POLITECNICO DI MILANO

Keep your distance

Surname: Mohamed

First name : Hesham

ID:10654249

Surname: Hamed

First name : Ahmed

ID:10682755

Supervised by: Prof Edoardo Longo

I- Introduction:

In order To deal with the social distancing problem, this project aim to design and implement a software solution using TinyOs , Node-Red and simulated using Cooja.

The software should have the following requirement

1- Each mote should broadcast a message with its ID every 500ms.
2- Each mote should send an Alert message when receiving other broadcast messages from the other motes.
3- Each mote should send a notification to the owner of the mote on his personal phone with the ID of the other motes using Node Red and IFTTT
4- Each mote should save the data received from the other motes ,containing the Sender ID and the time.

II- Summary

In this project I will discuss how to implement such a requirement using Nesc , Cooja, Node-Red in the following

III- Nesc:

a. Interfaces:

Interface	Component	Function
Boot	MainC	Interface that notifies components when TinyOS has booted (initialized all of its components).
Receive	AMReceiverC	Interface for handling received messages
AMSend	AMSenderC	Interface for handling messages to be sent
AMControl	ActiveMessageC	Interface that is used for switching between the on and off power states of the component providing it
MilliTimer	TimerMilliC	Interface that can fire events according to the user requirement
Packet	AMSenderC	Interface that is used to prepare messages
	PrintfC	Used for Debugging purposes and used by cooja to relay messages to node Red
	SerialStartC	

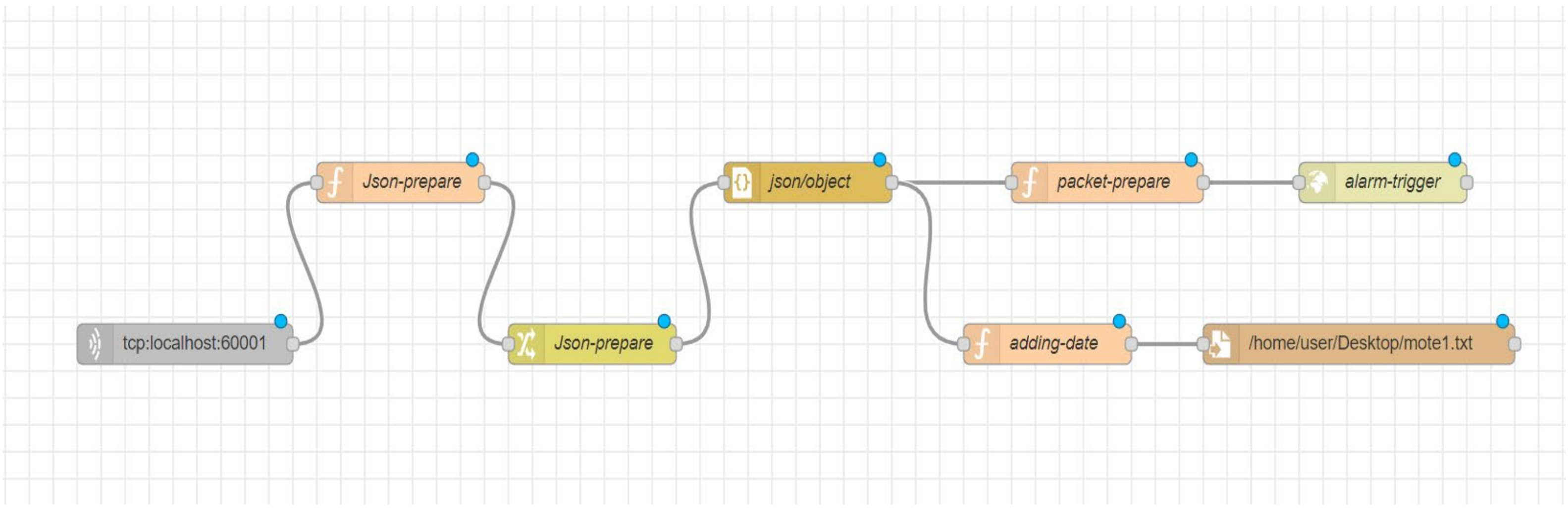
b. **Nesc Algorithm**: The implemented algorithm is pretty simple ,each mote send a broad cast message with his own ID and in reception from other mote it relay this message to Cooja .

IV- Cooja:

a. In Cooja we create a new simulation
b. Adding 5 sky motes and load the .Exe file generated from compiling the Source code
c. Assigning a different Server Socket to each mote
d. Start the simulation.

V- Node-Red:

In node-red we take the received messages from nesc and preprocess this data in order to send a notification to the user mobile phone and store the data in json format , the nodes used are the following,



1- Tcp in node :to read the received packets from the other motes
2- Json-prepare nodes: to preprocessing the messages by removing the garbage generated by Cooja and prepare the message to be transformed into Json Format
3- Json/object Node: to convert from Json string to the javascript object format
4- Packet-prepare Node: to set up the message header and content for the http request
5- Add-date Node: it’s a function to add the date to the messages before storing it in a file
6- Alarm-trigger node: its an http post request in order to generate a notification for the user

VI- Recommendation :

The Data being generated at very fast pace which may lead to memory problem and sending the data to a sink will add more power consumption , my recommendation to solve this problem is to process the data periodically to generate more compact and useful information from these data , with very simple queries we can get to know how much time spent with each person . for example assume that a user tested positive for COVID19 , the most valuable information we can get is to know the people he/she have met and how much time they have been together for each person in order to stop spreading it.