

## SEISMIC READINGS FOR EARTHQUAKE DETECTION IN A DISTRIBUTED WIRELESS SENSOR NETWORK (WSN)

### OVERVIEW:

#### I. Preamble

A seismometer represents an instrument which responds to ground noises and shaking such as caused by earthquakes, volcanic eruptions, and explosions. These sensors are typically combined with a timing device and a recording device to form a seismograph. Seismometers play a crucial role in seismology, which is a scientific study of earthquakes. Figure 1 illustrates a screenshot of seismic activities which were recorded along Australia/New Zealand in August/September 2022. These activities provide vital information to governments on early warning of earthquakes. This [link](http://ds.iris.edu/seismon/zoom/index.phtml?rgn=Australia_NZ) contains latest earthquakes in the Australia NZ Region as a reference on the type of data produced by these sensors (i.e., latitude, longitude, magnitude, depth, etc.)

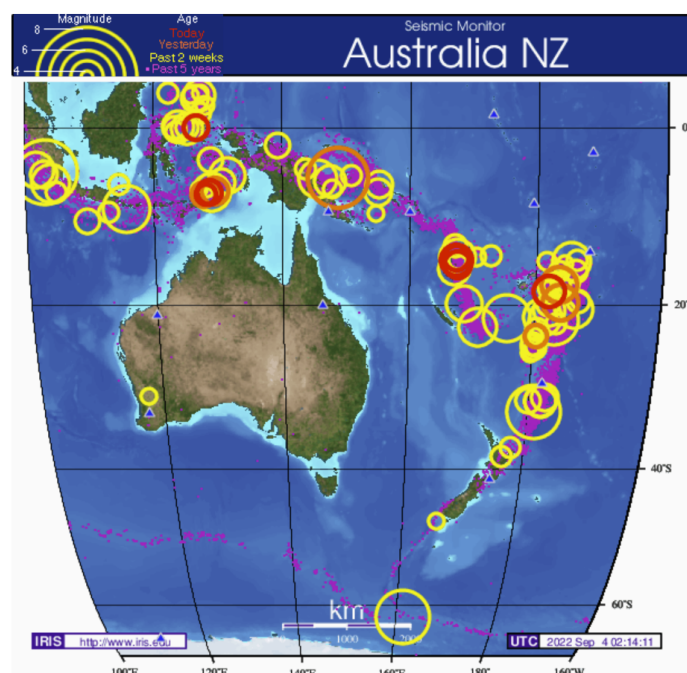


Figure 1. Seismic activities which were recorded along Australia/New Zealand on August/September 2022. Image sourced from [http://ds.iris.edu/seismon/zoom/index.phtml?rgn=Australia\\_NZ](http://ds.iris.edu/seismon/zoom/index.phtml?rgn=Australia_NZ)

However, the location or placement of seismometers are crucial as these sensors are sensitive towards noise and require careful calibration. As such, in an attempt to accurately detect underwater seismic activities, scientists have set up seafloor based seismic sensors.

Figure 2 illustrates a typical seafloor based seismic sensor setup, which consists of a surface buoy and a seafloor ocean bottom seismometer. Data collected from a seismometer could be transmitted to nearby seismometer or to a ground (or base) station for further processing.

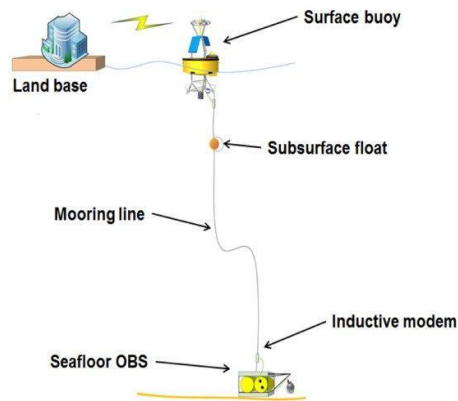


Figure 2. A seafloor seismic sensor setup which consists of a seafloor ocean bottom seismometers (OBS) sensor and a surface buoy.  
Image sourced from <https://www.mdpi.com/1424-8220/18/4/1132/htm>

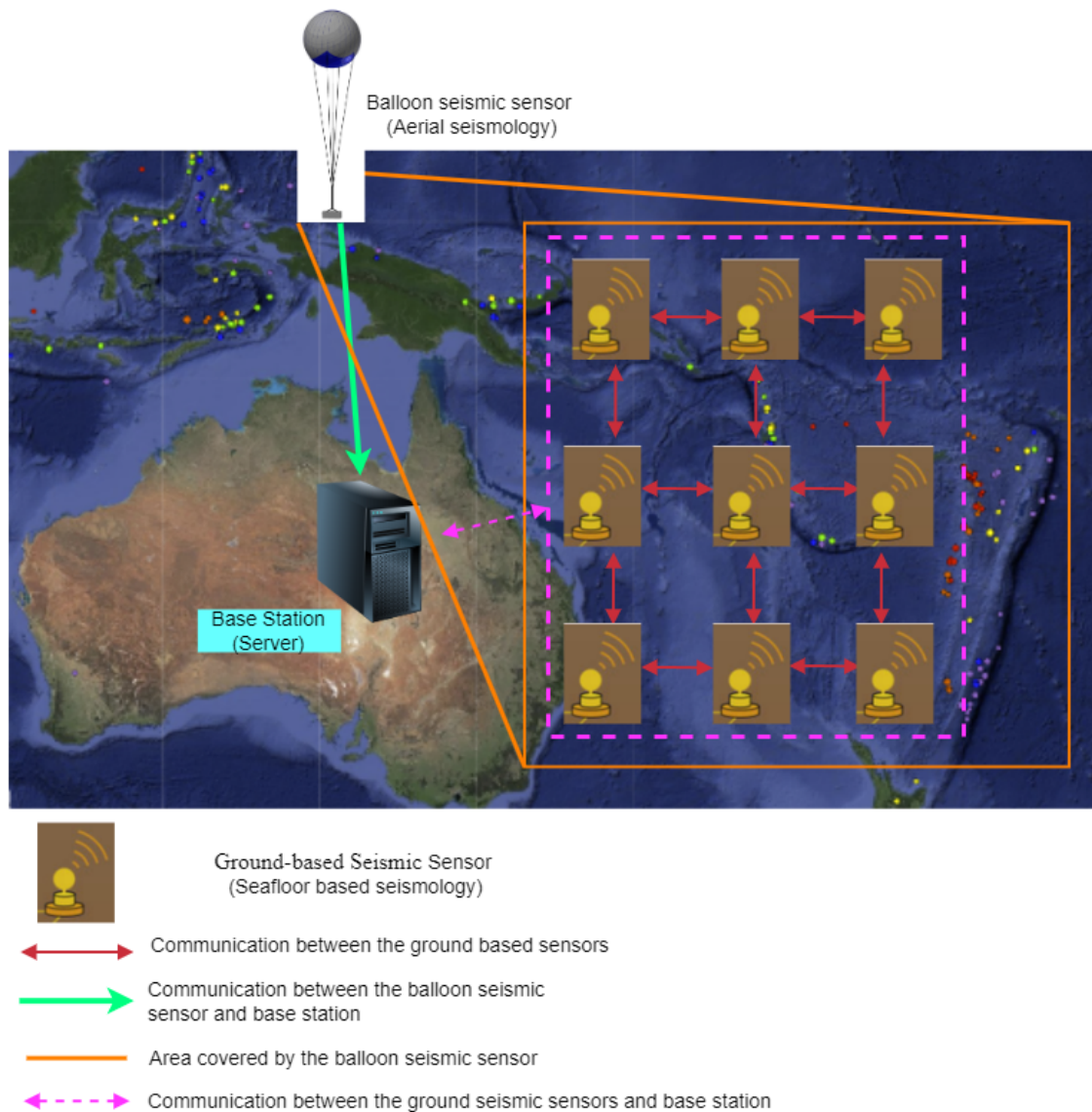


Figure 3. Overview of the wireless sensor network to measure seismic readings in detecting earthquakes. In this illustration, the seafloor seismic sensors are positioned in a  $3 \times 3$  grid configuration. In addition, a balloon seismic sensor covers or observes the same grid layout where the seafloor seismic sensors are located. Note that balloon seismic sensor only communicates with the base station and does not communicate with any of the seafloor seismic sensors. The map and icons in this figure were sourced from: [http://ds.iris.edu/seismon/zoom/index.phtml?rgn=Australia\\_NZ](http://ds.iris.edu/seismon/zoom/index.phtml?rgn=Australia_NZ), <https://www.sciencedirect.com/science/article/pii/S0094576516302119> & <https://royalsocietypublishing.org/doi/epdf/10.1098/rsta.2011.0214>

In this assignment, you are required to simulate a wireless sensor network (WSN) of a set of interconnected seismic sensors. Figure 3 illustrates an example overview of the WSN and a balloon seismic sensor (aerial seismology). In this figure and for the purpose of illustration, nine seafloor based seismic sensors are positioned in a  $3 \times 3$  Cartesian grid layout. The number of actual sensors may vary based on the area being observed. The sensors are positioned such that each sensor communicates with its immediate adjacent sensors (e.g., up, down, left, and right in the Cartesian grid layout). All seafloor seismic sensors in this grid can communicate with the base station and vice-versa. In Figure 3, the magenta-coloured dotted box/arrow shows that the seismic sensors send their readings to the base station (two-way communication).

In addition, a balloon floating in the Earth's ionosphere observes the same Cartesian grid layout where the seafloor seismic sensors are currently positioned (as illustrated in Figure 3). This balloon uses a [barometer](#) to measure seismic readings, as part of a new method being researched to detect earthquakes using [aerial seismology](#). The base station compares the received data from the seafloor seismic sensors and seismic readings from the balloon. This comparison is done to ascertain if there is a conclusive or an inconclusive match between the information sent by the seafloor seismic sensors and the balloon.

The aim of this group (of two or three members) 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:

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

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).

## II. 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 2,  $m = 3$  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 seafloor seismic sensor, and each node communicates with its immediate adjacent nodes. These nodes can exchange data through *unicast* and *broadcast* modes of communications.

All nodes 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 sensor 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 sensor node and base station can be represented as a MPI process.

### 1.0) Simulating the seafloor seismic sensor node:

- A single MPI process simulates one sensor node. For instance, in a  $3 \times 3$  Cartesian grid layout, there will be 9 MPI processes. Each node will have a (x, y) coordinate (e.g., Node 1 - (0,0), Node 2 - (0,1), Node 3 - (0,2), ... Node 9 (2, 2)). You could use MPI virtual topologies here.
- Each node periodically produces a simulated sensor reading. This reading consists of date, time, latitude and longitude of the earthquake point, [magnitude](#), and [depth](#) of the quake from the sensor. You can [generate random float values](#) to simulate the sensor readings and you can refer to this [link](#) as a reference. The following table lists sample readings (simulated at a cycle of 10 seconds):

*Table 1. A list of simulated value from a seismic sensor node. These values are randomly generated.*

YYYY	MM	DD	HH	MM	SS	Lat	Long	Magnitude	Depth (km)
2022	09	05	10	01	10	-15.36	167.50	4.75	5.25
2022	09	05	10	01	20	-15.36	166.50	4.85	5.55
2022	09	05	10	01	35	-15.35	167.55	4.55	6.25
2022	09	05	10	01	40	-15.35	167.51	3.90	5.75
2022	09	05	10	01	40	-15.36	167.52	4.25	5.78

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

- If the generated magnitude at each time interval average exceeds a predefined [threshold](#) (e.g., magnitude > 2.5), this constitutes a possible earthquake. The node will then send a request to its immediate adjacent neighbourhood nodes to acquire their readings for comparison purposes. To reiterate, the neighbourhood nodes refer to immediate top, bottom, right and left adjacent nodes.
- Upon receiving the readings from its neighbourhood nodes, the node compares these readings to its own reading to check if the readings are similar. Comparisons here are based on:
  - Computing the absolute difference in latitude and longitude coordinates of reported seismic events between the nodes.
  - Computing the absolute difference in magnitude readings of reported seismic events between the nodes.

**Note:** You could also include depth as a third comparison factor.

- Should the difference in location and magnitude from at least two or more neighbourhood nodes match the readings of the local node (within a threshold which you can determine), the node sends a report (i.e., alert) to the base station. To compute the distance between two geographical coordinate points in C, you may refer and use the code in this [link](#) with proper citation to the code in your report.
- The report sent to the base station should contain as much information as possible about the possible alert. This information includes the timestamp (including the date & time) at which an alert is detected, sensor value readings (e.g., magnitude, depth, location) and number of messages compared with the neighbourhood 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.
- 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.
- Simulation should work for dynamic value of m x n nodes and threshold settings. At start up, the program allows the user to specify the grid size (m x n) and threshold values.
- If you are aiming for HD or upper HD: The node uses a thread (i.e., POSIX or OPENMP) to send or receive MPI messages between its adjacent nodes. This thread is created by the sensor node and terminates properly at the end of the program.

## 2.0) Simulating the balloon seismic sensor

- A single POSIX thread simulates the balloon seismic sensor (you may opt to use OpenMP as an alternative to POSIX thread). Note that the balloon covers the same area of the seafloor seismic sensors. For simulation purposes, this thread is created by the base station node (refer to point 3.0 below for details about the base station).
- This thread periodically produces seismic readings. This reading consists of date, time, latitude and longitude of the earthquake point, magnitude, and depth of the quake from the sensor. However, the generated magnitude always exceeds the predefined threshold (e.g., magnitude > 2.5).
- The following table lists sample seismic readings produced by the thread (simulated at a cycle of 1 second) for the entire Cartesian grid layout.

*Table 1. A list of simulated value from the balloon seismic senso. These values are randomly generated.*

YYYY	MM	DD	HH	MM	SS	Lat	Long	Magnitude	Depth (km)
2022	09	05	10	01	1	-15.37	167.49	3.75	5.15
2022	09	05	10	01	2	-15.40	166.51	3.85	5.25
2022	09	05	10	01	3	-15.31	167.52	4.01	6.15
2022	09	05	10	01	4	-15.21	167.40	3.95	5.98

Note: You can set your own cycle (e.g., 2 seconds or 3 seconds or 5 seconds, etc.).

- The information in (c) is stored in a shared global array, which can also be accessed by the base station 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).
- The thread repeats (a) to (d) until upon receiving a termination signal from the base station. Once the thread receives a termination signal, the thread exits.

## 3.0) Simulating the base station

- A single MPI process simulates the base station node.
- The base station node also creates the thread which simulates the balloon seismic sensor (refer to point 2.0 above).
- The base station node periodically listens for incoming reports from the seafloor seismic sensor nodes.
- Upon receiving a report from a sensor node, the base station compares the received report with the data in the shared global array (which is populated by the balloon seismic sensor).
  - If there is a match, the base station logs the report as a conclusive event (*conclusive* alert).
  - If there is no match, the base station logs the report as an inconclusive event (*inconclusive* alert).

**Note:** You can set your own threshold values when comparing the content of the received report and data in the shared global array. However, comparisons between the node report and shared global array should factor in the difference in location and magnitude. You could also include depth as a third comparison factor.

- 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, total number of messages (for this simulation)) to an output file. Doing so may assist in proving the correctness and efficiency of your implementation. You can also include IP addresses of the node and/or base stations, but including a loopback address (127.0.0.1) is not considered here.

- f) The base station program allows the user to specify a sentinel value at runtime to allow a proper shutdown of the base station, balloon sensor and sensor nodes. **Note:** CTRL+C is not allowed.
- g) Continuing from (f), the base station sends a termination message to the sensor nodes to properly shutdown. The base station also sends a termination signal to the balloon seismic sensor to properly shutdown.
- h) 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 sensor nodes. This thread is created by the base station and terminates properly at the end of the program.

#### **4.0) Fault detection (Only for teams of three members)**

- a) A fault detection mechanism is simulated whereby the sudden failure of a node is detected by the base station and/or adjacent sensor nodes.
- b) Once a faulty node is detected, a method is implemented to notify the base station to log the fault and to gracefully shutdown the entire network.

**Note: 4.0) Fault detection** is for teams of three members. If you belong to a team of two members, you are not required to implement this feature.

**Important:** The aforementioned description represents the baseline in meeting the requirements of the assignment. The marking rubric will include specific requirements to obtain a HD for a criteria in this assignment.

### **III. Submission:**

**Teamwork:** Yes. You will work in a team of two (or three) members to complete this assignment. Your team formation is based on the same team formation used in the weekly lab activities and in the first assignment. Each team member within a group will produce the same set of report, code files, log files and any other supplementary materials. However, each team member is required to make a submission independently in Moodle.

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

**Write-up Task – E-folio style report layout:**

- I. *Methodology* – Description of the methods and algorithms used to simulate the seafloor seismic sensor nodes, balloon seismic sensor and base station. Include illustration to describe the design, and flowchart (or pseudocode) of the algorithm(s).
- II. *Results tabulation* – Tabulation or illustration of the results using tables, graphs, or screenshots of the log file content.
- III. *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. Include any additional observations to support the analysis.

Maximum number of words in the report: 1,500 words for two member teams, 2,000 words for three member teams. The word count excludes figures, tables, and references (if any).

**Demo Interview Task:** The group needs to attend a demo and interview session held during Week 12 lab session. The interview is individually assessed.

Please refer to FAQ which are enclosed with the assignment specifications in Moodle. In addition, please refer to the assessment rubric which is enclosed together with this specification. This rubric should assist you to attain a better understanding of the marking guide for the assignment.

**Please do not attempt to plagiarise or collude your assignment work.**

Please refer to the University's [policy](#) on academic integrity. You are required to attach a group assignment coversheet with your submission. Faculty group assignment coversheets can be found [here](#). Please include the coversheet as the first page in your assignment report before submitting the report and code into Moodle.

Submission is to be made in Moodle for FIT3143. An assignment submission link will be provided in Moodle (under the Assessment section). **Although this is a group work (i.e., students within a team will share the report, code, and log files), each student is required to make a submission in Moodle.** The following files are required for submission to Moodle:

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

Please make sure the FIT group assignment coversheet is appended into the report (first page). Please refer to Question 17 in the FAQ for further details about appending the coversheet.

- 2) 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.

**2) 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.

**The Assignment-2 submission is due on Tuesday, Week 12 of the semester (11:55 PM Clayton/Malaysia time). All demo interviews must be completed during Week 12's Lab session.**

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

**PROVISIONAL MARKING GUIDE**

**Part A: Assignment Code and Demonstration (18 marks) – Two Member teams**

1.0 Simulating the seafloor seismic sensor nodes	6 marks
2.0 Simulating the balloon seismic sensor	2 marks
3.0 Simulating the base station	5 marks
Overall code structure, code comments, and running on CAAS	2 marks
Code demonstration and Q&A in Week 12 (Individually assessed)	3 marks

**Part A: Assignment Code and Demonstration (18 marks) – Three Member teams**

1.0 Simulating the seafloor seismic sensor nodes	5 marks
2.0 Simulating the balloon seismic sensor	2 marks
3.0 Simulating the base station	4 marks

4.0 Fault detection and response	2 marks
Overall code structure, code comments, and running on CAAS	2 marks
Code demonstration and Q&A in Week 12 (Individually assessed)	3 marks

**Part B: Assignment Report (7 marks) – All teams**

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

**Note:** Penalty (i.e., marks deduction) will be applied for poor grammar, high similarity index for the report and late submission. Please refer to the marking rubric for further details.