IoT Project: "Keep your distance"

Prof. Matteo Cesana - Year 2020/2021

Gabba Rohit [codice persona: 10706944]
Tortorelli Giuseppe [codice persona: 10582962]

1 TinyOS

In order to keep track of the consecutive messages we decided to identify them with an incremental identifier.

The message struct is composed of the fields:

- *id*: the id of the sender mote
- *msg_number*: the id of the message sent from that mote

We decided to assign three arrays of fixed length to each mote:

- *near_motes_ids*: to save the id numbers of the near motes
- *near_motes_counters*: to save the number of consecutive messages received from that mote
- *near_motes_msgs_number*: to save the id number of the message that we expect to receive from that mote

Every time a mote sends a message, the id to be assigned to the message is incremented.
At the receiving end of a message, the *near_motes_ids* array is scaned to find the id of the sender mote.

If the the Id is not found then it is added to the first spot available in the array; otherwise we procede by checking the id of the message received.

We compare the expected id number from *near_motes_msgs_number* array with the value received from the mote. If the two values don't match then the counter that keeps track of the consecutive messages is restarted with the value 1.

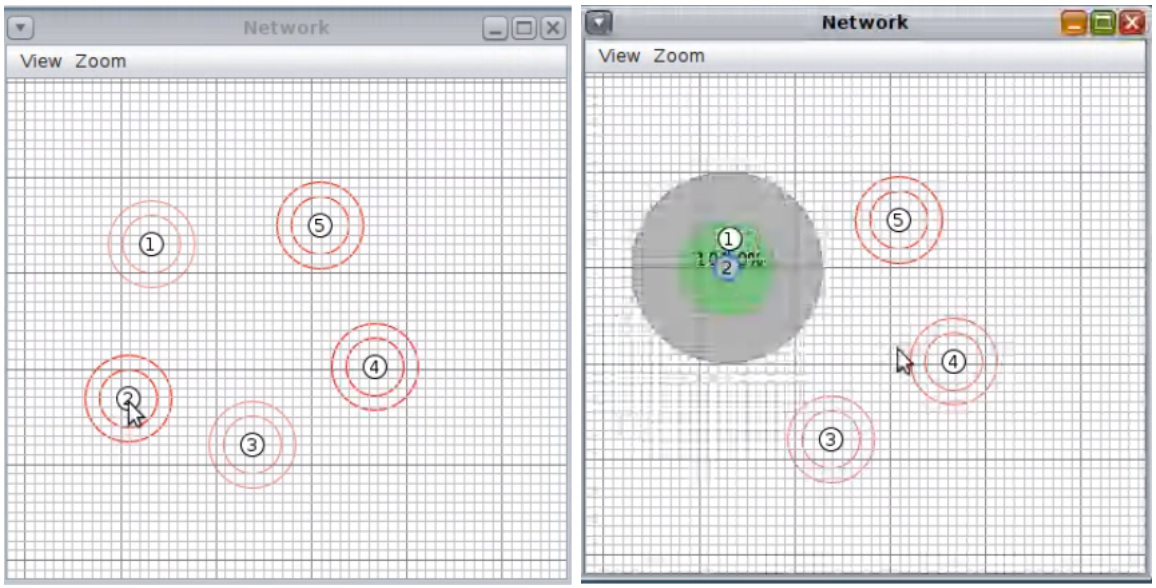
Otherwise check the value of the *near_motes_counters* array: if 10 consecutive messages have been received then an alarm is launched and the counter is restarted, otherwise the counter is incremented.

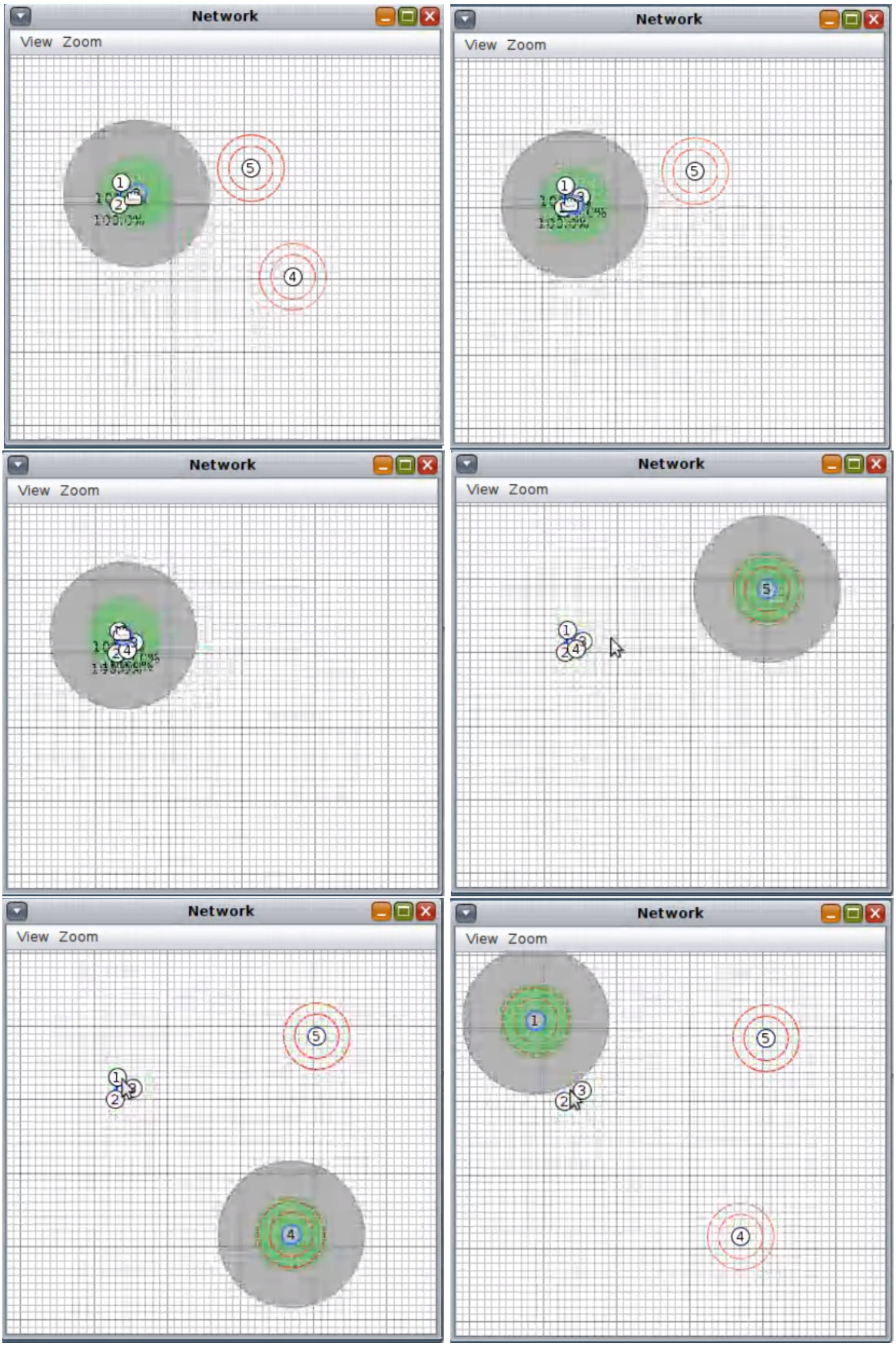
2 Cooja

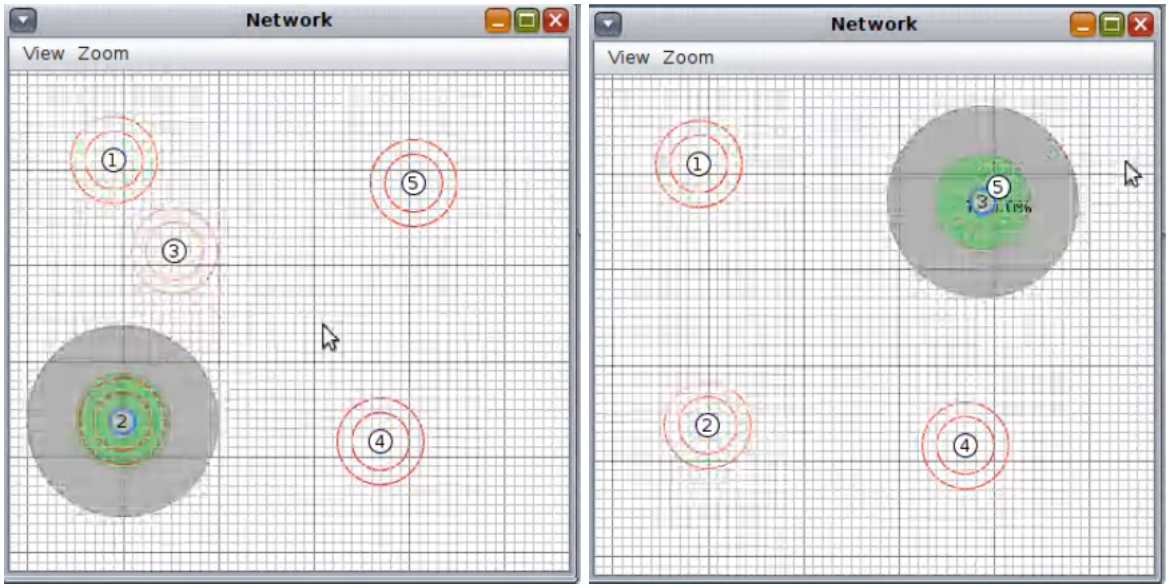
We tested the functioning of the system by simulating it in *Cooja* with 5 *Sky motes* and trying various configurations.

During the simulation we assumed that the messages are received without any interferences, therefore we ignored the cases where the counter is automatically reset because of a non correct delivery of a message.

Here are some screenshots of the configuration tried during the simulation (ordered as in the log file)




Network
View Zoom
Network
View Zoom
Network
View Zoom
Network
View Zoom
Network
View Zoom
Network
View Zoom



We have assigned a socket to each mote with a different port number (from 6001 to 6005) towards the ip of *Node-RED*.

3 Node-RED

We created 5 (number of motes chosen for the simulation) *tcp in* nodes, each of them listening on the port of the associated mote.
Then we used the *function* node to filter the alarm messages from the log sent by *Cooja* and extract the Ids of the involved motes.
Finally we used the *htttp request* node to send a POST request towards *IFTTT* by specifying our API key.

4 IFTTT

Once received the messages through a *Webhooks* applet listening on the *ALARM* event, we sent a notification on *IFTTT* app for the cellular phone containing the message:

"ALARM mote x: I'm close to mote y! COVID"

Wheree x is the id of the mote that sent the alarm.