IoT 2023 Challenge 3

Pasquale Castiglione
10657816

Lorenzo Campana
10605775

Introduction

The goal of this challenge was to implement a routing algorithm in order to be able to send a packet from Mote1 to Mote2.

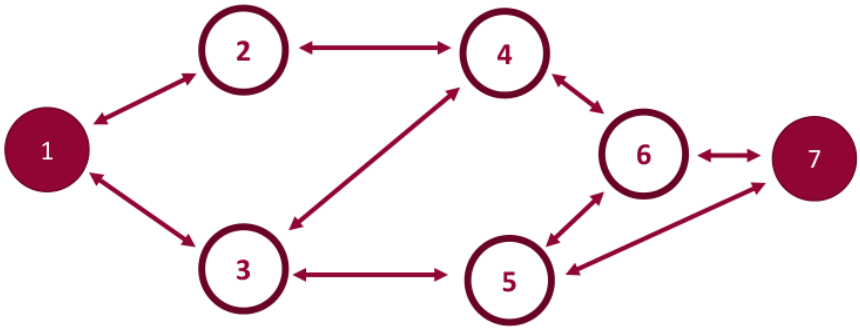

Figure 1: Network Topology

Code

RadioRoute.h

In the header file, the data structures for the messages and for the routing table were defined.

```
typedef nx_struct radio_route_msg {
        nx_uint16_t type;
        nx_uint16_t sender;
        nx_uint16_t destination; // node_requested
        nx_uint16_t value; // cost
} radio_route_msg_t;
```

```
typedef struct route_entry_t {
        uint16_t next_hop;
        uint16_t cost;
} route_entry_t;
```

RadioRouteC.nc

Variables

`waiting_packet` and `waiting_address` were defined in order to store the content and the destination address of the message to be sent when the routing table is empty.

Functions

- `initializeRoutingTable`
- `addRoutingEntry`
- `dbgPacketInfo`

Send Event

The sending of a message is implemented in the `actual_send` function. Before the sending of a message, the variable `locked` is evaluated in order to check if the radio module is in use or not. If the radio is free, the sending node checks if a route for the destination is present in its routing table. In case of data message with null entry in the routing table, `waiting_packet` and `waiting_address` are set and a routing request sent in broadcast.

Receive Event

Received messages are handled according to their type:

- **Data Message**
  If the receiver is not the intended destination of a message then it checks if a route to the destination is present in its routing table. If a route is present then it forwards the message to the next hop.
- **Route Request**
  If the receiver node is the intended destination of a route request or if the node has a non empty entry in the routing table, then it answers back sending a `Route Reply` broadcast message. Otherwise it broadcast again the route request message to its neighbors.

- **Route Reply**
  When a `Route Reply` is received the nodes check their routing table and they update them if the received cost for the destination is lower than the stored one. Also, the nodes check weather there is a `waiting_packet` with a `waiting_address` equals to the updated entry destination address. In this case, it proceeds with the packet forwarding to the updated next hop.

Led Control

Results