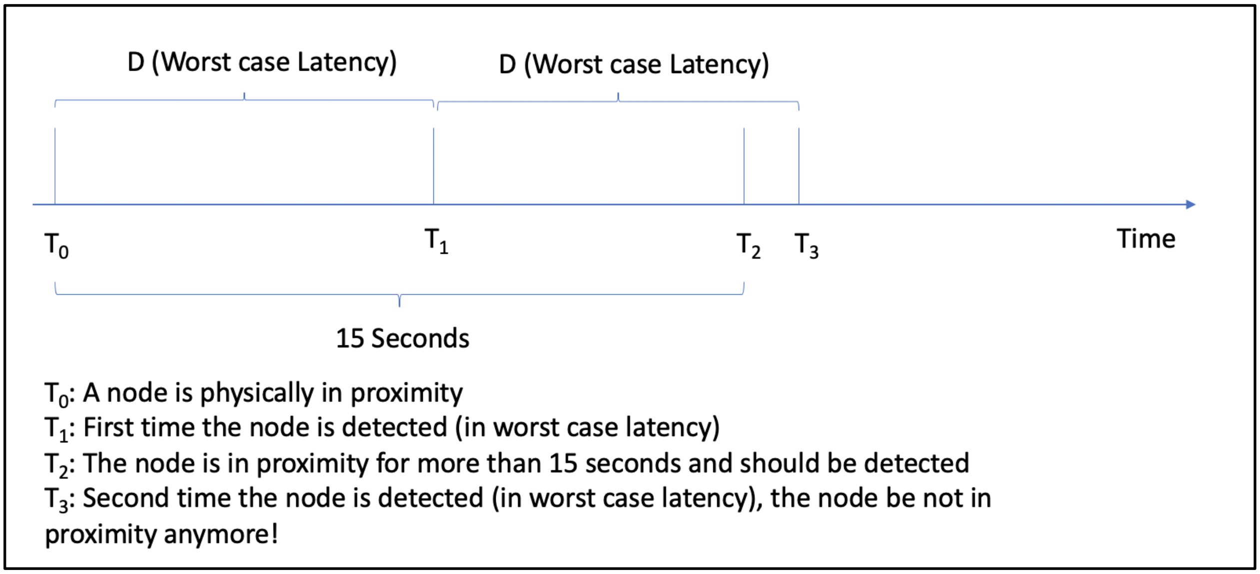

Fig. A Situation where we might miss a node who is actually in close contact

The diagram shows a possible situation of detection. It is worth noting that we need two consecutive detections of a node in proximity, and compare the time interval between them to detect the node as in "close contact". Assume we take the longest discovery time (D) to discover the node, if D > 7.5 seconds, two consecutive detections would take more than 15 seconds. The second time we discover the node, the node might not be in proximity anymore and we will not print "DETECT". As a result, we might miss this node. Hence, we decide to set the discovery delay upper bound of our Quorum-based Protocol to be 7.5 seconds.

According to the formula of the Quorum-based Protocol, $Delay\ Upper\ Bound\ D\ = \Theta T\ /\ (f^2)$ where Θ is a constant depending on the device (Mobility lecture notes, p42), T is the time slot length and f is the duty cycle. To figure out the T and f we need to use in order to achieve an upper bound of 7.5 seconds, we first need to calculate the Θ. In Assignment 4, we found that when we set the time slot length to 0.1 seconds and N size to be 13 (duty cycle thus 14.79% using the formula $f\ = (2N\ - 1)\ /\ N^{\ 2}$), generally we can guarantee the discovery of a node within 10 seconds. Hence we plug-in the values and calculate out the Θ for our device should be 10 * (0.1479 ^ 2) / 0.1 = 2.19.

With this Θ value, we can thus find possible combinations of T and f that allow us to achieve an upper delay bound of 7.5 seconds by solving the polynomial equation of T * (N ^ 4) / (2N - 1) ^ 2 =7.5. We thus form 5 different settings of T and f, and tested the Power Consumption of these settings on Cooja. We run the simulation for 1 minute. Following is a screenshot of Cooja Powertrace log of our experiment.

Time	Mote	Message
59:10.696	ID:1	454404 P 1.0 709 7907572 108419521 495888 13989386 0 0 7417 156420 344 10076 0 0 (radio 12.-1174% / 6.35% tx 0.05% / 0.20% listen 12.-1179% / 6.15%)
59:15.550	ID:2	455044 P 2.0 710 7869347 108621586 564134 14025721 0 0 16921 146916 1274 35292 0 0 (radio 12.-1165% / 22.31% tx 0.11% / 0.77% listen 12.-1176% / 21.54%)
59:15.696	ID:1	455044 P 1.0 710 7924673 108566260 497156 14024673 0 0 17098 146739 1268 35287 0 0 (radio 12.-1171% / 22.31% tx 0.05% / 0.77% listen 12.-1176% / 21.53%)
59:20.549	ID:2	455684 P 2.0 711 7876807 108777966 564481 14035796 0 0 7457 156380 347 10075 0 0 (radio 12.-1164% / 6.36% tx 0.11% / 0.21% listen 12.-1175% / 6.14%)
59:20.696	ID:1	455684 P 1.0 711 7932130 108722643 497503 14034748 0 0 7454 156383 347 10075 0 0 (radio 12.-1170% / 6.36% tx 0.05% / 0.21% listen 12.-1176% / 6.14%)
59:25.550	ID:2	456324 P 2.0 712 7893523 108925090 565700 14069462 0 0 16713 147124 1219 33666 0 0 (radio 12.-1198% / 21.29% tx 0.11% / 0.74% listen 12.-1173% / 20.54%)
59:25.696	ID:1	456324 P 1.0 712 7949057 108869556 498715 14068410 0 0 16924 146913 1212 33662 0 0 (radio 12.-1167% / 21.28% tx 0.05% / 0.73% listen 12.-1173% / 20.54%)
59:30.549	ID:2	456964 P 2.0 713 7901055 109081398 566076 14080049 0 0 7529 156308 376 10587 0 0 (radio 12.-1197% / 6.69% tx 0.11% / 0.22% listen 12.-1172% / 6.46%)
59:30.695	ID:1	456964 P 1.0 713 7956586 109025867 499091 14078997 0 0 7526 156311 376 10587 0 0 (radio 12.-1166% / 6.69% tx 0.05% / 0.22% listen 12.-1172% / 6.46%)
59:35.550	ID:2	457604 P 2.0 714 7908825 109237468 566451 14091471 0 0 7767 156070 375 11418 0 0 (radio 12.-1196% / 7.19% tx 0.11% / 0.22% listen 12.-1171% / 6.96%)
59:35.696	ID:1	457604 P 1.0 714 7964354 109181939 499465 14090417 0 0 7765 156072 374 11416 0 0 (radio 12.-1165% / 7.19% tx 0.05% / 0.22% listen 12.-1171% / 6.96%)
59:40.550	ID:2	458244 P 2.0 715 7925505 109384628 567662 14125135 0 0 16677 147160 1211 33664 0 0 (radio 12.-1193% / 21.28% tx 0.11% / 0.73% listen 12.-1168% / 20.54%)
59:40.696	ID:1	458244 P 1.0 715 7981216 109328917 500674 14124071 0 0 16859 146978 1209 33654 0 0 (radio 12.-1199% / 21.27% tx 0.06% / 0.73% listen 12.-1168% / 20.54%)
59:45.550	ID:2	458884 P 2.0 716 7932981 109540992 568010 14135210 0 0 7473 156364 348 10075 0 0 (radio 12.-1192% / 6.36% tx 0.11% / 0.21% listen 12.-1167% / 6.14%)
59:45.696	ID:1	458884 P 1.0 716 7988687 109485286 501022 14134146 0 0 7468 156369 348 10075 0 0 (radio 12.-1198% / 6.36% tx 0.06% / 0.21% listen 12.-1167% / 6.14%)
59:50.550	ID:2	459524 P 2.0 717 7949897 109687916 569278 14170497 0 0 16913 146924 1268 35287 0 0 (radio 12.-1189% / 22.31% tx 0.11% / 0.77% listen 12.-1164% / 21.53%)
59:50.696	ID:1	459524 P 1.0 717 8005817 109631996 502283 14169438 0 0 17127 146710 1261 35292 0 0 (radio 12.-1195% / 22.31% tx 0.06% / 0.76% listen 12.-1164% / 21.54%)
59:55.550	ID:2	460164 P 2.0 718 7957340 109844313 569625 14180569 0 0 7440 156397 347 10072 0 0 (radio 12.-1188% / 6.35% tx 0.11% / 0.21% listen 12.-1200% / 6.14%)
59:55.695	ID:1	460164 P 1.0 718 8013256 109788397 502630 14179510 0 0 7436 156401 347 10072 0 0 (radio 12.-1194% / 6.35% tx 0.06% / 0.21% listen 12.-1200% / 6.14%)
1:00:00.550	ID:2	460804 P 2.0 719 7964770 110000723 569971 14190644 0 0 7427 156410 346 10075 0 0 (radio 12.-1187% / 6.36% tx 0.11% / 0.21% listen 12.-1199% / 6.14%)
1:00:00.696	ID:1	460804 P 1.0 719 8020682 109944811 502976 14189585 0 0 7423 156414 346 10075 0 0 (radio 12.-1193% / 6.36% tx 0.06% / 0.21% listen 12.-1199% / 6.14%)
1:00:05.550	ID:2	461444 P 2.0 720 7981685 110147648 571189 14224303 0 0 16912 146925 1218 33659 0 0 (radio 12.-1184% / 21.28% tx 0.11% / 0.74% listen 12.-1196% / 20.54%)
1:00:05.696	ID:1	461444 P 1.0 720 8037781 110091552 504186 14223240 0 0 17096 146741 1210 33655 0 0 (radio 12.-1190% / 21.28% tx 0.06% / 0.73% listen 12.-1196% / 20.54%)
1:00:10.550	ID:2	462084 P 2.0 721 7989365 110303820 571592 14236010 0 0 7677 156172 403 11707 0 0 (radio 12.-1183% / 7.39% tx 0.12% / 0.24% listen 12.-1195% / 7.14%)
1:00:10.696	ID:1	462084 P 1.0 721 8045457 110247728 504589 14234947 0 0 7673 156176 403 11707 0 0 (radio 12.-1189% / 7.39% tx 0.06% / 0.24% listen 12.-1195% / 7.14%)
1:00:15.550	ID:2	462724 P 2.0 722 8005838 110451175 572813 14269669 0 0 16470 147355 1221 33659 0 0 (radio 12.-1180% / 21.29% tx 0.12% / 0.74% listen 12.-1192% / 20.54%)
1:00:15.696	ID:1	462724 P 1.0 722 8062380 110394633 505802 14268603 0 0 16920 146905 1213 33656 0 0 (radio 12.-1186% / 21.28% tx 0.06% / 0.74% listen 12.-1192% / 20.54%)
1:00:20.550	ID:2	463364 P 2.0 723 8013521 110607332 573160 14279741 0 0 7680 156157 347 10072 0 0 (radio 12.-1179% / 6.35% tx 0.12% / 0.21% listen 12.-1192% / 6.14%)
1:00:20.696	ID:1	463364 P 1.0 723 8069819 110551034 506149 14278675 0 0 7436 156401 347 10072 0 0 (radio 12.-1185% / 6.35% tx 0.06% / 0.21% listen 12.-1192% / 6.14%)
1:00:25.550	ID:2	464004 P 2.0 724 8020942 110763751 573507 14289817 0 0 7418 156419 347 10076 0 0 (radio 12.-1179% / 6.36% tx 0.12% / 0.21% listen 12.-1191% / 6.15%)
1:00:25.696	ID:1	464004 P 1.0 724 8077237 110707456 506496 14288751 0 0 7415 156422 347 10076 0 0 (radio 12.-1184% / 6.36% tx 0.06% / 0.21% listen 12.-1191% / 6.15%)
1:00:30.551	ID:2	464644 P 2.0 725 8037829 110910704 574776 14325108 0 0 16884 146953 1269 35291 0 0 (radio 12.-1176% / 22.31% tx 0.12% / 0.77% listen 12.-1188% / 21.54%)
1:00:30.696	ID:1	464644 P 1.0 725 8094308 110854225 507756 14324042 0 0 17068 146769 1260 35291 0 0 (radio 12.-1181% / 22.30% tx 0.06% / 0.76% listen 12.-1188% / 21.54%)
1:00:35.549	ID:2	465284 P 2.0 726 8045536 111066837 575122 14335186 0 0 7704 156133 346 10078 0 0 (radio 12.-1175% / 6.36% tx 0.12% / 0.21% listen 12.-1187% / 6.15%)
1:00:35.695	ID:1	465284 P 1.0 726 8102011 111010362 508102 14334120 0 0 7700 156137 346 10078 0 0 (radio 12.-1180% / 6.36% tx 0.06% / 0.21% listen 12.-1187% / 6.15%)
1:00:40.562	ID:2	465925 P 2.0 727 8056551 111220014 575900 14356987 0 0 11012 153177 778 21801 0 0 (radio 12.-1173% / 13.75% tx 0.12% / 0.47% listen 12.-1185% / 13.27%)
1:00:40.696	ID:1	465924 P 1.0 727 8114290 111161923 508878 14355571 0 0 12276 151561 776 21451 0 0 (radio 12.-1179% / 13.56% tx 0.06% / 0.47% listen 12.-1185% / 13.09%)

Fig. Screenshot of the Cooja Simulation's Powertrace Log

Our test results are shown in the following table. We record the accumulated energy consumption and summed them up in the total column.

N Size	Timeslot length (Seconds)	Accumulated CPU Energy Consumtion (mJ)	Accumulated Low Power Mode Energy Consumption (mJ)	Accumulated transmission energy consumption (mJ)	Accumulated listen energy (mJ)	Accumulated idle transmission energy consumption (mJ)	Accumulated idle listen energy consumption (mJ)	Total (mJ)
12	0.1	4428468	113537025	1592	18990928	0	0	136958013
16	0.05	7923140	110042353	501223	14206292	0	0	132673008
9	0.15	4613196	113352297	20707	24854050	0	0	142840250
7	0.2	4442855	113522638	3131	31377216	0	0	149345840

6	0.25	4569720	113395773	15767	36103540	0	0	154084800

Fig. Table of accumulated power consumption for quorum based protocol with different settings but same upper bound

From the above results, we calculated that the total energy consumption is least when N Size = 16 and Time slot is 0.05. Hence, for power consumption reduction, we selected these values for our Quorum-Based Protocol.

Furthermore, we also compared the energy performance of two protocols, the Birthday protocol and the Quorum-based Protocol using Cooja. To ensure fairness, two protocols run at the same duty cycle. We select the setting that has the least cost for Quorum-based protocol (N Size = 16, time slot = 0.05s, duty cycle = 12.10%). Accordingly, we set the wake time of the birthday protocol to be 0.05s. Then according to the duty cycle formula (Duty cycle = Wake time / (Wake time + Sleep Slot * Sleep Cycle)), we calculate the sleep slot to be 0.04 seconds. The results of our experiments are shown in the below table.

Method	Setting	Total Energy Consumption
Birthday	Wake Time = 0.05s, Sleep Slot = 0.04s, Duty cycle = 12.10%	132547589
Quorum Based Protocol	N Size = 16, Time slot = 0.05s, Duty cycle = 12.10%	132673008

We observed that Quorum-based Protocol had only a slightly higher energy consumption (125 J) than the Birthday protocol. The difference between these protocols in terms of energy consumption is not so significant compared to the difference between different T and f. Hence, we opted to proceed with the Quorum-based protocol as this slight increase in energy consumption is trivial and compensated with the deterministic discovery and latency discovery time upper bounding provided by the Quorum-based protocol.

Accuracy and Robustness of our System

Obstacle	Time interval for Detection (s)	Time interval for Detection (s)	Time interval for Absence (s)	Time interval for Absence (s)	Average Detect Time (s)	Average Absence Time (s)
No obstacle	17.4	18.3	33.3	34.1	17.9	33.7
Inside pocket	24.4	25.7	44.3	40.1	25.1	42.2
Multiple Nodes	20.2	18.5	35.3	35.4	19.4	35.4

To evaluate our system, we measured the average time interval for detection and absence when the sensor tag is moved within proximity and outside of proximity. We also tested the system in different environments where the Sensortag was placed in a pocket and when there were 3 Sensortags simultaneously running.

We observed that our system is generally able to sense when the sensor is in proximity and when it has left albeit with some additional latency in each of the conditions. The reason for this latency is largely caused by instances of RSSI value inconsistencies during experimentation where although the Sensortag has moved outside of 3m or within 3m, the RSSI value received still remains below or above the threshold value we set. This caused our program logic which depends on the RSSI value to be unable to output the Detect and Absent statement immediately at the 15s and 30s mark.

Summary

Challenges Faced

Based on our experimentation during the project as well as the previous assignments, we were able to conclude that while RSSI is not very effective for measuring how far an object is, it can be used to detect whether a node is in proximity. Nevertheless, this effectiveness is also subject to external factors such as the physical and radio environment, or the orientation of the sensor tag. This gave rise to some limitations in our original implementation of the algorithm and difficulties in choosing the RSSI threshold for detecting whether a node is in proximity.

Our original implementation of the algorithm was based on detecting consecutive packets that were above the RSSI threshold and keeping track of the timestamp of the first packet that was transmitted. We found that when the environment was varied, the fluctuations in the RSSI values caused the state of the node to change and this caused the timer to be reset and negatively affected the robustness of the system. To deal with such fluctuations, we included a simple stack in the node and used the average of the last 3 RSSI values received to determine the state of the sensor tag, and it was able to improve the robustness of our system to some degree.

There was also a lot of deliberation on the parameters to be chosen so as to achieve the balance of lower power and good robustness. Generally, better robustness would come at the cost of higher power consumption. To achieve this balance, we decided to choose the settings with a delay upper bound that we thought was reasonable (through experimentation and calculations based on the previous assignment) and slightly tweak the values of MIN_CONTACT and ABSENT_LIMIT to account for the possible latencies.

Conclusion

In conclusion, we implemented the Quorum-based Protocol for our tracing application. The tracing token will discover nearby nodes, obtain the RSSI value using a dynamic average window and judge whether the node is within the 3-meter range using the RSSI threshold we set based on our experiments in different environments. Once the node is within proximity for a period longer than the limit set, we say that we "DETECT" the node is in close contact; we use a similar approach to detect when the node is "ABSENT".

To lower the power consumption, we chose a combination of duty cycle and time slot length that cost the least energy consumption under the maximum discovery delay bound we set. Furthermore, upon comparing the birthday protocol and the Quorum-based Protocol, we decide to use the Quorum-based Protocol finally since it provides a deterministic discovery bound and its energy consumption is not significantly larger than the Birthday Protocol.