Implementation and Analysis of Wireless Sensor Network Simulation using Message Passing Interface and Open Multi Processing

Sulthan De Neiro Raihan Syah Bungin
School of Information Technology
Monash University
Malaysia
29906164

Nicholas Yeo Wei Ming
School of Information Technology
Monash University
Malaysia
29458021

***Abstract*—We design, implement and evaluate a simulation of a Wireless Sensor Network(WSN) of temperature sensors that identifies high temperatures and verifies said readings by validating the results with adjacent sensors and an infrared satellite, all using Message Passing Interface(MPI). Based on the message passing model of parallel computing, MPI is used to simulate sensors, a base station and a satellite using individual threads. Each individual processing thread only has access to their local memory, but can communicate with other threads by sending and receiving messages. We use open-mpi, a widely used and open source implementation of MPI to implement our code. We show, through the analysis of our implementation, that the message passing model of parallel computing is both viable and efficient for the use in WSNs.**

***Index Terms*—Wireless Sensor Network, Message Passing, Parallel Algorithms, Parallel Processing, Simulation, Environmental Monitoring, Fires**

I. Introduction

Wildfires have been part of our planet's history since they became prevalent on a global scale around 400-350 million years ago [1]. They have proven to be extremely destructive, especially recent fires in Australia [2], The Amazon Rainforest [3] and California, United States [4]. They lay waste to acres of forests, destroying habitats, causing countless death of fauna and flora, and releasing huge amounts of carbon dioxide to the atmosphere. The impacts of wildfires on humans are also devastating, causing untold lost of life, property, and mental well-being .This is made even worse because of the creep of the construction of property in heavily forested areas [5].

There also has been a sizable voice within the scientific community arguing that wildfires are only going to become more prevalent as an effect of climate change [5]–[7]. However Stefan H. Doerr argues in [1] that contrary to popular sentiment, there actually is evidence that there is a trend of decreasing prevalence of wildfires.

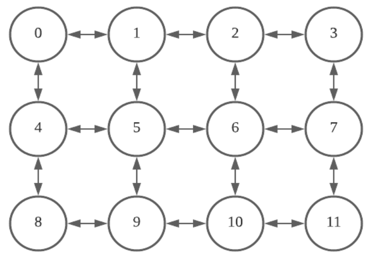
Regardless, wildfires take a significant toll on both humans and the environment. Thus, there is a need for early detection systems for such phenomenon. Our objective is to design, build and test a simulation of a wireless sensor network that can act as an early warning system for wildfires. That way, governments and the respective bodies would be able to act fast to either put out the fires, or failing that, have more time to evacuate the danger areas. One of the problems that can occur in this situation is that the higher number information being sent may affect the communication time needed to receive the information. One could hypothesize that the higher number of messages tends to increase the average communication time due to many information being held which may affect the next reported alert. Another problem is that the alert reported from the sensor may not actually be true due to unwanted issues which may require the system to use a second detection system such as satellite. It can be hypothesized that an increasing number of simultaneous reports cause the information being queued and lower the probability of the alert being true.

In the following sections, we will design and implement a dynamic two dimensions grid architecture to simulate the fire alert system by utilizing the concept of wireless sensor networks, base stations, and satellite data generation using the combination of Message Passing Interface (MPI) and Open Multi Processing (OpenMP) and test the hypothesis stated above.

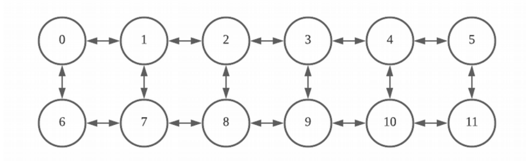
II. Design scheme for sensor network

Figure 1: Grid Architecture of 12 Sensor Nodes

(a) 3 x 4 Grid



(b) 2 x 6 Grid



A. Grid Architecture to Handle Communication Between Processes

Hoefler et al. [10] stated that one of the ways to have scalable processes is using the concept of nearest neighbor architecture. The system has dynamic capabilities to be expanded for any size. It also provides availability to run parallel processes without interrupting its functionality. Therefore, based on the suggested architecture from this research, the two dimensional grid will be adopted for our system.

The number of total nodes for sensor and base station is determined by the user input. There are three arguments for the user input. The first two arguments determine the size of the wireless sensor network while the last argument determines the number of iterations for the system to run. The first argument determines the size of the row while the second argument determines the size of the column of the grid. Equation 1 determines the number of processors that are used inside the system.

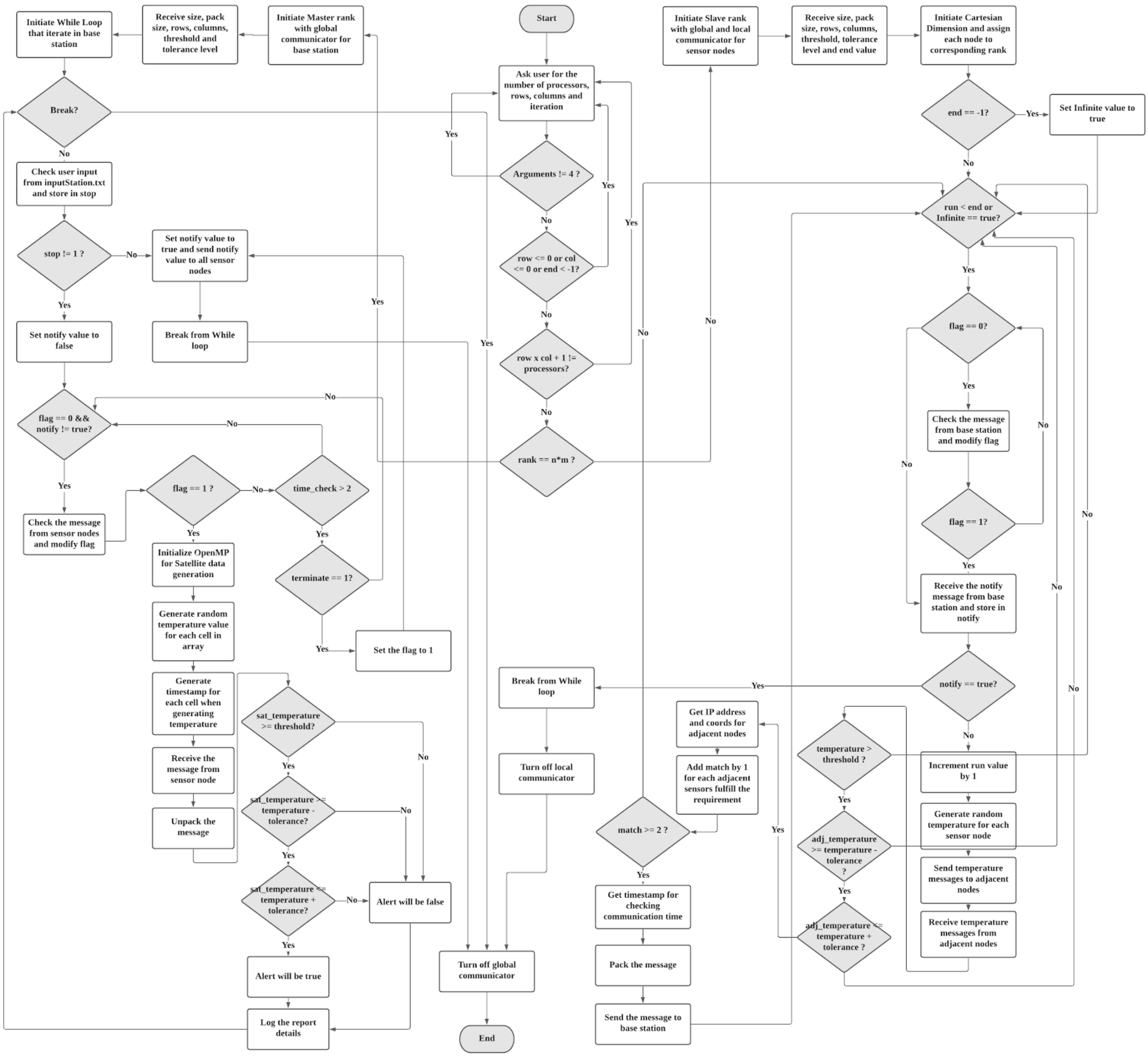
$$Number\_of\_processors = (\ rows\ x\ columns\ ) + 1 \quad (1)$$

Note: The extra one processor that was added will be allocated for the base station.

Utilizing the MPI, the base station has the process with the rank of (n+m). The reason for selecting the $(n+m)^{th}$ as the rank for the base station is to simplify the sensor nodes creation due to the **MPI_Cart_create** that starts the grid creation from 0 instead of 1.

The structure of the sensor nodes are based on the number of rows and columns that were stated by the user before the program executed. Figure 1 visually shows the possible arrangement of the 12 sensor nodes architecture excluding the base station. Figure 1a visualizes how the structure is arranged when the width is 4 and the height is 3,

Fig. 2 : Wireless Sensor Networks Architecture



while Figure 1b visualizes the arrangement when the width is 6 and the height is 2.

After the initialization of the sensor nodes, due to the connection between each node with the adjacent top, left, right or bottom, each node can communicate throughout each other. Each of the sensor nodes is also connected to the base station. The index for the rows and columns inside the wireless sensor network can be determined by Equation 2 and 3.

$$index\_row = rank \,//\, columns \quad (2)$$

$$index\_column = rank \ mod \ columns \quad (3)$$

Finding the ranks of the four adjacent nodes can be achieved by using the **MPI_Cart_shift** function. Using this function, it would be unnecessary to manually calculate the adjacent nodes using a custom formula. Therefore, this will improve the system inside the sensor nodes to reduce the computational power needed to achieve information.

B. *Wireless Sensor Network Detection and Alert Verification*

This section discusses the wireless sensor network event detection and the verification of the information. The system is split up into two different components. The first component is the base station. This component will handle the alert verification, information logging, and system termination. The second component is the sensor node. This component will handle the event detection, information packing, and sending message between components.

Figure 2 illustrate the system architecture of wireless sensor network detection and alert verification. At the beginning of the execution, the system asks for the user to input the length and width for the size of the sensor nodes grid architecture dynamically. The system will check if the proper arguments are given to avoid error later in the program. Then, the MPI Environment will be initialized and create processes that are determined by the user input. The number of processes are determined in Equation 1 in the section II-A.

The MPI will split into two communicators where one for global communication and the other as local communication. The global communicator is used by the base station and sensor nodes, while the local communicator handles the node to node communication inside the sensor nodes. The implementation using two different communicators will reduce the chance of the communicator disrupting the entire system entirely if one of the communicators functions incorrectly.

1.) Base Station: The base station is initialized at the (n x m)th process that holds the values of the rows,columns, size of the packed data, threshold and tolerance level. These values are used to receive the messages from the sensor nodes and verify the alert messages. Then, it initiates the loop that is able to be terminated based on two conditions. First, the loop stops if there is user input that notify the system termination. Second, the loop stops if all sensor nodes reach the end of iteration.

In order for the first condition to work, the loop checks if the user has input "1" in the text file. If it does, the base station will send notifications to all sensor nodes to be terminated, and the base station will stop the loop and end the program. For the second condition, the system termination occurs when the base station no longer accepts any messages from sensor nodes. After waiting for 2 seconds, the base station will terminate its own loop and end the program. The program will continue if any of the two conditions is not fulfilled.

For each iteration, the base station generates satellite data and compares it with the received alert message sent from the sensor nodes to verify the alert. The message contains the necessary information that will be logged in the report details. The simulation of satellite data generation will be explained in Section II-C.

2.) Sensor Nodes: The sensor nodes is initialized for (n x m) - 1 times for each process starting from 0 to (n*m)-1. These processes hold the values like base station in addition with the iteration limit value. The sensor nodes will initiate a dimension based on the size of n x m and initialize the local communicator for the sensor to sensor communication.

Before the iteration starts, the sensor will check the value of the iteration limit to determine the two different modes that can occur. If the iteration limit value is -1, then the program can only be terminated by fulfilling the first condition of base station termination. Otherwise the sensor can be terminated by fulfilling any of the two conditions of base station termination.

The main loop is initialized and iterates based on the mode that it selected. For each iteration, each sensor checks the termination message from the base station. If it exists, the sensor terminates their action, turns off local communicators, and ends their process. If no termination message exists, the sensor continues the process.

For each iteration, each of the sensors generates a random value of temperature. The temperature will be compared to the adjacent sensor nodes using the concept from section II-A. The inter process communication is implemented using the **MPI_Send** and **MPI_Recv** which then check if the values that being compared fulfill the requirements. The range to fulfill the requirements can be determined by Equation (4) and (5).

$$range\_min = temperature - tolerance \quad (4)$$

$$range\_max = temperature + tolerance \quad (5)$$

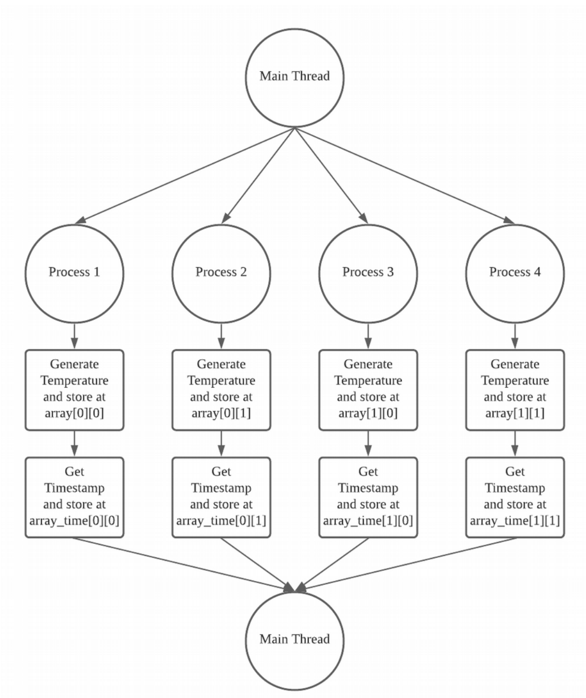
Note: The temperature refers to the temperature of the current sensor.

If the current sensor temperature is above the threshold, and the adjacent temperature is within the range, then the condition is fulfilled to include that adjacent sensor as part of the matching comparison. The current sensor then gets the IP address and increments the counter of the matched comparison. This action is repeated equal to the number of adjacent nodes that the current sensor has.

If the sensor node detects two or more matching comparisons, the sensor will send the necessary information to the base station as the alert. Before sending the information, the sensor packs the necessary information into one pack of data using the **MPI_Pack**. This implementation guarantees that each sensor only sends one message to the base station whenever an alert occurs. Therefore, it reduces the number of sending messages which increases the system efficiency.

C. *Infrared Image Satellite Data Generation*

Fig. 3: Satellite Parallel Data Generation



The satellite data generation is the subprocess that happens inside the base station based on Figure 2 and the detail action is illustrated in Figure 3. The satellite data generation is implemented using OpenMP which performs parallel data generation based on the generated threads. The number of threads generated is based on the number of processes created for the wireless sensor network. Each of the threads generates the random temperature value and the timestamp for the data generation and stores the information in the global array. Then, The information in the global array can be accessed by the base station to verify the alert message by using comparison of temperature that fulfills the same requirement of the sensor nodes comparison in section II-B. Before the verification process, the thread that generated by the OpenMP will merge back into the main thread and end the OpenMP for that base station iteration.

III. Result and Discussion of the Implemented Design

A. *Setup for Testing Simulation*

In this section, we investigate the procedure that used to test and analyze the wireless sensor networks and satellite data generation. The simulation is tested in the native Ubuntu operating system and was run on an AMD Ryzen™ 5 2500U Processors that has 8 cores and logical threads.

Table I: System Specifications and Parameters

Specification	Value	Description
CPU	8	Number of CPU
Logical Cores	8	Number of logical cores
RAM	12	Memory for running the program
Parameter	**Value**	**Description**
Height	4	The height of the grid of the sensor nodes
Width	5	The width of the grid of the sensor nodes
Iteration	20	Number of iteration perform by the program
Interval	0.1	Wait time between each iteration (seconds)
Maximum Random	100	The maximum generated value for temperature
Threshold	80	The minimum temperature to be alert
Tolerance	10	Value to be added or subtracted from the temperature to be classified as alert
Pack Size	10000	Size of the buffer for sending and receiving data

Table I summarizes the predefined information to obtain the data for the analysis. These values will remain constant for any condition when the simulation is being tested. The tests are based on the hypothesis mentioned in section I.

In order to track the communication time based on the increasing number of simultaneous reported alerts, counters are needed to track the number of messages being received and calculate the average communication time within the same iteration. Calculation of the average communication time is based on Equation (6).

$$average\_time = communication\_time / messages \quad (6)$$

For the second test, in order to test the comparison of satellite data due to event times queued in the base station for an increasing number of simultaneous reported alerts, the counter from the previous test is reusable. The second test requires the program to keep track of the number of matches alert between satellite and sensor messages for each iteration. The probability of the true alert can be achieved from Equation (7).

$$probability = match\_alert / total\_messages \quad (7)$$

Note: Equation (6) and (7) does not exist in the original code and is only being used for testing simulation.

Each test was conducted 2 times to reduce the bias of the first run.

B. *Simulation Testing Result*

This section examines the obtained result from section III-A. The discussion for this section is divided into three parts. Each subsection respectively describes the base station log file, average communication time, and satellite verification on alert messages.

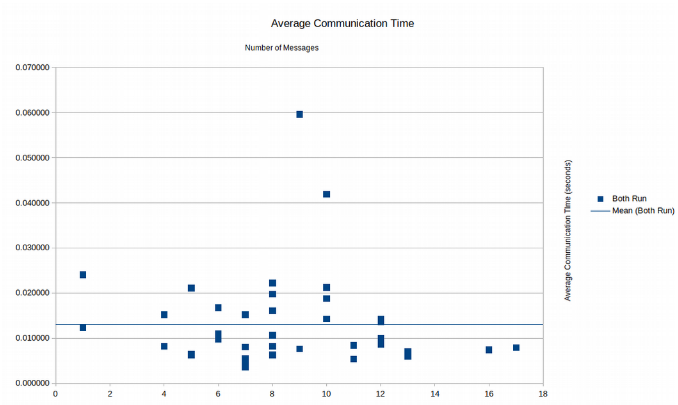
Fig. 4: Base Station Log

```
---------------------------------
---------------------------------
Iteration : 2
Logged Time :         Mon 2020-11-2 9:18:8
Alert Reported Time : Mon 2020-11-2 9:18:8
Alert Type : True

Reporting Node   Coord         Temp      IP
3                (0,3)         89        127.0.1.1

Adjacent Nodes   Coord         Temp      IP
4                (0,4)         95        127.0.1.1
2                (0,2)         85        127.0.1.1

Infrared Satellite Reporting Time (Celcius) : Mon 2020-11-2 9:18:7
Infrared Satellite Reporting (Celcius) : 91
Infrared Satellite Reporting Coord : (0,3)

Communication Time (seconds) : 0.019200
Total Messages send between reporting node and base station: 1
Number of adjacent matches to reporting node: 2
---------------------------------
---------------------------------
```

1.) Base Station Log File: The base station logs all the sending messages from each sensor node to the base station for each iteration. The information inside the message is illustrated in Figure 4. The first line of the message shows the value of the current iteration when sending the message. The next two lines displayed the time when the message was received and sent to the base station respectively. For this example, the difference of timing is not distinctive due to the message being sent and received within the same second.

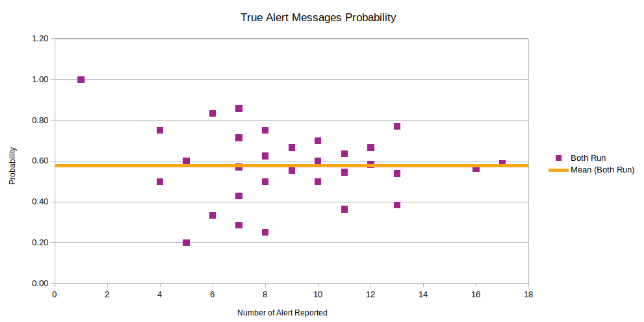
The next line indicates if the reported alert message is true or false. The shown indicator is based on the comparison of the message and the result from the satellite. The next two lines show the sensor node that is sending the message, the coordinates, the temperature, and the IP address of that sensor node. The next three lines display the information of the adjacent nodes that fulfill the requirements of matching comparison based on the statement in section II-B. Both of the IP addresses of the reporting and adjacent nodes are the same due to the system being tested on the same device. The next three lines show the result of the satellite reporting time, temperature, and coordinates respectively. The coordinate of the satellite is identical with the reporting node. The next three lines display the communication time, total message being sent and the number of matching adjacent nodes respectively. The communication time value is based on the difference of logged and alert reported time that displayed in higher precision. Based on the explanation in section II-B, the total messages sent between the reporting node and base station will remain the same.

Fig. 5: Average Communication Time



2.) Average Communication Time: Figure 5 visualizes the result of the average communication time for all iterations. The graph combined the information from both test runs. The horizontal axis represents the number of messages in one iteration while the vertical axis represents the average communication time in seconds within an iteration. The horizontal line inside the graph represents the average value of the average communication time.

Fig. 6: Alert Message Probability



3.) Satellite Verification on Alert Messages: Figure 6 illustrates the probability of the true alert messages. Both test run information is combined within the same graph. The horizontal axis represents the number of alert messages while the vertical axis represents the probability of those messages being true within one iteration. The horizontal line represents the average probability of messages being true.

C. Result Analysis and Discussion

1.) Analysis of Average Communication Time: Based on the hypothesis in section I, it is assumed that higher value of messages being sent increases the duration of the average communication time. Referring to Figure 5, the average of communication time varies based on the number of messages being sent from sensor nodes to the base station. The figure shows that the two highest average communication times are achieved when the number of messages is at 9 and 10. Alternatively, the events with the two highest number of messages produce lower values of average communication time compared to the average value of the events. One could argue that the lower values of the number of messages tend to have higher variance and lower accuracy on the precision. This could be observed through the lowest value of the average communication time is achieved when the number of messages is at 7 and the highest value is obtained when the number of messages is at 9. The highest difference of the average communication time is achieved when the difference number of messages is only 2. Alternatively, the higher values of the number of messages increase the accuracy of the data and could potentially reach the mean of the average communication time. Therefore, the higher number of simultaneous reports does not increase the average communication time.

2.) Analysis of Alert Message being True: According to the hypothesis, it is assumed that the higher number of messages being received may decrease the chance of the alert being true. Referring to Figure 6, the highest probability of alert being true is achieved when the number of messages is 1. Since the probability of alert being true when there is only 1 message is half, therefore we exclude this data from our analysis due to the randomness value. The figure shows that the higher the number of messages being reported does not decrease the probability of the alert being true. Alternatively, it reduces the probability difference between the mean and the selected event for each number of messages. Therefore, the higher number of simultaneous reports does not decrease the probability of the alert being true.

IV. Conclusion

Firstly, grid architecture based communication is capable of parallel message passing. The message that is being sent and received is not being limited to the local system but capable of being transmitted to the other system. The implemented system is capable of generating satellite data in parallel which reduces the time needed to compare two information from different systems. This system can be improved by reducing the information exchange between the sensor nodes however it requires complex systems that have higher space allocation.

Conclusively, it is not true that the number of simultaneous reports increase the average communication time and decrease the probability of alert being true. However, future work may be required due to the low number of values being tested in this system. Another factor that can be considered is that this simulation does not take into account the need of real life sensors to be able to determine their location, instead using matrix to determine the location of sensors. Leading localization techniques include Global Positioning Systems or Anchor Nodes [8]. By extending our simulation to include map and terrain simulation, localization techniques could be tested to improve result accuracy. Another potential improvement is to extend operation time, by reducing the message size as research showed that the message transmitter consumes the most power [9]. Thus, through the implementation of data compression algorithms like Huffman coding or multi-objective evolutionary optimizations, we would be able to decrease power consumption significantly.

Therefore, this research opens up the possibility for many improvements on the design and implementation of grid based architecture and wireless sensor network.

REFERENCES

[1] *S. H. Doerr and C. Santín, 'Global trends in wildfire and its impacts: perceptions versus realities in a changing world', Philos. Trans. R. Soc. B Biol. Sci., vol. 371, no. 1696, Jun. 2016, doi: 10.1098/rstb.2015.0345.*

[2] *'Australian wildfires killed, displaced 3 billion animals, scientists say', NBC News. https://www.nbcnews.com/news/world/australian-wildfires-declared-among-worst-wildlife-disasters-modern-history-n1235071 (accessed Nov. 02, 2020).*

[3] *A. Moloney, 'FEATURE-Bolivia's forest people burned by "perfect recipe" for wildfire disasters', Reuters, Oct. 29, 2020.*

[4] *'A power company's electrical equipment started one of the most destructive fires in California history - CNN'. https://edition.cnn.com/2020/10/30/us/southern-california-edison-responsible-woolsey-fire/index.html (accessed Nov. 02, 2020).*

[5] *J. E. Halofsky, D. L. Peterson, and B. J. Harvey, 'Changing wildfire, changing forests: the effects of climate change on fire regimes and vegetation in the Pacific Northwest, USA', Fire Ecol., vol. 16, no. 1, p. 4, Jan. 2020, doi: 10.1186/s42408-019-0062-8.*

[6] *A. Borunda, 'The science connecting wildfires to climate change', Science, Sep. 17, 2020. https://www.nationalgeographic.com/science/2020/09/climate-change-increases-risk-fires-western-us/ (accessed Nov. 01, 2020).*

[7] *A. P. Williams et al., 'Observed impacts of anthropogenic climate change on wildfire in California', Earths Future, vol. 7, no. 8, pp. 892–910, 2019.*

[8] *J. Yick, B. Mukherjee, and D. Ghosal, 'Wireless sensor network survey', Comput. Netw., vol. 52, no. 12, pp. 2292–2330, Aug. 2008, doi: 10.1016/j.comnet.2008.04.002.*

[9] *T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, 'Practical data compression in wireless sensor networks: A survey', J. Netw. Comput. Appl., vol. 35, no. 1, pp. 37–59, Jan. 2012, doi: 10.1016/j.jnca.2011.03.001.*

[10] *T. Hoefler, R. Rabenseifner, H. Ritzdorf, B. R. de Supinski, R. Thakur,and J. L. Tr äff, "The scalable process topology interface of mpi 2.2,"Concurrency and Computation: Practice and Experience, vol. 23, no. 4,pp. 293–310, 2011.*