Al Najah National University
Second semester 2017/2018
Final Project

IoT enabled Autonomous Smart Water-meter Networking

Supervisor:
Dr. Othman M.Othman

Students:
Yara Al-Qadah
Lama Abu Zahid
Tharwat A’araj

ABSTRACT
Traditional water meter reading is a time consuming and error prone process. For such reasons, the world is moving towards smart water meter reading based on **Internet of things (IoT)**. Which is an emerging field in research and industry that makes use of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity. Which enables these objects to connect and exchange data. This paper focuses on the communication between smart water meters. By building a self organizing network that collects data from water meters in the same area using radio communication module by dividing the area to clusters using K-means algorithm. Then, the node that has the highest number of connected nodes and has WiFi connection will be selected as a cluster head (CH) so it will collect the data from other nodes. Then transfer the collected data over the Internet to the server using its WiFi module. This process will be done within minimal delay.

INTRODUCTION
The world is moving towards automating things using IoT to make life easier and more comfortable. For example, many countries are looking at Smart Grid as an advanced solution for delivering power with lower cost and energy efficiency. In definition, Smart Grid is a form of electricity network utilizing digital technology. It connects between suppliers, distributors and consumers to deliver electricity from suppliers to consumers in a smart way that save energy, reduce costs and increase reliability. In the same manner that IoT is used to enhanced the distribution of electricity; this project will focus on using IoT to enhance water-meter reading.

Many papers were written on this topic. For example, in [1] the paper proposes a wireless meter reading system by using both the multilevel relay and the concentrator based on LoRa communication. However, LoRa communication has limited network size based on parameter called as duty cycle[2]. It is defined as percentage of time during which the channel can be occupied. Also, [3] try to perform automated water-meter reading for updating the consumption information from field to the database utility for billing and payment. This smart water-meter approach differs from traditional methodologies by using simple IoT hardware in conjunction with smartphone app. But, this paper focus on the hardware more than the communication between meters or the used protocol. At [4] developed communication in wireless systems for water metering by using ZigBee which provides more accuracy than the traditional analog water meters, but it is more expensive. Also, using 6LoWPAN protocol which 6LoWPAN allows the transmission of IPv6 packets over IEEE 802.15.4 links. Besides, it performs encapsulation and header compression of IPv6 packets in order to allow the participation of low power devices in IOT. But 6LoWPAN protocol could not be deployed easily because it requires extensive knowledge of stack and the workability of IPv6 [5].

This project aims to resolve the mentioned problem, by proposing an ad hoc communication network that connects the smart water meters using heterogeneous nodes which use two communication facilities (RF and WiFi). First, the RF network is used because of its low power consumption, which makes it suitable for propagating readings towards a sink node that is connected to the Internet. While the WiFi network is used due to its high availability in homes, high throughput, large coverage area and thus it can act as a sink to send the data directly to server on the Internet. However, it has high power consumption.

Moreover, the proposed solution supports self-organizing network that enables the two networks to work in a complete harmony as if they are a one network. This is done in a way that helps to overcome any changes in the network and continue to work without any human intervention.
The communication between these nodes will be done by dividing the network into clusters using k-means clustering which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

DESIGN

In this paper two main wireless communication modules were used. The first module, is nRF24L01 [6] which is a single chip radio transceiver for the world wide 2.4 - 2.5 GHz ISM band. The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator, a demodulator, modulator and Enhanced ShockBurst™ protocol engine. Also, it has low current consumption only 9.0mA at an output power of -6dBm and 12.3mA in RX mode and has a range of 100 meters.

The second module, is ESP8266 [7] is a low-cost Wi-Fi chip it can work on 3-3.6V at maximum, it is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to a Wi-Fi network in order to meet users' continuous demands for efficient power usage, compact design and reliable performance in the IoT industry. Also, ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Moreover, it has a range of 366 meters. These two modules will be connected using Arduino Uno [8], which is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. The connection is shown in (Figure 1).

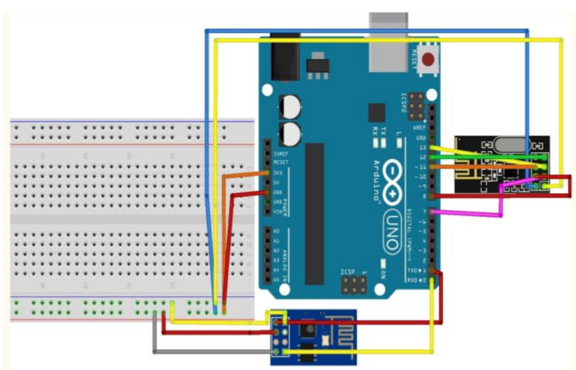

Figure 1 (arduino,ESP8266 and nRF24L01 node connection)

First of all we have some assumptions that we have to clarify. We assume that the power will be supplied to the modules using water generator so our main focus will be to build autonomous network that can work without user intervention. So, our main focus will be on the number of connected nodes and wifi connection.
This project will be built using clustering. So, the nodes will be divided into some clusters and then some nodes, called cluster-heads, will be selected to be the head of theses clusters. This node will be responsible for collecting data from other sensors in its cluster and send these data to the Base Station.

Many methods has been set to choose the CH for example, it can be selected randomly and change it periodically between the cluster nodes. Also, CH can be selected based on power to reduce the energy utilization and network performance when all the sensor nodes of the network are sending data to the base station or central collection centre. However, in our project the CH will be elected based on two parameters. First, the number of connected nodes in order to decrease the transmission power. Second, the WiFi connectivity.

The protocol begins by sending **broadcast message** (Figure 1.1) which contains node ID to all nodes (Figure 2.a) in the same coverage area. Then, each node calculate the number of connected nodes and whether it has a WiFi connection or not (Figure 2.b), and send this value to **initial node** which is the node with the least ID (Figure 2.c) using **connected message** (Figure 1.2). After that, this node based on the received information it will choose an **initial CHs** (Figure 2.d) which are the highest, lowest and the middle number of connected nodes Then, the initial node will sends the ID of initial CH to all the nodes by sending **CH message** (Figure 1.3).

Source ID (4 byte)	Destination ID (4 byte)	Packet size (2 byte)	Message type (1 byte)

Figure 1.1

Source ID (4 byte)	Destination ID (4 byte)	Data(# conn node, WiFi) (4 byte)	Packet size (2 byte)	Message type (1 byte)

Figure 1.2

Source ID (4 byte)	Destination ID (4 byte)	Data(CH ID) (4 byte)	Packet size (2 byte)	Message type (1 byte)

Figure 1.3

Every node check the received message if it contains its ID or not. And according to this, if its the same; this node is the initial CH.
After the broadcast stage. The protocol will apply k-means algorithm to divide the network into clusters. So, every node calculate the distance between every node and the initial CHs. Then, clusters will be formed based on the minimal distances between the nodes and the initial CH(Figure 2.e). **CH** is chosen based on the highest connected nodes and WiFi availability (Figure 2.f). After forming clusters, the nodes that are not in the same coverage area of the CH will elect a **secondary CH** that is connected to one of the CH intermediate node and has the maximum number of connected nodes. This step will be repeated until coverage. Then, each node send **collect message** (Figure 1.4) which contains water meter reading to secondary CH then send it to main CH (Figure 2.g) which collect data and send it directly to the server using the WiFi module (Figure 2.h).

Source ID (4 byte)	Destination ID (4 byte)	Data(water meter reading) (4 byte)	Packet size (2 byte)	Message type (1 byte)

Figure 1.4

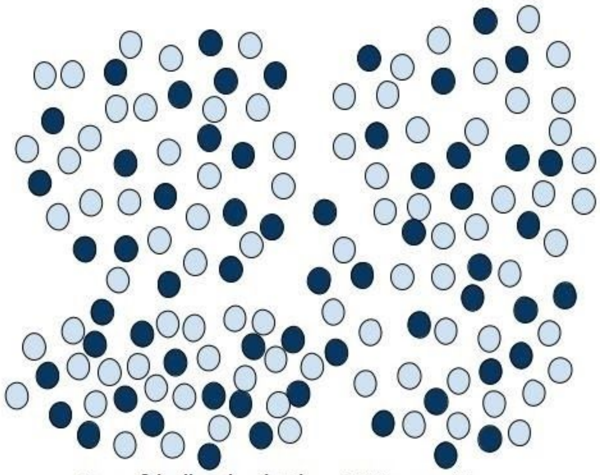
Figure 2.b all nodes that have WiFi connection

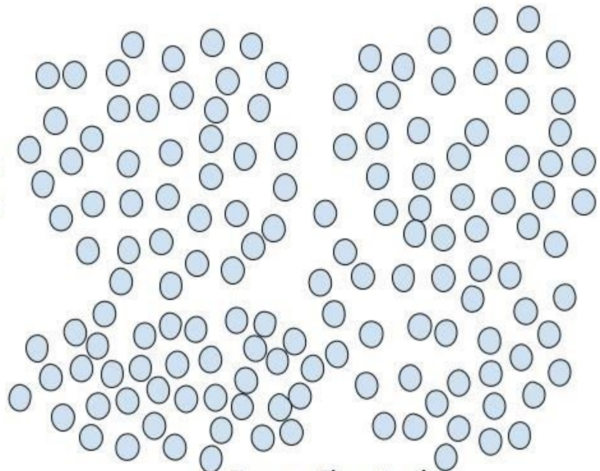
Figure 2.a The network

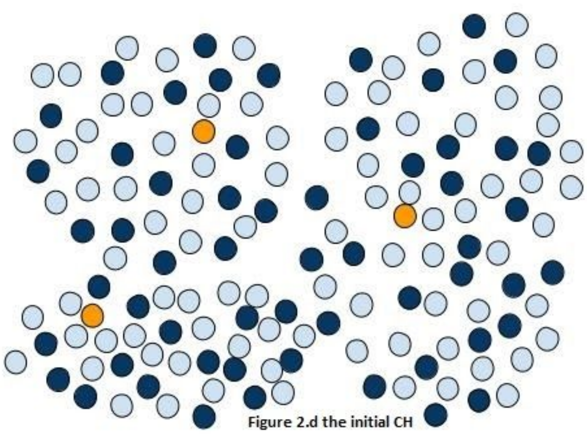
Figure 2.d the initial CH

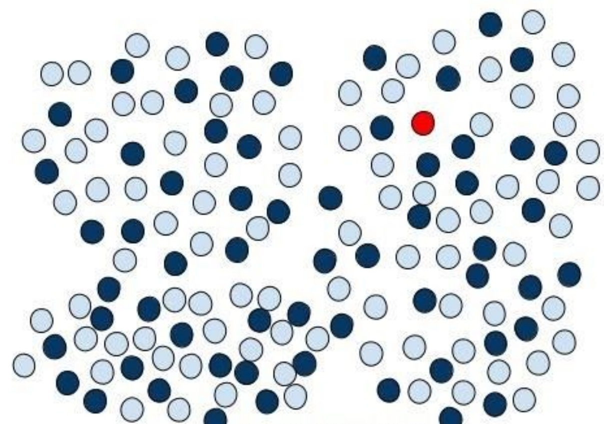
Figure 2.c the initial node

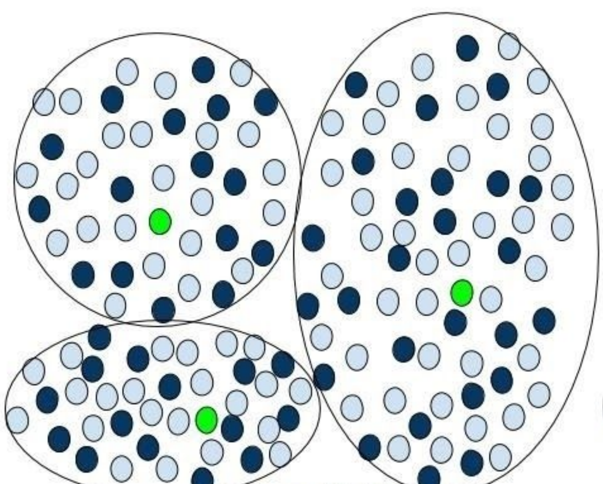
Figure 2.f the CHs

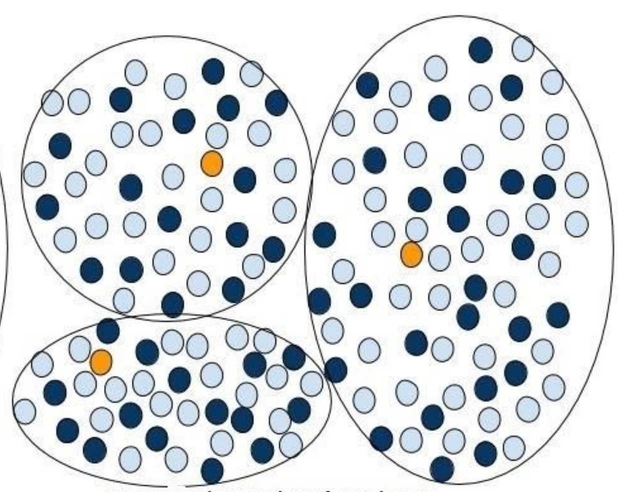
Figure 2.e the initial CHs form clusters

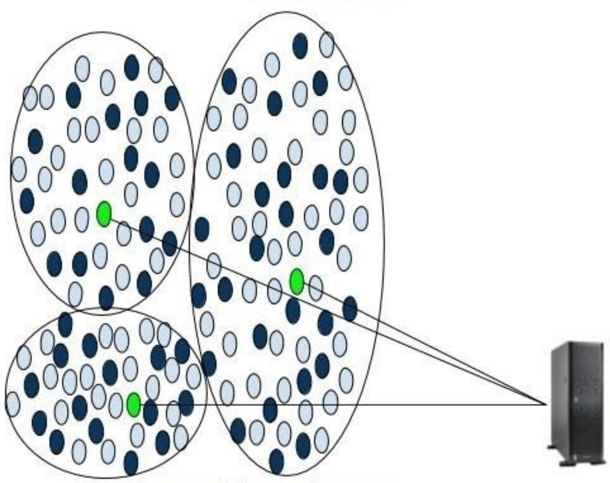

Figure 2·h CHs send data to the server

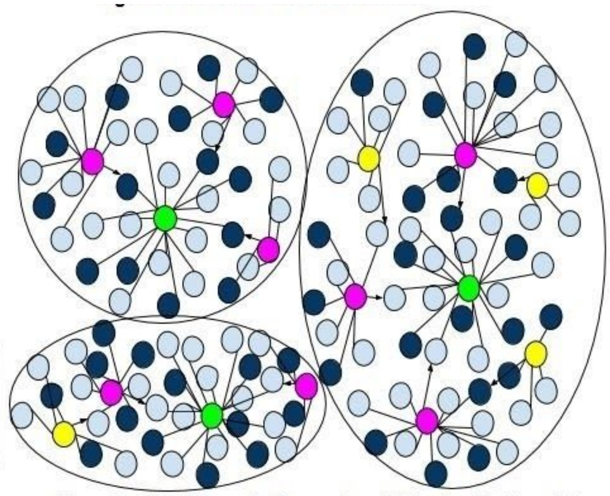

Figure 2.g CHs, 1st ans 2ed secondery CH start collecting data

Figure 3 shows the flow chart for this protocol.

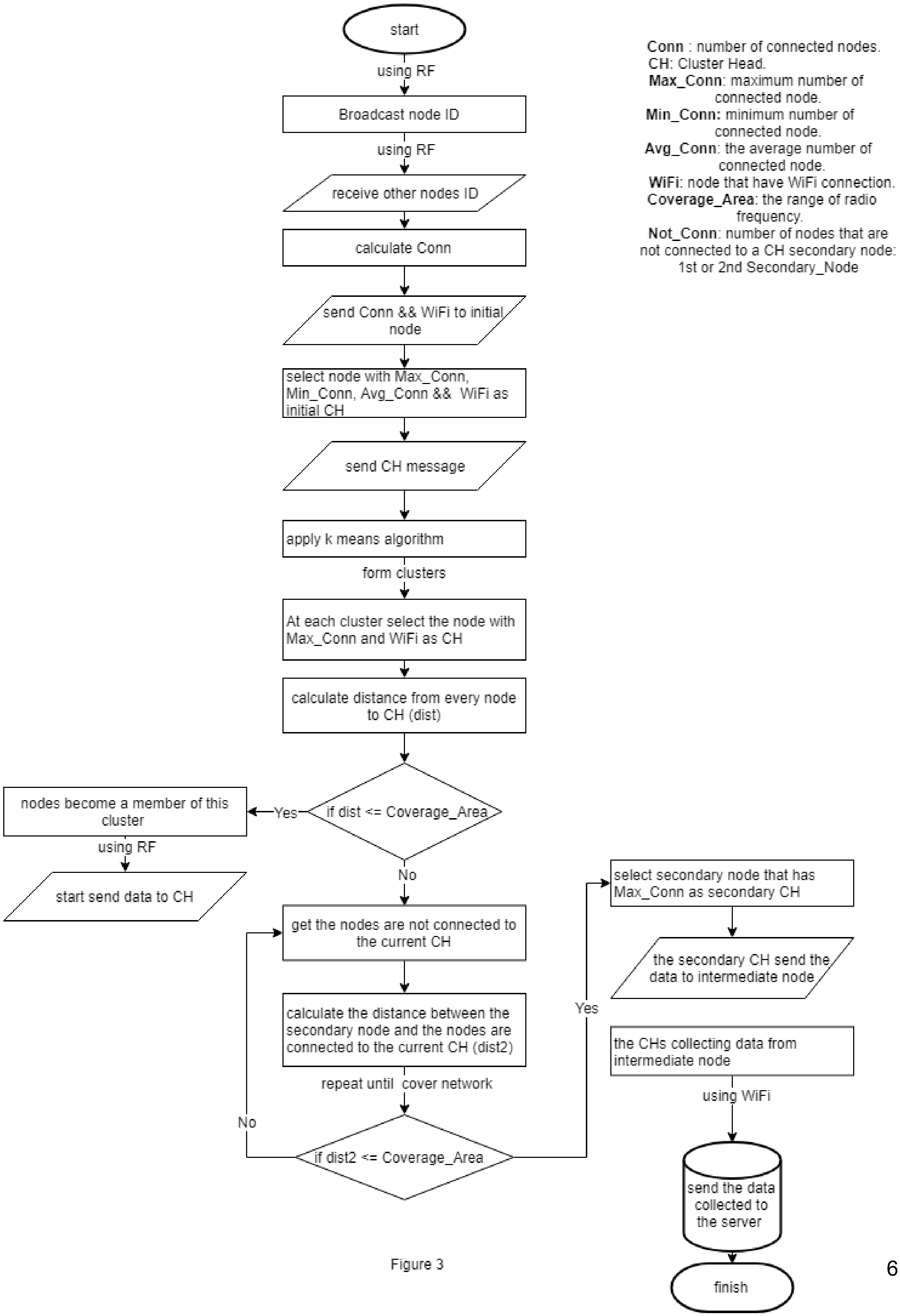
start
using RF
Broadcast node ID
using RF
receive other nodes ID
calculate Conn
send Conn && WiFi to initial node
select node with Max_Conn, Min_Conn, Avg_Conn && WiFi as initial CH
send CH message
apply k means algorithm
form clusters
At each cluster select the node with Max_Conn and WiFi as CH
calculate distance from every node to CH (dist)
if dist <= Coverage_Area
Yes
nodes become a member of this cluster
using RF
start send data to CH
No
get the nodes are not connected to the current CH
calculate the distance between the secondary node and the nodes are connected to the current CH (dist2)
repeat until cover network
if dist2 <= Coverage_Area
No
Yes
select secondary node that has Max_Conn as secondary CH
the secondary CH send the data to intermediate node
the CHs collecting data from intermediate node
using WiFi
send the data collected to the server
finish
Conn : number of connected nodes.
CH: Cluster Head.
Max_Conn: maximum number of connected node.
Min_Conn: minimum number of connected node.
Avg_Conn: the average number of connected node.
WiFi: node that have WiFi connection.
Coverage_Area: the range of radio frequency.
Not_Conn: number of nodes that are not connected to a CH secondary node: 1st or 2nd Secondary_Node

Figure 3

The nodes go through many states during its working cycle. So, there are three main stages: **Broadcast**, **cluster**, **send data** (Figure 4.1).

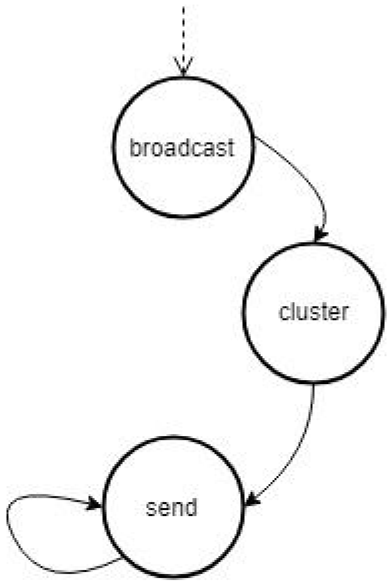

Figure 4.1

The broadcast stage (Figure 4.2) start with idle state, then it moves to send state in order to send its ID then it waits for some time then moves to receive state and receive other nodes ID. After that it move to the next stage.

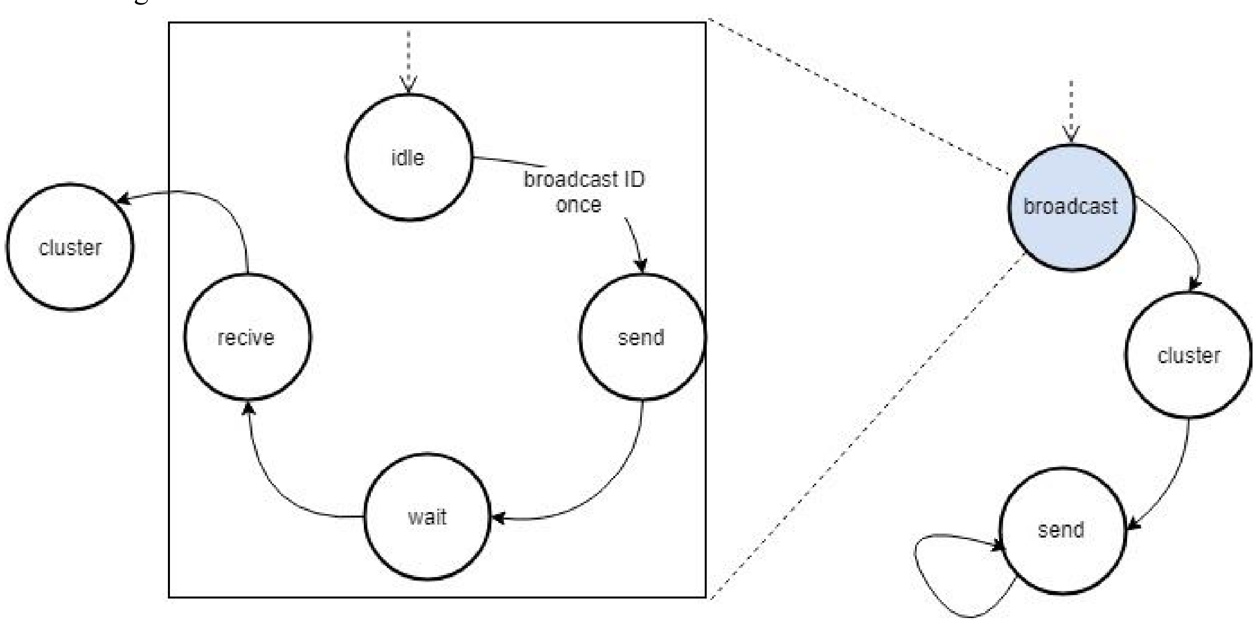

Figure 4.2

Then forming cluster stage. In this stage there is two state diagram one for original node (Figure 4.3) and another for initial cluster head. The original node starts with send state and send the number of connected nodes and if it has WiFi connection or not. And after some time, the state change to wait state. Then it moves to receive state and check the received message which contain ID, if the ID received is my ID it moves to receive state. Otherwise, it moves to send state.

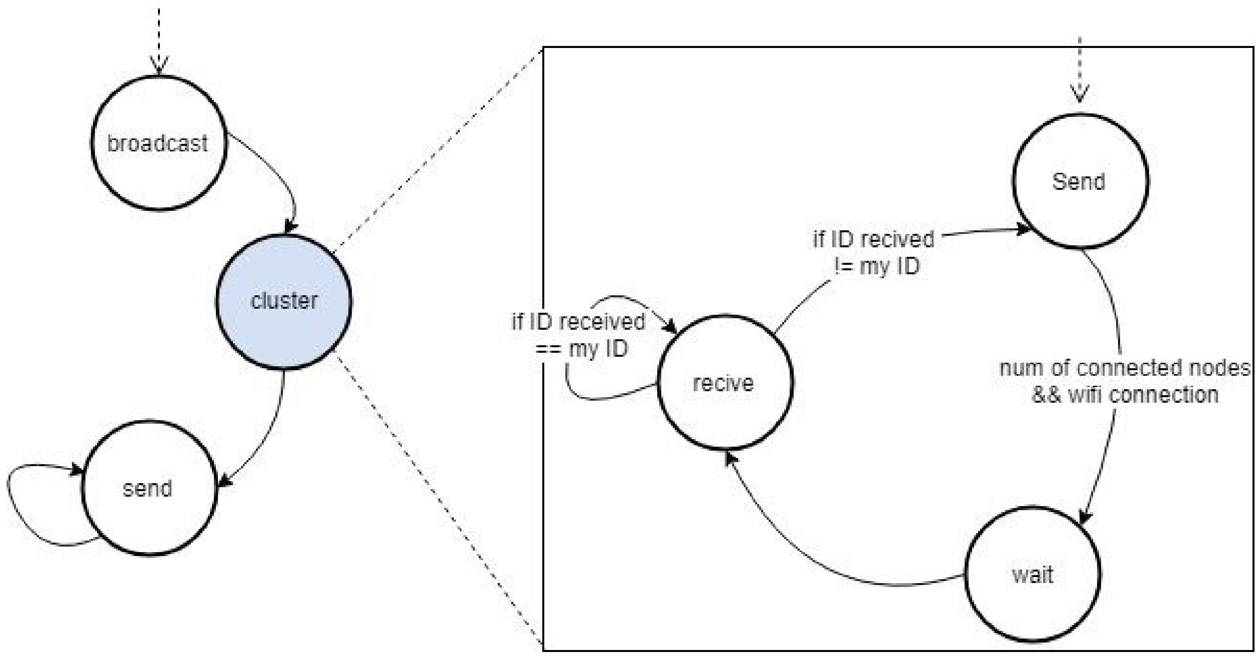

Figure 4.3

For initial CH (Figure 4.4) it starts with receive state and receive number of connected nodes from other nodes. Then it compare between the received message at the calculate state and select the node with the highest number of connected nodes and has WiFi connection. Then, send the ID of the selected node. Then it check the sent message which contain ID, if the ID received is my ID it moves to receive state. Otherwise, it moves to send state.

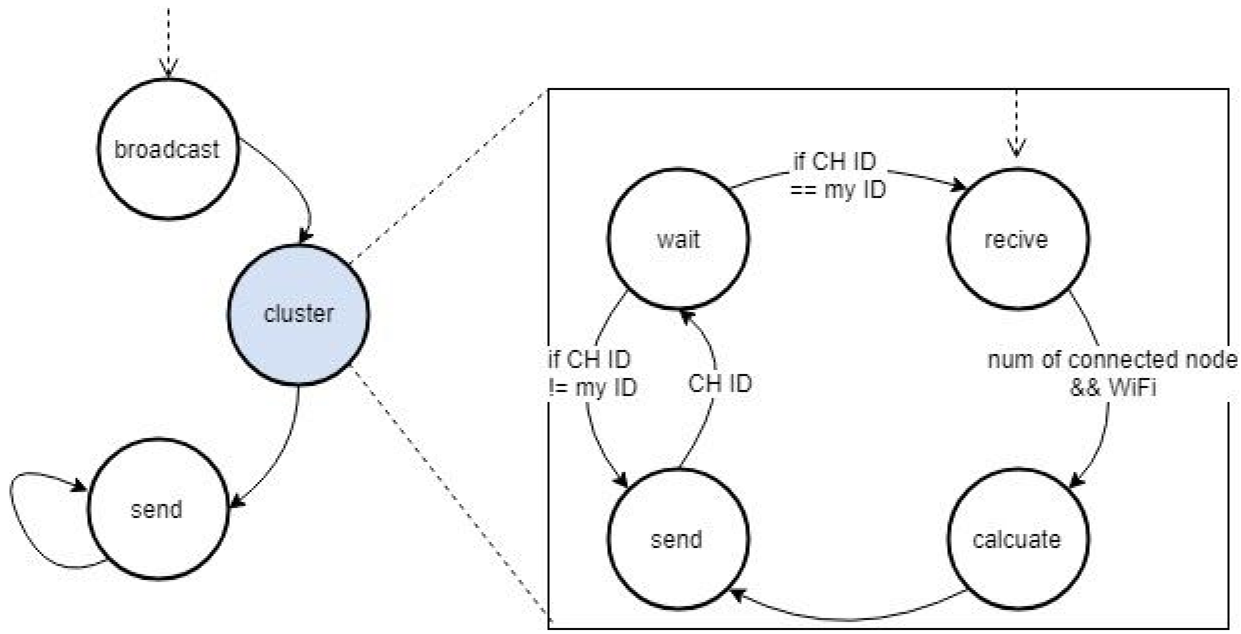

Figure 4.4

Finally collect data stage. For original node (Figure 4.5), it sends data (water meter reading) at the send state. Then it waits for sometime and moves to finish state. Also, this stage has state for CH, intermediate nodes, and secondary cluster. The state for these node (figure 4.6) start by receiving data from other nodes at the receive state. Then moves to collect state and send it at the send state.

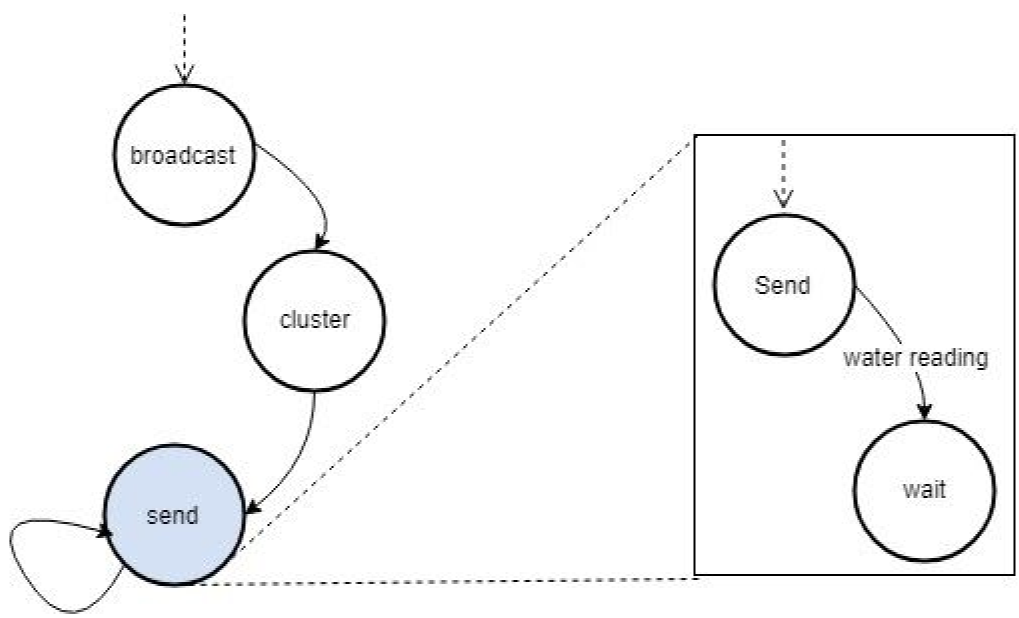

Figure 4.5

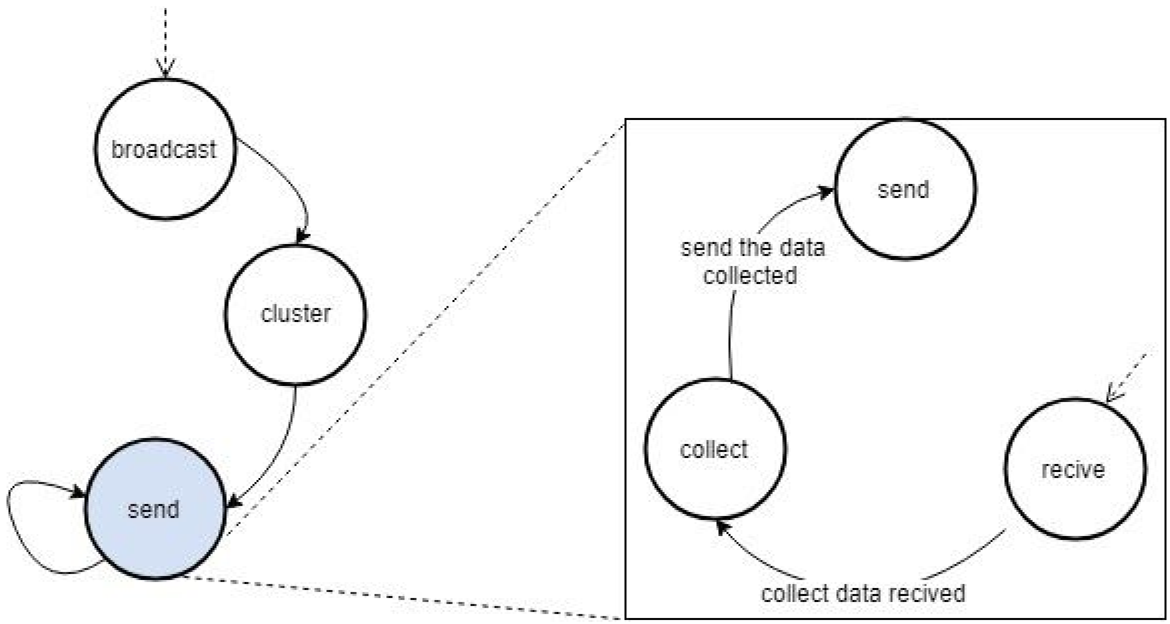

Figure 4.6

EVALUATION

We apply this protocol using ns2 simulation and evaluate delivery ratio and end to end delay of our protocol and randomly selected cluster the results was as the following:

First, end-to-end delay as we can see in Figure 5 (where x axis represents simulation time and y axis represents end-to-end delay in milliseconds) the two curves have roughly the same delay however our protocol has higher delay because it require more processes and more iteration to form the network.

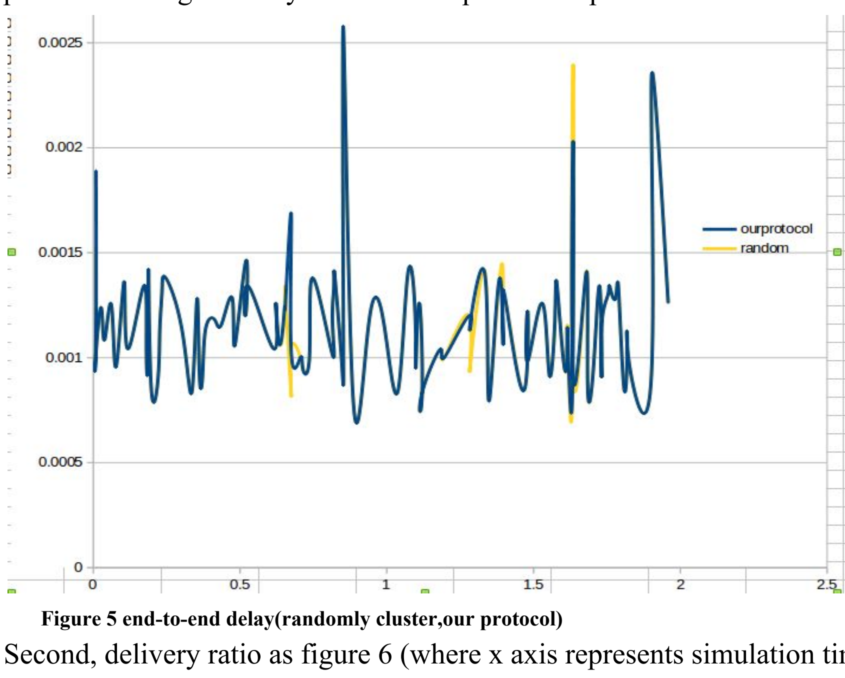

Figure 5 end-to-end delay(randomly cluster,our protocol)

Second, delivery ratio as figure 6 (where x axis represents simulation time and y axis represents delivery ratio) shows the curves started very high because of broadcast messages then it decrease to because of cluster formation so the number of received packets will decrease. When the network became stable our protocol has higher delivery ratio

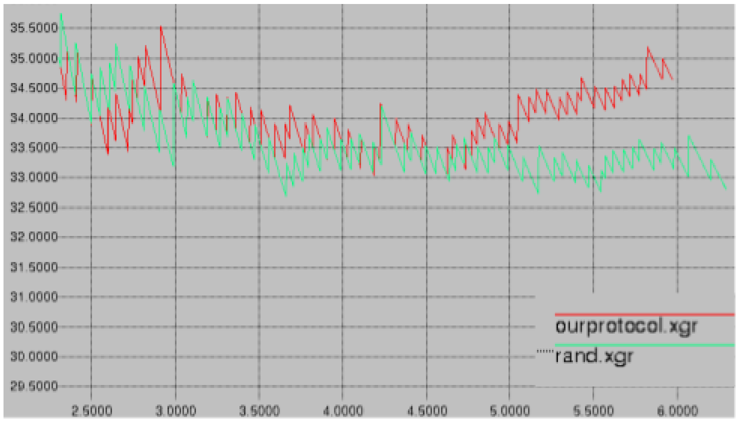

Figure 6 delivery ratio (randomly cluster,our protocol)

FUTURE WORK:

Despite our effort to cover the whole graph sometimes there will be isolated areas that we can't reach. So, we think about to solve this problem using mobile car that collect data from such areas. Also, we can change the algorithm to choose the number of clusters by itself.

CONCLUSION

This project was applied by configuring two heterogeneous network that work in harmony using RF and WiFi technology. Also, the network divides nodes to clusters using k-means algorithm. However, CH doesn't chosen randomly like k-means it is chosen based on the number of connected and availability of WiFi. Finally, our protocol was applied using ns2 simulator and was evaluated in terms of delay and delivery ratio and as we saw the results was better for our protocol.

REFERENCES

[1]Research on Water Meter Reading System Based on LoRa Communication
[2]http://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Lora-or-LoRaWAN.html
[3] M Suresh, U. Muthukumar and Jacob Chandapillai, “A Novel Smart Water-Meter based on IoT and Smartphone App for City Distribution Management ”, Data Acquisition Systems Laboratory, Centre for Water Management Fluid Control Research Institute. Palakkad, Kerala. 678 623. INDIA.
[4]An IoT based 6LoWPAN enabled Experiment for Water Management
[6]http://www.teomaragakis.com/hardware/electronics/how-to-connect-an-esp8266-to-an-arduino-uno/
[7]https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf
[8]https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
[9]https://www.farnell.com/datasheets/1682209.pdf