

ASSIGNMENT COVER SHEET

| | | | |
|-----------------------------|----------------------------|--------------------------|-----------------------------|
| Student's name | (Family name) Kang | (Given names) Hong Bo | |
| ID number | 32684673 | E-mail | hkan0015@student.monash.edu |
| Unit code & name | FIT3143 Parallel Computing | Unit code | FIT3143 |

| | |
|--|---|
| Title of assignment | Assignment – 2 (You may include an appropriate title as per your report) |
| Lecturer/tutor | Dr. Shu Min Leong (Jessie) |
| Is this an authorised group assignment? <input type="checkbox"/> Yes <input checked="" type="checkbox"/> 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? <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No | |
| Tutorial/laboratory day & time | Tuesdays, 8 am - 10 am |
| Due date: 16 October 2023 | Date submitted 13 October 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/...../.....

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

| | |
|---------------------------|--|
| Student Statement: | <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> provide to another member of faculty; and/or submit it to a plagiarism checking service; and/or 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. |
| | SignatureKang Hong Bo..... Date.....13/10/2023..... |
| 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 |

FIT3143 Semester 2, 2023
Assignment 2 – Report
ELECTRIC VEHICLE CHARGING GRID: OPERATED BY A DISTRIBUTED
WIRELESS SENSOR NETWORK (WSN)

Include the word count here (for Sections A to C): A: 438 B: 500 C: 474

A. Methodology

For this assignment, we were asked to design a parallel algorithm on wireless sensor networks. To do that, we are going to use MPI as a library, drawing inspiration from research work on optimizing communication patterns and scalability in parallel computing (Hoeffler et al., 2010). This approach aligns with the principles of MPI, which have been refined over the years to address issues related to scalability and communication optimization.

Overview

In these assignments, we need to build a $m \times n$ grid that can communicate with a base station, and each node of $m \times n$ will have k ports inside to determine how many available ports are in this station. A brief drawing of the layout of the network is given below.

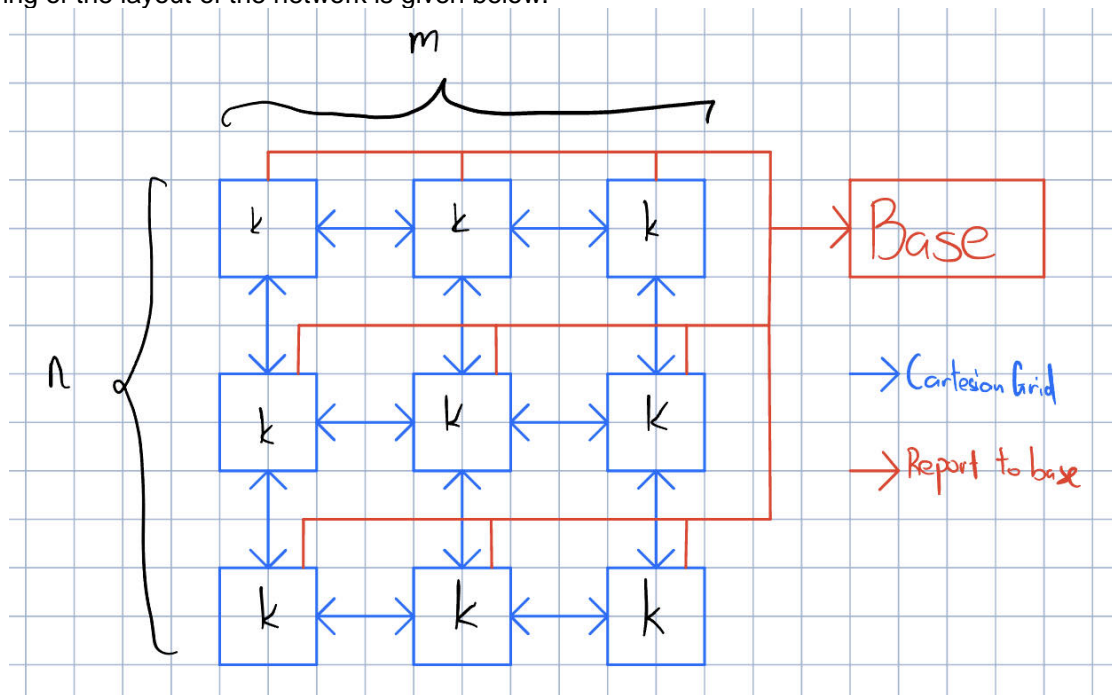


Fig 1. Sketch of the network

Each blue box is a charging station, and the station will contain a k number of charging ports. All stations can communicate with each other to know their availability and the base station which we need to report things to it.

Flow Chart

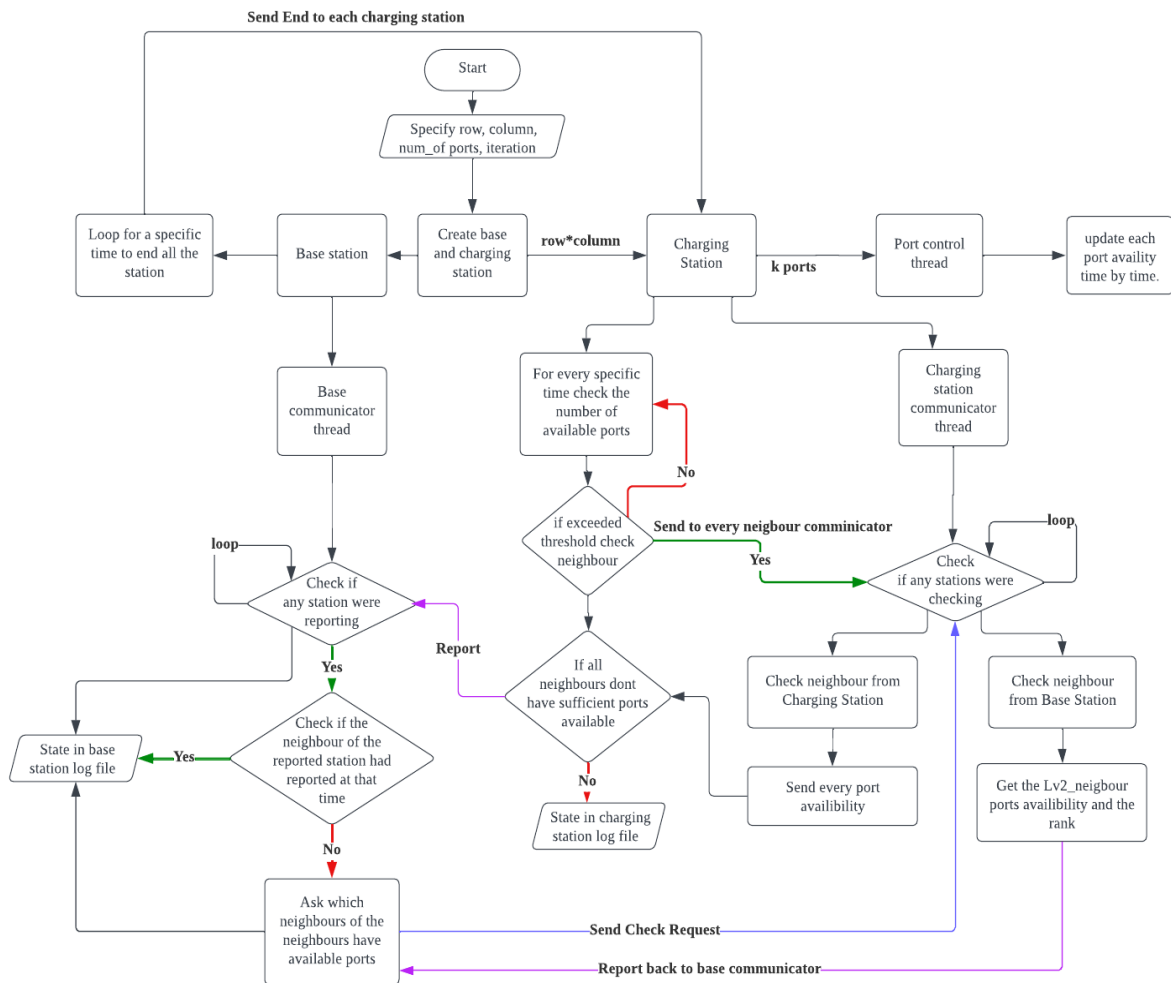


Fig2. Flow Chart of the implementation of the WSN model.

Main

Firstly, since we are using MPI to implement this program, the user can prompt how many processors to run and the executable file can take in at most 4 arguments which will decide how many rows and columns (rows * columns must be equal to the size of the processors -1), how many ports will there be in the station and the time to end the process. My executable file could take in none, two, three and four arguments. None will check for the number of processors and get the grid size by the square root of the number of processors -1. Two arguments indicate row and column, the third argument indicates the number of ports, and the fourth argument indicates the number of cycles. All of the default settings of the number of ports are 3 and the cycle is 12. After initiating the variables of the process, we separate the processors into one base and charging stations where the base will be the last processor (size 1).

Base Station

The base station will have a loop that decides when all the stations should terminate. It will also create a base station communicator to receive reports from charging stations ask about the availability of other stations and send the available ports to the reported station. It will also save the reported station time by time so that for the next reporting station if it needs to check the station that has been reported at that time, the base station will not ask that specific station and proceed to the station that hasn't reported yet. The base station communicator will save the details in the log file for each report received.

Charging stations

Every charging station will create cartesian grids with rows and columns using MPI libraries. They will have k^{th} ports which will be updated by the POSIX thread constantly. Each charging station will check for the availability of their charging ports and if it exceeds the threshold (20% left), the station will ask their neighbour stations if there are available ports, if their neighbour doesn't have sufficient available ports then it will send reports to the base station communicators. Besides, each charging station will have its communicator which will receive any message from other charging stations or the base station. The charging station will have a log file which will determine the time and available ports in the log file.

B. Results Tabulation

Log File

Files that keep details of each iteration or report. Screenshots of log files are given below.

```
Report from station 0 at time 2
Neighbor 0: -2
Neighbor 1: 2
Neighbor 2: -2
Neighbor 3: 1
station_report_state[0]: 1
station_report_state[1]: 0
station_report_state[2]: 0
station_report_state[3]: 0
Station 0 is checking for station 2
Station 0 is checking for station 1
Report from station 1 --- Neighbor rank: 3 has 1 available ports
Report from station 2 --- Neighbor rank: 3 has 1 available ports
```

Image 1. Base station log file (Simulation 1)

In this base station log file, we can know which station was reported and when it was reported. The neighbour of the reported station will also be shown in the log file. The station report state will determine whether the base station needs to check for the neighbour or not. Then the report from the neighbour will be listed last

The charging station log file will consider the availability of ports, the status of checking neighbours, and the status of reporting to the base.

```
Station 0 iteration 1
Time: 2
Port 0:0
Port 1:1
Port 2:1
Station 0 has 1 available ports
Station 0 is checking neighbor 2
Station 0 is checking neighbor 1
Station 0 - Time spent for checking neighbor: 0.000636
Iter 1 - Station 0 neighbours has no available ports
Sending my rank 0 iter 1 to base station
Station 0 received from base station
Station 0 has no available ports
Station 0 - Time spent for checking lv2 neighbor: 0.005711
```

Image 2. Charging station 0 log file (Simulation 1)

The charging station checks its charging ports if it has less than 20% of ports left in the station it will prompt for the neighbour's station charging ports. The duration of checking with other neighbours was specified in the log file with units of nanoseconds. After checking with the neighbours, if the neighbours don't have sufficient charging ports (more than 20%). The charging station will report to the base and will receive the availability of the other stations. The duration of reporting to the base and receiving messages from the base was written in the log file.

Summary of Results

Local Device

For results in the local device, I will run two simulations with different sizes of grids with constant ports and times.

Simulation 1: `mpirun -np 5 a2 2 2` (default setting 3 ports, 12 iterations)

For this simulation, it will form a 2 by 2 cartesian grid, each node will consist of 3 ports.

| Simulation 1 | | Row | 2 | C.S | 4 | | |
|--------------|--------|--------|-------|-------|-------|-------|--|
| | | Column | 2 | | | | |
| | | Ports | 3 | | | | |
| | | Time | 12 | | | | |
| Time | Report | NFS | ETS 0 | ETS 1 | ETS 2 | ETS 3 | |
| 1 | 0 | 2 | 0 | 0 | 1 | 1 | |
| 2 | 4 | 4 | 1 | 1 | 1 | 1 | |
| 3 | 0 | 2 | 1 | 1 | 0 | 0 | |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | |
| 5 | 0 | 2 | 0 | 1 | 0 | 1 | |
| 6 | 1 | 3 | 0 | 1 | 1 | 1 | |
| 7 | 1 | 3 | 1 | 1 | 1 | 0 | |
| 8 | 0 | 2 | 1 | 0 | 0 | 1 | |
| 9 | 1 | 3 | 0 | 1 | 1 | 1 | |
| 10 | 1 | 3 | 1 | 1 | 1 | 0 | |
| 11 | 0 | 2 | 1 | 0 | 0 | 1 | |
| 12 | 1 | 3 | 0 | 1 | 1 | 1 | |

Table 1. Results of Simulation 1

Simulation 2: `mpirun -np 7 a2 2 3` (default setting 3 ports, 12 iterations)

For this simulation, it will form a 2 by 3 cartesian grid, with 3 ports in each node.

| Simulation 2 | | Row | 2 | C.S | 6 | | | |
|--------------|--------|--------|-------|-------|-------|-------|-------|-------|
| | | Column | 3 | | | | | |
| | | Ports | 3 | | | | | |
| | | Time | 12 | | | | | |
| Time | Report | NFS | ETS 0 | ETS 1 | ETS 2 | ETS 3 | ETS 4 | ETS 5 |
| 1 | 3 | 5 | 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 0 | 3 | 1 | 1 | 1 | 0 | 0 | 0 |
| 3 | 0 | 3 | 1 | 0 | 0 | 0 | 1 | 1 |
| 4 | 1 | 4 | 0 | 0 | 1 | 1 | 1 | 1 |
| 5 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 3 | 5 | 1 | 1 | 1 | 1 | 1 | 0 |
| 10 | 1 | 4 | 1 | 1 | 1 | 0 | 0 | 1 |
| 11 | 1 | 4 | 1 | 0 | 0 | 1 | 1 | 1 |
| 12 | 0 | 3 | 0 | 1 | 1 | 1 | 0 | 0 |

Table 2. Results of Simulation 2

The table above shows how many reports (Reports) were sent to the base station at a specific time, how many stations were non-free stations (NFS) for other stations to refer to, and whether the station exceeded the threshold (ETS).

To determine whether the stations report to the base or not, we have a log file that saves the report from the charging station (C.S) time by time. To determine whether it is an NFS we also did a log file for each station which saves the details of the ports and if the station's ports have insufficient ports (less than the threshold) it will be considered as (ETS).

CAAS

For results in CAAS, I run two simulations. One simulation has the same size as simulation 2 in the local device and the other simulation is run with a larger grid size.

Simulation 1: Node: 2, ntasks:7, CPUs-per-task:3, ntasks-per-node:4 (rows: 2 cols:2 ports:3 cycle:12)

| Simulation 1 | | Row | 2 | C.S | 6 | | | |
|--------------|--------|--------|-------|-------|-------|-------|-------|-------|
| | | Column | 3 | | | | | |
| | | Ports | 3 | | | | | |
| | | Time | 12 | | | | | |
| Time | Report | NFS | ETS 0 | ETS 1 | ETS 2 | ETS 3 | ETS 4 | ETS 5 |
| 1 | 0 | 4 | 1 | 1 | 0 | 0 | 1 | 1 |
| 2 | 1 | 4 | 0 | 0 | 1 | 1 | 1 | 1 |
| 3 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 4 | 1 | 1 | 1 | 1 | 0 | 0 |
| 5 | 0 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| 6 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 1 |
| 7 | 1 | 4 | 0 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 4 | 1 | 1 | 1 | 1 | 0 | 0 |
| 9 | 0 | 4 | 1 | 1 | 0 | 0 | 1 | 1 |
| 10 | 0 | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 11 | 0 | 3 | 1 | 1 | 1 | 0 | 0 | 0 |
| 12 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Table 3 Results of Simulation 1 in CAAS

Simulation 2: Node:2, ntasks:10, CPUs-per-task:3, ntasks-per-node:5 (rows: 3 cols:3 ports:3 cycle:12)

| Simulation 2 | | Row | 3 | C.S | 9 | | | | | | |
|--------------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | Column | 3 | | | | | | | | |
| | | Ports | 3 | | | | | | | | |
| | | Time | 12 | | | | | | | | |
| Time | Report | NFS | ETS 0 | ETS 1 | ETS 2 | ETS 3 | ETS 4 | ETS 5 | ETS 6 | ETS 7 | ETS 8 |
| 1 | 2 | 6 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 6 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 3 | 0 | 4 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 4 | 0 | 3 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 5 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 6 | 1 | 5 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 7 | 0 | 4 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 8 | 0 | 4 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 9 | 0 | 3 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 11 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 12 | 0 | 4 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |

Table 4 Results of Simulations 2 in CAAS

Tables 3 and 4 represent the results of simulations in CAAS and it had something different from the local device as we need to run the code using "srun" instead of using "mpirun". Using "srun" we will need to specify how many nodes, tasks, CPUs per task and tasks per node in the job files.

Time Taken Results (in nanoseconds)

Summary of the time for checking neighbours and reporting to the base station in the table.

1. Checking Neighbours

| Time checking neighbour | | | | |
|-------------------------|-----------|-----------|-----------|-----------|
| Time | Station 0 | Station 1 | Station 2 | Station 3 |
| 1 | 0 | 0 | 0.000889 | 0.000779 |
| 2 | 0.000636 | 0.00082 | 0.000806 | 0.000856 |
| 3 | 0.000986 | 0.000831 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0.001106 |
| 5 | 0 | 0.000889 | 0 | 0.000961 |
| 6 | 0 | 0.000842 | 0.004838 | 0.00079 |
| 7 | 0.000643 | 0.000942 | 0.0013 | 0 |
| 8 | 0.001332 | 0 | 0 | 0.001254 |
| S9 | 0 | 0.001027 | 0.000897 | 0.000596 |
| 10 | 0.000884 | 0.001417 | 0.000888 | 0 |
| 11 | 0.001278 | 0 | 0 | 0.001205 |
| 12 | 0 | 0.001177 | 0.001174 | 0.000626 |

Table 5.1 Time checking neighbours (Simulation 1 in local device)

| Time checking neighbour | | | | | | |
|-------------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Time | Station 0 | Station 1 | Station 2 | Station 3 | Station 4 | Station 5 |
| 1 | 0.000633 | 0.001232 | 0.001715 | 0.00064 | 0.002387 | 0 |
| 2 | 0.000832 | 0.001225 | 0.001111 | 0 | 0 | 0 |
| 3 | 0.001323 | 0 | 0 | 0 | 0.001535 | 0.000837 |
| 4 | 0 | 0 | 0.001077 | 0.001125 | 0.001408 | 0.000717 |
| 5 | 0.000428 | 0.002734 | 0.00067 | 0.001206 | 0.001052 | 0.00111 |
| 6 | 0.00064 | 0.000997 | 0.000633 | 0.000458 | 0.000978 | 0.000776 |
| 7 | 0.000669 | 0.000968 | 0.000438 | 0.000536 | 0.001357 | 0.000453 |
| 8 | 0.000584 | 0.000899 | 0.001356 | 0.000634 | 0.001254 | 0.000683 |
| 9 | 0.000857 | 0.00111 | 0.001196 | 0.000641 | 0.001482 | 0 |
| 10 | 0.000931 | 0.001295 | 0.000604 | 0 | 0 | 0.000865 |
| 11 | 0.001038 | 0 | 0 | 0.000694 | 0.001601 | 0.000995 |
| 12 | 0 | 0.001599 | 0.00087 | 0.001193 | 0 | 0 |

Table 6.1 Time checking neighbours (Simulation 2 in local device)

| Time checking neighbours | | | | | | |
|--------------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Time | Station 0 | Station 1 | Station 2 | Station 3 | Station 4 | Station 5 |
| 1 | 0.000023 | 0.000076 | 0 | 0 | 0.000141 | 0.000115 |
| 2 | 0 | 0 | 0.000142 | 0.001704 | 0.00165 | 0.000584 |
| 3 | 0.000017 | 0.000086 | 0.000556 | 0.000574 | 0.000625 | 0.000349 |
| 4 | 0.000018 | 0.00007 | 0.000102 | 0.000221 | 0 | 0 |
| 5 | 0.000018 | 0.000288 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0.000248 | 0.000312 |
| 7 | 0 | 0 | 0.000207 | 0.000336 | 0.000131 | 0.000164 |
| 8 | 0.000016 | 0.000106 | 0.000182 | 0.00019 | 0 | 0 |
| 9 | 0.000012 | 0.000235 | 0 | 0 | 0.000171 | 0.000195 |
| 10 | 0 | 0 | 0.000302 | 0.000226 | 0.000111 | 0 |
| 11 | 0.000026 | 0.000203 | 0.000276 | 0 | 0 | 0 |
| 12 | 0.000024 | 0 | 0 | 0 | 0 | 0 |

Table 7.1 Time checking neighbours (Simulation 1 in CAAS)

| Time checking neighbours | | | | | | | | | |
|--------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Time | Station 0 | Station 1 | Station 2 | Station 3 | Station 4 | Station 5 | Station 6 | Station 7 | Station 8 |
| 1 | 0.000018 | 0.000019 | 0.00038 | 0.000442 | 0.000442 | 0 | 0 | 0 | 0.000039 |
| 2 | 0.000014 | 0.000017 | 0.000108 | 0 | 0 | 0 | 0.000322 | 0.000173 | 0.000012 |
| 3 | 0.000022 | 0 | 0 | 0 | 0.000119 | 0.000161 | 0.000353 | 0 | 0 |
| 4 | 0 | 0 | 0.000095 | 0.000245 | 0.000229 | 0 | 0 | 0 | 0 |
| 5 | 0.000026 | 0.000022 | 0.000374 | 0 | 0 | 0 | 0 | 0.00029 | 0.000016 |
| 6 | 0.000016 | 0 | 0 | 0 | 0 | 0.000162 | 0.000148 | 0.000107 | 0.000013 |
| 7 | 0 | 0 | 0 | 0.00013 | 0.000177 | 0.000149 | 0.000255 | 0 | 0 |
| 8 | 0 | 0.000023 | 0.000105 | 0.000416 | 0.000195 | 0 | 0 | 0 | 0 |
| 9 | 0.000017 | 0.000015 | 0.000265 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0.000015 | 0 | 0 | 0 | 0 | 0 | 0 | 0.000387 | 0.00002 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0.000107 | 0.000237 | 0.000176 | 0 |
| 12 | 0 | 0 | 0 | 0.00011 | 0.000106 | 0.000375 | 0 | 0.00011 | 0 |

Table 8.1 Time checking neighbours (Simulation 2 in CAAS)

The tables above show that each time station took how long to communicate with their neighbours. If the time is zero means their ports are sufficient and don't communicate with the neighbours.

2. Reporting to the Base station

| Time reporting to base | | | | |
|------------------------|-----------|-----------|-----------|-----------|
| Time | Station 0 | Station 1 | Station 2 | Station 3 |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0.005711 | 0.010847 | 0.004838 | 0.012791 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0.004161 |
| 7 | 0.005861 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0.004257 |
| 10 | 0.007123 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0.005124 |

Table 5.2 Time reporting to base (Simulation 1 in local device)

| Time reporting to base | | | | | | |
|------------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Time | Station 0 | Station 1 | Station 2 | Station 3 | Station 4 | Station 5 |
| 1 | 0.00564 | 0.008946 | 0 | 0.007642 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0.005891 |
| 5 | 0.024307 | 0.021105 | 0.01363 | 0.025517 | 0.020019 | 0.007677 |
| 6 | 0.004857 | 0.005141 | 0.005829 | 0.004591 | 0.005965 | 0.005176 |
| 7 | 0.00623 | 0.006054 | 0.006449 | 0.0059 | 0.006037 | 0.006567 |
| 8 | 0.00551 | 0.005858 | 0.004764 | 0.005531 | 0.004982 | 0.005301 |
| 9 | 0.006619 | 0.006435 | 0 | 0.007645 | 0 | 0 |
| 10 | 0 | 0 | 0.004548 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0.005213 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 6.2 Time reporting to base (Simulation 2 in local device)

The table above shows how long it takes to report and receive the data from the base station.

| Time reporting to base | | | | | | |
|------------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Time | Station 0 | Station 1 | Station 2 | Station 3 | Station 4 | Station 5 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0.000247 |
| 3 | 0.000732 | 0.000762 | 0.000788 | 0.000387 | 0.000848 | 0.000506 |
| 4 | 0.00055 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0.004857 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0.000329 |
| 8 | 0.000606 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 7.2 Time reporting to base (Simulation 1 in CAAS)

| Time reporting to base | | | | | | | | | |
|------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Time | Station 0 | Station 1 | Station 2 | Station 3 | Station 4 | Station 5 | Station 6 | Station 7 | Station 8 |
| 1 | 0.001483 | 0.000694 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.000234 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 8.2 Time reporting to base (Simulation 2 in CAAS)

C. Analysis & Discussion

Analysis of different stations sending reports to the base station.

Based on Reports from [Reporting to the base station](#), the time taken for each station to get the message back from the base station has small increases for each station to report at the same time period where the first reported station will take the least time and the last station that reports to the base will take the longest time. Example from [Table 7.2](#) time 3, the first reported station is Station 3 which has a time of 0.000506 ns and the last station reported is Station 4 which has a time of 0.000848ns. This is due to each station sending reports to the base station at the same time, but the base station was checking them 1 by 1. But at the same time since we will save the reported station in that time, the other stations that are reporting at that time will take less time to get the message back from the base station.

Analysis of the effect of different numbers of neighbour

In this assignment, we are asked to build charging stations that can communicate with the 4 adjacent neighbours. Theoretically, the time taken for a station with a higher number of neighbours will consume more time but eventually based on the results some of the run times of stations with 4 neighbours run faster than stations with fewer neighbours. For example, refer to [Table 8.1](#) time 8, Station 4 in a 3*3 grid is the centremost station with 4 neighbours it runs 0.000195 ns. But station 3 which has 3 neighbours runs for 0.000105ns. This situation is probably because station three might be checking for some stations where the station is checking for the other stations.

Comparison of local device and CAAS platform.

To compare the local device and CAAS platform, I will focus on the results of [Checking Neighbours](#) since it will be more specific to observed rather than the results of reporting to the base station since it will be affected by the time for checking neighbours. Based on Tables [6.1](#) and [7.1](#), these two tables indicate the grid size of 2 rows * 3 columns but they are run on different platforms 6.1 runs in local devices and 7.1 runs in CAAS. By having a quick look at all the data in the tables we can find out that the data from the local device takes more time than CAAS. For example, by taking the results of stations 2 and 3 in time 4 we can observe that CAAS has an approximate 10 times speed up.

Discussion

For further improvement in the WSN-designed EV charging station, we can separate the base station task into some small base stations that may be controlling the message of different rows and columns, and these small base stations will summarize the reports to the base. This would shorten the communication time for every station to wait for other stations to communicate with the main base station.

D. References

Hoefler, T., Rabenseifner, R., Ritzdorf, H., de Supinski, B. R., Thakur, R., & Träff, J. L. (2011). The scalable process topology interface of MPI 2.2. *Concurrency and Computation: Practice and Experience*, 23(4), 293-310. <https://onlinelibrary.wiley.com/doi/full/10.1002/cpe.1643>