MONASH University

FIT3143 – Assignment 2

Marks: 20 marks and 20% of all marks for the unit.

Due Date: **Week 11**, **Friday 18/October/2019, 11:55 PM Melbourne time**.

Submission: (i) The assignment submission must be made through Moodle for this unit **by the due date**. (ii) The submission of this assignment must be in the form of a **single PDF file, C code file/s and log file/s**.

Extensions: *No extensions will be given.*

Lateness: Late penalty of 10% per day after the due date, including the weekends.

Authorship: This assignment is an individual assignment and the final submission must be identifiably your own work. Breaches of this requirement will result in an assignment not being accepted for assessment and may result in disciplinary action.

Cover Sheet: A completed individual assignment covered sheet is required with the submission. This coversheet should be appended with the assignment report (PDF file). Please refer to the submission details at the end of this file.

Assignment Question

Event detection in a fully distributed wireless sensor network -WSN

I. WSN Description

The wireless sensor network comprises **20 nodes** and a **base station**. These nodes are arranged in a **4 x 5** (rectangular-shaped) grid.

Distance between the adjacent nodes is kept as such that these nodes can wirelessly communicate. The adjacent nodes can exchange data through *unicast* and *broadcast* modes of communications.

Communication between non-adjacent nodes is not possible.

Every node in the WSN can however independently exchange data with the base-station (e.g. through a satellite link). Base-station for this WSN is an additional computer node that is entrusted with the task of gathering data from all the 20 nodes in the WSN.

Event Detection Criterion: In order for an event to be recorded by the WSN, at least ***three*** adjacent nodes, to a reference node, must simultaneously report their activations to the base-station (explanatory figure in the end). The base station then collects all the event reports and writes these to its log file.

Assignment Project: Develop a Message Passing Interface (MPI) C code that simulates the operation of this WSN including the base station in an efficient manner. The criterion for measuring efficiency in this exercise is in finding a communication scheme that minimises the messages to the base-station whilst satisfying the WSN's event detection criterion (stated above). In addition, the events and/or messages sent between adjacent nodes and to the base station are to be encrypted. You can choose and implement any C based encryption algorithm. You are also allowed to use available encryption algorithms (in C programming language) provided you properly cite the owner of the algorithm (in your C source code and in your report). More importantly, you are required to demonstrate the use of Open Multi-processing (OpenMP) to improve the performance of the applied encryption algorithm.

Hints:

1. Assume that a set of MPI tasks (processes) represents the WSN and each MPI task can represent a WSN node.
2. In order to simulate random occurrences of events within the WSN, each WSN node may be provisioned with an independent random number generator with the condition that at least **three** adjacent nodes must produce the same random number, at the sampling time, to constitute an event.
3. Write the key performance metrics e.g. the simulation time, number of events detected, number of messages/event 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.

II. Submission:

Programming Task: Write, and demonstrate in the lab, a MPI + OpenMP program that effectively addresses all the points in the problem statement. Assume that a set of MPI processes represents the WSN and each MPI process represents a WSN node.

Write-up Task - Report layout (Up to 2000 words):

I. Abstract (brief overview of the assignment, chosen method and summary of results).

II. Introduction (Basics of inter-process communication (IPC) and OpenMP).

III. Design Scheme for IPC (Include illustration to describe e the selected scheme, and flowchart (or pseudocode) of the parallel algorithm).

IV. Encryption algorithm used to encrypt the messages and acceleration of the encryption algorithm using OpenMP.

V. Results and Discussion of the implemented design scheme. Describe the content of the messages which were exchanged between the nodes and messages that were sent to the base.

VI. Conclusion.

The report should be formatted using an IEEE template.

Please refer to the assessment rubric which is enclosed together with this specification. This rubric should assist you to attain a better understanding on the marking guide for the assignment.

Please do not attempt to plagiarise or collude your assignment work.
Note that it is a University requirement (http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-managing-plagiarism-collusion-procedures.html) for students to submit an assignment coversheet for each assessment item.

Faculty Assignment coversheets can be found at http://www.infotech.monash.edu.au/resources/student/forms/. 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. Each student is required to submit the following files:

a) Assignment report (.PDF file which includes the coversheet). Drag and drop this file into the Moodle submission page.

b) Assignment code files (.h and .c file extensions). You are allowed submit multiple files in your Moodle assignment page. As such, if you are submitting multiple C code files, simply drag and drop these files into the Moodle submission page.

c) Log file/s containing the messages which were exchanged between the nodes and messages which were sent to the base.

III. Supplementary explanation:

The wireless sensor network comprises 20 nodes **and** a base station. These nodes are arranged in a 4 x 5 (rectangular-shaped) grid.

Each node is an MPI Process, so now you have **total 21 MPI processes**.

Let’s assume your map/grid looks like the following:

	A	**B**	**C**	**D**	**E**
1	A1	B1	C1	D1	E1
2	A2	B2	C2	D2	E2
3	A3	B3	C3	D3	E3
4	A4	B4	C4	D4	E4

What is an “adjacent node”?

All the **immediate neighbours in top-bottom and left-right directions** are “adjacent nodes”.

- For node B2, adjacent nodes are: B1, A2, C2, B3.
- For node A2, adjacent nodes are: A1, B2, A3.
- For node A4, adjacent nodes are: A3, B4.

<u>**Note: Communication between diagonal nodes are not allowed.**</u>

Communication scheme:

Each node can only communicate with an adjacent node directly. Non-adjacent nodes (such as B3 and D3) can’t communicate directly.

Communication between

- Adjacent Node ←→ Adjacent Node
- Node ←→ Base Station

may take place using any of the following methods:

- Blocking or non-blocking Send/Receive
- Broadcast
- Map/Reduce style functions
- Any other method that you think is appropriate.

The goal here should be, to pass **as fewer messages as possible**. Note that messages sent to the adjacent nodes and to the base station should be encrypted.

What is an “Event”?

The program/simulation will run for *x* number of iterations (where *x* = you decide!) of *y* millisecond each (where *y* = you decide!). Every node in the grid will generate a random number within a range such that:

range_lower_bound <= randomly_generated_number <= range_upper_bound

where range_lower_bound = you decide!
and range_upper_bound = you decide!

You need to adjust the time interval, and range bounds such that there is a good chance that a few nodes throughout the grid, will end up generating the same random number.

An event occurs at a node *N* when **minimum three** of *N*’s adjacent nodes end up generating the same random number.

Let’s assume nodes B1, C2 and B3 generated the same random number. That would count as an occurrence of the event at node B2.

Similarly, if nodes A1, B2 and A3 generated the same random number, that would count as an occurrence of the event at node A2.

Clearly, the event will never occur at the nodes located at four corners of the grid (such as node A4) since you require minimum three adjacent nodes to generate the same random number, and the nodes located at the corners have only two adjacent nodes.

What to measure?

Below are the metrics you should keep track of, for every iteration:

- The time taken to encrypt/decrypt a message before sending or after receiving. The use of OpenMP here should demonstrate improvements in encryption and/or decryption processing time.
- Number of messages passed throughout the network
- Number of events occurred throughout the network
- Details of nodes involved in each of the events (reference node and its adjacent nodes)

At the end of the program, writing these metrics to a log file would help us evaluate the efficiency and correctness of your program.