



FIT3143 - ASSIGNMENT 2

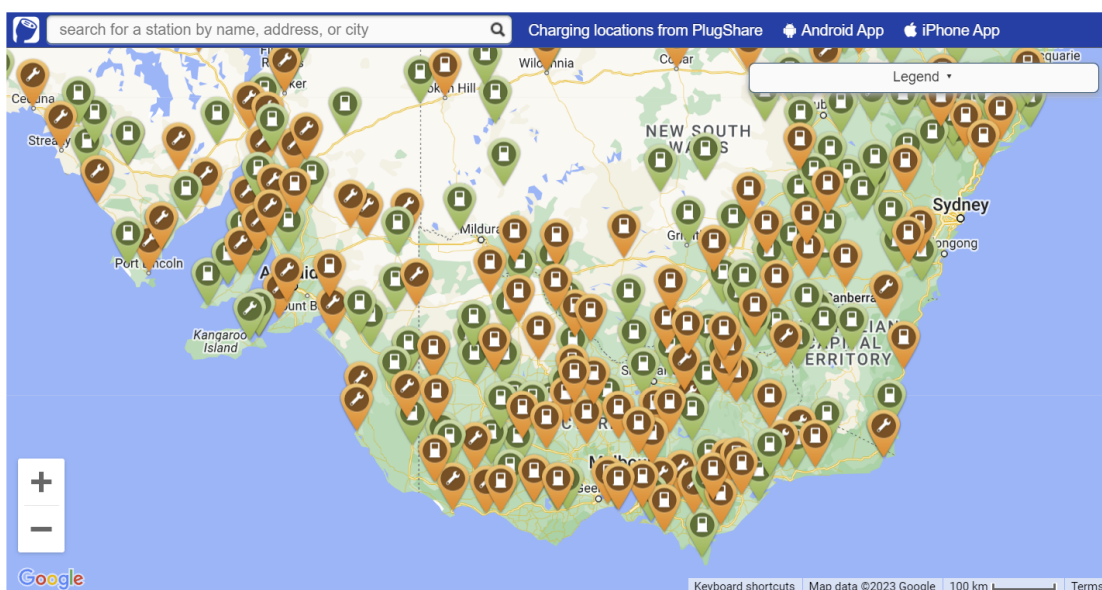
ELECTRIC VEHICLE CHARGING GRID: OPERATED BY A DISTRIBUTED WIRELESS SENSOR NETWORK (WSN)

Unit learning outcomes (LO) for this assignment:

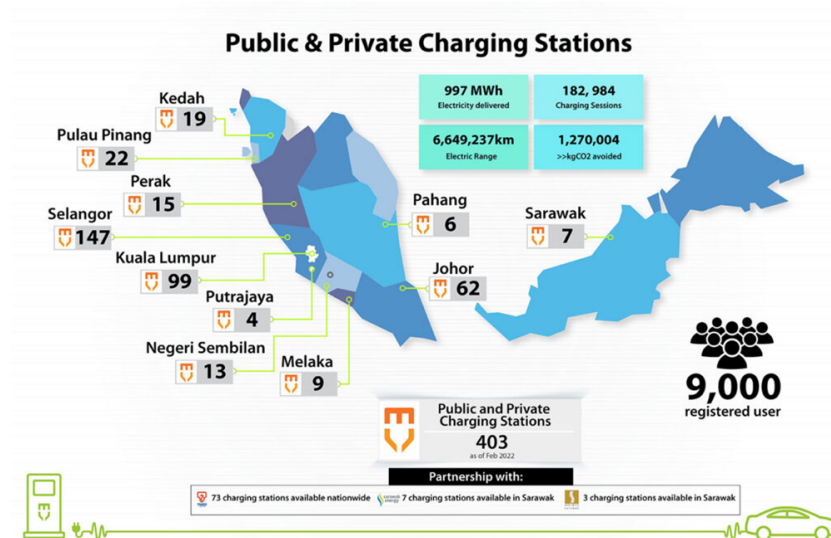
- a) Design and develop parallel algorithms for various parallel computing architectures (LO3).
- b) Analyse and evaluate the performance of parallel algorithms (LO4).
- c) Apply technical writing and presentation to effectively communicate parallel computing to a range of academic and expert audiences (LO5).

I. Preamble

As the electric vehicle (EV) market continues to grow, the availability and accessibility of EV charging stations become critical factors in shaping the adoption and convenience of electric vehicles. EV charging station navigation refers to the ability to efficiently locate and navigate to nearby charging stations. Figures 1a and 1b illustrate the distribution of the EV charging stations in Victoria and in Malaysia. Effective navigation systems hold significant importance for optimal route planning and provide seamless urban mobility, thus contributing to the growth of the EV market and the transition to sustainable transportation.



(a)



(b)
 Figure 1. Number of charging stations available in (a) Victoria and in (b) Malaysia. Images sourced from <https://myelectriccar.com.au/>
<https://www.mgtc.gov.my/what-we-do/low-carbon-mobility-2/chargev/>

To ensure the optimal selection of electric car charging ports, an Electric Vehicle Charging Navigation System (EVCNS) Center will be coordinating the selection of the EV charging stations based on the real time information received such as the availability of charging stations and distance from the users.

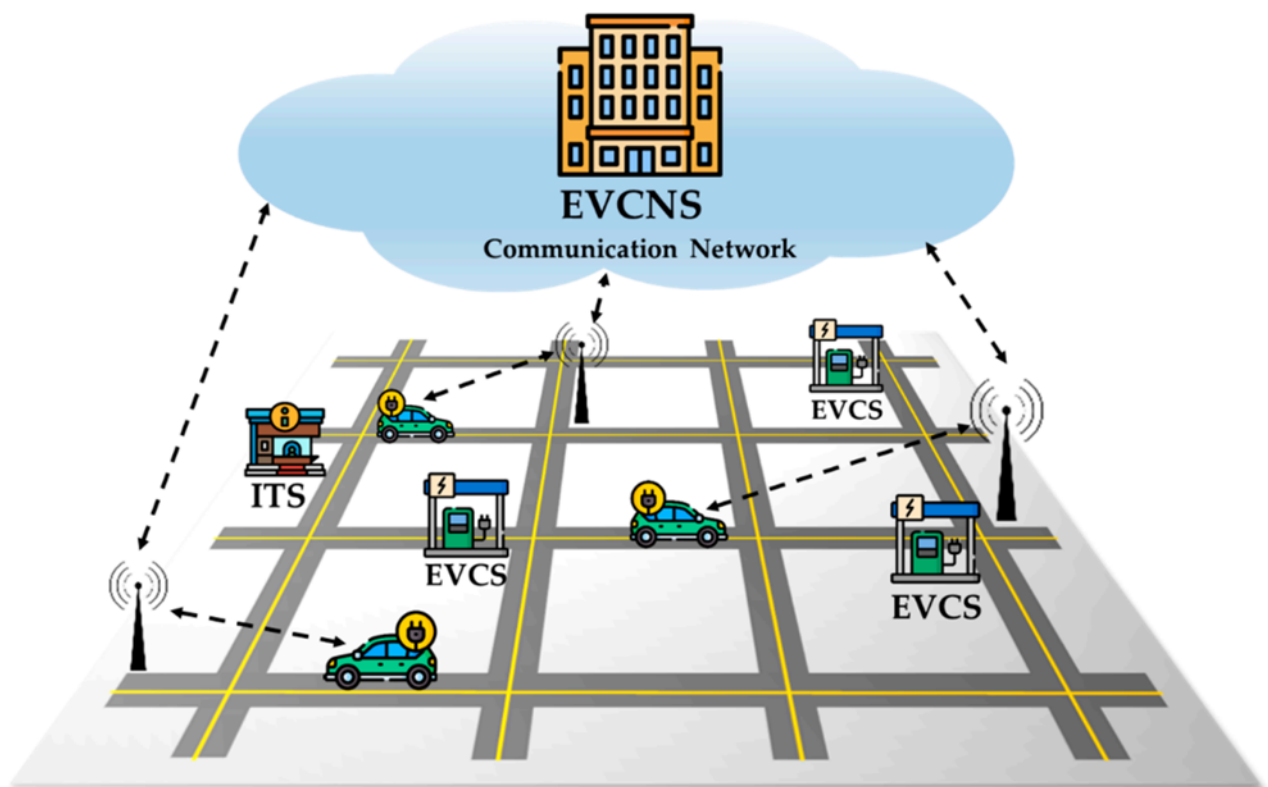


Figure 2. An electric vehicle charging navigation system (EVCNS) which consists of the EV charging station (EVCS) and intelligent transport system (ITS) that monitors the real-time traffic condition. Image sourced from <https://www.mdpi.com/1996-1073/13/23/6255>

Figure 2 illustrates an EV charging navigation system which consists of an intelligent transport system (ITS) and the EV charging stations. Data collected from EV charging stations could be transmitted to nearby EV charging navigation system centers or to a base station for further processing.

II. Objective of this assignment

In this assignment, we are attempting to simulate a wireless sensor network (WSN) of a set of interconnected EV charging nodes. Figure 3 illustrates an example overview of the WSN. In this figure and for the purpose of illustration, nine EV charging nodes are positioned in a 3×3 Cartesian grid layout. The number of actual nodes may vary based on the area being observed. The charging nodes are positioned such that each node communicates with its immediate adjacent nodes (e.g., up, down, left and right in the Cartesian grid layout). Each EV charging node has a number of in use or free charging ports. If all ports (or almost all ports) are in full use, the node will prompt for neighbour node data. If the received data from the neighbours also show that their ports are heavily utilized, then the reporting node will alert the base station that the node and its quadrant are all being used up. The base station receives the data from the nodes and logs the data.

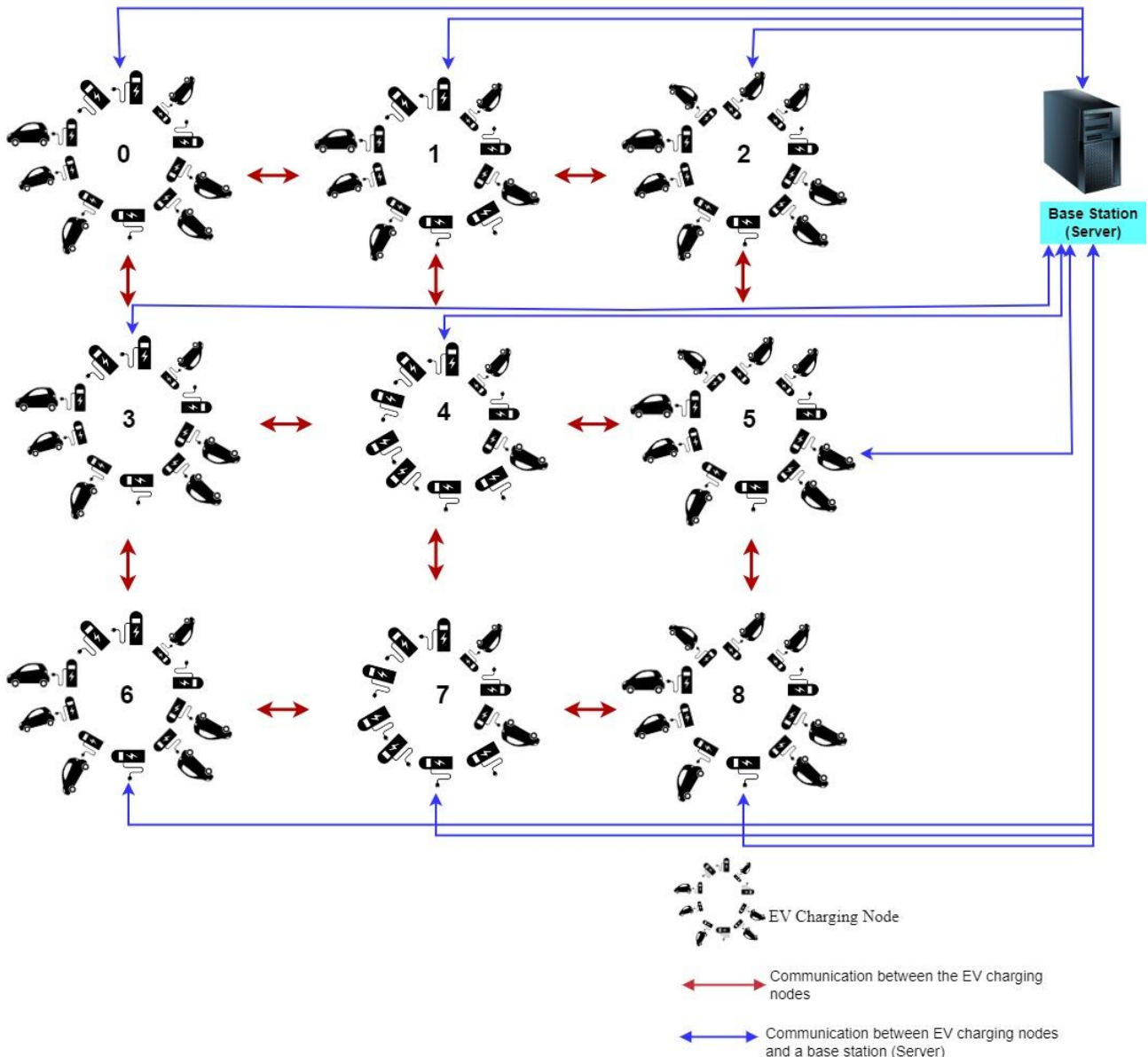


Figure 3. Overview of the wireless sensor network as the electric charging grid. In this illustration, the EV charging nodes are positioned in a 3×3 grid configuration. Each EV charging node has a number of in use or free charging ports. Each EV charging node communicates between the neighbours and the base station.

In addition, the base station compares the received data from the charging nodes. If the node and its quadrant are all being used up, the base station sends the data of the next nearest available charging nodes to the comparing node.

The aim of this individual assignment is to simulate the aforementioned setup. A simulation model needs to be designed and implemented using a combination of Message Passing Interface (MPI) and POSIX/OpenMP in C or C++ programming language. Detailed specifications are in the subsequent sections.

Objectives of the assignment:

- To design and implement a distributed computing system based on simulating a wireless sensor network;
- To design and implement parallel algorithms in a distributed computing system;
- To analyse and evaluate the performance of the parallel algorithms and approaches in a distributed computing system;
- To apply technical writing to effectively communicate parallel computing.

III. WSN Description

The deployed wireless sensor network comprises $m \times n$ nodes in a cartesian grid, and a base station. In the example illustration in Figure 3, $m = 3$ and $n = 3$. However, the values of m and n vary as specified by the user during program runtime. Each node in this cartesian grid simulates the behaviour of the EV charging node, and each node communicates with its immediate adjacent nodes. These nodes can exchange data through both unicast and broadcast modes of communications.

Each node in the WSN can independently exchange data with the base-station. Base-station for this WSN is an additional computer node that is entrusted with the task of gathering data from all charging nodes in the WSN. All communications among nodes, and between the nodes and the base station are achieved using Message Passing Interface (MPI). In essence, each charging node and base station can be represented as an MPI process.

1.0) Simulating the EV charging node:

- A single MPI process simulates one EV charging node. For instance, in a 3×3 Cartesian grid layout, there will be 9 MPI processes. Each node will have a (x, y) coordinate (e.g., Node 0 - (0,0), Node 1 - (0,1), Node 2 - (0,2), ... Node 8 (2, 2)). It will be good to use MPI virtual topologies here.
- Each EV charging node has k number of in use or free charging ports simulated with POSIX threads (you may opt to use OpenMP as an alternative to POSIX thread). Each thread periodically updates its value, indicating its availability. The following table show the sample charging ports availability of an EV charging node, where $k=5$ and simulated at a cycle of 10 seconds:

Table 1. Sample of charging ports availability for an EV charging node with 5 ports

YYYY	MM	DD	HH	MM	SS	Availability
2023	09	07	10	01	10	1
2023	09	07	10	01	20	2
2023	09	07	10	01	30	4
2023	09	07	10	01	40	5
2023	09	07	10	01	50	5

Note: You can set your own cycle (e.g., 2 seconds or 5 seconds or 10 seconds, etc.)

- c) The information in (b) is stored in a shared array, which can also be accessed by all threads within the same node. The array has a fixed size, and you can decide the size of this array. Once the array is full, the thread removes the first entered data from the array to make way for the new data (first in, first out approach).
- d) If all ports (or almost all ports) are in full use, the node will prompt for neighbour node data. To reiterate, the neighbour nodes refer to immediate top, bottom, right and left adjacent nodes (if exist).
Note: You can define the number of cycles for the port to remain occupied and also the number of ports in use to trigger the request from the neighbour node.
- e) If the received data from the neighbours shows a vacancy on the respective neighbour nodes, the reporting node shall indicate the nearest available neighbour nodes. If the received data also show that all of their ports are heavily utilized, then the reporting node will alert the base station that the node and its quadrant are all being used up.
- f) The report sent to the base station should contain as much information as possible about the alert. This information includes the timestamp (including the date & time) at which a fully occupied charging node (including its quadrant) is detected and number of messages exchanged with the neighbour nodes. You should demonstrate efficiency when reporting an alert message to the base station. In this context, you should minimize the number of calls to the MPI Send (or ISend) functions by a node to the base station when reporting an alert condition.
- g) The node should receive a message from the base station about the nearest available EV charging nodes apart from its neighbour nodes.
- h) Each node repeats parts (a) to (g) until upon receiving a termination message from the base station. Once the node receives a termination message, the node cleans up and exits.

2.0) Simulating the base station

- a) A single MPI process simulates the base station node.
- b) The base station node periodically listens for incoming reports from the EV charging nodes.
- c) Upon receiving a report from an EV charging node, the base station checks for the nearest neighbour nodes that are available based on the neighbouring nodes of the reporting node. For instance, if the reporting node is Node 0, the neighbour nodes are Node 1 and 3. The base station will check whether there's any reports received from Node 1 and 3 (i.e., Node 2, 4 and 6):
 - If no report is received from the nodes within a predefined period, the base station will suggest the available nodes to the reporting node.
 - If there are reports received from all of the nodes within a predefined period, the base station will send a message notifying the reporting node that there are no available nodes nearby.**Note:** You can define the time frame of the predefined period.
- d) The base station writes (or logs) the key performance metrics (e.g., the simulation time, number of alerts detected, number of messages/events with senders' adjacency information/addresses, suggested available nodes and the total number of messages (for this simulation)) to an output file. Doing so may assist in proving the correctness and efficiency of your implementation.
- e) The base station has an option to send a termination message to the EV charging nodes to properly shutdown for maintenance.
- f) If you are aiming for HD or upper HD: The base station uses a thread (i.e., POSIX or OPENMP) to send or receive MPI messages from the EV charging nodes. This thread is created by the base station and terminates properly at the end of the program.

Important: The aforementioned description represents the baseline in meeting the requirements of the assignment.

IV. What needs to be done

A) **Programming Task:**

- a) Design, write and demonstrate an MPI + thread program that effectively addresses all the points in Section III (WSN Description). Assume that a set of MPI processes represents the WSN and each MPI process represents a WSN node and the base station.

B) **Report**

- a) Methodology – Description of the methods and algorithms used to simulate the EV charging nodes and base station. Include illustration to describe the design, and flowchart (or pseudocode) of the algorithm(s).
- b) Results tabulation – Tabulation or illustration of the results using tables, graphs, or screenshots of the log file content.
- c) Analysis & discussion - Provide an analysis on the behaviour of the simulator. Describe the content of the messages which were exchanged between the nodes and messages that were sent to the base station. Include any additional observations to support the analysis.
- d) Maximum number of words in the report: 1,500 words. The word count excludes figures, tables, and references (if any).

C) **Code demonstration and interview**

- a) Demonstrate the program during the lab session (after the assignment submission due date). Exact dates will be announced in Moodle and/or Ed Discussions.
- b) Interview by the teaching staff to ascertain the authenticity of the work completed by the student.

V. Submission

Individual Assignment: Yes. As part of the authentic assessment structure for technical and taught based units, each student is to work individually in this assignment. While you may have discussions with your peers about the assignment, you must produce and submit your own work. Submission of the assignment is made via Moodle.

The following files are required for submission to Moodle:

1) **A compressed file (i.e., zipped) containing the following:**

- a) Code file(s) (.h and .c/.cpp file extensions)
- b) Log file(s)
- c) Any other files (e.g., Makefile) which may support your assignment work.

2) **The Assignment report (in PDF). Please do not compress this file. Drag and drop the PDF report into the submission area.**

Turnitin similarity check is enabled for the report submission. Once you upload a report as a draft, Turnitin will check for similarity. Based on the similarity report, you can amend your report to reduce its similarity index and reupload your report. Therefore, it is important that you do not compress your report into a zipped file. Once you submit your report for grading, you are not able to make any amendments to it. Please contact your tutor if you would like your submission to be reverted to draft mode.

The Assignment-2 submission is due on Friday, Week 11 of the semester (11:55 PM Clayton/Malaysia time). All demo and interviews will be scheduled in Week 12 (unless special consideration is granted or an alternative arrangement is made).

VI. Provisional Marking Guide

This assignment is worth 25 marks (or 25 percentage points) of your overall unit score.

Part A: Assignment Code and Demonstration (17 marks)

1.0 Simulating the EV charging nodes	10 marks
2.0 Simulating the base station	7 marks

Part B: Assignment Report (8 marks)

Methodology	3 marks
Results tabulation	2 marks
Analysis and discussion	3 marks

Note: Penalty (i.e., marks deduction) will be applied for poor grammar, high similarity index for the report and late submission. The marking guidelines contains further breakdown of the marks.

VII. Academic Integrity & Generative AI

Please do not attempt to plagiarise or collude your assignment work. Do not plagiarise! We will check your work against your peers, and also submissions made in previous years.

Please refer to the University's [policy](#) on academic integrity.

Generative AI tools cannot be used for any assessments in this unit.

In this unit, you must not use generative artificial intelligence (AI) to generate any materials or content in relation to your assessment.