IoT Project 2020

Two different options are available for the implementation of the IoT project:

- **Single person project.**
- **Group projec**t; each group **must** be composed of **3 people.**

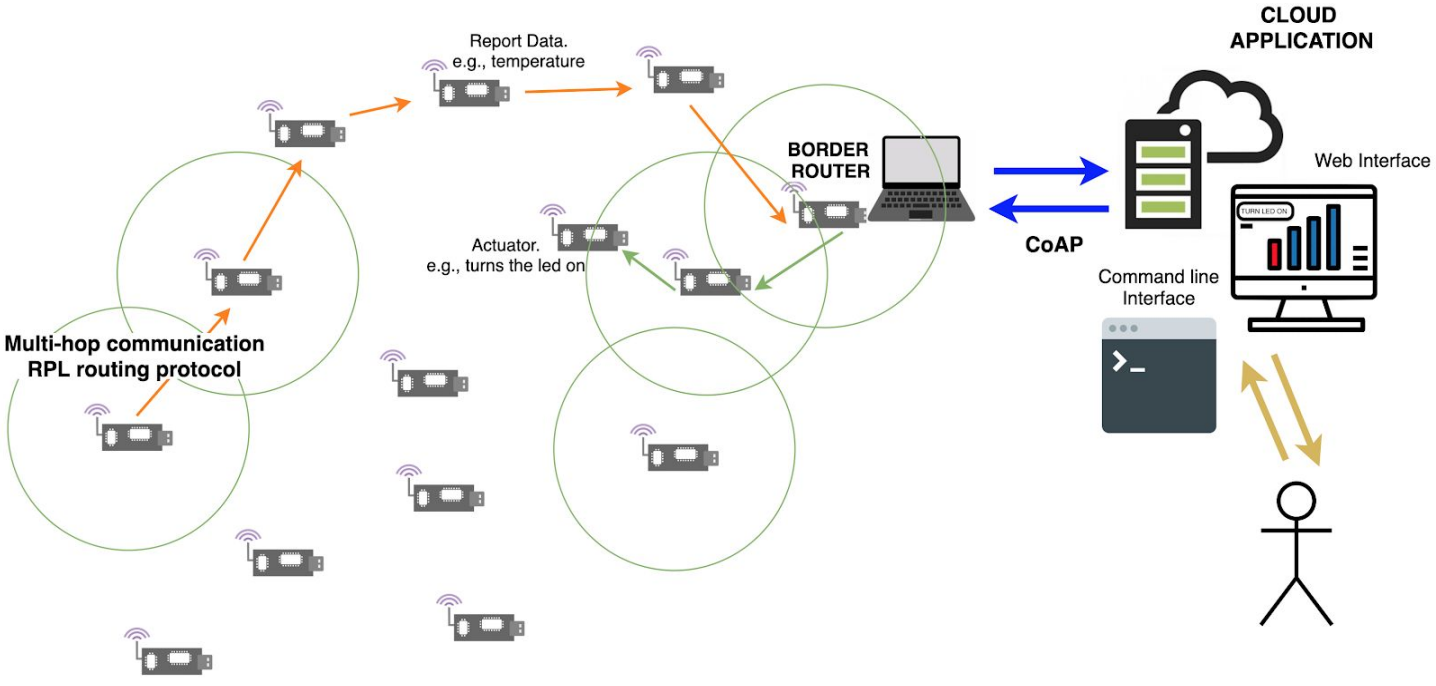
No other options are available.

Objective: Development of a complete IoT system referring to one the following real-life use cases:

- Home automation
- Smart Agriculture
- Smart Urban Waste
- E-health
- Smart product management

Additional use-cases can be defined by the student(s).
You can find a brief description of the use cases at the following link(https://drive.google.com/file/d/1M3tDf0YmXP4P9jb7PwT3hVbkOQcNLPVs/view?usp=sharing).

System scheme and structure



The system will comprise the following components:

- A set of IoT devices, including both sensors to collect data from the physical environment/devices to which they are attached and actuators for remote control and actuation on an environment/a device to which the actuator is attached. The IoT devices will be implemented exploiting the Contiki-NG operating system and emulated via Cooja

- A ‘cloud’ application that will interact with the sensors/actuators to offer certain functionalities depending on the application and use-case. The cloud application will be developed using Californium and, for the sake of simplicity, will run on VM where the IoT devices are emulated

Functionalities

The application developed through the IoT system has the following functionalities:

- Sensor and actuators will expose their functionalities through the CoAP protocol as CoAP servers.
- Resources and their data must be continuously monitored through CoAP Observing in order to retrieve the representation of the resources over time. Updated data must be shown to the user.
- IoT devices must register to the cloud application at bootstrap. This is performed via CoAP by issuing, as CoAP clients, a request to the cloud application. To this aim the cloud application must expose a registration interface. The request will provide all the details required to identify the sensor/actuator and its functionalities.
- The cloud application must expose an user interface to show the data collected in real-time and allow the user to change the status of actuators. The user interface should be a **command line interface** for the **single person project** or a more complex **web interface** for the **group project**.
- [**Only for group projects**] Some type of sensors might implement machine-to-machine interactions, i.e. by exchanging data directly by themselves without the intervention of the cloud application. In this case the cloud application must expose an additional interface, the look-up interface, through which IoT devices can retrieve the IP address of other IoT devices belonging to a given category, e.g. all the temperature sensors, all the light bulbs, etc.

In order to have an idea on how implement the registration and lookup interfaces exposed by the cloud application you can refer to the **CoRE Resource Directory** draft, which provides an example of how these interfaces could be implemented in a standard directory (https://tools.ietf.org/pdf/draft-ietf-core-resource-directory-24.pdf).