POLITECNICO DI MILANO

DEPARTMENT OF ELECTRONICS, INFORMATION AND BIOINGINEERING

M.Sc. Course in

Internet of Things

Final Project

Keep Your Distance

Professor: Dr. Matteo Cesana

Tutor: Edoardo Longo

Students:

Hiva Amiri	10696153
Erfan Abbasi Zadeh Behbahani	10628157
MohammadJavad Ebrahimpour	10696374
https://github.com/hivaamiri/_polimi_iot/tree/master/_iot/Final_KeepYourDistance	Githublink

ACADEMIC YEAR 2019-2020

Objective

You are requested to design and implement a software prototype for a social distancing application using TinyOS and Node-Red and test it with Cooja. The application is meant to understand and to alert you when two people (motes) are close to each other. The operation of the software is as follow:

- Each mote broadcasts its presence every 500ms with a message containing the ID number.

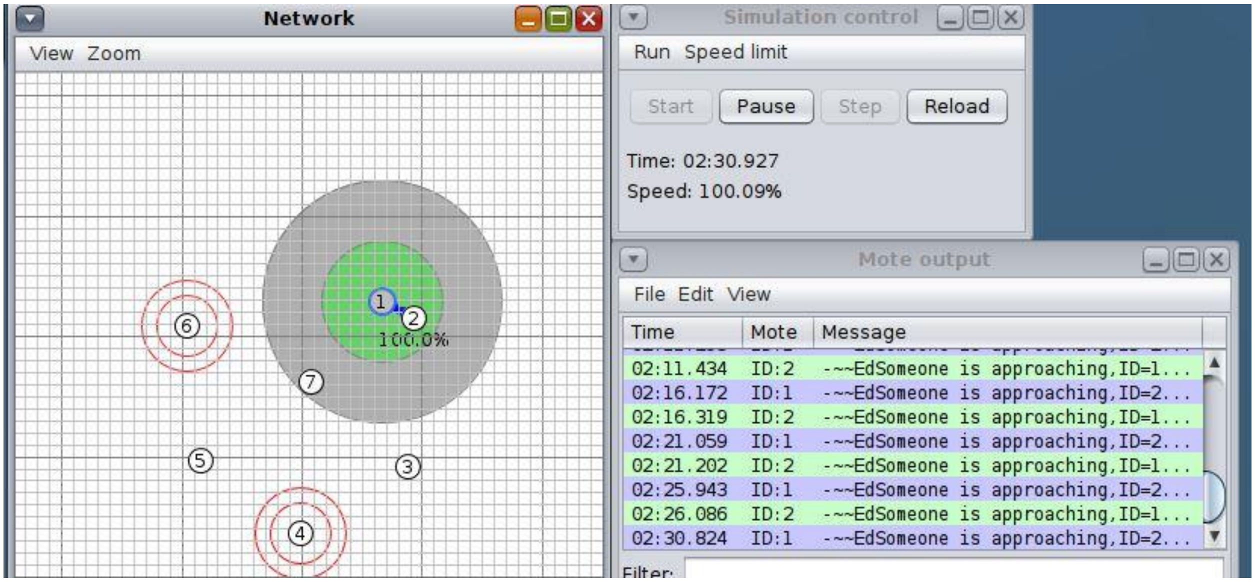
- When a mote is in the proximity area of another mote and receive a message, it (1) stores the received mote ID and (2) triggers an alarm. Such alarm contains the ID number of the near mote. It is shown in Cooja and forwarded to Node-Red with a different socket for each mote.

- Upon the reception of the alert, Node-red sends a notification through IFTTT1 to your mobile phone. Use at least 5 motes. Start the simulation with all the mote far away from each other an move them with the mouse testing different configurations.

Approach

The project of Activity 2 and Activity 5 are used to develop required objective. Most of the functions are similar to second hands-on activity which its source and documentation is accessible via this link.

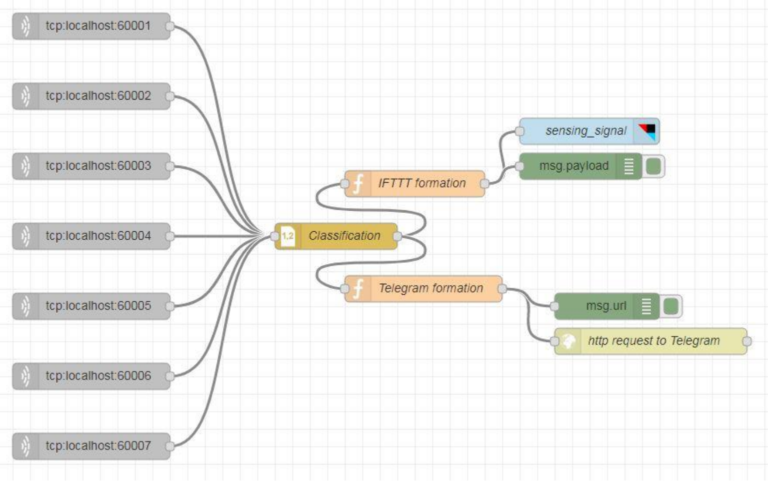
Simply, each mote broadcast its ID using TOS_NOD_ID constant over the channel, the other motes in the range will receive the message and can see who is close to them, actually who is in their radio range. Using a `printf` function on receiver side of each node, the ID of the sender and the motes in the vicinity will be printed on the serial. Since we are using Cooja simulator, the printed message is sent to Nodered using a serial server. The topology of the simulated network is shown on the picture which in this case both mote 1 and 2 are sensing each other and they are sending message to the serial output.



Time: 02:30.927
Speed: 100.09%

```
Time       Mote  Message
02:11.434  ID:2  -~~EdSomeone is approaching,ID=1...
02:16.172  ID:1  -~~EdSomeone is approaching,ID=2...
02:16.319  ID:2  -~~EdSomeone is approaching,ID=1...
02:21.059  ID:1  -~~EdSomeone is approaching,ID=2...
02:21.202  ID:2  -~~EdSomeone is approaching,ID=1...
02:25.943  ID:1  -~~EdSomeone is approaching,ID=2...
02:26.086  ID:2  -~~EdSomeone is approaching,ID=1...
02:30.824  ID:1  -~~EdSomeone is approaching,ID=2...
```

On the Nodered side, it listens to the TCP port number 60001 to 60007 which these ports are trying to communicate with Cooja serial server and read the data from mote 1 to 7 respectively. Based on the printed format which we defined on TinyOS source codes, a CSV function converts the value to javascript objects in two fields named “val and sender”. Val refers to the mote ID close to us and sender is the ID of the mote which sends the message to the serial output. The following picture is the scheme of nodes on Nodered.

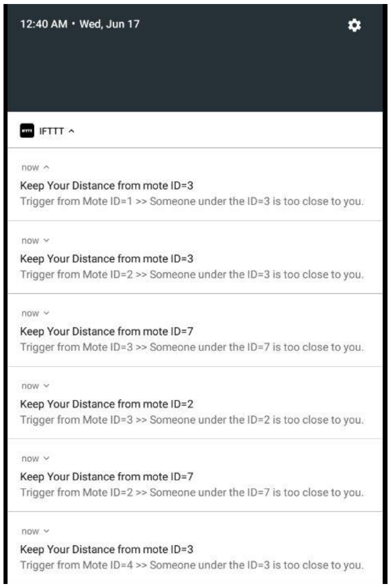


After creating an account on IFTTT, we can use webhooks to trigger a function which in our case, the trigger is an http request from Nodered containing the ID of the motes. In turn, after triggering, we can ask IFTTT to send a notification on our mobile phone, sending an email or make a phone call!

The line of code to send data using http request is

```
msg.payload = {"value1" : msg.payload.val, "value2" : msg.payload.sender}
```

Variables “value1” and “value2” are variables that can be used to send dynamic data to IFTTT which is this case value1 is the ID of the mote close to us and value2 is the ID of the sender of the message on the serial output of the TinyOS. Output of the IFTTT formation function is connected to an IFTTT node which it can be downloaded from Nodered library. The notification can be seen in the picture.



12:40 AM • Wed, Jun 17

IFTTT ^

now ^
Keep Your Distance from mote ID=3
Trigger from Mote ID=1 >> Someone under the ID=3 is too close to you.

now ˅
Keep Your Distance from mote ID=3
Trigger from Mote ID=2 >> Someone under the ID=3 is too close to you.

now ˅
Keep Your Distance from mote ID=7
Trigger from Mote ID=3 >> Someone under the ID=7 is too close to you.

now ˅
Keep Your Distance from mote ID=2
Trigger from Mote ID=3 >> Someone under the ID=2 is too close to you.

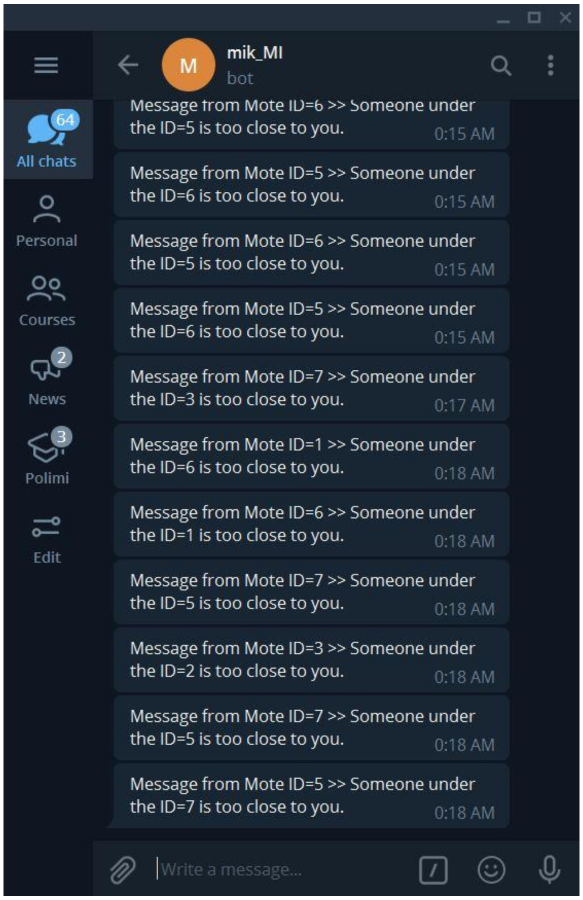
now ˅
Keep Your Distance from mote ID=7
Trigger from Mote ID=2 >> Someone under the ID=7 is too close to you.

now ˅
Keep Your Distance from mote ID=3
Trigger from Mote ID=4 >> Someone under the ID=3 is too close to you.

We put a step further and tried to send message on Telegram messenger using Telegram Bots which by making a bot on Telegram, you are provided with a token that allows you to send messages on bot conversation. The line of code which we used to send message on Telegram bot is

```
"https://api.telegram.org/bot1110030125:AAEk6WnrtoKNpRbGIC6vKZkXKqgmHtZXruI/sendmessage?chat_id=89042906&text=Message from Mote "+sender+" >> Someone under the " +data+" is too close to you.";
```

The "data" and "sender" are values which the TinyOS sends on serial output. Highlighted part is the token we need to send message on bot.The following picture is a screenshot of received message from Nodered.



Message from Mote ID=6 >> Someone under the ID=5 is too close to you. 0:15 AM

Message from Mote ID=5 >> Someone under the ID=6 is too close to you. 0:15 AM

Message from Mote ID=6 >> Someone under the ID=5 is too close to you. 0:15 AM

Message from Mote ID=5 >> Someone under the ID=6 is too close to you. 0:15 AM

Message from Mote ID=7 >> Someone under the ID=3 is too close to you. 0:17 AM

Message from Mote ID=1 >> Someone under the ID=6 is too close to you. 0:18 AM

Message from Mote ID=6 >> Someone under the ID=1 is too close to you. 0:18 AM

Message from Mote ID=7 >> Someone under the ID=5 is too close to you. 0:18 AM

Message from Mote ID=3 >> Someone under the ID=2 is too close to you. 0:18 AM

Message from Mote ID=7 >> Someone under the ID=5 is too close to you. 0:18 AM

Message from Mote ID=5 >> Someone under the ID=7 is too close to you. 0:18 AM

It worth mentioning that there are another way to sent message to IFTTT using http request or executing a command line like the following pattern

```
https://maker.ifttt.com/trigger/EVENT/with/key/VXNHHydKRTQFoUynnmZILc9
```

and for command line

```
curl -X POST https://maker.ifttt.com/trigger/EVENT/with/key/VXNyHHdKRTQFoUnnmZyILc9
```

which both are using POST, GET attributes of http request to do it but, we preferred to use built-in nodes on the project.