CS4628 - P2 - MULTIHOP EDGE NETWORK

Overview

The task in this practical is to build a wireless multi hop edge network based on the simulation environment *microbit_sim* introduced in the first lab. The network architecture for this exercise is shown in Figure 1. You can use the code from the first practical as a starting point to construct your software (correct code available of Canvas). A message generated by a node is transmitted wirelessly to the sink. Messages may have to be forwarded by other nodes to reach the sink. The sink receives the message and prints it out (For now we do not look at implementing a backend located in the Internet).

The nodes should form a DODAG, a destination-oriented directed acyclic graph so that messages can be forwarded by each node towards the sink. A directed acyclic graph does not have cycles; A DAG is a destination-oriented (DODAG) when there is a directed path in the DAG from any node to the sink.

A DODAG can be constructed as follows, using Figure 2 as example. The sink starts advertising its presence with a broadcast message (e.g. a JOIN message) which is received by node 6/5/2. In this message the sink advertises its rank value as 1. Node 6/5/2 update their rank to 2 (one plus the rank value received) and mark the upstream node as the sink (node 1) as shown in Figure 3.

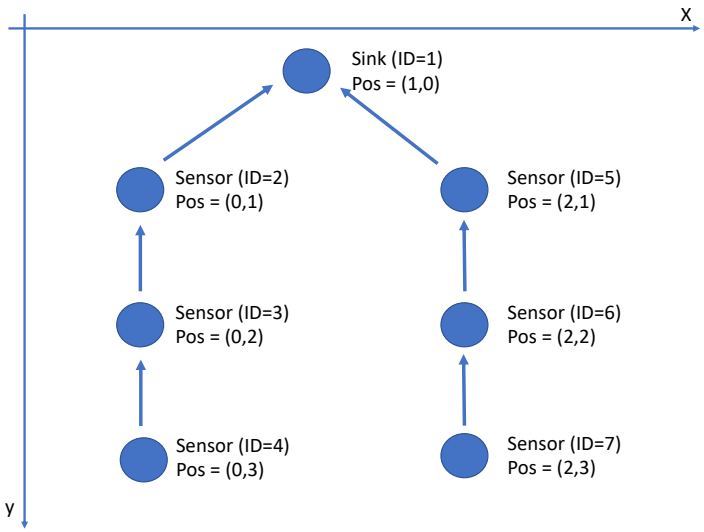

Figure 1. CS4628 IoT Architecture

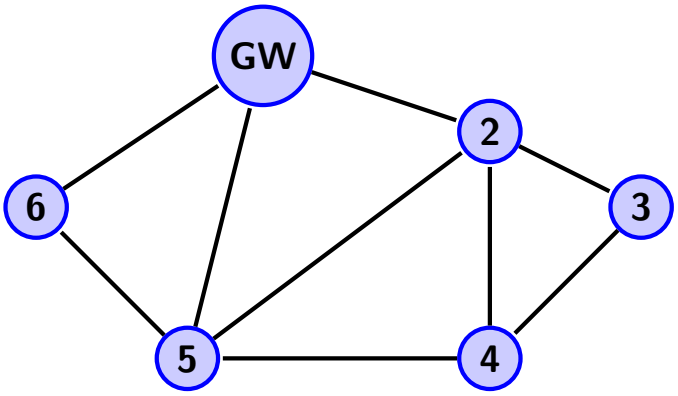

FIGURE 2. DODAG example.

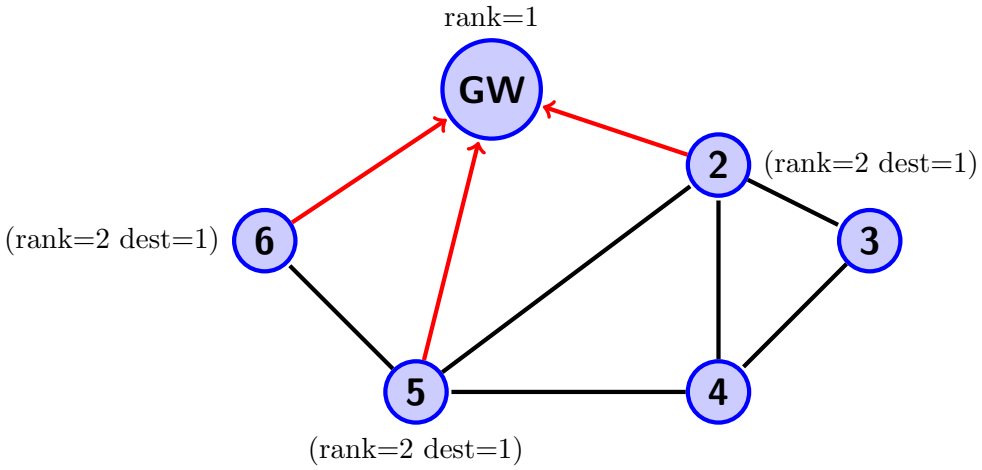

FIGURE 3. DODAG example, after JOIN broadcast from sink (GW).

Nodes 6/5/2 can now broadcast a JOIN message of their own. The JOIN transmitted by node 6 will be received by node 5 and the sink. These nodes will ignore the message as their rank value is better than the received rank (meaning closer to the sink). The JOIN transmitted by node 5 is received by nodes 6/2/4 and the sink. Node 4 will now set its rank value to 3 and the destination to 5 for message forwarding; all other nodes ignore this message as the rank is lower than their own. Now node 2 transmits the JOIN message which updates node 3 with rank 3 and destination 2; node 5 and 4 ignore the message. Finally node 3 sends a JOIN message which is ignored by everyone as no better rank can be achieved. The resulting DODAG is shown in Figure 4.

Nodes should not send the JOIN messages based on a previously received JOIN message as this would create a chain reaction (a broadcast storm). Instead nodes should use a periodic timer to transmit a JOIN; they should only transmit if their rank is set (i.e. they have attached to the network).

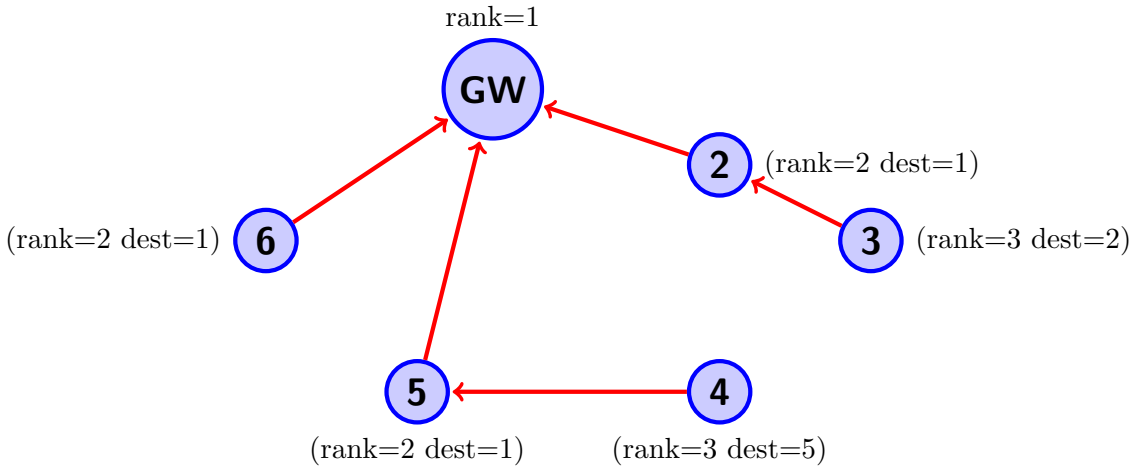

FIGURE 4. DODAG example, after JOIN broadcast from sink (GW) and all other nodes after that

When nodes receive a JOIN message with a lower rank value they should update their parent node and rank as a better topology is found, e.g. less hops to the sink. When nodes do not receive a JOIN message from their parent anymore, the parent node may have left and nodes should then consider any received JOIN message to attach themselves to the network again.

A node will send periodic temperature readings to the parent node which will then forward it along the path to the sink. Use the data field (payload) to record the routing path; i.e. each forwarding node should add its ID to the data field. Thus, when messages are received at the sink the topology can be determined (As a debugging feature).

Nodes should print the current rank (use rank 0 for a node that has not found a parent).

Messages are JSON encoded (they are a simple string); an example message type is DATA:

```
{"TYPE": "TEMP",
  "SRC": 3,
  "DST": 6,
  "LSRC": 3,
  "LDST": 6,
  "SEQ" : 4,
  "DATA": 31}
```

LISTING 1. Messages

For testing purposes try to implement an event in the simulator that can change a node configuration. For example, you could simulate that a node is repositioned after some time or that it changes the transmission range to see if your implementation can deal with a sudden change in the topology.

CS4628 Continuous Assessment - PART 2

Please submit an answer to the following question as Assignment 2. Your answer should not be longer than ONE page (You can use figures or code pieces to illustrate your answer).

Question P2 [4 MARKS]: Transmission Probability

Assume all nodes in a network use a slotted communications protocol. The synchronisation is provided by a coordinator node, but how this works is not important here. The nodes DO NOT use a listen before talk protocol to check if a slot is free. The probability for a node to transmit in a slot is p. There are N nodes. What is the probability of a successful transmission in a slot, assuming that a collision of one or more transmissions in a slot results in an unsuccessful transmission? Draw a graph depicting the probability of successful transmission dependent of the number of nodes N. Explain your answer.