MONASH University
Information Technology

GROUP ASSIGNMENT COVER SHEET

Student ID Number	Surname	Given Names
29617111	*Lian*	**Meiya**
28688139	*Zhang*	**Shenzhou**

***** Please include the names of all other group members.

Unit name and code	FIT3143 Parallel Computing		
Title of assignment	Assignment 2 Alter Detection in a distributed wireless sensor network (WSN)		
Lecturer/tutor	ABM Russel		
Tutorial day and time	Monday 10-12 a.m.	**Campus**	
Is this an authorised group assignment?	☒ **Yes** ☐ **No**		
Has any part of this assignment been previously submitted as part of another unit/course?	☐ **Yes** ☒ **No**		
Due Date **31/10/2020**		**Date submitted** **31/10/2020**	

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

Extension granted until (date) .... Signature of lecturer/tutor ....

Please note that it is your responsibility to retain copies of your assessments.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity Policy and Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
  i. provide to another member of faculty and any external marker; and/or
  ii. submit it to a text matching software; and/or
  iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Signature .. Shenzhou Zhang............. Meiya Lian.................. *Date*............31/10/2020..............................

* delete (iii) if not applicable

Signature ___Shenzhou Zhang__ Date:__31/10/2020___________________

Signature ___ Meiya Lian_______Date:__ 31/10/2020__________________

Simulation of alert detection in a distributed wireless sensor network (WSN)

Faculty of Information Technology
Melbourne, Australia

Meiya Lian

Shenzhou Zhang

***Abstract* — Wireless sensor network (WSN) refers to a group of sensors that are used for monitoring and recording different physical conditions of the environment and organize the collected data at a central location (a base node). It used to measure a lot of different conditions such as temperature, sound, pollution lever, wind and so on. This paper aims to simulate the aforementioned setup. It used a grid layout to represent different sensors and simulate the process of exchange information between each sensor and their immediate adjacent sensor (according to the grid). In this simulation program, MPI is used to exchange the information between the sensors and base station. In addition, threads will be used for simulating the data generated by the infrared imaging satellite.**

***Keyword* — wireless sensor network, WSN, MPI, grid based architecture, cartesian topology**

I. Introduction

Currently, the importance of communication between different points (especially in the program that the communication between two or more software) become more important than ever. Because the co-operative nature can be utilized for increasing computation speed, convenience and modularity. Inter process communication (IPC) is one of the mechanisms that allows processes to communicate with each other.

There are many forms of IPC, one of the most common used forms is MPI. MPI is a message-passing library interface specification and achieves compactness, functionality and proficiency on a signal system [1]. MPI will allow the user to communicate across a network by sharing the message (either Point-to Point or broadcast). Because the WSN architectures are the standard for low power sensor networks, to accomplish low power usage, each node in the WSN tries to reduce the amount of processing and communication done in them. In order to implement these WSNs, highly efficient IPC architectures are needed. Although there are many different IPC architectures, but since a sensor node can only communicate with the four adjacent nodes, so a grid based socket IPC architecture would be the beast method for the WSN. [2]

The objective of this paper is to design and implement an alert detection system in a distributed wireless sensor network. The message exchange between nodes and the base station is simulated by using the MPI library. Nodes representing the WSN network will be mapped to a cartesian topology. One of the processes in the system is representing the base station. Posix thread is used in this process for simulating the work done by the infrared imaging satellite. The communication between the satellite and the base station is implemented through the use of global variables between threads.

The hypothesis proposed is that for a given grid size, the rate of true alters will not be larger than the rate of false alters while the number of iterations increases and also different grip with the same number of iterations will also affect the rate as well.

Here is an overview about the following section: Section II introduces the design and architecture of the detection system. Section III analyses the performance of the architecture and Section IV concludes the report.

II Sensor Network's design

A. Overview

The whole system consists of two parts -- the WSN network and the base station with the infrared imaging satellite. After the user input the grid size and the number of iterations for the system to run, the system will firstly broadcast the values to all the processes.

Then, *MPI_Comm_split* is used to split the processors into two groups. (show in Fig.1) The two groups will then execute their corresponding functions. Figure 2 shows the flowchart of the base code of the system.

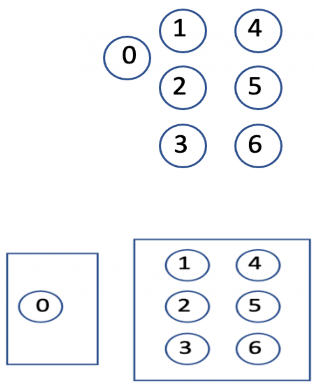


Fig. 1 before splitting(left) and after splitting(right) the system.

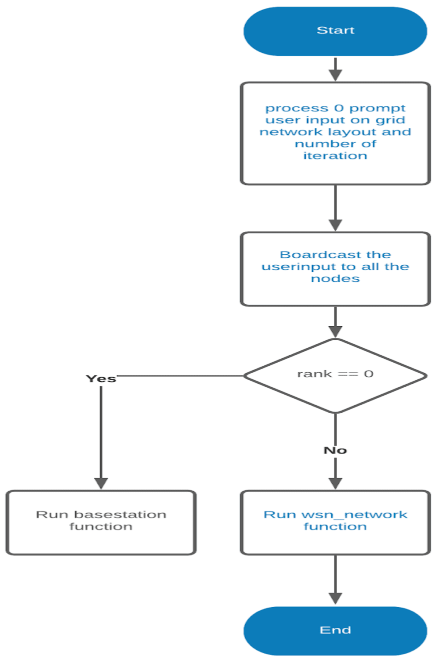


Fig. 2 Technical flowchart of the base code of the system

B. Sensor nodes

Cartesian topology is used for mapping the nodes into a grid network. Each node in the network is only allowed to communicate with its adjacent nodes and the base station only. The layout of the grid is dynamic, defined by the user during the program runtime. Figure 2 shows the two possible layouts for a network size of 9.

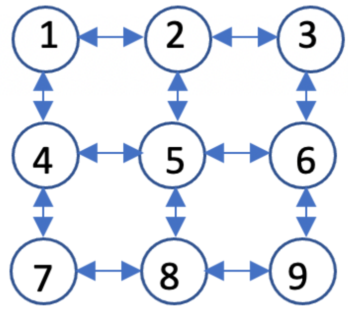


Fig. 3(a) 3*3 layout

Fig. 3(b) 9*1 layout

The coordinate for each node is defined below:

$$row\_index = (rank - 1) / width$$

$$column\_index = (rank - 1) \% width$$

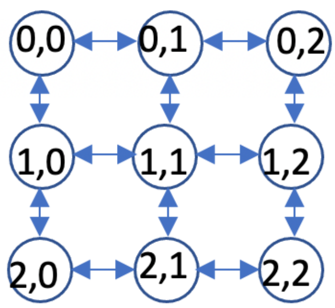


Fig. 4(a) coordinates of each node in example Fig. 2 (a)

Fig. 4(b) coordinates of each node in example Fig. 2 (b)

The working flow of each node is shown in Figure 6. In each iteration, the node will generate an integer value between 20 and 120 simulating the sensor reading. If the reading is greater than the predefined threshold(80), the sensor will send a message to each of its adjacent nodes to acquire their readings. The non-blocking *MPI_Isend* and *MPI_Irecv* are used in order to allow the node to process other node's requests while waiting for its neighbour nodes' values.

While the node is waiting for the reponses, it listens to the requests from its neighbours. The non-blocking *MPI_Irecv* and *MPI_Test* are used as it is unknown for each node at each iteration how many requests it will receive. In the implementation code, the source node is *MPI_ANY_SOURCE* as the sender of each request is also unknown. The pseudo code below depicts this process:

Algorithm 1: Listening other nodes' requests

```
MPI Irecv;
while within pre-defined time window do
    MPI Test;
    if receive request from a neighbour node then
        send self reading back to the requesting node;
        MPI Irecv;
    else
end
```

Fig. 5 Pseudo code for listening to other nodes' requests

After the above message-detecting loop, the node will directly check the responses it received. If more than one of the adjacent nodes have similar reading values, the node will then send a report to the base station. The report is packed using the *MPI_Pack*. The table below shows the content of the pack.

Item	Type	usage	Example
my_cart_rank	MPI_INT	Rank of the node in the cartesian layout	5
reading	MPI_INT	The node's reading value	85
count	MPI_INT	Number of adjacent nodes having similar reading	2
node 1	MPI_INT	Neighbour node 1's rank	2
node1's reading	MPI_INT	Neighbour node 1's reading	90
node 2	MPI_INT	Neighbour node 2's rank	6
node1's reading	MPI_INT	Neighbour node 2's reading	100
Generated time	MPI_LONG	Indicates when is the reading of the current node(in this example, node 5) generated	1604040458

Fig. 6 Example of messages sent by a node in one iteration in a 3*3 layout

Before the iteration starts, the node will firstly send the basic information to the base station, which is recorded by the base station and be used later for logging. The detail of the information is depicted in part (c).

This communication is done by MPI blocking send and receive, in order to make sure the information is properly received. This minimises the size of the packet sent each time by the node. Hence, reduces the cost of communication.

Each node keeps looping the above steps until it receives the termination message sent by the base station. The detection of the base station uses a similar approach to the method shown in Fig. 5.

C. Base station

Figure 8 describes the technical flowchart for the base station. Before the iteration starts, the station receives information from all of the nodes in the WSN network and forms a struct array to store the values. The index of the array indicates the rank of the corresponding node in the cartesian layout. For each node, the following data is stored:

1. Coordinates in the cartesian layout

2. MAC address of the node
3. IP address of the node

In each iteration, the base station keeps listening to receive alarm messages during the pre-defined iteration time window. Once an alarm is received, it passes the message to the function *logToFile*, which unpacks the message, extracts useful information and formats it to log. During the extraction process, the function also updates the global variable *totalMsgReceived* and *correctAlarm* for analyzing the overall performance of the sensor nodes.

After k iterations (which k is defined by the user), the base station will log a summary information including how many alarms received and the true alarm rate. Then, blocking *MPI_SEND* is used for sending the termination signal to all the nodes in the WSN network before the station cleans up and returns.

D. Infrared imaging satellite.

The high level implementation of the satellite is shown in figure 8. The satellite generates a list of values representing the reading in each coordinate every 1500 ms. In reality, these values will be sent to the base station. For simulation purposes, this communication is done by setting this value list to be a global variable that is accessible by both threads. Since the base station thread only requires to read the data, mutex lock is not used when the satellite thread

The index of the value list represents the node ranks in the WSN network. For each update, a random value is generated for each node, accompanied by a timestamp indicating the generated time of the value.

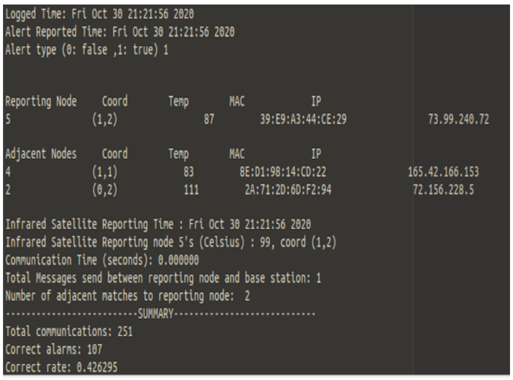
To make the satellite regenerate this data list every 1500ms, the thread will be put into sleep until 1500ms is passed after updating the whole list, for every iteration.

The thread will keep executing the above operations until the termination signal sent by the base station is arrived. This communication is also implemented by utilizing the global variables shared between the thread.

III Result & Discussion

The platform used for this test was the virtual machine. Memory for this VM is 2048 MB, including 10 processor. System is Ubuntu (64-bit).

20 runs were completed on Virtual Machine, starting with 40 iterations with each of them increased by 10 until 100, then will increase 100 each time in order to see the difference. And this will repeat for three different grids to see how the size of the grid will affect the result. So in total 60 runs made in order to test the system.



```
Logged Time: Fri Oct 30 21:21:56 2020
Alert Reported Time: Fri Oct 30 21:21:56 2020
Alert type (0: false ,1: true) 1


Reporting Node    Coord      Temp      MAC           IP
5             (1,2)            87       39:E9:A3:44:CE:29            73.99.240.72

Adjacent Nodes    Coord      Temp      MAC           IP
4             (1,1)        83       8E:D1:98:14:CD:22           165.42.166.153
2             (0,2)        111       2A:71:2D:6D:F2:94           72.156.228.5

Infrared Satellite Reporting Time : Fri Oct 30 21:21:56 2020
Infrared Satellite Reporting node 5's (Celsius) : 99, coord (1,2)
Communication Time (seconds): 0.000000
Total Messages send between reporting node and base station: 1
Number of adjacent matches to reporting node:  2
-------------------------SUMMARY--------------------------
Total communications: 251
Correct alarms: 107
Correct rate: 0.426295
```

Fig 9. station_log.txt

In the station_log.txt file, it will record in which iteration which node has reported to the base station and record all its information including its adjacent node's coordinates, temperature, MAC and IP address. Also including the communication time between the base node receiving the message and the report has been generated. In the end of

The above results were showed that the rate of true. Each picture represented different gird. Within each grid there were a wide range of iteration been tested.

From all the results had been tested, it clearly showed that the number of false alters are larger than true alters. For each grid, when the number of iterations were less then 100, the rate might be larger than 0.5, however once the iteration became larger, then the rate will decreasing to approximately 0.38-0.4.

1） 7 nodes (2*3 grid)

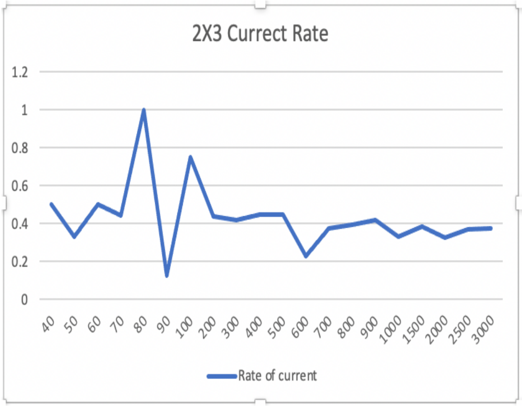


Fig 10. Grid result (2x3)

2） 21 nodes (2*10 grid)

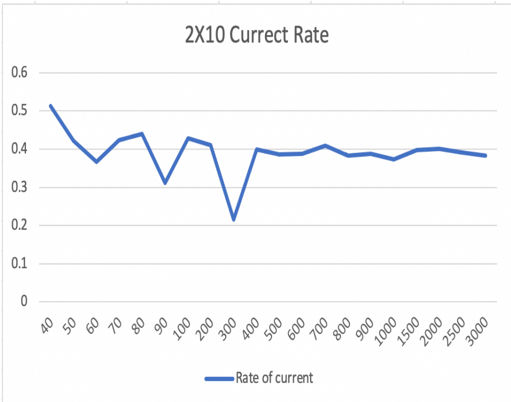


Fig 11. Grid result (2x10)

3） 41 nodes (5*8 grid)

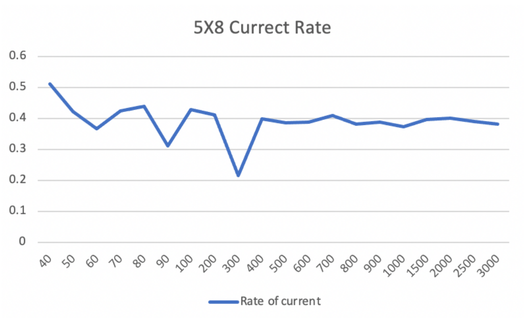


Fig 12. Grid result (5x8)

The above results were showed that the rate of true. Each picture represented different gird. Within each grid, there was a wide range of interactions been tested.

From all the results had been tested, it clearly showed that the number of false alters are larger than correct alters. For each grid, when the number of iterations was less than 100, the rate might be larger than 0.5, however, once the iteration increasing to a large number the rate will be stable between 0.37-0.4. Similarly, when we increase the number of nodes, this will be more obvious. When the number of nodes was 7, possible of 0.5 before the 100 iterations, however when the number of nodes increasing to like 41, the rate will stable while the iteration reached 50. The reason behind that is, the data generated for infrared satellites has a wide range even some difference will be allowed, but the range of the infrared satellite is still larger than the range of the sensor node.

VI Conclusion

In conclusion, the objective of this project is to design and implement an alert detection system in a distributed wireless sensor with a hypothesis proposed that the rate of true alter will be less than the rate of false alters. In section ii, introduced the design schema behind each part of the wireless sensor network including an overview of the whole system, sensor nodes, base station and infrared imaging satellite. The third part of this report showed the result of the different tests. In the test the two main factors changed were the number of iterations and the number of nodes. After comparing all the results, the hypothesis was accurate, which indicates the rate of false alerts will larger than the true alters. Some future work based on the outcomes of the simulation results can change the range of infrared imaging satellites. The range used in this system is quite broad, so it will make the false alert's appearances more than true. For the following experience, a smaller range can be used here to generate the temperatures, and this range can be according to the location of the network and what season, so it will be more actual compared to the one used now.

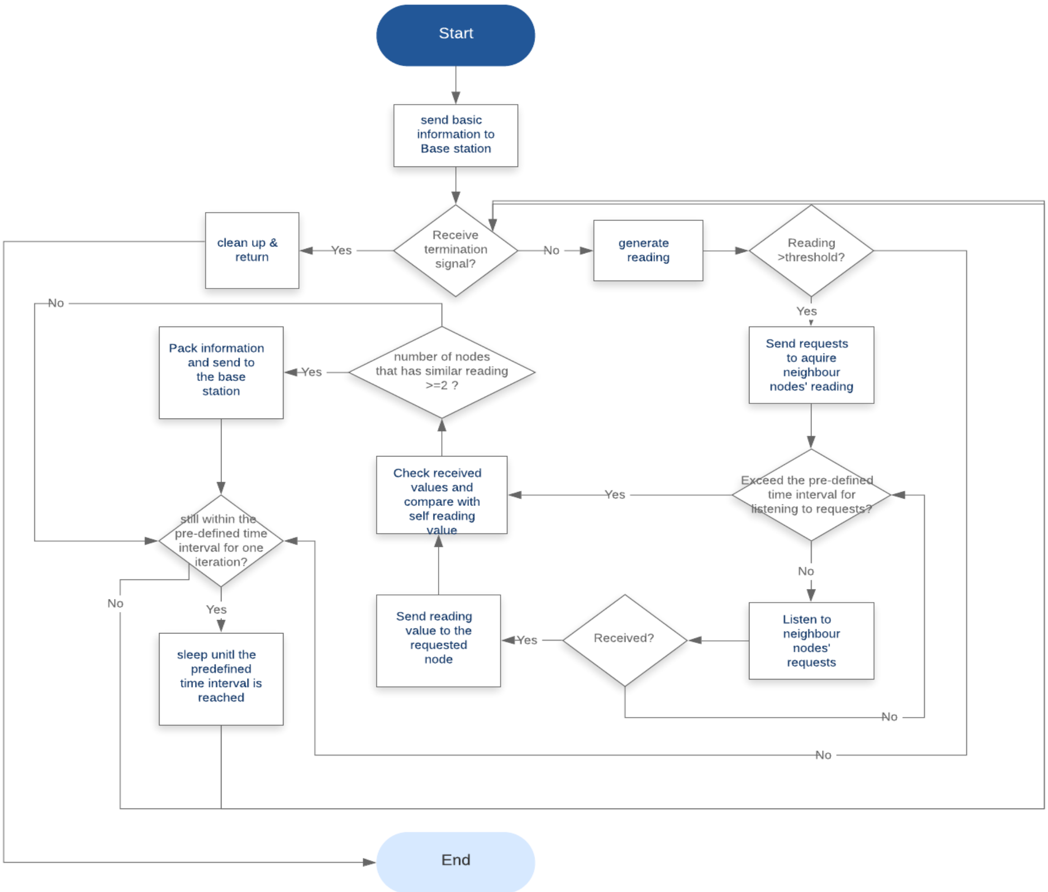


Fig. 7 Technical flowchart of the WSN node

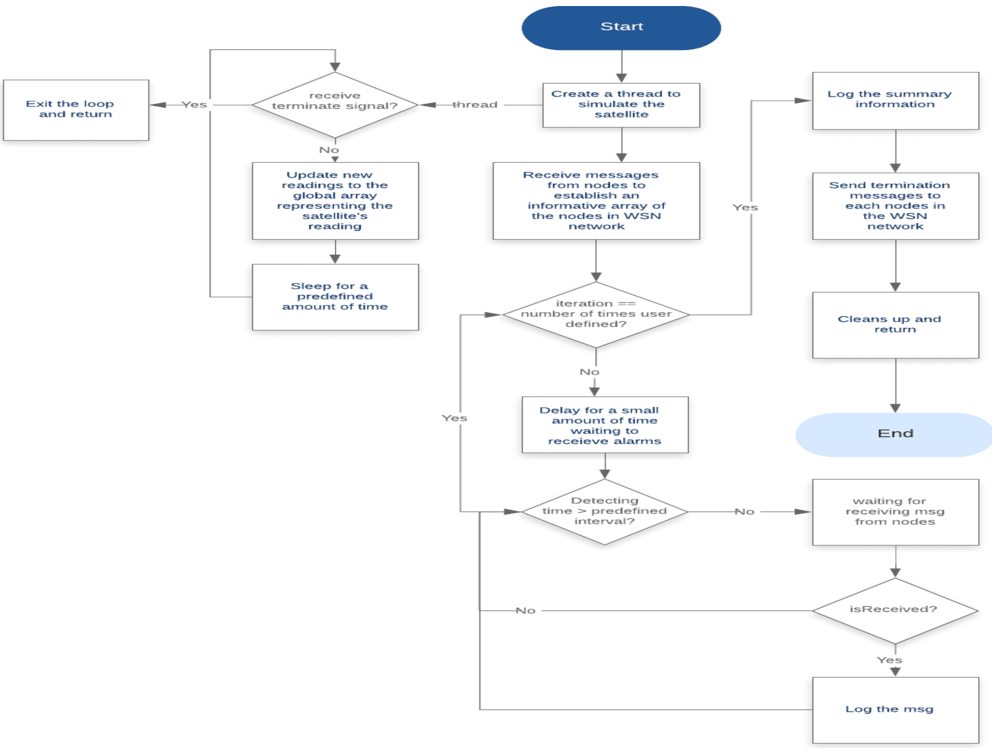


Fig8. Technical flowchart of the base station

REFERENCE

[1] M. Khan and M. A. Shah, "Inter-process communication, MPI and MPICH in microkernel environment: A comparative analysis," ***2017 23rd International Conference on Automation and Computing (ICAC)*****, Huddersfield, 2017, pp. 1-7, doi: 10.23919/IConAC.2017.8082077.dad**

[2]. Baskaran, V. (2020). ***Simulation and Analysis of Distributed Wireless Sensor Network using Message Passing Interface.*** **ArXiv.org, ArXiv.org, Jul 1, 2020.**