IS4151/IS5451 – Pervasive Technology Solutions and Development

AY 2019/20 Semester 2

Individual Assignment 2

Objectives

At the completion of this individual assignment, you should:

1. Develop a better understanding on how to employ single-board computers in various roles to create useful real-world solution in conjunction with single-board microcontrollers and cloud servers.

2. Acquire a deeper technical skillset on working with the wireless data communication and computational capabilities of general-purpose single-board computers such as Raspberry Pi.

3. Acquire a deeper design and technical skillset on using single-board computers to implement a real-world IoT system that exhibits the fog/edge computing architecture.

Opening Narrative

The coronavirus disease (COVID-19) pandemic has reached the shore of **Panem**. In order to protect the rich and elite citizens in The Capitol, President Coriolanus Snow has ordered a lockdown. All the inhabitants in the twelve outlying numbered Districts are to be quarantined within their own District and tagged with a 2.4 GHz radio-based COVID-19 tracker. Each District will have an edge processor that is to be operated by the local Peacekeepers to track the temperature and location of the inhabitants. The Capitol itself will host a cloud server that is capable of collating all the tracking information and provide President Snow with a global view of the inhabitants across the twelve districts.

General Requirements

You are required to develop an IoT system of COVID-19 trackers for Panem known as **HungerTrack**. This is an individual assignment and you are required to complete all project tasks on your own. The submission deadline is the end of Week 12, i.e., **Sunday, 12 April, 11:59 pm**. You will do a simple demonstration during the lab class in Week 13.

The HungerTrack system uses mico:bit as the primary COVID-19 tracker or sensor node for tracking the temperature and location of the inhabitants in a particular District. It is assumed that the micro:bit ambient temperature represents the temperature of the inhabitant wearing the tracker. Due to the large geographical area of Panem, an edge computing architecture will be utilised such that each District can respond to any incident in a timely manner. An edge processor that is to be implemented with Raspberry Pi will be setup in each District to allow the local Peacekeepers to monitor inhabitants within the District. A global cloud server that is to be implemented using a conventional computer will be setup in The Capitol. This allows President Snow to exercise global monitoring of the inhabitants in the twelve Districts.

The high-level schematic diagram of HungerTrack is depicted in Figure 1. For simplicity, the diagram only depicts The Capitol and three Districts. In Panem, there are twelve Districts altogether. Due to the transmission limit of the 2.4 GHz radio, it is necessary to setup a radio mesh network within each District so that all inhabitants in a particular District can be tracked. Each District is assigned its own radio group.

The IoT system is also required to track and monitor inhabitants from other Districts who had intruded into a District. For the purpose of this intrusion tracking, you may assume that only micro:bit devices from another radio group that is within 1 hop is considered an intruder.

The HungerTrack system consists of four main components that you would be developing:

- The COVID-19 tracker or node running a single micro:bit program.
- The edge processor radio controller running a single micro:bit program.
- The edge processor running multiple Python programs with a database.
- The cloud server running a single Python program with a database.

HungerTrack IoT System Operation Process

Each edge processor initialises by running one or more Python programs. If multiple Python programs are used, one of the programs will initiate the serial-over-USB connection handshaking with the radio controller. The radio controller initialises by running the micro:bit program. After initialisation, it is ready to accept tracking information sent or relayed by the trackers via radio, and forward the information over serial-over-USB to the edge processor.

Each COVID-19 tracker or micro:bit device initialises by running the micro:bit program. After initialisation, it is ready to transmit its identity (serial number and name) together with its temperature to the edge processor either directly or indirectly (through other trackers). A tracker is also required to transmit its location in terms of the number of hops away from the edge processor. For example, a tracker that transmits directly to the edge processor is considered as 1 hop away. Another tracker that transmits indirectly via another tracker is considered as 2 hops away.

The edge processor will record and store all tracker information in its local database. At a periodic interval, the edge processor will relay the tracker information to the cloud server for central storage and processing.

The edge processor will publish the latest trackers information on a web page for the local Peacekeepers in the District to monitor. No data export API is required at the edge processor.

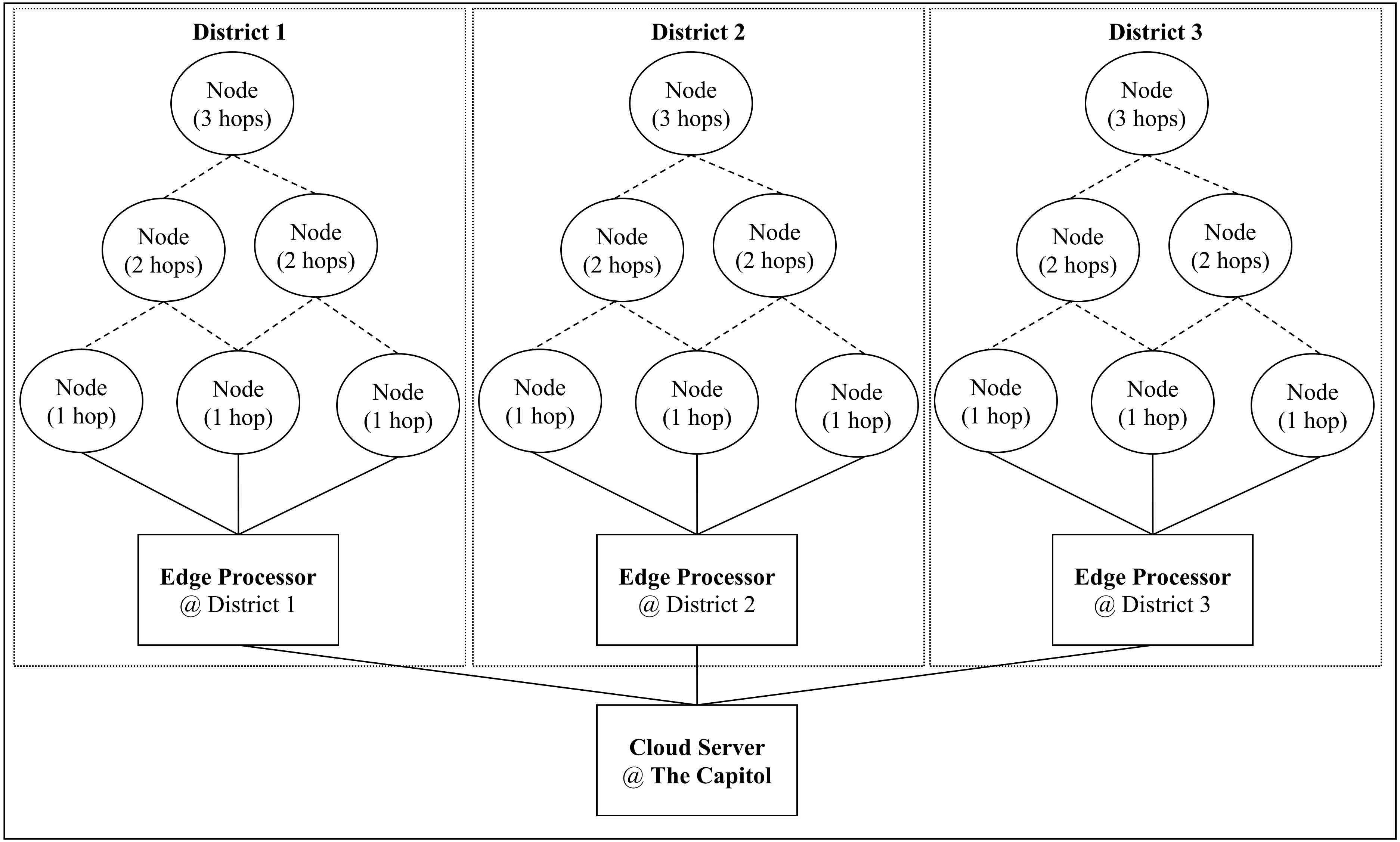
District 1
District 2
District 3
Node (3 hops)
Node (2 hops)
Node (1 hop)
Edge Processor @ District 1
Edge Processor @ District 2
Edge Processor @ District 3
Cloud Server @ The Capitol

Figure 1 – High-level schematic diagram of the HungerTrack system.

The edge processor will actively monitor the following two types of events in its District:

1. An inhabitant or intruder with temperature greater than 38 degree Celsius.

2. An intruder has entered within 1 hop of the edge processor.

Upon detecting a Type 1 event, a local lockdown alarm will be automatically activated. The cloud server will also be notified of these two events.

The cloud server initialises by running the Python program. After initialisation, it is ready to accept tracker information reported by the edge processors and store them in the global database. The cloud processor will publish the latest tracker information collated from all the edge processors on a web page for President Snow and other high-ranking officials at The Capitol monitor. No data export API is required at the cloud server.

Whenever a Type 1 or 2 event is received by the cloud server, a global lockdown alarm can be manually activated by President Snow and his officials at The Capitol. A local or global lockdown alarm can only be deactivated from the cloud server.

A COVID-19 tracker will also automatically trigger a personal warning to the inhabitant if s/he is within 1 hop of another inhabitant whose temperature is 36-38 degree Celsius.

A high-level schematic diagram of the HungerTrack installation at a District is depicted in Figure 2 below.

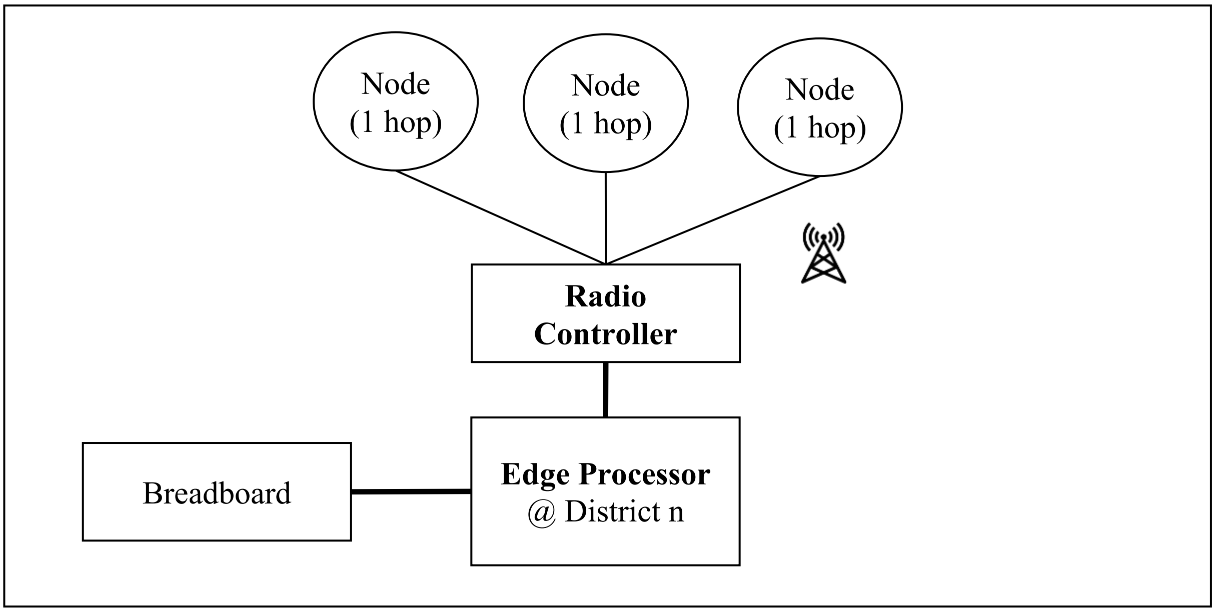

Figure 2 – High-level schematic diagram of the HungerTrack installation at a District.

For the actual prototype implementation, you are required to implement the setup for i) one District using two trackers; ii) another District using one tracker.

HungerTrack IoT System Expected Output

The COVID-19 tracker that is implemented using micro:bit should utilise its 25 LEDs to display the various status at the node according to the guidelines in Table 1.

X,Y	1	2	3	4	5
1					
2					
3					
4					
5					
Information	Local temperature – Integer value ranging from (30, 40]: • 30 and below – Turn off all 10 LEDs. • 31 to 40 – Turn on 1 LED for each 1 degree Celsius starting from (1,1) to (1, 5) and then (2,1) to (2,5).		Personal warning – Turn on (3,1) to (3,5)	Local lockdown alarm – Turn on (4,1) to (4,5)	Global lockdown alarm – Turn on (5,1) to (5,5)

The edge processor should publish the following information in one or multiple web pages:

- A table listing all tracker information, one row per tracker showing the following attributes:
  - Device ID
  - Device Name
  - District (i.e., radio group)
  - Temperature
  - Fever (Y if > 38, N otherwise)
  - Intruder (N if radio group is same as District, Y otherwise)
  - Timestamp
- A table listing all Type 1 and Type 2 events:
  - Device ID
  - Device Name
  - District (i.e., radio group)
  - Event
  - Timestamp

The edge processor should also control three LEDs to be turned on as follows:

- Green LED – Turned on when there is no Type 1 event.
- Red LED:
  - Turned on when there is a Type 1 event, i.e., local lockdown alarm.
  - Turned on and blinking when there is a global lockdown alarm.
- Blue LED – Turned on when there is a Type 2 event.

The cloud server should publish the following information in one or multiple web pages:

- A table listing all tracker information, one row per tracker showing the following attributes:
    - Device ID
    - Device Name
    - District (i.e., radio group)
    - Temperature
    - Fever (Y if > 38, N otherwise)
    - Intruder (N if radio group is same as District, Y otherwise)
    - Timestamp

- A table listing all Type 1 and Type 2 events:
    - Device ID
    - Device Name
    - District (i.e., radio group)
    - Event
    - Timestamp
    - A button to deactivate local lockdown alarm (Type 1 event only)

- A web page showing the current global lockdown alarm status, and a button to activate/deactivate the global lockdown alarm.

Basic Reference Protocol

You are required to design and implement your own protocol based on the requirements given in this project specification.

Overall Assessment Criteria

The overall assessment criteria are listed below.

Criterion	**Maximum Possible Marks**
Implementation of the COVID-19 Tracker	2
Implementation of the Radio Controller	3
Implementation of the Edge Processor	2
Implementation of the Cloud Server	2
General Coding Techniques and Design Considerations	1
Total	**10**

Assignment Deliverables Submission

The deliverables to be submitted to LumiNUS Files tool at the end of Week 12 are to be placed in a single zip archive file with the following folders structure:

- **docs** subfolder containing:
  - Softcopy of a Microsoft Word document in DOCX format briefing explaining the design of your system, in particular your protocol.
  - Special instructions for setting up and running your system.
- **source** subfolder containing:
  - All project folders/source files that constitute your system.
- **readme.txt** containing:
  - Full name
  - Matriculation number.
  - Email.
  - Contact number.

Upload this zip archive file to the designated LumiNUS Files tool folder: "Deliverables Submission > Individual Assignment 2".

Your assignment deliverables must be submitted latest by **Sunday, 12 April, 11:59 pm**. No assignment will be accepted for assessment after this date/time. For the demonstration, you will be given the source files from your LumiNUS submission.

-- End of Individual Assignment 2 Specification --