Simulation and Analysis of Wireless Sensor Network using Message Passing Interface

Samarasekara Vitharana Gamage, Bhanuka Manesha
bsam0002@student.monash.edu
28993373
School of Information Technology
Monash University
Malaysia

***Abstract*—Phasellus ut venenatis ante. Nullam faucibus felis nec neque laoreet, sed luctus nulla gravida. Phasellus velit tellus, semper id viverra vitae, semper sit amet lorem. Nam congue efficitur tristique. Aliquam bibendum enim mattis nulla pellentesque venenatis. Vivamus varius, arcu id ultricies porttitor, dui arcu pretium est, et iaculis ligula lectus in urna. Aenean quis nunc at felis luctus condimentum. Mauris a risus massa. Ut varius sodales eros. In vel bibendum lectus. Mauris vitae ornare lectus. Nam placerat euismod condimentum. Morbi posuere risus a risus dictum, et tempor massa gravida. Proin pulvinar egestas odio, sed sollicitudin enim pharetra eget. Vivamus dui elit, viverra ac felis quis, blandit tristique nibh.**

***Index Terms*—Phasellus ut venenatis ante. Nullam faucibus felis nec neque laoreet, sed luctus nulla gravida. Phasellus velit tellus, semper id viverra vitae, Phasellus ut venenatis ante.**

I. Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur porttitor mauris nibh, eu congue nunc facilisis a. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin aliquam mauris mattis fermentum venenatis. Cras non interdum tellus, vitae molestie justo. Aenean at pellentesque leo. Curabitur efficitur congue velit, nec gravida erat feugiat eu. Quisque et volutpat nunc. Maecenas cursus neque non lacus elementum, vel sollicitudin eros tempor.

Suspendisse potenti. Maecenas id nulla non nulla tincidunt convallis. Vestibulum nec lorem massa. Donec elit quam, tempor ut magna a, pretium maximus turpis. Suspendisse condimentum erat in eleifend elementum. Vestibulum tempor aliquam nibh, sed pellentesque erat pellentesque eu. Vestibulum vitae ex felis. Nullam id purus ac dolor lacinia ullamcorper nec quis tellus. In non ante cursus, molestie arcu ac, fringilla diam. Aenean facilisis, est vel ullamcorper pellentesque, sapien sapien scelerisque urna, pellentesque accumsan mauris dui in tellus. Etiam consequat auctor blandit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris condimentum pretium lorem, eu finibus odio sagittis ut. Integer malesuada, mi a aliquet gravida, augue neque ornare tortor, vel sodales mi dolor ac turpis. Pellentesque quis suscipit sem, ac imperdiet mi. Duis accumsan, sapien vel consequat rhoncus, augue elit scelerisque magna, sit amet dapibus dolor libero id sem.

II. Theoretical Analysis and Inter Process Communication Design

This section explores and discusses about the Inter Process Communication Design used by the program. The Wireless Sensor Network uses a grid based architecture where each node is able to communicate with the adjacent node and the base station only.

A. Inter Process Communication Grid Architecture

The number of nodes used by the WSN depends on how many processes are allocated by the user at the start. If the user specifies the number of nodes as 21, then the system uses one node as the base and the rest as the sensor nodes. Using MPI we allocate rank 0^{th} process to the base station and the rest of the nodes as the sensor nodes.

The system uses a dynamic grid based architecture. So the layout of the nodes are based on the height and width specified by the user. Figure 1 illustrates a simple example of how the user can rearrange a 21 node architecture. Figure 1a illustrates how the grid is initialized when the width is 4 and height is 5, while Figure 1b illustrates when its 2 and 10 respectively. The algorithm for the dynamic node initialization is explained in Section aaaaa.

After initializing, each node is able to communicate with the adjacent four nodes i.e left, right, top and bottom nodes and also with the base station. The base station is able to communicate with each node individually. The adjacent nodes of each of the sensor nodes are calculated using the height and width specified by the user. Equation 1 and 2 can be used to determine the row index and column index of a given node in the given sensor network. *Do note that we first minus 1 from the current rank, this is due to our base station being rank 0.*

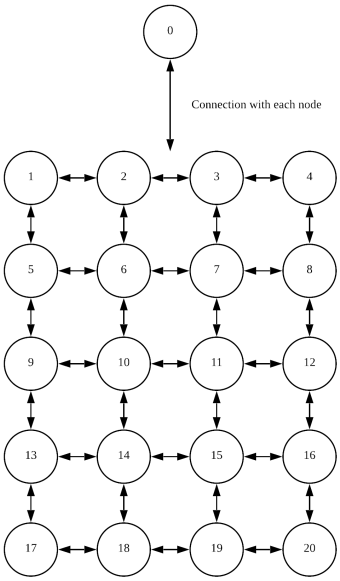
$$rowindex = (\text{rank} - 1) \quad // \quad \text{width} \tag{1}$$

$$columnindex = (\text{rank} - 1) \quad mod \quad \text{width} \tag{2}$$

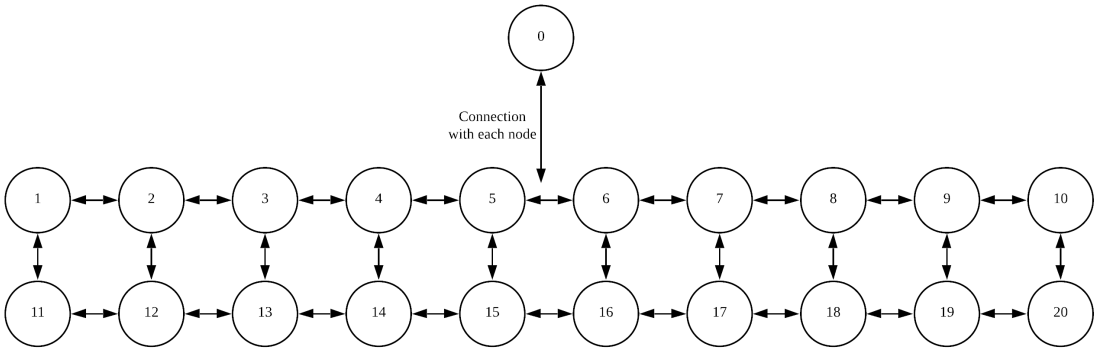
After determining the row index and the column index, we can use Equations 3, 4, 5 and 6 are used to get the four adjacent nodes. *Do note that the shift in the ranks due to base station is in rank 0 is accounted for.*

Fig. 1: Dynamic Grid Layout

(a) 4 X 5 grid



(b) 10 X 2 grid



$$left = rowindex \times WIDTH + columnindex \quad (3)$$

$$right = rowindex \times WIDTH + columnindex + 2 \quad (4)$$

$$top = (rowindex - 1) \times WIDTH + columnindex + 1 \quad (5)$$

$$bottom = (rowindex + 1) \times WIDTH + columnindex + 1 \quad (6)$$

Using this process, we do not need to send the rank values to the base station when an event occur. The base station can use the rank of the incoming node to generate the four adjacent nodes in order.

Published papers..

B. Encryption/Decryption and its Effects

Due to the increase risk of cyber attacks, all wireless networks needs to be encypted as its is easier to attack and steal unencypted data. So, all the messages are encrypted throughout the wireless sensor network. After comparing multiple encryption standards, Advanced Encryption Standard (AES) was used to encrypt the messages.

AES is a symmetric block cipher which can encrypt and decrypt information. Encryption part converts data into cipher text form while decryption part converts cipher text into text form of data. AES used different 128/192/256 bit keys to encrypt and decrypt data. For the WSN implementation we will be using AES 192, which uses a 192bit key. Since AES is the current standard for wireless communication, financial transactions and e-business we will be using it instead of other encryption algorithms **??**. AES has many modes of operations.

Examples for these are Electronic Codebook (ECB), Cipher Block Chaining (CBC), Propagating CBC (PCBC), Cipher Feedback (CFB), Output Feedback (OFB) and Counter (CTR).

For the WSN, we will be using an online implementation by kokke (ref), which use the Counter (CTR) mode. In order to improve the speed of the encryption process we will be using OpenMP to parallelize the process. The speed up analysis for the encryption algorithm is discussed in Section IV-C.

C. Wireless Sensor Network Event Detection Algorithm

Now we look into the algorithm used by the system to detect events in the WSN. The system is divided into two main sections.

- Base Station which logs all the events
- Sensor Nodes which triggers and checks for events

Initially the system generates sensor network grid dynamically using the *width* and *height* given by the user. This was discussed in Section II-A. Then both the sensor nodes and the base station start running their specific methods. First the IP Address and the MAC Address is sent from each node to be stored in the base station for logging purposes. These messages are encrypted using the AES encryption algorithm discussed in section II-B. The base station will then decrypt the message and store it in a dynamic array. Below is the data passed from the node to the base station.

- IP Address of the node
- MAC Address of the node

Now let us look into the algorithm used by the sensor nodes to generate and detect events.

1) Sensor Nodes: Firstly, each node generates a random number and sends it to the adjacent nodes. This message which contains the random number, sent between the nodes are also encrypted. Then the sensor node starts receiving random numbers from the adjacent nodes. If three or more random numbers are the same, an event is detected. The algorithm also takes into account the random numbers generated by the nodes in the previous iteration. So it works as a sliding window with a time frame depth of 2. Then the nodes will notify the base station by sending a message. This message contains the following details:

- The random number that triggered the event
- An array of four elements which corresponds to the iteration of the matched value. If the left adjacent node matched with the random number of the previous iteration, the array will contain the iteration number of the previous element in the 0^{th} index of the array.
- The MPI Wtime of the event detection to be used to calculate the communication time
- The date time string of the time when the event was detected to be used for logging

After sending the message to the server, it saves the random numbers from the current iteration.

2) Base Station: After the WSN is initialized, the base station starts checking for messages from any node. If the user has entered -1 as input for the iteration, the system will terminate checking for events only if user enters the keyword "stop". In order for this to work, the base station creates a POSIX thread, which enables it to check for user inputs as well log the events from the nodes. When the user inputs the keyword "stop", the thread then sends a message to the nodes to stop event detection gracefully.

flow chart for encryption and decryption

III. Methodology

In this section we look into the methodology used to test and analyze both the event detection criteria as well as the encryption algorithm used in the WSN. Table I summaries the parameters used by the program to get the data required for analysis. The buffer size of the messages used for testing the encryption is different from the actual buffer size which was used to get the results. The reason for this is the smaller buffer size is insignificant to see any speed up when parallelizing the code as they are in milliseconds. So for the encryption test case we used a buffer size of 160000, and for the WSN normal usage test case it was set to 320.

All of the test cases were run on a 2016 MacBook Pro with an Intel 6820HQ processor, which contains 4 cores and 8 logical cores. Figure 4 in the Appendix shows the screen shot of the console. Another approach for the test case would be to use the dynamic stopping approach. Figure 5 in the Appendix shows the screen shot of the console for this approach.

TABLE I: Parameters used for generating the data to be analyzed

Parameter	Value	Description
Width	4	The width of the sensor node grid
Height	5	The height of the sensor node grid
iterations	100	Total number of iteration performed by the system
interval	1	Wait time between each iteration
MAXRANDOM	4	Limit the random numbers generated to between 0 - 3
packsize	320 and 160000	Size of the buffer for sending and receiving data

IV. Results and Discussion

This section analyses the results obtained using the methodology described in Section III. The analysis is broken down into three main sub sections, each explaining the different aspect of the result.

A. Summary of Events

When an event is triggered by the sensor, the system generates two types of log files.

- Base Station generates the main log file which logs the event information and the simulation information

- Each sensor node logs the the original message, encrypted message and the encryption time

1) Base Station Log File: The base station logs two types of information, the event detection details and the simulation details at the end. First lets look into the log file to understand each part of it.

Listing 1 shows an event triggered at the 98 iteration of our test. This can be observed from the first line of the log which determines the iteration number. The next two lines of the logs displays the time when the event was logged on the file and the actual time when the event is detected respectively. This is useful when the node and base station is present in two geographical locations and the time to send the message effects the logging time. But in this example, since the node is a process inside the same machine, we are not able to see a difference. The next two lines denote the node that was triggered i.e. the node that sent the message to the base station.

```
1  ------------------------------------------
2  Iteration : 98
3  Logged Time :           Mon 2019-10-14 10:49:21
4  Event Occured Time :    Mon 2019-10-14 10:49:21
5
6  Activated Node
7  10    78:4f:43:5b:c2:c3   10.156.12.253
8
9  Adjacent Nodes
10 9     78:4f:43:5b:c2:c3   10.156.12.253   98
11 11    78:4f:43:5b:c2:c3   10.156.12.253   97
12 14    78:4f:43:5b:c2:c3   10.156.12.253   98
13
14
15 Triggered Value : 3
16 Communication Time (seconds) : 0.000409
17 Decryption Time (seconds) : 0.000048
18 Total Messages with server: 451
19 Total Activations per Message: 3
20 Total Activations : 1371
21 ------------------------------------------
22
```

Listing 1: Base station log file

B. Inter Process Communication Analysis

charts here node diagram here

C. Encryption Analysis

explain speed up here check byte log files

V. Conclusion

VI. Future Work

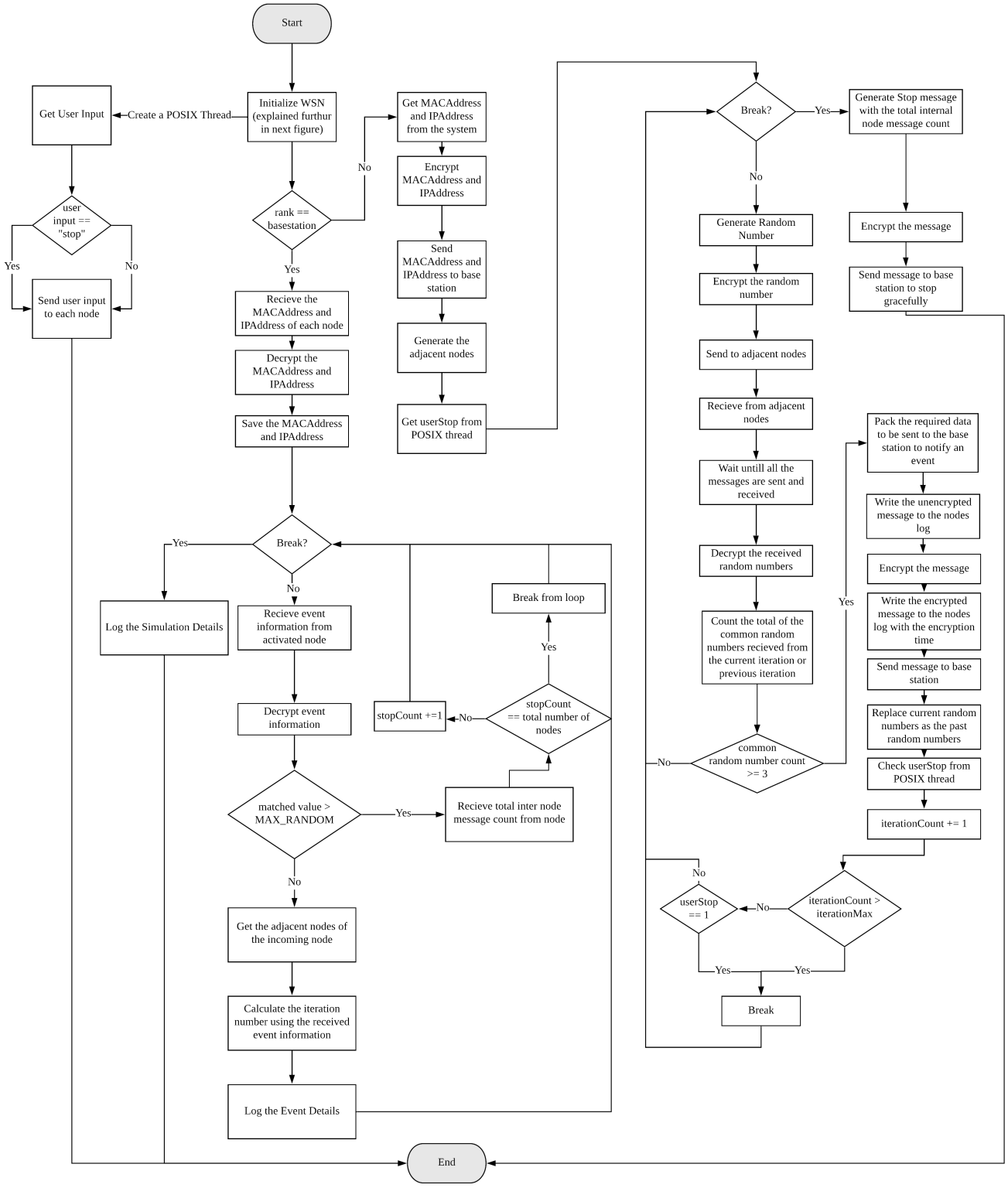


Fig. 2: Technical Flowchart for WSN Inter Process Communication

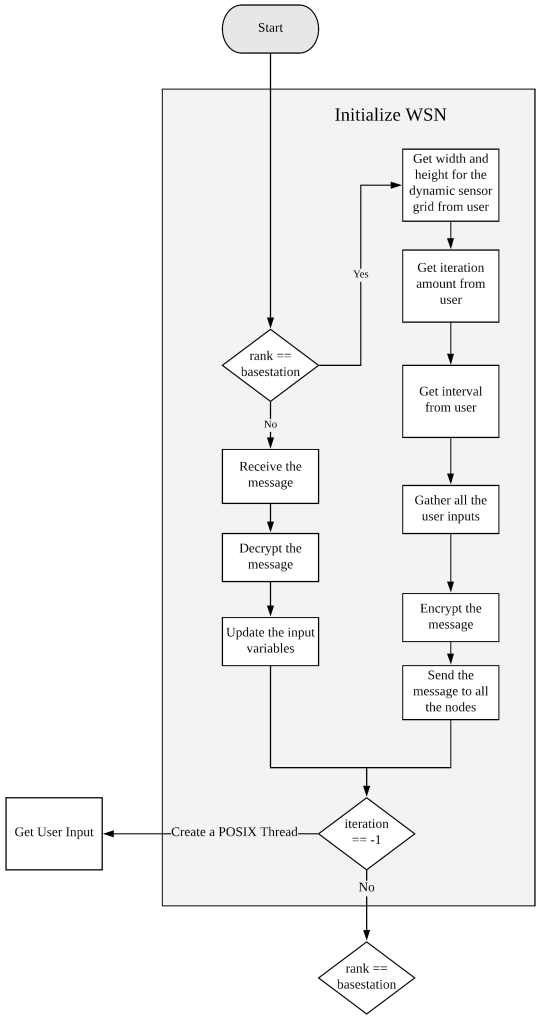


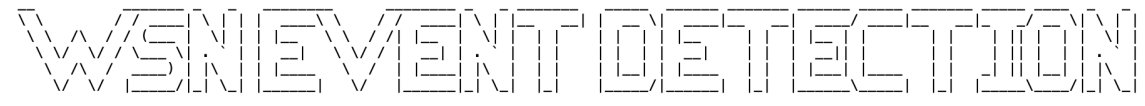
Fig. 3: Technical Flowchart for Initializing the WSN

APPENDIX

```
What is the shape of the 20 node grid ? (width height) :
4 5
Creating node grid of size (4,5) and base station using 21 nodes

How many iterations does the nodes search for (integer value, -1 for until "stop" is entered)? :
100
How often does each iteration happen (seconds) ? :
1
Running 10 iterations at 1 second intervals
```

Fig. 4: Screenshot of the console used for the generating the results



```
What is the shape of the 20 node grid ? (width height) :
4 5
Creating node grid of size (4,5) and base station using 21 nodes

How many iterations does the nodes search for (integer value, -1 for until "stop" is entered)? :
-1
How often does each iteration happen (seconds) ? :
1


Please enter "stop" to end the simulation.....
stop

Stopping the simulation.
```

Fig. 5: Screenshot of the alternative approach to run the program until user enters stop