Contents

Todo list

Chapter 1

Introduction

The case chosen for this assignment was project 3 regarding simulator for wireless communication. Based on the assignment description, the report will contain information about all the phases of the project; Analysis, Design, how the design was implemented, major decisions that we needed to take and finally the live product.

The wireless network simulator involved designing a software that simulates wireless communication between nodes. Furthermore, the nodes are to be places in different position in a user defined space and they are to communicate with each other. In the assignment description, the program is required to have these additional aspects:

- User must choose different type of nodes such as base stations, mobile phones, drones, satellites...
- A subset of nodes may transmit a block of data selecting a certain power, frequency, bandwidth and time of transmission
- A subset of nodes may listen to incoming data and try to decode the data
- A GUI is used to insert and remove nodes. It is also used to start and stop data communication and to show communication quality information of each node

We then proceeded to carry on with the first stage of the waterfall model which is Analysis. During the Analysis, we discovered additional considerations that we would need to take into account. This resulted in the group coming up with a use-case diagram and a class diagram. Then the simulator was implemented in python. During this project, the group had encountered some challenges which would be discussed in the report.

Chapter 2

Analysis

In the first step of the waterfall model, requirement and object analysis is carried out.

2.1 Defining use cases

Based on the description given, actors and use cases are identified and then clearly defined.

2.1.1 Actors

Based on the project description provided, the only actor that we have found would be the user who is running the simulation. At first we had considered the database to be an actor of the system but later discovered that it would not be that appropriate to have the database as an actor to the system.

2.1.2 Use cases

Next, we have defined the various use cases that we want the user to be able to do, which will be described in the following sections.

Add node

When the user wants to add the node, they would need to specify the type of node added. The node added could be either a IoT Node or a network gateway.

Start Simulation

When the user wants to start the simulation, they would need to provide the following information:

- Number of nodes in the network
- Size of the 2D grid where the nodes will be placed
- Option of verbose output

See simulation statistics

When the user wants to see simulation statistics, the system would need to check if any simulation has occurred previously. If there is no prior simulation, then the system would display an error. Otherwise, the system would display the simulation statistics such as throughput, delay, and packet loss.

2.1.3 Use case diagram

Based on the factors mentioned above, a use case diagram was developed. The use case diagram can be seen below.

2.1.4 FURPS

The FURPS were utilized to better define the system's requirements. The system's overall summary was also created using this technique. This approach also aided in providing a fresh and more enlightening perspective on the issue and the requirements.

Functionality: what should the system do?

- The system should be able to place the nodes in a grid and transmit packets from one node to the gateway.
- The system would need to run the simulation for a fixed amount of time

Usability: what kind of UI is needed?

- A screen would be required to display the simulation statistics
- A device should receive user input eg.keyboard

Reliability: what is the tolerance of the system to failures?

- The system should be able to check if the simulation has been run before displaying simulation statistics

Performance: which response times, accuracy, availability, resource usage should the system have?

- The GUI needs to be run on a simple computer
- The simulation statistics needs to be stored in a server

Sustainability: what adaptations may be needed?

- The system need to give the user the ability to change the simulation conditions after a run has been completed

2.2 Domain model

To aid the group in identifying the various objects involved in the problem, a domain model was developed. Firstly, the identified actor, which is the user, is represented as a GUI in the model. This also constitutes the front end of the system. Regarding the back end of the system, we would need one object to simulate the network and another one to generate/store the simulation statistics. Ultimately, we would need an object for the devices themselves and one for the overall network. Here it might also be a viable option to differentiate between a node object and a gateway object that then both can inherit attributes and methods from the device object when making the class diagram. Then, in order to control everything and tie the simulation to gather, it would make sense to incorporate a simulation object that handles the simulation itself.

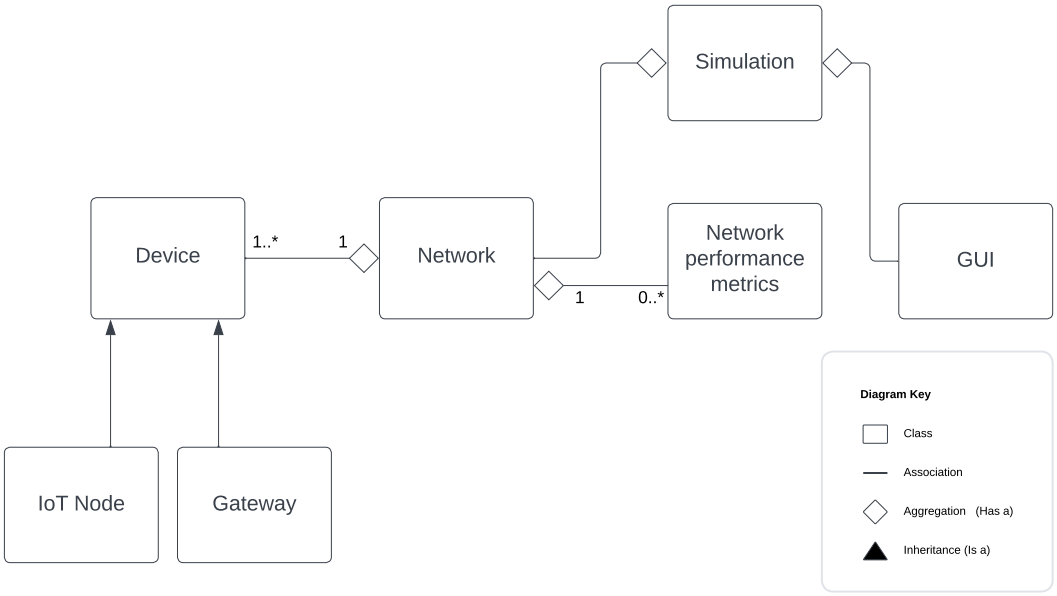

Figure 2.1: Diagram of the domain model.

2.3 Class diagram

The entirety of the class diagram can be seen in Figure 2.2. The parts that were not implemented have their text colored red. This also implies that the class diagram is made in accordance to how the implementation has been done

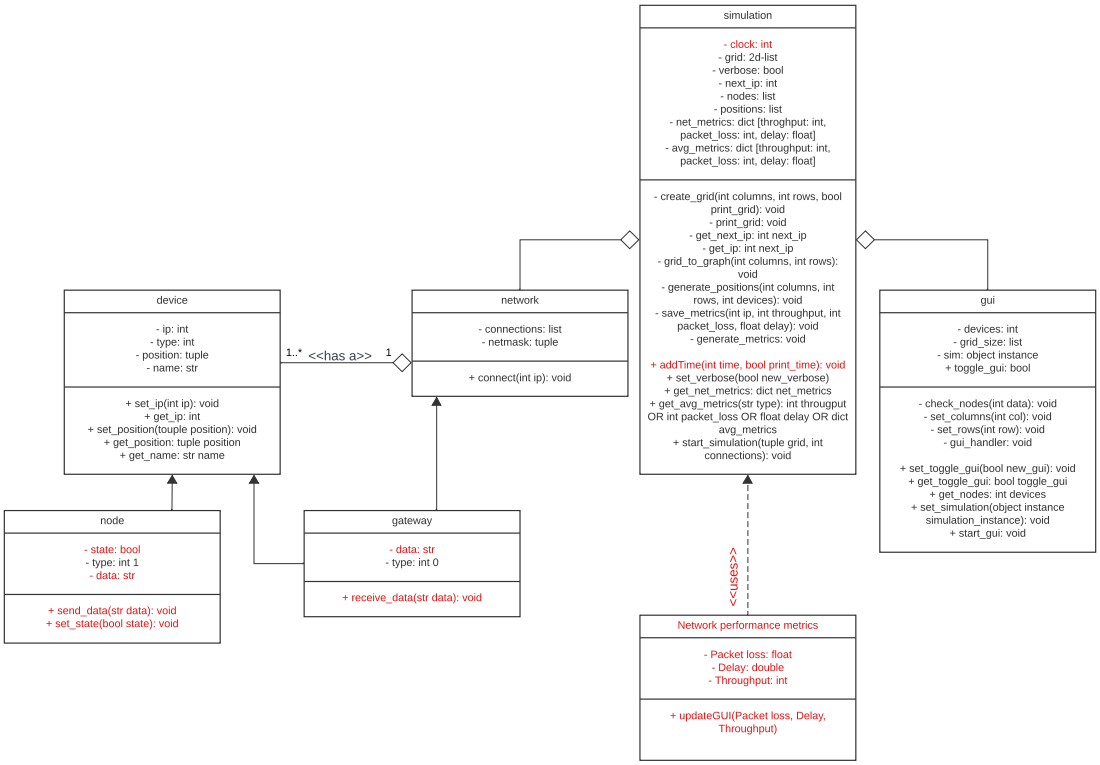

Figure 2.2: Diagram of the classes.

2.3.1 GUI

2.3.2 Simulation

2.3.3 Network

2.3.4 Device

Node

Gateway