Analysis and Evaluation of Bluetooth Mesh

Introduction

As we focus in this course in Next Generation Networks, our group would like to concentrate on a rising technology: Bluetooth Mesh Networking. The official Bluetooth webpage states:

> "Bluetooth mesh networking enables many-to-many (m:m) device communications and is optimized for creating large-scale device networks. It is ideally suited for building automation, sensor network, asset tracking, and other IoT solutions that require tens, hundreds or thousands of devices to communicate with one another."
>
> (https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/topology-options/le-mesh/mesh-faq/)

Not only it is related to our NGN course but also represents a fundamental standard in the Internet of Things (IoT) world.

Analysis

Our project aims at understanding how to Bluetooth standard works combined with a meshed used of the network. To do so we initially thought about using the existing Bluetooth Mesh standard from Bluetooth SIG. After finding out that most of the microcontroller in the market (including the Raspberry Pi) do not support this official technology, we decided to follow a different path: we will create our own Bluetooth mesh concept.

Our meshed network is a constellation of multiple devices that communicate with each other applying different topologies: unicast, multicast and broadcast should be made available. Therefore, we plan to introduce a routing table that allows nodes of the network (the devices) to choose autonomously where to send information without a central entity deciding the routes. There will be just one device keeping an eye on the whole network: the provisioner. In the next paragraph the proposal will go into more details.

This scenario will apply to a chat service, that we will implement using the Serial Port Profile (SPP) of the Bluetooth devices. This chat application will be created by our team and will feature only very basic chatting features. It will be used to showcase the Bluetooth mesh functionality.

Scenario

As described in the previous paragraph we will have two main actors: the provisioner and the nodes. A node is a device that acts both as server and client. The server is the head of the node, providing routing information for the chat messages, while the client triggers the action of sending a message to the next node/s.

For this purpose, we plan to use MQTT-SN, a messaging protocol work on top of UDP (please notice the difference with MQTT that works with TCP/IP). Every device consists of an MQTT-SN Broker (with integrated Gateway to allow the transmission via Bluetooth Serial Port) and an MQTT-SN client. The broker decides whether to keep the message and send it to the internal client or forward it to the next device's client (see Figure 1: Overview of BLE Meshed Network with MQTT-SN via SPP.

The provisioner on the other hand monitors the network and is responsible for adding or removing nodes from the topology. This entity needs to communicate with the single devices, updating the network. The provisioner-nodes and nodes-nodes communications should succeed with MQTT-SN as well.

Another important and technically complex aspect is the automation of the Bluetooth paring and connection event once the network receives a new node. This part is not totally defined yet; we still need to determine an efficient way to perform it. In the initial development phases this step will be done manually. This means connection all the notes together “by hand”.

The service running of this topology is a chat. We want to take advantage of the Bluetooth Serial Port Profile (SPP). To open the ports and get device information as well as to extract and read the content of the serial terminal, we rely on the Python library PySerial.

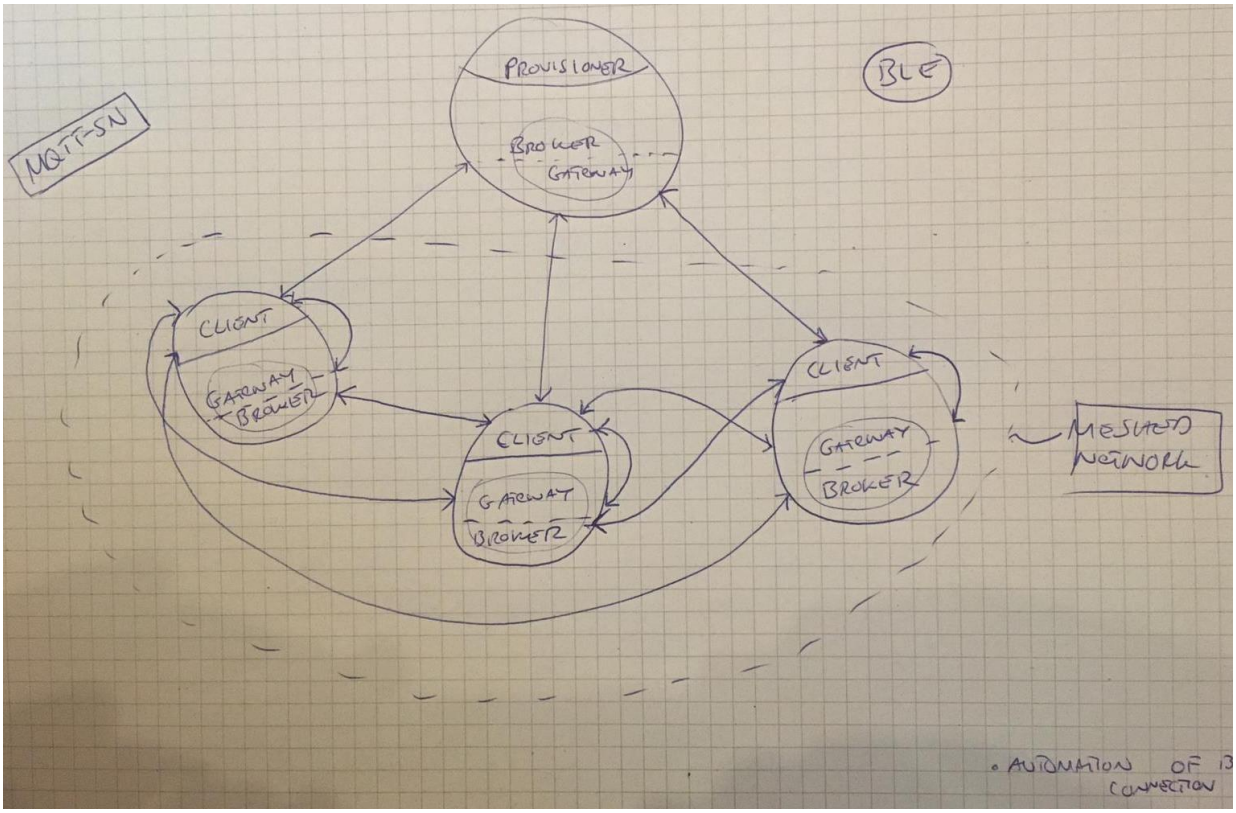

Figure 1: Overview of BLE Meshed Network with MQTT-SN via SPP

Workflow

Our initial intention is to study how Bluetooth Mesh works and how to establish a simple network. This will require a study period followed by tests with devices like smartphones, laptops, and Raspberries. We expect to close this part on 26th April.

Secondly, we will apply our new knowledge to implement the server/client MQTT-SN via SPP setup for the nodes. This is the most complex and long part of our project since we have to build the concept from scratch. We plan to conclude it on 24th May.

Then we will develop the provisioner. At the same time, we will automate the process of pairing and connecting the Bluetooth devices. For this we expect to finish on 7th June.

Finally, once the concepts show its first results, we will be able to perform and network analysis, showing how the hops between the devices work and the role of the provisioner. For this part we plan to finish 2 weeks before report submission.

Tasks Division

Below we will describe a possible and meaningful scenario that involves the use of hardware autonomously at our own places.

1) Study and Analysis of Bluetooth Mesh - All together

3) MQTT-SN via SPP client and broker + provisioner - Alessandro

4) Automation of BL connection + Network Analysis – Samuel

Notice that we will probably cooperate on the different tasks to help each other and learn as much as possible.

Modification

After Theodor Fischer's renounce we decided to cancel the UI for the chat. We will just rely on the terminals. Moreover, we plan to realize the provisioner no more in second phase but in the third. Finally, Samuel will perform alone BLE Automation and Network Analysis while Alessandro both provisioner and client/server nodes concept.