

IoT 2023 Challenge 3

Pasquale Castiglione
10657816

Lorenzo Campana
10605775

1 Introduction

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

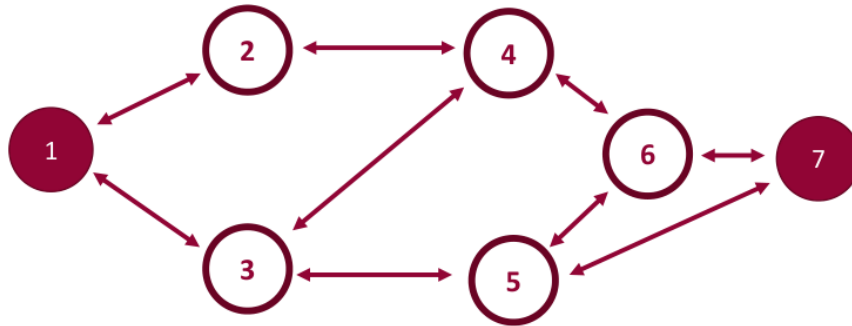


Figure 1: Network Topology

2 Code

2.1 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;
```

2.2 RadioRouteC.nc

Two variables: `waiting_packet` and `waiting_address` were defined in order to store the message and the destination address to send when the routing table was empty. At the beginning, right after the boot, the radio module was activated and a five seconds timer was started for the Mote 1.

2.2.1 Send

Before the sending of a message, the variable `locked` is checked in order to check if the radio module is in use or not. If the radio is free,

2.2.2 Receive

Received messages are handled according to their type:

- **Data Message**

If the receiver node is not the intended destination 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 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 update it if the received cost for the destination is lower than the stored one.

2.3 Led Control

3 Results