LINGI2146 - Project

1. Scenario

You have been contacted by a popular technology company. They want to expand into the promising market of *Home Automation*. They contracted you to develop a proof of concept for a new ecosystem of *Internet of Things* (IoT). This ecosystem would consist of interconnected devices, such as:

- Connected lamps
- Proximity sensors
- Connected door locks
- Temperature sensors
- …

All devices form a network connected to the internet. More precisely, they use the 6LoWPAN/RPL protocols to form a wireless IPv6 *personal-area network* (PAN). Devices should be able to appear and disappear easily in the network. They are also connected to a server, in charge of the control of the ecosystem. It provides an interface for users to interact with their IoT devices. Your job is to design and develop the software running this ecosystem.

2. Implementation

2.1 Communication protocol

The first part of the project is the design of the communication protocol between devices and with the server. There are different types of devices and some of them only send data (e.g. a temperature sensor), some only receive commands (e.g. a lamp), and some can do both. You have seen during the course the protocols MQTT, a publisher/subscriber sensor network protocol, and CoAP, a resource-oriented REST network protocol. The protocol you will design should be a competitor to these protocols, for IoT devices in the ecosystem.

Messages in the ecosystem are exchanged as datagrams, on top of UDP/IPv6. You need to describe the format of these datagrams. Your protocol should be efficient, to respect the limited network bandwidth and resources of the devices. Your protocol may also consider security and reliability aspects of the communication.

Your goal is to design the communication protocol in the ecosystem and describe the format of all messages sent through the network.

2.2 IoT Devices

For the second part of the project, you will design the (simulated) devices in the ecosystem. As a proof of concept, you need to design at least four type of devices, with at least one sensor and one connected object. Your design should however be easily expandable to more device types as the company develops in the market.

For this proof of concept, you will not be able to have access to actual hardware as the company did not create it yet. You will instead simulate the ecosystem in Cooja, as you have seen during the practical sessions of the course. You will not be able to perform the exact actions of each device, but you can use any close-enough behavior you may seem fit. For example, you could light up the LEDs or a Z1 node for a connected lamp or detect activity when the user button of a Z1 node is clicked for a proximity sensor.

One node in your RPL network should act as a gateway, to enable the communication between the server running on your machine and the devices in Cooja. For this part, you can use the RPL border router seen during TP4 as root of the RPL network. The devices you will create can then be nodes in the RPL network created by the border router. You have also seen during TP4 how nodes can receive UDP messages with RPL. You will need to look up inside the Contiki OS library how you can send UDP messages to the server.

Warning: because we are creating an IPv6 network, you will need to have IPv6 communication from Cooja to your host machine. In a default Docker configuration, IPv6 is disabled by default. You can verify this by inspecting the parameters of a docker container, you will see that you have no IPv6 gateway address. You can enable IPv6 in docker by editing the `/etc/docker/daemon.json` file on your machine (create it if it does not exist). Add the following configuration (leave other configurations if there are any):

```
{
  "ipv6": true,
  "fixed-cidr-v6": "fd00::/64"
}
```

After that, restart the docker service: `sudo systemctl restart docker`. You will be able to find the IPv6 gateway address, that can be used to contact your host machine from the nodes in Cooja, in the parameters of the container. See this link[1] for more information.

Your goal is to define the list of supported devices in the ecosystem, and the capabilities of each device. You should also implement the communication protocol for each type of device, as well as the actions related to their capabilities.

2.3 Control server program

The third part of the project is the design of the control server. This program implements the communication protocol to interact with the IoT devices. It gathers data from the devices and sends them commands. For this purpose, you could make the server run continuously, or make it run only at appropriate times.

The server presents an interface to let users interact with the devices. For example, the user should be able to turn on/off a connected lamp or get the last reading of a temperature sensor. For the proof of concept, a *command-line interface* (CLI) is fine.

Your goal is to design the server interface and implement it to allow interactions with the ecosystem.

2.4 Automation

As of now the only actions on the ecosystem are initiated by the user through the server interface. For the last part of the project, we would want to be able to create some automations in the ecosystem. For example, a connected lamp could turn on automatically when a proximity sensor detects some activity.

These automations can be centralized and controlled by the control server program, or you could make one device connect directly to another. With our example, this would mean that once the proximity sensor detects some activity, it would send a message directly to the connected lamp to turn on without going through the server. Discuss all design decisions such as this one in your report.

Your goal is to design a way in the ecosystem to handle automations and demonstrate it in your simulation. As an optimization, you can permit the user to define automations within the interface of the server.

3. Deliverables

The deadline for the project is May 14. You will work in groups of three students. You have to deliver a short report in PDF format (~5 A4 pages) describing:

- The communication protocol and its message format
- The supported IoT devices and their capabilities
- The control center program and its interface
- How messages are exchanged between them
- How automations are handled in the ecosystem
- Any other design decision relevant to the project

Keep the report short and only give the essential information i.e., your design decisions, not source code.

The report **must** contain a link to a github or bitbucket repository read-accessible to us with the **commented** source code of the code for your devices and the server.

Don't forget to put the names of all group members on the cover page of the report.

[1] https://medium.com/@skleeschulte/how-to-enable-ipv6-for-docker-containers-on-ubuntu-18-04-c68394a219a2#8f4f