

IoT Projects (COMO) - 2018/2019

Alessandro Redondi
alessandroenrico.redondi@polimi.it

December 10, 2018

1 General Rules, Grading and Deadlines

Grade composition of the entire course:

- 26 out of 30 points are assigned based on the written exams
- up to 8α out of 30 points are assigned based on projects (parameter α depends on project delivery date, see below).

General rules:

- Projects are NOT mandatory. One student can decide not to take any project; in this case, that maximum grade he/she can get will be 26/30.
- Projects can be developed in groups of maximum 2 people.

Projects delivery deadlines and grading:

- July 26, 2019, $\alpha = 1$, this means that if you deliver your project by July you can get the full 8 points
- September 30, 2018, $\alpha = 0.5$, this means that if you deliver the project after July 31, 2018, but before September 30, 2018, you can get up to 4 points
- after September 27, 2018, $\alpha = 0$, this means that you don't get any additional points after this deadline.

The students willing to take the project assignment must choose among the project proposals listed in the following section or propose an original project and write to alessandroenrico.redondi@polimi.it describing the project you intend to implement.

2 Proposed Projects

You will have to deliver the following items:

- Complete source code of the project
- Self-explanatory log file, showing that your project works. Try to be as detailed as possible when preparing the log file (i.e., use debug/print statements in all the crucial phases of your project)
- Project report (max. 3 pages), which summarizes your approach in solving the problem, including figures when needed. Don't include source code in the project report.

Projects will be evaluated based on the rationale and technical depth of the design choices, correctness of the source code and organization and clarity of the project report.

2.1 Project 1. pub/sub based sensor network - up to 8 points

You are required to implement a sensor network acting according to a pub/sub paradigm similar to the one used by MQTT and test its correct operation using a simulator (TOSSIM or COOJA).

The requirements of this project are:

1. Create a topology with 8 nodes (1 broker and 7 clients).
2. Implement broker functionalities for the following MQTT operation: CONNECT/CONNACK, SUBSCRIBE/SUBACK, PUBLISH (with QoS 0 and QoS 1). You can describe all other MQTT related stuff (e.g., RETAIN flag, LWT, QoS 2 functionalities etc.)
3. Test the operation of the system with client subscriptions and publish on any topic (you can assume only topic WITHOUT wildcards)
4. Implement rule-based subscription functionalities. Assuming a client publishing numbers on any topic (e.g., /temperature), another client can subscribe to `>value/temperature` and receive only messages whose payload is greater than `value`.

2.2 Project 2. LoraWAN-like sensor networks - up to 6 points

Implement and showcase a network architecture similar to LoraWAN. The requirements of this project are:

1. Create a topology with 5 sensor nodes, 2 gateway nodes and one network server node, as illustrated in Figure 1.
2. Each sensor node periodically transmits (random) data every 30 seconds, which is received by one or more gateways. Gateways just forward the received data to the network server.
3. Network server keeps track of the data received by gateways, taking care of removing duplicates. An ACK message is sent back to the forwarding gateway, which in turn transmits it to the nodes. If a node does not receive an ACK within a 1 second window, the message is retransmitted.
4. The network server node should be connected to Node-RED, and periodically transmit data from sensor nodes to the Node-Red dashboard, where it can be displayed.

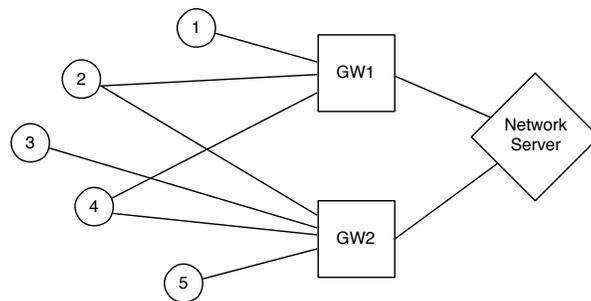


Figure 1: Network topology.

2.3 Project 3. Data collection with Thingspeak - up to 4 points

In this project, you are required to implement a system for data collection using TinyOS, Node-RED and Thingspeak. The requirements of this project are:

1. Create TWO simulated WSN, each one with a sink node and two sensor nodes (one equipped with a temperature sensor and the other with a humidity sensor, you can use the provided `TempHumSensorC.nc` component, whose operation is demonstrated in the `SensorTestAppC.nc` application). Simulate the two WSNs in Cooja attaching EACH sink node to a Node-RED socket (use the SERVER serial socket tool in Cooja). Data should now be available in Node-RED.
2. Use the functionalities provided by Node-RED to transmit the data from the simulation to ThingSpeak. Each WSN should be linked to a different channel, and each sensor node in each WSN should be logged on a different Field.
3. Use Node-Red functions to read the data from ThingSpeak and write a function to send an alert email when the average value of the Temperature from the two WSN exceed a predefined threshold.