ASSIGNMENT COVER SHEET

Student's name	(Family name) LIM	(Given names) ZHENG HAUR	
ID number	32023952	**E-mail**	zlim0038@student.monash.edu
Unit code & name	FIT3143 Parallel Computing	**Unit code**	FIT3143

Title of assignment	**Assignment – 2 *(Distributed Wireless Sensor Network Design for EV Charging Grid)***
Lecturer/tutor	Dr. Vishnu Monn / Dr. Leong Shu Min
Is this an authorised group assignment? ☐ **Yes** ☒ **No** If this submission is a group assignment, each student must attach their own signed cover sheet to the assignment.	
Has any part of this assignment been previously submitted as part of another unit/course? ☐ **Yes** ☒ **No**	
Tutorial/laboratory day & time	Tuesday, 1:00 pm
Due date: 16 Oct 2023 *(Extended 18 Oct 2023)*	**Date submitted: 18 Oct 2023**

All work must be submitted by the due date. If an extension of time to submit work is required, a Special Consideration Application (In-semester Assessment Task) must be submitted.

Has an extension been approved? **Yes** ☒ **No** ☐ **If yes, please give the new submission date** 18/10/23

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

Student Statement:

- I have read the University's Student Academic Integrity Policy and the University's Student Academic Integrity: Managing Plagiarism and Collusion Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Assessment and Academic Integrity Policy
- I have taken proper care of safeguarding this work and made all reasonable effort to ensure it could not be copied.
- I acknowledge that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
    - i. provide to another member of faculty; and/or
    - ii. submit it to a plagiarism checking service; and/or
    - iii. submit it to a plagiarism checking service 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 .... *Date*.... 18 October 2023 ....

Privacy Statement

The information on this form is collected for the primary purpose of assessing your assignment. Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au

MONASH University

FIT3143 Semester 2, 2023
Assignment 2 – Report
Distributed Wireless Sensor Network Design for EV Charging Grid

Note: Please refer to Assignment specifications, FAQ and marking guide for details to be included in the following sections of this report.

Include the word count here (for Sections A to C): 1357

A. Methodology (443)

B. Results Tabulation (466)

C. Analysis & Discussion (448)

D. References

Note: You may opt to customize this report to include additional sub-sections or any additional formatting where necessary. Please ensure that your report is properly formatted.

A. Methodology

The main design for the Distributed Wireless Sensing Network for the EV Charging Grid consists of a base station and a **m*n** size grid of charging nodes. Each node in the grid can communicate with their immediate adjacent nodes and the base station. In this design, each node as well as the base station is represented by a **MPI process**.

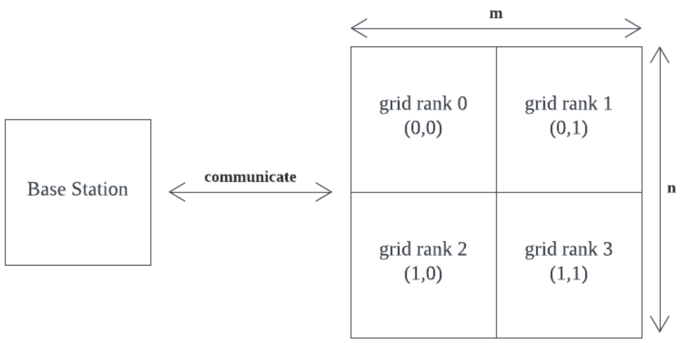


Figure 1: Diagram showing a 2x2 grid and base station.

All nodes including the base station will repeat a cycle of a period of time (e.g., 10 seconds) and during each cycle, the nodes will start by computing their number of ports available. If a node has a lower number of ports than a set amount, then it will prompt its adjacent nodes for their availability as well. If, however, all adjacent nodes are fully occupied as well, then it will send a report to the base station and the base station will reply with a list of available nearby nodes (nodes that are adjacent to the adjacent nodes). The general algorithm is shown in more detail in *Figure 2*.

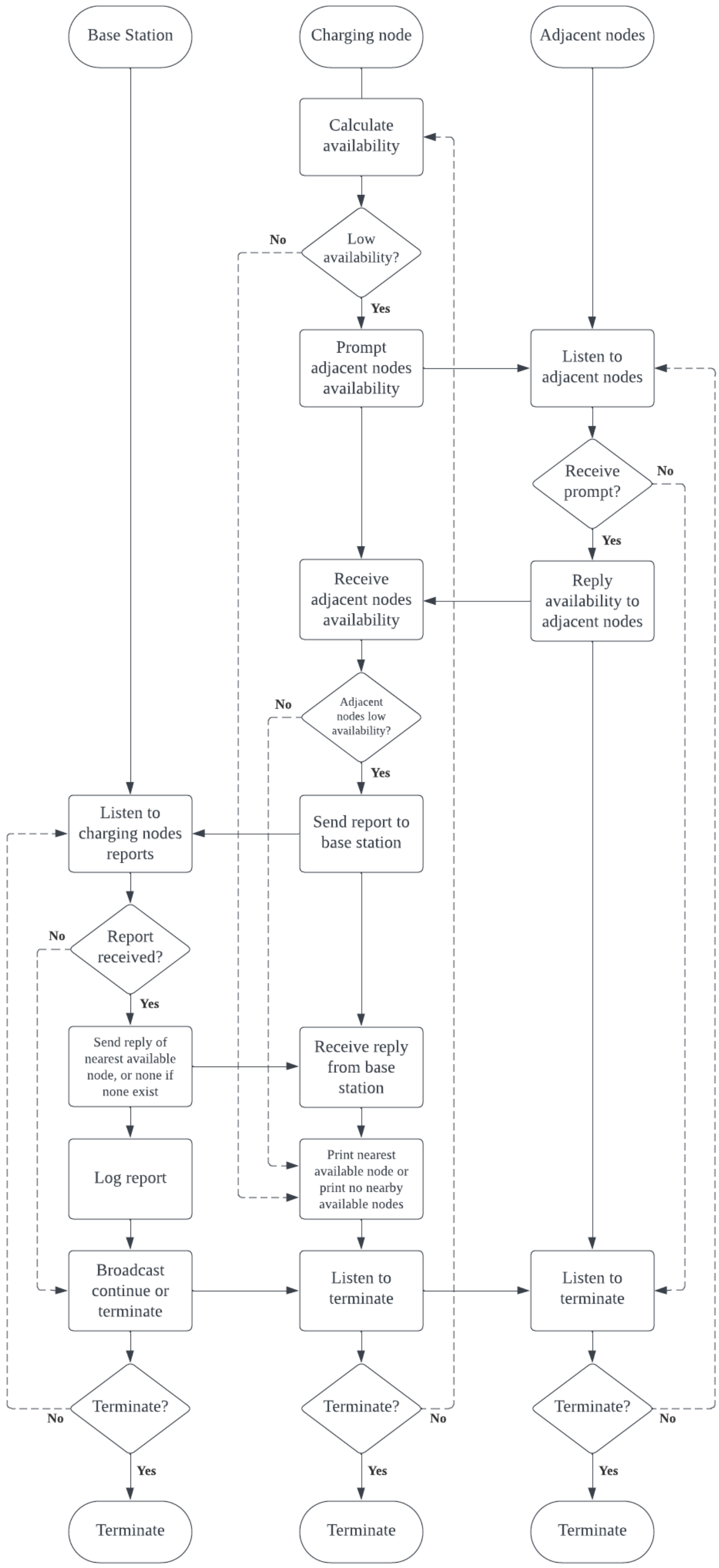


Figure 2: Flowchart of the general interaction between each MPI Process.

i. Charging Nodes:

The charging nodes are built using a 2D cartesian grid virtual topology. This creates a grid of size **m*n** where each process has their own grid rank and coordinates.

```
Availability Table: 2
1697530973      0
1697530978      0
1697530983      0
1697530988      1
0       0
Availability Table: 2
1697530973      0
1697530978      0
1697530983      0
1697530988      1
1697530993      1
Availability Table: 2
1697530998      0
1697530978      0
1697530983      0
1697530988      1
1697530993      1
```

Figure 3: 3 iterations of availability table for grid node rank 2.

At the start of each cycle, each node will calculate their availability based on probability in our simulation and this is stored into a table where the columns are the time and the availability. The table has a fixed size, and the data is overwritten from the first row when the table is full as shown in *Figure 3*. After computing availability, they will prompt each of their neighbours if their availability is below the threshold. If all neighbours are also below the threshold, they will send a report to the base station and wait for a reply. Then, each node will output their nearest available charging node, or none.

```
ITERATION: 3

[GRID 0] currently have 0 availability(s).
[GRID 0] CURRENTLY LOW ||| neighbors all unavailable.
[GRID 3] currently have 1 availability(s).
[GRID 2] currently have 0 availability(s).
[GRID 2] CURRENTLY LOW ||| nearest available node: 3.
[GRID 1] currently have 0 availability(s).
[GRID 1] CURRENTLY LOW ||| nearest available node: 3.
[GRID 0] CURRENTLY LOW ||| nearest available node (neighbor of neighbor): 3.
```

Figure 4: Output of all grid nodes for each iteration

ii. Base Station:

```
-----------------------------------------------------------------
Iteration: 3
Logged time: Tuesday, 2023-10-17 05:59:40
Alert reported time (Local time): Tuesday, 2023-10-17 05:59:35
Number of adjacent nodes: 2
Availability to be considered full: 0

Reporting Node  Coord   Port Value  Available Port
0               (0, 0)  5           0

Adjacent Nodes  Coord   Port Value  Available Port
2               (1, 0)  5           0
1               (0, 1)  5           0

Nearby Nodes    Coord
3               (1, 1)
3               (1, 1)

Available station nearby:
Communication Time (seconds): 0.5851198640
Total Messages send between reporting node and base station: 2
```

Figure 5: A log entry from the base station for each report

The Base Station is represented by an independent **MPI process**. The base station listens to incoming reports in each cycle from each node, and then computes the nearest available node to the reporting node and sends a reply. Each report is logged into a log file where each entry is as shown in *Figure 5.* The base station also controls the termination of all nodes at the end of each cycle.

B. Results Tabulation

The results are tabulated from **6 executions of the program on the local computer** and **3 executions of the program on the cluster computing setup (CAAS)**.

Specifications of each run:

- Size of grid
  - Local: 2x2 and 2x3
  - CAAS: 4x4
- Number of charging ports: 5
- Number of iterations: 10
- Cycle Period: 5 seconds
- Availability to be considered low: 0

Specifications of Platform:

- Local Computer:
  - Logical Processors: 8
  - Base Clock Speed: 2.50GHz
  - System Memory: 16GB
- CAAS
  - Tasks: 20
  - Memory: 20GB
  - CPUs per Task: 8
  - Tasks per Compute Node: 2

Run	Grid Size	Average Communication Time (seconds)	Number of Reported Messages
Local 1	2x2	0.583202814	4
Local 2	2x2	1.085092796	15
Local 3	2x2	0.620997774	4
Local 2x2 (Avg.)	**2x2**	**0.917095404**	**7.67**
Local 4	2x3	0.791596118	14
Local 5	2x3	0.963910821	12
Local 6	2x3	0.721061794	16
Local 2x3 (Avg.)	**2x3**	**0.813958671**	**14**
CAAS 1	4x4	0.730024671	37
CAAS 2	4x4	0.862899372	24
CAAS 3	4x4	0.549931689	18
CAAS (Avg.)	**4x4**	**0.729357825**	**26.33**

Table 1: Program execution results.

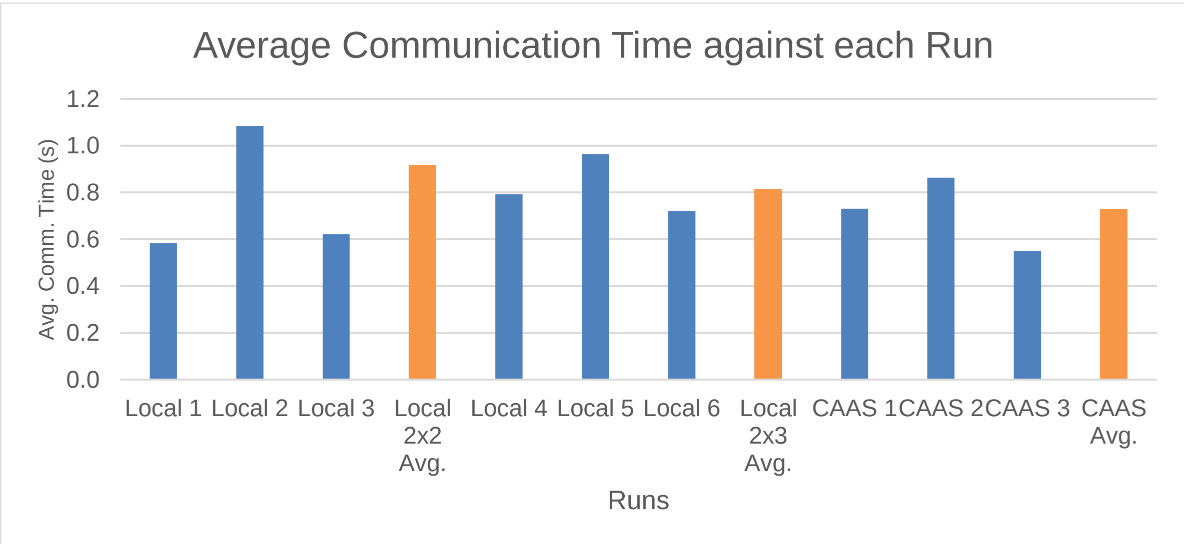
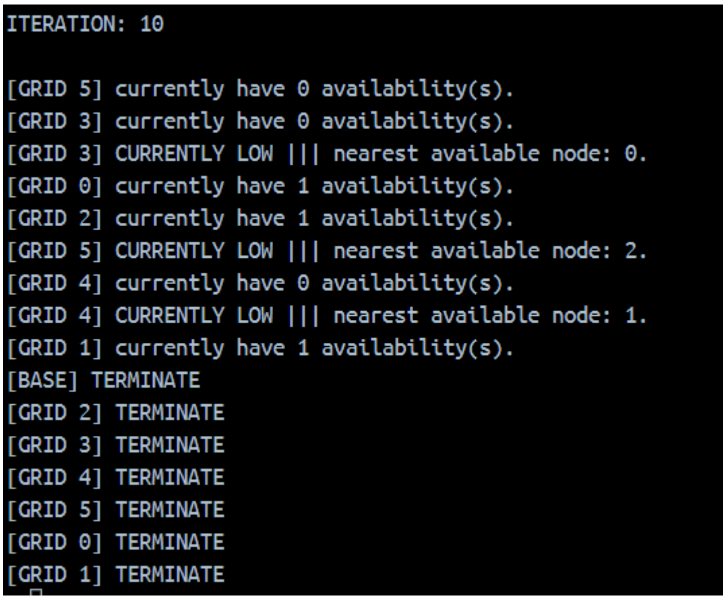


Figure 6: Graph of average communication time against each run.

Table 1 records the average communication time for each of the reported message and reply from the log files of the base station of each test run. There are 3 runs for each grid size, 2x2, 2x3 and 4x4 using the local machine and CAAS. *Figure 6* is the graph created from the values of *Table 1.*



```
ITERATION: 10

[GRID 5] currently have 0 availability(s).
[GRID 3] currently have 0 availability(s).
[GRID 3] CURRENTLY LOW ||| nearest available node: 0.
[GRID 0] currently have 1 availability(s).
[GRID 2] currently have 1 availability(s).
[GRID 5] CURRENTLY LOW ||| nearest available node: 2.
[GRID 4] currently have 0 availability(s).
[GRID 4] CURRENTLY LOW ||| nearest available node: 1.
[GRID 1] currently have 1 availability(s).
[BASE] TERMINATE
[GRID 2] TERMINATE
[GRID 3] TERMINATE
[GRID 4] TERMINATE
[GRID 5] TERMINATE
[GRID 0] TERMINATE
[GRID 1] TERMINATE
```

Figure 7: Proper termination for each MPI Process.

Run	**Grid Size**	**Termination Successful**
Local 1	2x2	Yes
Local 2	2x2	Yes
Local 3	2x2	Yes
Local 4	2x3	Yes
Local 5	2x3	Yes
Local 6	2x3	Yes
CAAS 1	4x4	Yes
CAAS 2	4x4	Yes
CAAS 3	4x4	Yes

Table 2: Table showing if termination was successful for every run.

Number of Reported Message per Iteration	Average Communication Time (Seconds)
1	0.798561
2	0.811899
3	0.921747
4	0.871792
5	0.870004
6	0.665415
12	0.738333
13	0.85496
16	0.72392

Table 3: Table showing the number of reported messages per iteration and average communication time in seconds.

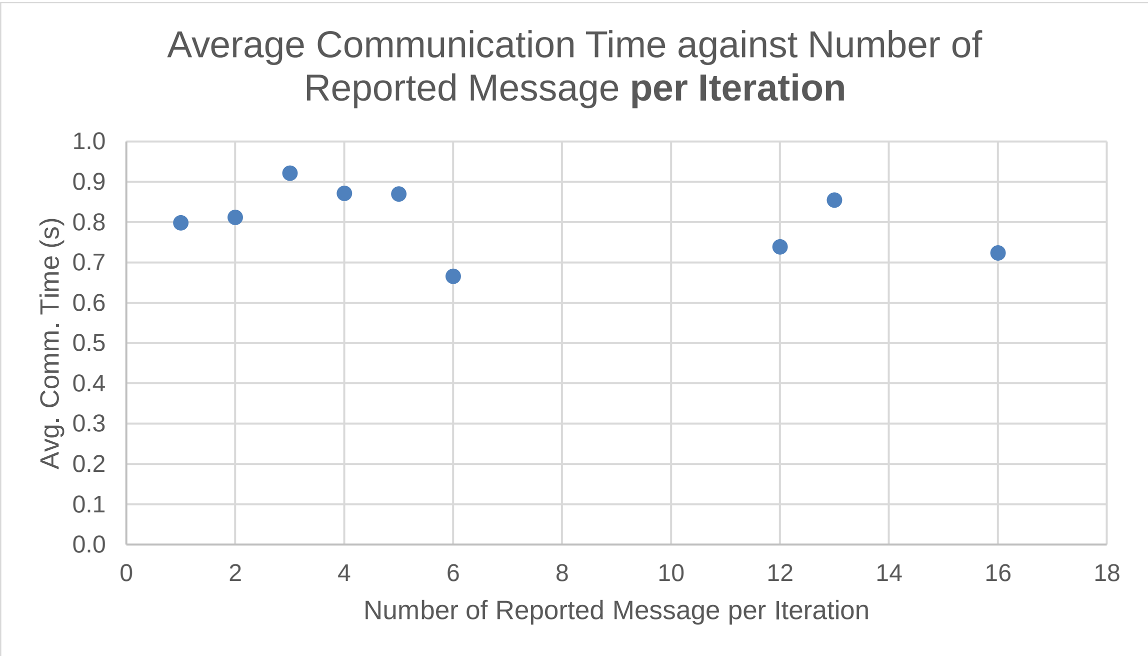


Figure 8: Graph showing the average communication time against the number of reported messages per iteration.

The values tabulated in *Table 3* is calculated manually by reading the log file from the base station. The number of reported messages per iteration is counted manually and the average communication time in seconds is recorded. *Figure 8* is a graph that displays the results tabulated in *Table 3*.

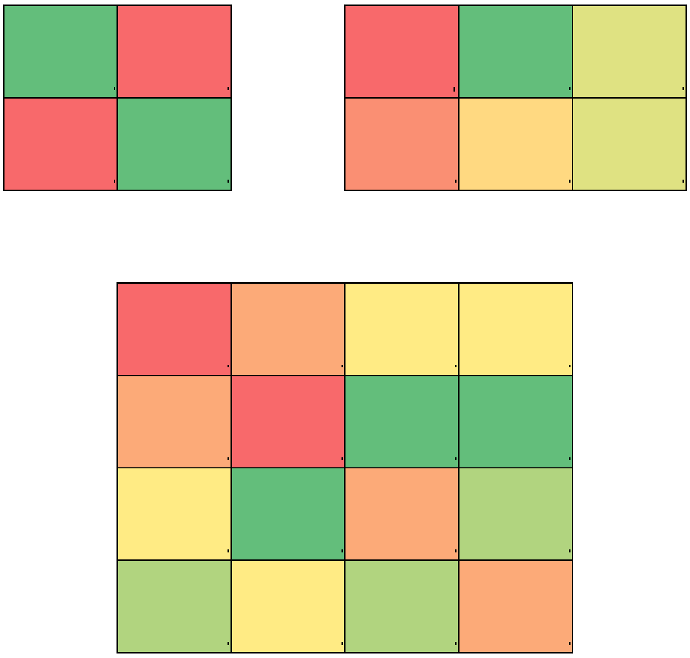


Figure 9: Heatmap of reporting nodes to base station.

Figure 9 heatmap is generated by counting the number of report messages from each node for each grid sizes from the base station log files. Red represents a higher number of report messages and green indicates a lower number of report messages.

MONASH University

C. Analysis & Discussion

The first hypothesis was for the average communication time taken to increase for an increasing number of reported messages in the same iteration.

From the results tabulated in *Table 1* and plotted into a graph as shown in *Figure 6*, we could observe that the average communication time per report from each charging station node to the base station is roughly the same across multiple grid sizes from 2x2 to 4x4. The increasing grid size should increase the number of reporting nodes and thus, increase the simultaneous report messages in each iteration, but the results tabulated could be inferred that the communication time is roughly constant with increasing grid sizes.

Furthermore, the results do not reflect upon the hypothesis as shown in *Figure* 8, the communication time is relatively constant across the different number of simultaneous report messages. This might be due to several reasons such as the grid size not being large enough for the simultaneous report messages to cause a communication congestion. In addition, in this simulation, the receiving of report messages and the sending of replies is done in parallel using OpenMP and thus, this will reduce the time taken for communication in high demands. Hence, the observation was a constant communication time despite of the increasing number of simultaneous incoming and outgoing report messages.

The second hypothesis is that the amount of communication of the charging nodes grid and the base station would be the densest at the edges of the grid. *Figure 9* shows the heatmap of communication of charging nodes in grid to the base station and observing it, we could see that the hypothesis is indeed true. This is most likely because the nodes at the edge of the grid have lesser adjacent nodes, and thus is more likely for all their adjacent nodes to be low in charging station availability. Hence, they would communicate with the base station to find a nearby available charging station nodes.

Lastly, the last hypothesis is that the amount of communication of the charging nodes within the grid would be the densest near the middle of the grid. However, due to the communication within the grid from one node to another not being logged, we would be unable to verify the veracity of this hypothesis.

The limitation of this simulation is the need for 1 processor for each MPI process to represent each node and the base station. Despite using CAAS for having a cluster of compute nodes, we are still limited on the grid size as the number of computing nodes grows exponentially with the grid size. Therefore, larger grid sizes such as 10x10 would be inherently difficult to simulate.

D. References

RookieHPC. (n.d.) *MPI Documentation.* GitHub. Retrieved from https://rookiehpc.org/mpi/docs/index.html