MONASH University

TSUNAMI DETECTION IN A DISTRIBUTED WIRELESS SENSOR NETWORK (WSN)

OVERVIEW:

I. Preamble

A tsunami represents a series of waves in a water body caused by the displacement of a large volume of water, generally in an ocean or a large lake. Their destructive power can be enormous and therefore various government and non-government organisations around the globe are working together to set up sea based wireless sensors to detect abnormal ocean wave heights (or sea water column height) which would indicate a tsunami. Such sensors are called tsunameters. The sea water column height readings from these sensors are then compared against similar readings from a satellite orbiting our planet. Should these readings be within a predefined threshold and time range, an alert is raised to the relevant authorities to evacuate coastal areas which could be affected by an oncoming tsunami.

Figure 1 illustrates a typical tsunameter setup, which consists of a surface buoy and a bottom pressure recorder. Data collected from a tsunameter could be transmitted to nearby tsunameter or to a satellite which relays the information to a ground station for further processing.

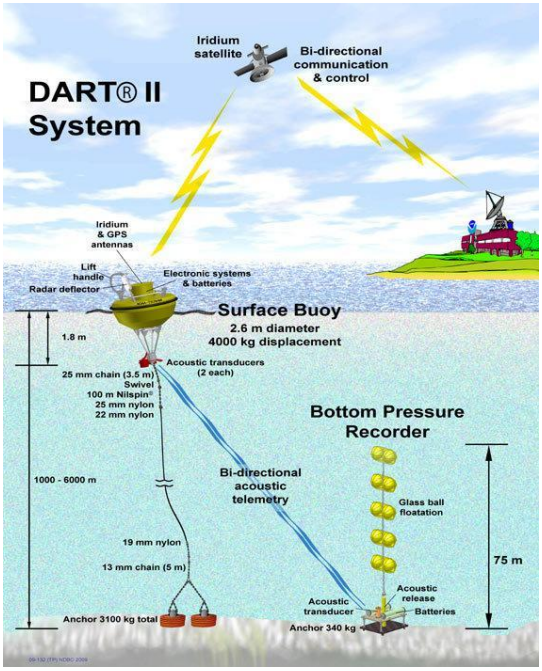

Figure 1. A tsunameter setup which consists of a surface buoy and a bottom pressure recorder. Image sourced from https://www.researchgate.net/publication/224119445_Performance_of_the_first_year_of_the_completed_US_operational_deep-sea_tsunameter_network

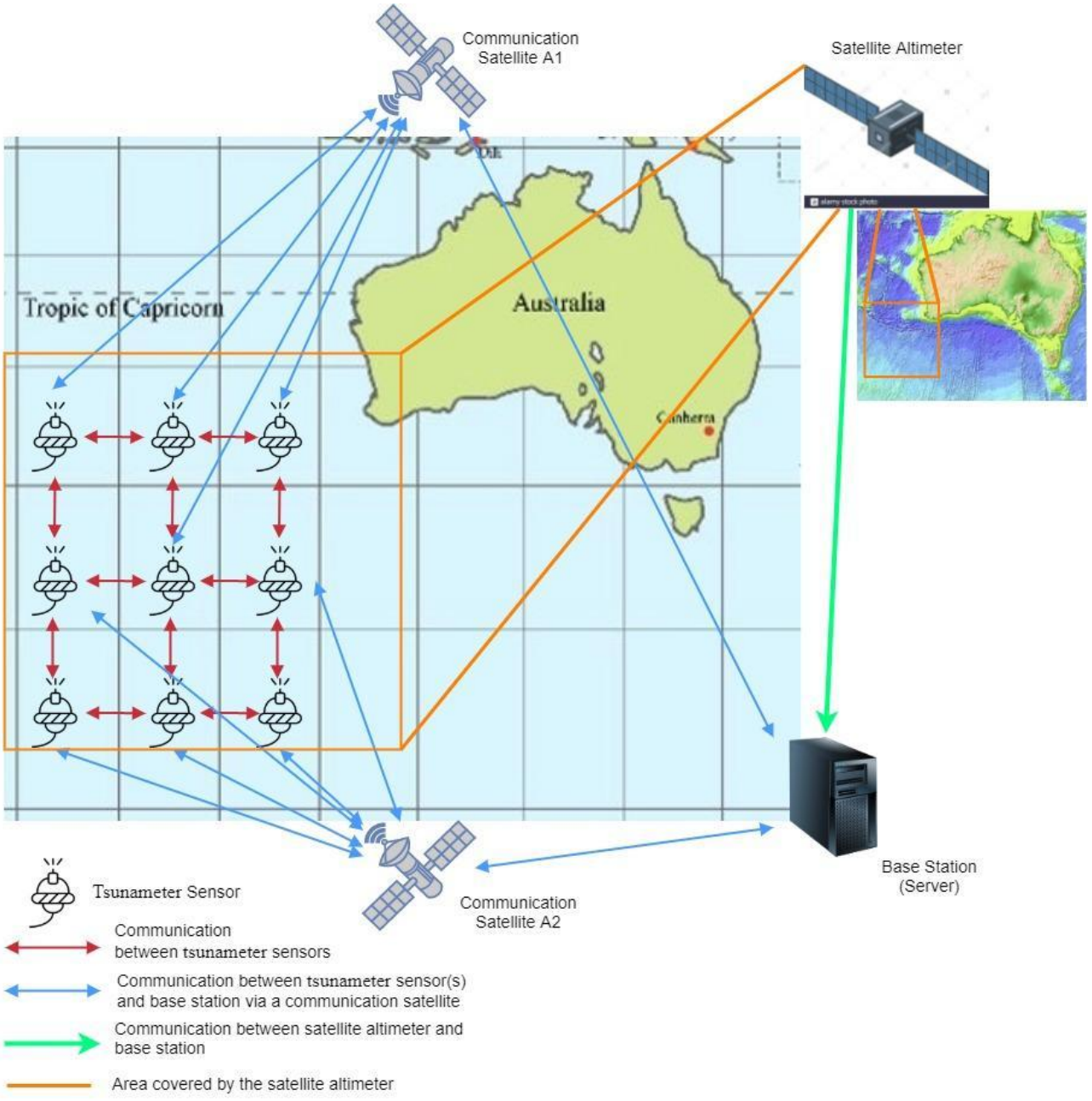

Figure 2. Overview of the wireless sensor network to detect a tsunami. In this illustration, the tsunameter sensors are positioned in a 3 × 3 grid configuration. Note that Communication Satellites A1 and A2 are actually the same satellite at different position. For the purpose of illustration, two communication satellites are drawn here to reduce the number of overlapping lines between the tsunameter sensors and the communication satellite. In addition, the satellite altimeter covers or observes the same grid layout where the tsunameter sensors are located. Note that satellite altimeter only communicates with the base station and does not communicate with any of the tsunameter sensors. The maps in this figure were sourced from: https://www.ergosum.co/cooking-across-latitudes/ & https://www.star.nesdis.noaa.gov/socd/lsa/AltBathy/

In this assignment, we are attempting to simulate a wireless sensor network (WSN) of a set of interconnected tsunameters. Figure 2 illustrates an example overview of the WSN and satellite network. In this figure and for the purpose of illustration, nine tsunameter sensors are positioned in a 3 × 3 Cartesian grid layout. The number of actual sensors may vary based on the area of the ocean 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). However, all the sensors can send and receive messages to and from a base station via a communications satellite.

In addition, a satellite altimeter orbiting the earth observes the same Cartesian grid layout where the tsunameter sensors are currently positioned (as illustrated in Figure 2). This satellite uses radar to periodically measure the sea water column height and only sends the data which exceeds a certain threshold to the base station. The base station compares the received data from the tsunameter sensors and satellite altimeter. This comparison is done to ascertain if there is a match or a mismatch between the information sent by the tsunameter sensors and the satellite altimeter.

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 computing 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* × *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 tsunameter sensor, and each node communicates with its immediate adjacent nodes. These nodes can exchange data through *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 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 tsunameter sensor node:

a) A single MPI process simulates one sensor 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 1 - (0,0), Node 2 - (0,1), Node 3 - (0,2), … Node 9 (2, 2)). It will be good to use MPI virtual topologies here.

b) Each node periodically produces a simulated sensor reading. This reading could be a simulated sea water column height. You can generate random float values to simulate the sensor readings. The following table lists sample sea water column height values (simulated at a cycle of 10 seconds):

Table 1. A list of simulated sea water column height value from a tsunameter sensor node

YYYY	MM	DD	HH	MM	SS	Height (m) – Randomly generated
2021	09	05	10	01	10	5632.858
2021	09	05	10	01	20	5635.950
2021	09	05	10	01	35	5639.125
2021	09	05	10	01	40	5620.189
2021	09	05	10	01	40	5610.105

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

c) The generated random values based on the simulation cycle are added to a simple moving average.
d) If the moving average exceeds a predefined threshold (e.g., > 6000 meters), this constitutes a possible event. 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.
e) Upon receiving moving average readings from its neighbourhood nodes, the node compares these readings to its own reading to check if the readings are similar.
f) Should the readings from at least two or more neighbourhood nodes match the reading of the local node (within a predefined tolerance range, e.g., 100 meters or more), the node sends a report (i.e., alert) to the base station.
g) 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., current moving average sea water column height value) 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.
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 satellite altimeter

a) A single POSIX thread simulates the satellite altimeter (you may opt to use OpenMP as an alternative to POSIX thread). For simulation purposes, this thread is created by the base station node (refer to point 3.0 below for details about the base station).
b) This thread periodically produces random sea water column height readings which exceed the predefined threshold (e.g., > 6000 meters). This thread also generates random coordinates within the range of the cartesian grid layout.
c) The following table lists sample sea water column height values produced by the thread (simulated at a cycle of 1 second) for the entire Cartesian grid layout.

Table 1. A list of simulated sea water column height value from a satellite altimeter

YYYY	MM	DD	HH	MM	SS	Coordinate (randomly generated within the size of the grid, e.g., 3×3 grid)	Height (m) - Randomly generated, always exceeds the pre-defined threshold
2021	09	05	10	01	10	0,0	6001.889
2021	09	05	10	01	11	1,2	6100.540
2021	09	05	10	01	12	2,2	6050.594
2021	09	05	10	01	13	2,1	7892.199
2021	09	05	10	01	14	1,1	7100.155
2021	09	05	10	01	15	2,2	7050.694
2021	09	05	10	01	16	2,1	6892.239
2021	09	05	10	01	17	1,1	6100.215

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

d) 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).

e) 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) A single MPI process simulates the base station node.

b) The base station node also creates the thread which simulates the satellite altimeter (refer to point 2.0 above).

c) The base station node periodically listens for incoming reports from the sensor nodes.

d) 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 satellite altimeter).
   - If there is a match, the base station logs the report as a matched event (true alert).
   - If there is no match, the base station logs the report as a miss-matched event (false alert).

e) 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.

f) The base station has an option to send a termination message to the sensor nodes to properly shutdown.

g) The base station also has an option to send a termination signal to the satellite altimeter to properly shutdown.

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 sensor nodes, satellite altimeter 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 FAQs 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. **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.

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.

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.
Click here to access the submission link for the assignment.

The Assignment-2 submission is due on Monday, 18th October 11:55 PM AEDT. All demo interviews must be completed in Week 12.

This assignment is worth 25% of your final grade.

MARKING GUIDE

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

1.0 Simulating the tsunameter sensor nodes	5 marks
2.0 Simulating the satellite altimeter	2 marks
3.0 Simulating the base station	5 marks
Overall code structure and code comments	2 marks
Code demonstration and Q&A in Week 12 (Individually assessed)	4 marks

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

1.0 Simulating the tsunameter sensor node	4 marks
2.0 Simulating the satellite altimeter	2 marks
3.0 Simulating the base station	4 marks
4.0 Fault detection and response	2 marks
Overall code structure and code comments	2 marks
Code demonstration and Q&A in Week 12 (Individually assessed)	4 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.