**Bus Tracking System Without Using GPS**

Submitted in partial fulfilment of the requirements

of the degree of

##### **BACHELOR OF ENGINEERING**

in

**COMPUTER ENGINEERING**

By

ROSHNI JAIWAR 33

VIRAJ KADAM 37

ROHAN PARANGE 60

SHWETA YADAV 94

Supervisor:

**Mr. SAURABH PATIL**

(Asst.Prof., Computer Engineering Department, XIE)



Xavier Institute Of Engineering, Mahim

Computer Engineering Department

University of Mumbai

2016-2017

**CERTIFICATE**

This is to certify that the project entitled

“**Peer-to-Peer Messaging Application using Wi-Fi”**

is a bonafide work of

ROSHNI JAISWAR 33

VIRAJ KADAM 37

ROHAN PARANGE 60

SHWETA YADAV 94

submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **“Bachelor of Engineering”** in **“Computer Engineering”**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor/Guide

( Mr. Saurabh Patil )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of Department Principal

( Prof. Sushama Khanvilkar ) ( Dr. Y.D. Venkatesh )

**Project Report Approval for B.E.**

This project report entitled  **Bus Tracking System Without Using GPS** by

ROSHNI JAISWAR 33

VIRAJ KADAM 37

ROHAN PARANGE 60

SHWETA YADAV 94

is approved for the degree of **Bachelor of Engineering** in **Computer Engineering** by the **University of Mumbai** during the academic year 2018-2019**.**

**Examiner**

1.

2.

Date:

Place:

**Declaration**

We declare that this written submission represents our ideas in our own words and where other’s ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penalty action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Roshni Jaiswar (34)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Viraj Kadam (37)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Rohan Parange (60)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Shweta Yadav (94)

Date:

**Abstract**

Reliability in public transport is of great importance today. Many traveler of public transport spend an ample amount of time waiting at bus stops. Our project focuses on presenting a solution that can tackle the above problem by providing them with expected bus arrival time. We focus implementing solution for same using IoT technology, but without having GPS module imposed on any vehicle. As we will be using static position property of Bus stops for tracking bus. Estimated bus arrival time will be disseminated on bust stops, benefits traveler to take appropriate decision. The bus stops will be made smart by using single processing unit having capability of fetching required bus data from cloud. The unit at Bus stop also display bus position information and calculate arrival time. The bus mobility units have a small micro controller OBU (On-Board Unit) enabled with Wi-Fi connectivity receiver module. As the bus arrives on a bus stop, its OBU establish Wi-Fi connection with Smart Bus stop Unit and data exchange will be initiated. The bus will send its unique bus number and arrival time stamp. The smart unit at bus stop propagates same information on the cloud. The every other bus stop will access the stored data on the cloud by firing required bus query and calculate the estimated arrival time. The prototype will be implemented by having raspberry as single processing unit at bus stop and bus OBU will be nodeMCU board. We have used raspberry pi 3B, nodeMCU(ESP12),cloud service.

Our project is on bus tracking system without using GPS, the bus stop is static we utilize the property to getting estimating time of the bus arrival. In the architecture diagram initially the bus is at bus depot, ones the bus number is selected the system get activated. After activation bus exchanges the bus number and time with the bus depot. The bus depot is connected to the cloud, the bus depot updates the information on the cloud the next corresponding bus stop fetches the information from cloud and calculates approximate time that will required by the bus to reach at it. The time is calculated using the fixed distance between the depot and bus stop. In the next scenario, the bus is at bus stop it again connected to the on board unit (OBU) at the bus stop. It does same thing its exchanges the arrival time and bus number with OBU on the bus stop its update the information to the cloud and this information is fetch by the next corresponding bus stop from the cloud. The calculates time that will be required for the bus to reach of this bus stop.

**Table of contents**

1. **Introduction**

**1.1 Problem Definition**

**1.2 Aim and Objective**

**1.3 Scope of the project**

**1.4 Existing System**

**2. Review of Literature**

**3. Description**

**3.1 Analysis**

**3.1.1. Use-case Diagram**

**3.1.2. Activity Diagram**

**3.2 Design**

**3.2.1. Block Diagram**

**3.2.2. Flow chart**

**3.3 Implementation Methodology**

**3.3.1. Port hopping sequence**

**3.3.2. Port Hopping process**

**3.3.3. Api used**

**3.3.3.1. Wifi manager**

**3.3.3.2. WifiP2P manager**

**3.3.4. Security**

**3.4 Details of Hardware and Software**

**3.4.1. Software Requirements**

**3.4.2. Hardware Requirements**

**4. Implementation plan for next Semester**

**Appendix**

**References**

**Acknowledgement**

**Bus Tracking System Without Using GPS**

1. **Introduction**

1.1 Problem Definition:

Daily many people travel through buses, they spend a lot of time waiting at the bus stop. Bus mostly delay and as we are not aware about the arrival of the bus, we are unable to make decision related to our travelling through which we can easily reach to our destination .We are building a Bus tracking system which will help people to get the information related to bus arrival .By knowing the information people can make decision whether they wanted to wait for the bus, choose another bus change the route or wanted to travel from another vehicle. We are not using GPS within the system. As bus stops are static and Buses are mobile these property can be used to make the system without using GPS. This will help people to access the information without using internet.

1.2 Aims/Objectives:

The primary objective of this paper is to make public transport more reliable by facilitating the traveller with approximate arrival time of the buses at the bus stop This paper presents a simple and cost effective solution to make public transportation services ‘smart’. The required solution is aimed to not only facilitate improvement in the services, but should also be a driving factor for increase in trust on the public bus transport systems.

1.3 Scope of the project

* We are making a system which will deliver information of the bus smartly by tracking it’s prior estimated time to reach the bus stop.
* Standing at the bus stop without knowing the arrival time is a major problem.hence the system aims to provide the user with the estimated time arrival of the buses.
* Knowing the arrival time will help the user in taking decision for route selection public transportation.

1.4 Existing System

Typically, bus tracking systems show data related to the arriving buses, time to arrival and estimated departure time based on location of bus that is obtained via Radio Frequency (RF) transceiver . There are such transceivers on every bus and bus stops so that the buses communicate the location to bus stops. The microprocessors at bus stop then calculate estimated time for arrival and display it on a screen.

SMS services on GSM are used in some solutions as means for communication . There are GSM modules with Global Positioning System (GPS) on the bus. These periodically send the location to databases through SMS. The location data can then be accessed by sending SMS to receivers on the database servers.

Location data can be obtained by GPS based tracking devices mounted on vehicles. These systems use Hyper Text Transfer Protocol (HTTP) to send location data to databases . Android devices in the buses stream location data to servers, which can be accessed through android applications or web-portals. However, none of these systems can handle an increase in requests. The data consumption for sending the data from buses to servers is high. The backend will crash if there is an increase in the number of users using the service. Therefore there is a need to explore a solution that will be scalable and efficient.

.

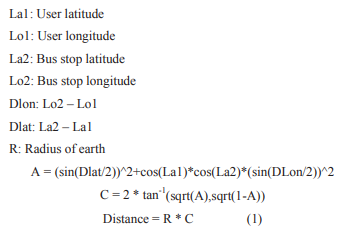
1. **Review of Literature**

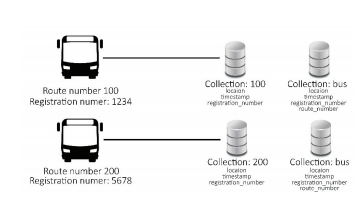
As we know that now-a-days about 60% of the people are making their transit as a Bus. The services of real-time tracking and arrival time prediction of buses is a very useful for bus riders, especially in cities.

In our Bus System communication is the main milestone to be achieved. Collecting real time location data is the major issue to tackle due to involvement of hardware which will need heavy investment. The traditional solution used in existing systems is installing GPS and General Packet Radio Service (GPRS) enabled devices in every bus. These devices are costly.

. Due to heavy traffic and road work etc., most of the buses are not in time. At the bus terminus people have to wait for long time without having knowledge of when the bus will arrive. Anybody who wants to use the public transportation system, they can’t find the time of arrival of particular bus at the particular destination even plan their departure from home accordingly. Due to unexpected delays in traffic jams the bus arrival time cannot be predicted. Our main aim is to develop a system to which the user’s waiting time reduced for bus and will provide him/her with all necessary details regarding the arrival/departure time of the bus.ha

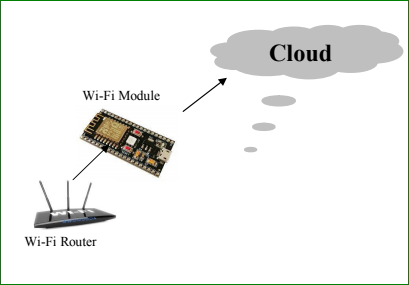
In this paper we are proposing the cheaper alternative for GPS. We are not using GPS. As the number of buses are more and the number of bus stops are less and they are static, this property of bus stop being static is being used for connection and getting the location of the bus .Whenever bus arrive at bus-stop the esp32 installed over the bus get connected to the bus stop and the location of the bus stored in the database according the bus number and route is share via cloud and fetched by other buses.Thus no internet is required.





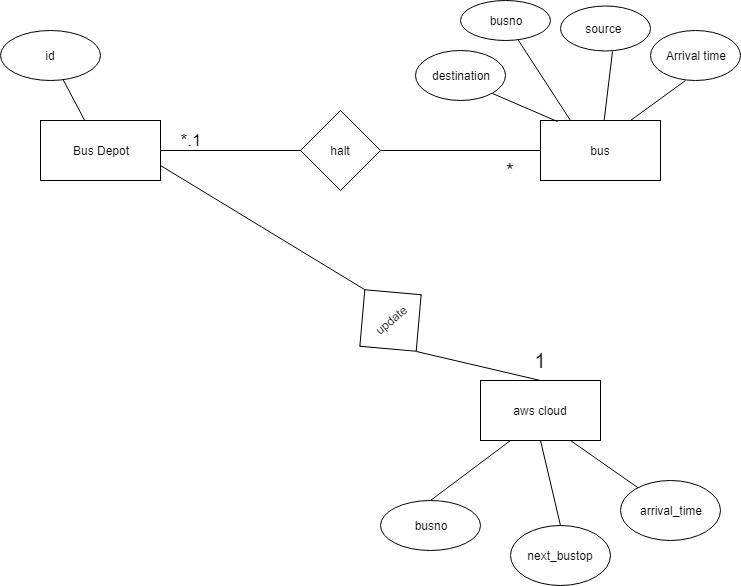
The other drawback of using GPS is requires lot of parameters to know the exact location of the bus such as longitude,latitude and altitude which is then needed to be stored in database and this requires lot of storage and lot of query processing and formulation while retrieving the location which again increases the cost .

most of the proposed system have used wifi module in the bus . the Wi-Fi module gets connected to the router which is installed in the bus terminal. The location is determined with the help of latitude and longitude of the bus terminal and transmits that data to the cloud through the Router. Here the wifi module is responsible for updating the cloud ,since the buses are dynamic it may happen that the wifi connectivity is lost before the data gets updated on the cloud.



**3.1 Analysis**

3.1.1. ENTITY RELATIONSHIP DIAGRAM



An ER diagram is a means of visualizing how the information a system produces is related.ER diagram specifies the entity and the attributes used in the system.The entity is an object or concept about which you want to store information.

Entities are:-

* Bus Depot-The bus depo act as a initial point from where the bus starts to travel. The bus depot and bus stops has a OBU which provide wireless connectivity for information transfer and receive the data the bus and update to the cloud.
* Bus - Once the bus number is selected the OBU unit get activated. The esp32 on the Bus get connected to the OBU unit on the bus depot or bus stop and shares the Bus number and initial time and update it on the cloud .
* CLOUD - It stores and provide the updated data .

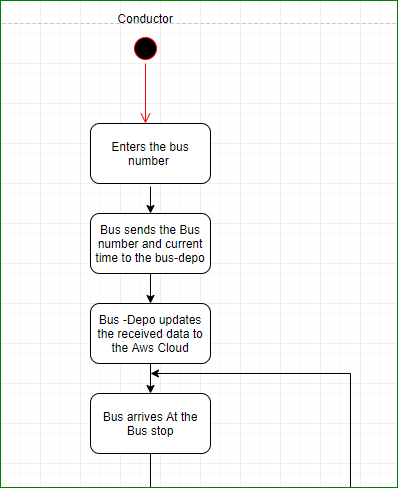
Attributes are:-

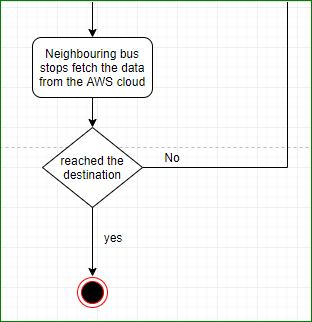
* id - It stores the unique id alloted to bus depot
* busno - It is the unique number given to the bus for specific routes.
* arrival\_time - this stores the arrival time of bus
* source - the start location of bus
* destination - the end location of bus
* next\_bustop - the adjacent next bus stops at which we have to display time .

Relationship:-

* Bus halts at the bus stop and get connect to it and share the data.
* Bus depot updates the information that he gets after connecting to the bus,i.e the busno,time on the aws cloud.

3.1.2. ACTIVITY DIAGRAM





**Fig. 1**

Fig. 1 depicts the activity diagram shows the task that are been performed

* firstly,the conductor is the actor,it start the system by entering the bus number

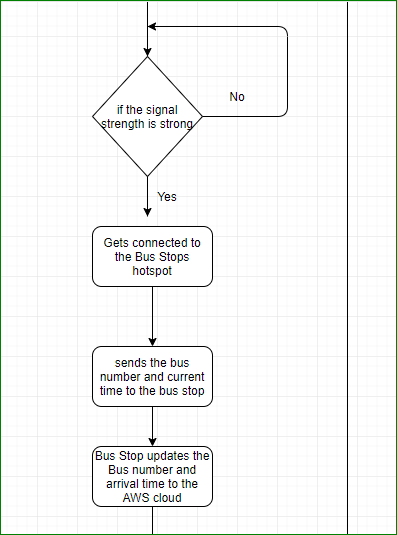


Fig. 2

Fig. 2 is the activity diagram of the system. The activity diagram has been divided into three sections: Sender, System Network and receiver. The user will type the intended message and send it in the first state. In the second state the sender’s device will send the initial sequence which has been hashed. In the third state the sender will synchronise it’s port hopping scheme.

In the fourth state the sender will send the encrypted values of port address, encrypted port range which is restricted by the values used in NAT (Network Address Translator) and PAT (Port Address Translator) and the initial port sequence to the receiver/intermediate node. In the fifth state the port hopping scheme will check for available ports in order to perform the transfer of data between the nodes. In the sixth state the message (encrypted) is sent over multiple ports. In the sixth state i.e at the receiver end the message is then decrypted. It is only in the seventh state that the decrypted message can be read by the receiver. In the eighth state the message and all its data is deleted/erased from all the intermediary nodes.

3.2 Design

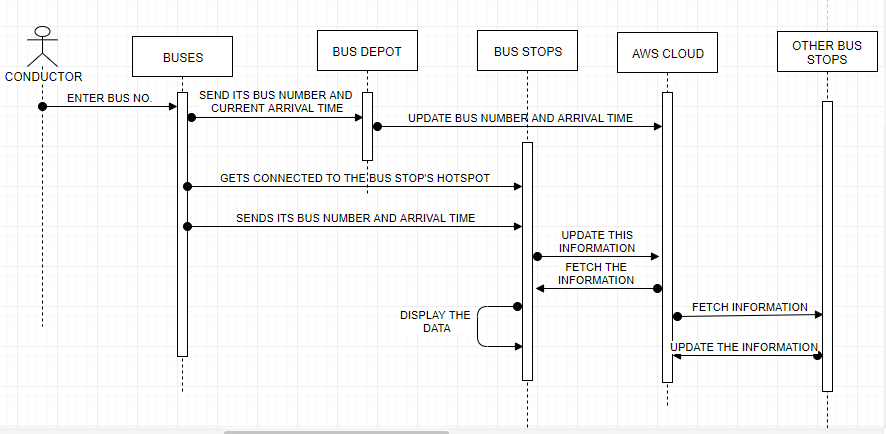
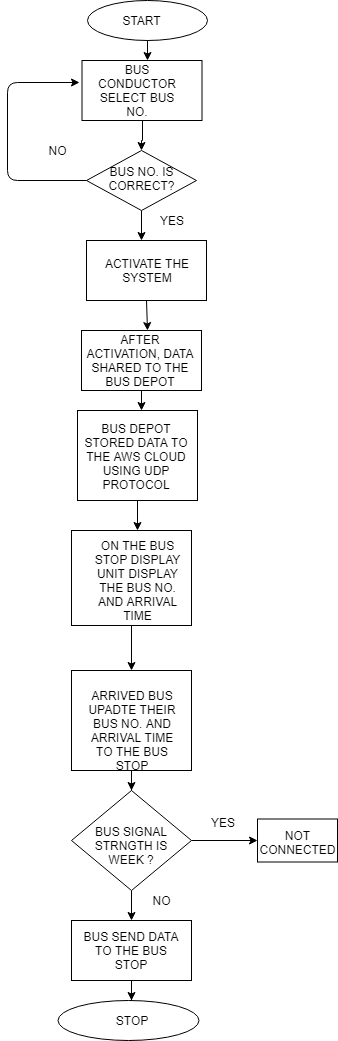


Fig. 3

Fig. 3 is the block diagram of the system. This is as simple as it gets, the diagram depicts the communication between two primary nodes; the sender and the receiver. In this typical scenario, the sender sends a message to a receiving node. This transmission is made possible by relaying the information from one secondary node to another- also called as the intermediate transition node. The transition nodes relay this information, until the message is received by the receiving node. Once receiving the message, the receiver node sends an acknowledgement which is relayed back to the sender node in a similar fashion. If no acknowledgement is received by the sender node within the time limit, it has to retransmit the original message assuming that the message was lost.

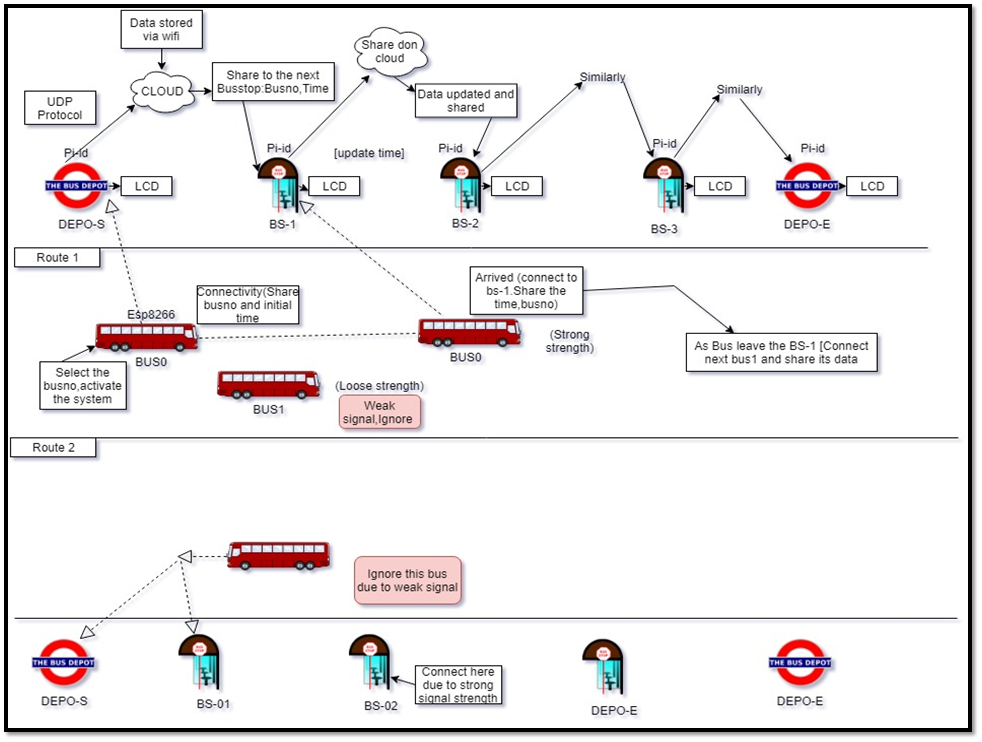


Fig. 4

The figures above are the flowcharts of the system. It depicts the flow of actions and data in the system. The sender uses the application to send the message. The application thereby uses the transmission module to relay the data to other intermediate nodes until the message is received by the receiver node. The receiver node then sends an ack packet to the sender node which travels back to the sender node in a similar fashion. If there is no ACK from the receiver, the sender has to retransmit the message assuming that the message was lost during the process.

The transmission is a key module as it plays an important role in every transmission. This module first creates a sub-network then checks whether the receiving node is present in the network. If yes, it directly forwards the message to that node. If not, then the transmission module elects an intermediate node and relays the message to that node. Now, the intermediate node is the host and creates a sub-network. It then processes the same way like the sending node did in the process of finding and delivering the message to the appropriate receiving node.

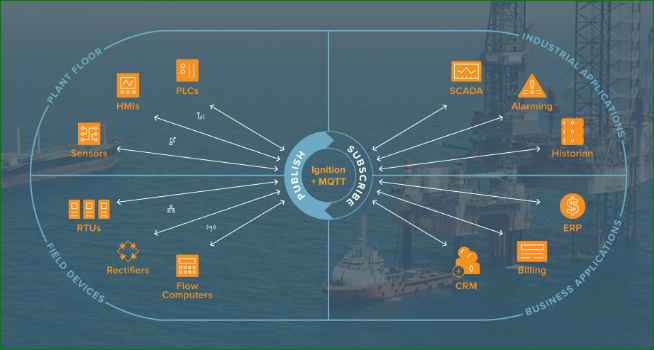
**3.3 Implementation Methodology**

3.3.1. MQTT protocol

MQTT is a publish/subscribe protocol that allows edge-of-network devices to publish to a broker. Clients connect to this broker, which then mediates communication between the two devices. Each device can subscribe, or register, to particular topics. When another client publishes a message on a subscribed topic, the broker forwards the message to any client that has subscribed.

MQTT is bidirectional, and maintains stateful session awareness. If an edge-of-network device loses connectivity, all subscribed clients will be notified with the “Last Will and Testament” feature of the MQTT server so that any authorized client in the system can publish a new value back to the edge-of-network device, maintaining bidirectional connectivity.

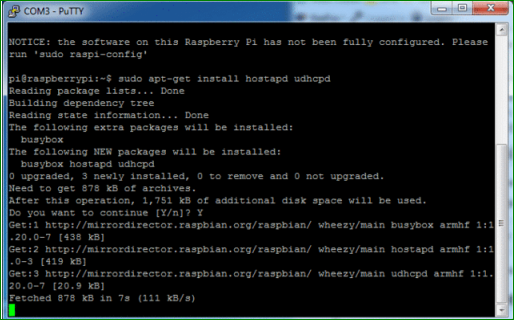
Diagrammatic Representation:

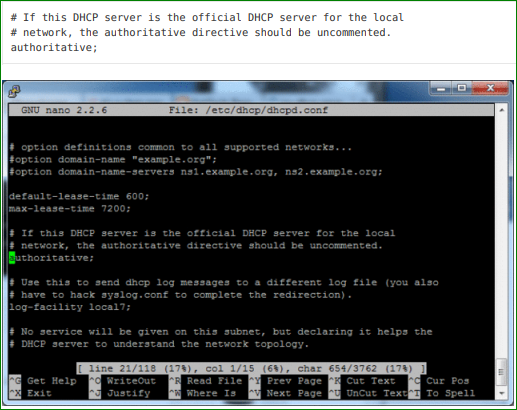


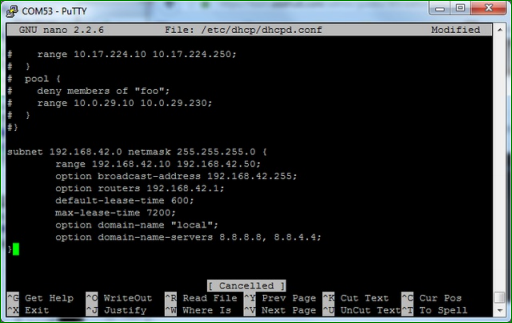
3.3.2. DHCP Server:

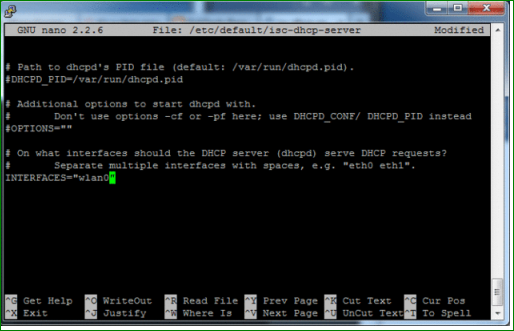
The Raspberry (present at the Bus-stop) will used as a DHCP Server. It will provide internet to the esp32 module present in the buses. The Esp32 module will get connected to the hotspot of the raspberry pi.

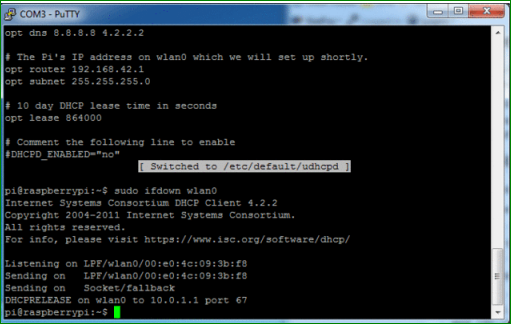
code to set raspberry pi as dhcp server

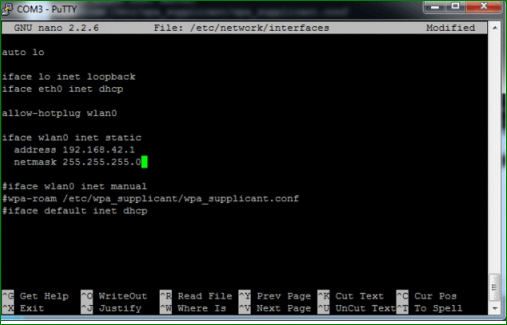


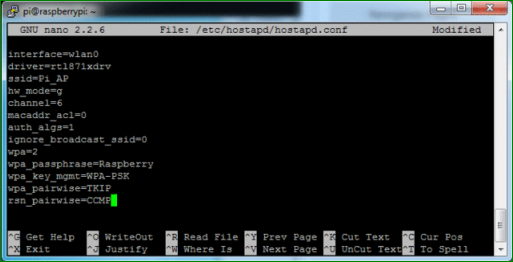


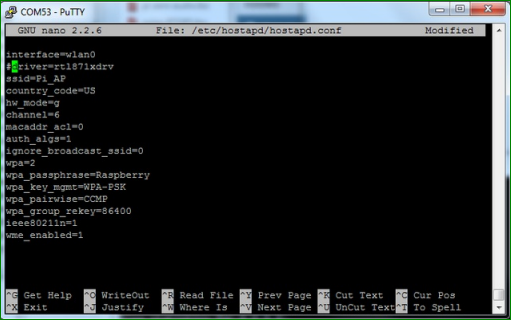












3.3.3.

**3.4 Details of Hardware and Software Requirements**

3.4.1. Software Requirements

* AWS cloud for storing and retrieving the bus number and its arrival time.As this information has to be shared to the neighbouring bus stops aws cloud is used for this purpose.

3.4.2. Hardware Requirements

* Raspberry Pi

This requirement is imposed because the application will be developed for the mobile devices as of now. Future updates and increments can cover devices like phablets and tablets.

* Esp32

The ardiuno will be installed in the buses and it will be responsible for exchanging the bus no and the current time with Raspberry pi.

* LCD

The bus number and the bus time will be displayed on the LCD . LCD will be at the bus stops .

**4. Implementation for next semester**

* The building and testing of our full fledged application will be our main priority.
* The designing of software shall be done using Android Studio and other applications if necessary.
* We will try to find an optimised algorithm to achieve maximum efficiency and throughput.
* Try to implement to send messages through bluetooth.

**Future Scope**

Currently, we are planning to develop this application to be used in a college campus atmosphere, basically to send and receive simple text messages, however in the future, we plan to increase the said range of Peer-to-Peer network which will definitely prove to be beneficial in times of a natural calamity or urgency to send emergency broadcast messages when the Internet is not readily available or not available at all.

Secondly, the application will support only text messages at the very base level, we can enhance this feature by including the ability to send and receive multimedia messages as well, in the form of future updates.

If the range can be increased to a remarkable level, then it could also be used in villages and other platforms and locations where network accessibility is poor, this would in turn help people living in remote areas to communicate better; specially in places where the density of the residents is comparatively far less than the density of the people residing in the cities and thereby implying that not many nodes will be available to transfer or relay messages from the sender to the receiver.

The use of Internet can also be incorporated as a switch to send packets which are not being delivered because of technical or topographical difficulties.

It could also be tweaked to send larger packets from sender to receiver.

Lastly, we are developing this application for Android users only, which means, the mobile users of other operating systems like iOS and Windows will not be able to avail this application and use its benefits. We are looking forward to develop the same application with the same features for any prominent mobile operating system out there to maximize the functionality of this application and facilitating a smooth transition from the native or current messaging applications to our proposed messaging application.

All in all, this application can serve as an alternative to all the messaging applications present today, providing a unique and a cost-effective solution to send the message across.

**References**

[1] Phone-to-Phone Communication Utilising Wi-Fi Hotspot in Energy-Constrained Pocket Switched Networks.

[2] Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs by P. Prasithsangaree and P. Krishnamurthy Telecommunications Program University of Pittsburgh, PA 15260.  
[3] United States Patent Application Publication Oran. Pub. no. US2007/0070996 A1.

[4] www.developer.android.com

**Acknowledgement**

The research was supported by Xavier Institute of Engineering. We are thankful to our professors who provided expertise that greatly assisted the research. We are also grateful to Mr. Saurabh Patil for guiding us through the process and giving insightful feedbacks, and Prof. Kunal Meher our project moderator. We have to express out appreciation to again name for sharing their pearls of wisdom with us during this course of this research. We are also immensely grateful \_\_\_\_\_\_

for their comments on earlier versions of the manuscript, although any errors are our own and should not tarnish the reputation of these esteemed professionals.