

## 2.1 Project 1. Lightweight publish-subscribe application protocol

You are requested to design and implement in TinyOS a lightweight publish-subscribe application protocol similar to MQTT and test it with simulations on a star-shaped network topology composed of 8 client nodes connected to a PAN coordinator. The PAN coordinator acts as an MQTT broker.

The following features need to be implemented:

1. Connection: upon activation, each node sends a CONNECT message to the PAN coordinator. The PAN coordinator replies with a CONNACK message. If the PAN coordinator receives messages from not yet connected nodes, such messages are ignored. Be sure to handle retransmissions if msgs get lost (retransmission if CONN or CONNACK is lost).
2. Subscribe: After connection, each node can subscribe to one among these three topics: TEMPERATURE, HUMIDITY, LUMINOSITY. In order to subscribe, a node sends a SUBSCRIBE message to the PAN coordinator, containing its node ID and the topics it wants to subscribe to (use integer topics). Assume the subscriber always use QoS=0 for subscriptions. The subscribe message is acknowledged by the PANC with a SUBACK message. (handle retransmission if SUB or SUBACK is lost)
3. Publish: each node can publish data on at most one of the three aforementioned topics. The publication is performed through a PUBLISH message with the following fields: topic name, payload (assume that always QoS=0). When a node publishes a message on a topic, this is received by the PAN and forwarded to all nodes that have subscribed to a particular topic.
4. You are free to test the implementation in the simulation environment you prefer (TOSSIM or Cooja), with at least 3 nodes subscribing to more than 1 topic. The payload of PUBLISH messages on all topics can be a random number.
5. The PAN Coordinator (Broker node) should be connected to Node-RED, and periodically transmit data received on the topics to Things-

peak through MQTT. Thingspeak must show one chart for each topic on a public channel.

**Final report** The final report must include the full TinyOS code, the log of the simulation (.log file in case of TOSSIM and some relevant screenshots for Cooja), the Node-Red code export (include also the details in case extra packages have been used), the ThingSpeak public channel URL, and a report that explains the approach and the assumptions done.